Note:
Before using this information and the product it supports, read the "Notices" topic at the end of this information.

Eighth Edition (July 2018)
This edition applies to Version 3 Release 1 of IBM IMS Index Builder for z/OS (program number 5655-R01) and to all subsequent releases and modifications until otherwise indicated in new editions.
This edition replaces SC18-9101-06.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
Contents

About this information ...................................................... v

Part 1. Introduction and product setup ................................ 1

Chapter 1. IMS Index Builder overview ................................. 3
What's new in IMS Index Builder ........................................ 4
What does IMS Index Builder do? ....................................... 9
IMS Index Builder terminology ......................................... 11
IMS Index Builder components ......................................... 15
IMS Index Builder architecture .......................................... 16
IMS Index Builder business scenarios .................................. 17
IMS tools product integration ........................................... 18
IMS Index Builder restrictions .......................................... 19
Service updates and support information .............................. 20
Product documentation and updates ..................................... 21
Accessibility features ..................................................... 23

Chapter 2. Configuring and activating IMS Index Builder ........... 25
Hardware and software requirements ................................... 26
Configuring the operating environment ............................... 27
Main storage requirements ............................................... 27
Enabling inter-region communication (APF authorization) ....... 27
Activating and verifying IMS Index Builder installation ........... 28
Activating IMS Index Builder ........................................... 28
Verifying IMS Index Builder installation .............................. 29
Configuring for DBRC command authorization ...................... 30
Configuring the IMS Tools Knowledge Base server ................. 31
Configuring IMS Tools Online System Interface .................... 32

Chapter 3. Migrating from IMS Index Builder Version 2.3 to Version 3.1 . . 33

Part 2. Using IMS Index Builder ........................................... 35

Chapter 4. IMS Index Builder functions ................................. 37
Creating secondary indexes ............................................. 38
Creating new secondary indexes ....................................... 40
DFSURWF1 and HPSRSIDX output files .............................. 42
Building secondary indexes using DFSURIDX as input .......... 45
Building partitioned secondary indexes using HPSRWF as input 48
Initializing secondary indexes ......................................... 50
Reorganizing and recovering secondary indexes ................... 51
Building a HIDAM primary index ...................................... 52
Building a HIDAM primary index on an alternate primary index data set 54
Building PHIDAM partition primary indexes ......................... 56
Building HALDB partition ILDS ....................................... 57
Building indexes and ILDSs for IMS catalog databases ............ 58
Selecting the scan method ............................................... 59
Issuing the IMS commands automatically ............................ 61
Using the IMS catalog .................................................. 63

Chapter 5. JCL and input control statements ............................ 65
IMS Index Builder JCL .................................................... 66
EXEC statement ....................................................... 66
DD statements .......................................................... 66
Secondary index DD statements ......................................... 71
Data set group DD statements ......................................... 71
IMS Index Builder input control statements ........................ 72
Required control statements .......................................... 73
Optional control statements ............................................ 77
Control statements incompatible with Version 3.1 ............... 97
JCL and input control statements by task summary ............... 98

Chapter 6. Defining the subordinate address space procedure .......... 99
RACF considerations .................................................... 101

Chapter 7. Defining runtime parameters ................................. 103

Chapter 8. Index records with duplicate keys ......................... 111

Part 3. Troubleshooting .................................................... 113

Chapter 9. Gathering diagnostic information ............................ 115

Chapter 10. IMS Index Builder output and control flow ................. 117
IMS Index Builder output files .......................................... 118
Control flow events reported on IIUSNAP ............................ 119

Chapter 11. IMS Index Builder reason codes ............................... 121
IMS Index Builder return codes and error handling ............... 122
IMS Index Builder initialization phase primary reason codes ... 124
IMS Index Builder initialization phase secondary reason codes . 128
IMS Index Builder control flow event codes ......................... 130
IMS Index Builder processing phase failure reason codes ........ 133
IMS Index Builder verification phase failure reason codes ........ 144
IBM® IMS™ Index Builder for z/OS® (also referred to as IMS Index Builder) enables you to build (or rebuild) IMS secondary indexes, Hierarchical Indexed Direct Access Method (HIDAM) primary indexes, and Indirect List Data Sets (ILDS).

IMS Index Builder supports full-function non-partitioned databases and partitioned high availability large databases (HALDB).

IMS Index Builder is a component of IBM IMS Database Solution Pack for z/OS, IBM IMS Database Utility Solution for z/OS, and IBM IMS Recovery Solution Pack for z/OS. IMS Index Builder is also available as a separately orderable product.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform these tasks:
• Plan for the installation of IMS Index Builder
• Install and operate IMS Index Builder
• Customize your IMS Index Builder environment
• Diagnose and recover from IMS Index Builder problems
• Design and write applications for IMS Index Builder
• Use IMS Index Builder with other IMS products

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

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

Always check the IMS Tools Product Documentation page for complete product documentation resources:


The IMS Tools Product Documentation page includes:
• Links to [IBM Knowledge Center](http://www.ibm.com/support/docview.wss?uid=swg27020942) for the user guides ("HTML")
• Links to the PDF versions of the user guides ("PDF")
• Program Directories for IMS Tools products
• Recent updates to the user guide, known as "Tech docs" ("See updates to this book!")
• Technical notes from IBM Software Support, known as "Tech notes"
• White papers that describe product business scenarios and solutions
Part 1. Introduction and product setup

IBM IMS Index Builder for z/OS (also referred to as IMS Index Builder) is an IMS tool that helps streamline index maintenance and recovery. The tool creates multiple indexes in one job step and eliminates the need to image copy indexes.

Topics:
- Chapter 1, “IMS Index Builder overview,” on page 3
- Chapter 2, “Configuring and activating IMS Index Builder,” on page 25
- Chapter 3, “Migrating from IMS Index Builder Version 2.3 to Version 3.1,” on page 33
Chapter 1. IMS Index Builder overview

IBM IMS Index Builder for z/OS (also referred to as IMS Index Builder) enables you to build (or rebuild) IMS secondary indexes, Hierarchical Indexed Direct Access Method (HIDAM) primary indexes, and Indirect List Data Sets (ILDS).

IMS Index Builder supports full-function non-partitioned databases and partitioned high availability large databases (HALDB).

Important: Throughout this information, references to HIDAM databases also apply to Partitioned Hierarchical Indexed Direct Access Method (PHIDAM) databases, unless stated otherwise. Likewise, references to Hierarchical Direct Access Method (HDAM) databases also apply to Partitioned Hierarchical Direct Access Method (PHDAM) databases, unless stated otherwise.

Topics:
- “What's new in IMS Index Builder” on page 4
- “What does IMS Index Builder do?” on page 9
- “IMS Index Builder terminology” on page 11
- “IMS Index Builder components” on page 15
- “IMS Index Builder architecture” on page 16
- “IMS Index Builder business scenarios” on page 17
- “IMS tools product integration” on page 18
- “IMS Index Builder restrictions” on page 19
- “Service updates and support information” on page 20
- “Product documentation and updates” on page 21
- “Accessibility features” on page 23
What's new in IMS Index Builder

This topic summarizes the technical changes for this edition.

New and changed information is indicated by a vertical bar (|) to the left of a change. Editorial changes that have no technical significance are not noted.

SC18-9101-07 (July 2018)

This edition covers the following functional enhancements provided by APARs PI59810, PI61398, PI65714, PI72375, PI73092, PI83668, PI77681, PI88851, and PI92715.

<table>
<thead>
<tr>
<th></th>
<th>Support for IMS Version 15.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>New control statement, PROGMON, to specify the interval for issuing load progress messages. For more information, see <a href="#">“PROGMON control statement” on page 85</a>.</td>
</tr>
<tr>
<td></td>
<td>Enhancement to support the PARTNAME control statement when building PSINDEX databases. For more information, see <a href="#">“PARTNAME control statement” on page 85</a>.</td>
</tr>
<tr>
<td></td>
<td>Support for update mode, to update only the existing records in PSINDEX databases. For more information, see <a href="#">“PROC control statement” on page 73</a>.</td>
</tr>
<tr>
<td></td>
<td>Enhancement to support retrieving DBDs from the IMS catalog directory data sets instead of DBD libraries. For more information, see <a href="#">“Using the IMS catalog” on page 63</a>.</td>
</tr>
<tr>
<td></td>
<td>New control statement, ZIIPMODE, to offload eligible workloads to zIIP processors. For more information, see <a href="#">“ZIIPMODE control statement” on page 95</a>.</td>
</tr>
<tr>
<td></td>
<td>Restrictions for using IMS exit routines documented in <a href="#">“IMS Index Builder restrictions” on page 19</a>.</td>
</tr>
</tbody>
</table>

SC18-9101-06 (March 2016)

This edition covers the following functional enhancements provided by APARs PI27638, PI28218, PI32727, PI48799, PI51722, PI56192, and PI58843.

| | Support for IMS Version 14. |
| | Support for HALDB 8 GB OSAM data sets. |
| | New control statement, COMPAUTH. See [“COMPAUTH control statement” on page 77](#). |
| | Security improvement. All the libraries concatenated to the STEPLIB and JOBLIB must be APF-authorized without exception. The DFSRESLB DD statement is no longer needed in IMS Index Builder JCL. |

SC18-9101-05

This edition covers the following functional enhancements provided by APARs PM74179, PM75258, PM76131, PM78145, PM86783, PM89774, PM92801, PM94114, PM97423, PI05347, PI06319, PI10953, PI15217, PI18256, PI21471, and PI24092.

| | When IMS Index Builder builds a HALDB primary index or an ILDS, it issues database authorization requests with EXCLUSIVE access intent for the HALDB partition instead of READ access intent. See [“DBAUTH control statement” on page 78](#). |
• If the DBAUTH YES,RECOV control statement is specified, IMS Index Builder issues the CHANGE.DBDS RECOV command to turn on the RECOVERY_NEEDED flag before requesting database authorization. See “DBAUTH control statement” on page 78.

• New control statements for overriding the job name of subordinate address space and the volume count for each sort stripe data set. See the following topics:
  - “IIURSORT control statement” on page 79
  - “JOBNSCAN control statement” on page 82
  - “JOBNSORT control statement” on page 83
  - “STRIPE control statement” on page 88

• Support for the IMS Tools Online System Interface of IBM IMS Tools Base for z/OS. By using the interface, IMS Index Builder can automatically issue IMS commands (/DBD, /DBR, and /STA) so that you do not need to issue these IMS commands manually. See the following topics:
  - “Issuing the IMS commands automatically” on page 61
  - “TOSIDBD control statement” on page 89
  - “TOSIDBR control statement” on page 91
  - “TOSISTA control statement” on page 92
  - “TOSIWAIT control statement” on page 93
  - “TOSIXCFGRP control statement” on page 93

• New parameter *NO for ITKBLOAD and ITKBSRVR control statements. When ITKBLOAD *NO is specified, the IMS Tools Knowledge Base library that is concatenated to the STEPLIB DD is not used. See the following topics:
  - “ITKBLOAD control statement” on page 81
  - “ITKBSRVR control statement” on page 82

• Revised installation and operational prerequisites. See “Hardware and software requirements” on page 26.

SC18-9101-04

This edition covers the following functional enhancements provided by APARs PM46849, PM46957, PM49620, PM49636, PM53350, PM56986, PM60941, PM64823, PM66456, and PM68881.

• Supports parallel sorting of PSINDEXes. You can sort the partitions in parallel to improve the performance of building PSINDEXes. For more information, see “MAXTASKS control statement” on page 83.

• New reason codes (901, 902, and 903) and an error message (IIUB080E) have been added to alert when block size is zero.

• ILDS control statement, which specifies to initialize Indir ect List Data Sets (ILDs) without loading ILE records, has been added. For more information, see “ILDS control statement” on page 80.

• A new message (IIUB081I) to alert when the processed HALDB has logical relationships and ILE records need attention.

• WFP output files of IMS High Performance Load can be used as input for building PSINDEXes. For more information, see “Building partitioned secondary indexes using HPSRWFP as input” on page 48.

• Configuration steps for enabling the DBRC command authorization have been added. For more information, see “Configuring for DBRC command authorization” on page 30.
Indexes and indirect list data sets (ILDS) for IMS catalog databases can be built. For more information, see "Building indexes and ILDSs for IMS catalog databases" on page 58.

**SC18-9101-03**

This edition covers the following functional enhancements provided by APARs PM17738, PM17740, PM19218, PM21662, PM22320, PM23259, PM23498, PM26050, PM27122, PM30368, PM31946, PM34562, PM37055, PM37987, and PM42233:

- Support for IMS Version 12.
- The CLASS control statement specifies the high-level qualifier of stripe data sets. For more information, see "CLASS control statement" on page 77.
- The ITKBLOAD control statement specifies the name of the load module library for IMS Tools Knowledge Base. For more information, see "ITKBLOAD control statement" on page 81.
- The DBAUTH control statement specifies whether IMS Index Builder issues the IMS DBRC authorization request for the prime database and the indexes to prevent transactions and programs from updating the database. The IMS DBRC authorization request can be issued in IMS Version 10 or later. For more information, see "DBAUTH control statement" on page 78.
- The SVCDUMP control statement specifies whether IMS Index Builder requests an SVC dump file after a system abend or a user abend. For more information, see "SVCDUMP control statement" on page 88.
- The IIURDFLT MAXTASKS entry of the runtime option module (IIURPRMS) customizes the IMS Index Builder run. For more information, see "IIURDFLT MAXTASKS entry" on page 107.
- The IIUB062I message is added to indicate a dynamic allocation failure of a stripe data set and its cause.

This edition also contains the following documentation changes:

- Restrictions for using IMS Index Builder are now documented in "IMS Index Builder restrictions" on page 19.
- IMS Index Builder does not support shared secondary indexes. Information about this limitation is added to "IMS Index Builder restrictions" on page 19.
- Revised the instructions to activate IMS Index Builder. See "Activating IMS Index Builder" on page 28.

**SC18-9101-02**

This edition covers the following functional enhancements provided by APARs PK84777, PK89282, PK90640, PK98192, PK98315, PM01560, PM01565, PM04676, PM04978, PM05407, PM06592, PM10758, PM12486, and PM13947:

- Description about the sequential scan technology that was introduced in IMS Index Builder Version 3.1 and the instruction to change the scan methods are added. For more information, see "Selecting the scan method" on page 59.
- The SIDXBUF control statement enables you to specify the number of buffers that are to be used in the load process for each secondary index. For more information, see "SIDXBUF control statement" on page 86.
- The TMRWAIT control statement enables you to specify the IIUTMRXT timeout value. For more information, see "TMRWAIT control statement" on page 89.
- The VIC control statement specifies whether IMS Index Builder issues the DBRC NOTIFY.UIC command to turn off the IC NEEDED flag. For more information, see "VIC control statement" on page 94.
A sample JCL is provided to replace the runtime option module (IIURPRMS). Also, several entries are added to the IIURPRMS module table. For more information, see Chapter 7, “Defining runtime parameters,” on page 103.

The verification function of IMS Index Builder verifies that the number of records that were loaded to each index matches the number of the scanned occurrences of the prime database. For more information, see “IMS Index Builder verification phase failure reason codes” on page 144.

Diagnostics Aid is supported for diagnosing and reporting IMS Index Builder problems. For more information about using the Diagnostics Aid, see Chapter 15, “Diagnostics Aid,” on page 165.

SC18-9101-01

Instruction to migrate from IMS Index Builder Version 2.3 to Version 3.1 is added. See Chapter 3, “Migrating from IMS Index Builder Version 2.3 to Version 3.1,” on page 33.

This edition covers the following functional enhancements provided by APARs PK72674, PK73170, PK73601, PK73802, PK82341, and PK84520:

- Support for IMS Version 11.
- The SORTFSZ control statement is used to support heavy sort loads, which is different from IMS Index Builder Version 2.3. For more information, see “IMS Index Builder input control statements” on page 72.
- SORTFSZ triggers striped sort mode.
- Run time optimized code (race code) is applied by default in IMS Index Builder encoding routines.
- A standard, single subordinate address space procedure, started task is invoked by job name.
- Runtime parameters have changed. For more information, see Chapter 7, “Defining runtime parameters,” on page 103.
- The number of STCs can be controlled with the MAXTASKS control statement.
- Stripe data sets have been introduced to enhance communications with subordinate address space. Stripes are internal work files with specific HLQ.
- Index records with duplicate keys are stored in stripes with LLQ "DUPLKEYS." Duplicate keys stripes are not deleted at termination.
- IMS Index Builder does not shadow the index data sets, so the CATLG statement is obsolete. However, in the DDEFPDS functionality, user catalogs are required to be supported by DD statements.

SC18-9101-00

- Support for enabling the IMS Tools Knowledge Base has been added. The IMS Tools Knowledge Base is a central repository for automatically collecting output reports that are generated. For more information, see “IMS tools product integration” on page 18 and the ITKBSRVR statement definition in “IMS Index Builder input control statements” on page 72.
- This release supports routing WTOs and messages from the subordinate address space to the master address space. For more information, see the IIUCAII DD statement definition in “IMS Index Builder JCL” on page 66.
- The IMS Index Builder load library is no longer needed in the IMS concatenation. For more information, see the STEPLIB DD statement definition in “IMS Index Builder JCL” on page 66.
• Allocation contention problems have been resolved in this release. The ALTER statement is no longer needed.

• Support for indirect list data sets (ILDS) is provided in this release with the BLD_ILDS keyword. For more information, see the PROC statement definition in “IMS Index Builder input control statements” on page 72.

• Support for rebuilding selected PSINDEX partitions has been added. For more information, see the INDEX statement definition in “IMS Index Builder input control statements” on page 72.

• The scope of sequential scan has been extended to allow for rebuilding of ILDS, primary indexes, and some secondary indexes. For more information, see Chapter 6, “Defining the subordinate address space procedure,” on page 99.

• The RACF® profile is now passed from the master address space to the subordinate address space. For more information, see Chapter 6, “Defining the subordinate address space procedure,” on page 99.

• IMS Index Builder will correctly process indexes with non-unique keys.

• This release of IMS Index Builder can process an unlimited number of PSINDEX partitions.

• The following DD statement is no longer supported:
  – SYSPRINT DD

• The following control statements are no longer supported. If they are used, warning messages are issued that the statement is no longer supported, and IMS Index Builder continues processing.
  – ALTER
  – CATLG
  – EXEC
  – NDXIONUFF
  – SORTE15
  – SORTE35
  – SORTID
  – SOROUT
  – SORTSTAT

• IDCAMS processing of the data sets is completed in the initialization and registration phases, before building the processing structures. IIUB053E is introduced to denote IDCAMS processing failure in the data set description block. IIUB100E reason 109 at initialization phase ends if IDCAMS failures have been encountered. In the registration phase, IIUB029E is used.

• DEL/DEF support is integrated to data set IDCAMS processing. See the DDEFPDS DD statement.

• Data set initializations are processed entirely during the initialization phase.

• The index load report has been enhanced. Separate report blocks are provided at data set/partition level, and a summary block is introduced at secondary index/PSINDEX level. The rejected records are reported in the summary block, and for PSINDEXes, the sum of the partition records is provided.

• IMS Index Builder will always use DBRC for HALDB databases. If the INPUT control statement specified DBRC=N, no notification is issued, regardless of the database type. In prior IMS Index Builder versions, notifications were always issued for HALDB databases.

• BLD_SECONDARY specifies that secondary index databases must be built. If the prime database does not have secondary indexes, an error is indicated, and IMS Index Builder stops at the initialization phase. In prior IMS Index Builder versions, an error was not indicated.
What does IMS Index Builder do?

IMS Index Builder streamlines index creation, reorganization, maintenance, and recovery. The tool creates multiple indexes in one job step, and allows you to rebuild IMS indexes, rather than recover or reorganize them using the traditional unload and reload processes used for the primary data store.

IMS Index Builder eliminates the need to image copy indexes, which means faster recovery and reorganization times and that new indexes can be added quickly without reorganizing the primary databases.

For any supported hierarchical direct (HD) database and for HISAM databases, IMS Index Builder scans the existing physical database and builds IMS primary and secondary indexes. For non-partitioned databases, it creates one or more new secondary index databases. A full database reorganization and initial HALDB load are no longer necessary.

IMS Index Builder provides an easy-to-use, one-step procedure for building all IMS index types including primary, secondary, and indirect list data sets.

Product features

Designed for high performance, IMS Index Builder supports partitioned and non-partitioned databases with the following features:

- Provides full support for IMS High Availability Database (HALDB) and for Database Recovery Control (DBRC).
- Integrates with IMS Tools Base IMS Tools Knowledge Base for storing and retrieving SYSPRINT output data.
- Extracts data that is needed for prefix resolution from the DFSURWF1 file and writes it to a new data set, improving the performance of prefix resolution.
- Builds non-PSINDEX IMS secondary indexes by using the DFSURWF1 file as input, which was created by the IMS reorganization reload, high performance load utility, or initial load utility.
- Builds non-PSINDEX IMS secondary indexes by using output file DFSURIDX from the IMS Prefix Resolution utility.
- Builds PSINDEX IMS secondary indexes by using the WFP output files that are generated by IMS High Performance Load.
- Provides a new sequential scan technology for building primary, secondary, and ILDS indexes. Comparisons to prior release functionality show improved elapsed time.
- Uses both parallel sort and parallel load whenever more than one index is being built, reducing the time that is needed to build multiple indexes of a single physical database.
- Applies parallel partition scan to HALDB.
- Builds a copy of the primary index of HIDAM database using the existing primary index as input.
- Creates new primary indexes of HIDAM and PHIDAM databases where input is based on a scan of the physical database without using the existing primary index.
- Builds and/or initializes IMS ILDS, primary, and secondary index data sets.
- Does not assume that ILDS and primary indexes exist and completely rebuilds them.
• Enables you to build the ILDS and primary and secondary indexes in one JOB step.
• Rebuilds selected PSINDEX partitions, and offers support for using Partition Selection Exits (PSE) when rebuilding PSINDEXes.
• Supports secondary indexes with non-unique keys.
• Supports IDCAMS delete and define functionality for data sets being built based on a user-provided procedure library so that no preliminary IDCAMS job steps are required.
• Improves usability by routing system messages from the subordinate address spaces to the main address space.
• Propagates RACF identity to subordinate address spaces so all database and data set access is performed with the same identity as the master address space.
• Issues IMS commands (/DBD, /DBR, and /STA) automatically to prevent updates to the databases while indexes are built and start the databases and indexes when indexes are built.

**Performance benefits**

IMS Index Builder is designed to minimize the elapsed time needed to build one or more secondary index databases. Elapsed time is a primary concern at most installations because the need for high data availability keeps shrinking the batch window.

IMS Index Builder achieves high performance by using parallel processing and by overlapping and integrating processing steps.

Significant performance features of IMS Index Builder are:
• IMS Index Builder uses subordinate address spaces (SAS) for parallel prime data scans and index data sorts.
• Records are loaded into each index database in parallel.
• Records to be sorted are passed to sort as they are read in or generated, which means that the scan steps and the sort steps overlap. When buffered sort is used, the records are not written on disk first, bypassing much of the I/O activity. Striped sort splits the sort process into consecutive steps.
• Run time optimized code (race code) is implemented in the index keys encoding process to reduce CPU time.

**Sorts processing**

IMS Index Builder generates sort control statements and internally calls the sort product that is provided by the user. When buffered sort is applied, IMS Index Builder provides a sort file size estimation.
• When indexes are reused, the size of the existing indexes is taken as the sort file size for the index.
• When indexes are deleted or defined, the allocations are used to estimate the index sort file size, and you can use the INDEX statement to override the estimation.
• For sequential scan internal sorts, the sum of the estimated index sort file sizes is used, and you can use the SORTFSZ control statement to override the estimated value.
IMS Index Builder terminology

IMS Index Builder includes several unique terms that you must understand before you begin to use IMS Index Builder.

Unique terms

Buffered sort
Sort exits are used to process buffers containing index records to be passed to and from sort.

Composed sort
A single sort process is used to handle index records composed of all indexes to rebuild.

DL/I Standard Data Language/I

HALDB High Availability Large Database.

HDAM Hierarchical Direct Access Method.

HIDAM Hierarchical Indexed Direct Access Method database.

Hierarchical scan
Obtains the prime data segments in logical sequence. The internal high performance unload (HPU) subset or the standard data language/I (DL/I) interface can be used. Hierarchical scan is run in MAS.

HISAM Hierarchical Sequential Access Method.

HPU Internal subset of High Performance Unload.

ILDS Indirect List Data Set.

MAS Master address space. The address space where IMS Index Builder receives control.

Mono sort
Single index or PSINDEX partition records are sorted using an E15 exit to obtain data from buffers, and an E35 exit to load the index or partition.

PHDAM Partitioned Hierarchical Direct Access Method.

PHIDAM Partitioned Hierarchical Indexed Direct Access Method.

Race code
Run time optimized machine code.

SAS Subordinate address space.

API SAS Address space that is dedicated to IMS Database Recovery Facility integration.

Scan SAS Address space that is dedicated to sequential scan sort SAS.

Sort SAS Address space that is dedicated to sort.
Sequential scan
Obtains the prime HALDB data segments in physical sequence, by partition, processing in multiple SAS.

Serial scan
Obtains the prime data segments in physical sequence, no parallel processing, runs in MAS.

Stripe
Work file for internal IMS Index Builder use.

Striped sort
Sorts are file oriented, index records are written to stripes to be passed to sort, and sort output is on stripes.

Twinning
When scan and sort run simultaneously in MAS.

Short names used in this information
To make this information easier to read, the following short names are used for product names and function names except where distinctions among them need to be made.

<table>
<thead>
<tr>
<th>Short name</th>
<th>Product and function name</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPIC</td>
<td>The Image Copy function of IMS HP Image Copy</td>
</tr>
<tr>
<td>HPPC</td>
<td>The HASH Check function of IMS HP Pointer Checker</td>
</tr>
<tr>
<td>IMS</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Version 13 Release 1 Database Manager (5635-A04)</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Database Value Unit Edition Version 13 Release 1 (5655-DSM)</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Version 14 Release 1 Database Manager (5635-A05)</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Database Value Unit Edition Version 14 Release 1 (5655-DSE)</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Version 15 Release 1 Database Manager (5635-A06)</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Database Value Unit Edition Version 15 Release 1 (5655-DS5)</td>
</tr>
<tr>
<td>IMS Database</td>
<td>IBM IMS Database Control Suite for z/OS, Version 3 Release 2 (5655-L08)</td>
</tr>
<tr>
<td>Control Suite</td>
<td></td>
</tr>
<tr>
<td>IMS Database</td>
<td>IBM IMS Recovery Solution Pack for z/OS, Version 1 Release 1 or later: IMS Database Recovery Facility (5655-V86, 5655-ISR)</td>
</tr>
<tr>
<td>Recovery Facility</td>
<td></td>
</tr>
<tr>
<td>or DRF</td>
<td></td>
</tr>
<tr>
<td>IMS Database</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td>Reorganization</td>
<td>• IBM IMS Database Reorganization Expert (5655-S35) provided in the following solution packs:</td>
</tr>
<tr>
<td>Expert</td>
<td>– IBM IMS Database Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Utility Solution for z/OS</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Database Reorganization Expert for z/OS, Version 4 Release 1 (5655-S35)</td>
</tr>
<tr>
<td>IMS Database</td>
<td>IBM IMS Database Solution Pack for z/OS, Version 2 Release 1 or later (5655-DSP)</td>
</tr>
<tr>
<td>Solution Pack</td>
<td></td>
</tr>
<tr>
<td>IMS Database</td>
<td>IBM IMS Database Utility Solution for z/OS, Version 1 Release 1 or later (5655-DBU, 5698-DUL)</td>
</tr>
<tr>
<td>Utility Solution</td>
<td></td>
</tr>
<tr>
<td>Short name</td>
<td>Product and function name</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IMS HP Image Copy</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Image Copy (5655-N45) provided in the following solution packs:</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Utility Solution for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Recovery Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Image Copy for z/OS, Version 4 Release 2 (5655-N45)</td>
</tr>
<tr>
<td>IMS HP Load</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Load (5655-M26) provided in the following solution packs:</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Utility Solution for z/OS</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Load for z/OS, Version 2 Release 1 (5655-M26)</td>
</tr>
<tr>
<td>IMS HP Pointer Checker</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Pointer Checker (5655-U09) provided in the following solution packs:</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Utility Solution for z/OS</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Pointer Checker for z/OS, Version 3 Release 1 (5655-U09)</td>
</tr>
<tr>
<td>IMS HP Unload</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Unload (5655-E06) provided in the following solution packs:</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Utility Solution for z/OS</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS High Performance Unload for z/OS, Version 1 Release 2 (5655-E06)</td>
</tr>
<tr>
<td>IMS Index Builder</td>
<td>The generic name for one of the following products:</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Index Builder (5655-R01) provided in the following solution packs:</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Database Utility Solution for z/OS</td>
</tr>
<tr>
<td></td>
<td>– IBM IMS Recovery Solution Pack for z/OS</td>
</tr>
<tr>
<td></td>
<td>• IBM IMS Index Builder for z/OS, Version 3 Release 1 (5655-R01)</td>
</tr>
<tr>
<td>IMS Parallel Reorganization</td>
<td>IBM IMS Parallel Reorganization for z/OS, Version 3 Release 1 and Release 2 (5655-M28)</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This product is replaced by IBM IMS Database Reorganization Expert for z/OS, Version 4 Release 1 (5655-S35)</td>
</tr>
<tr>
<td>IMS Recovery Solution Pack</td>
<td>IBM IMS Recovery Solution Pack for z/OS, Version 1 Release 1 or later (5655-V86, 5655-ISR)</td>
</tr>
<tr>
<td>IMS Tools Base</td>
<td>IBM IMS Tools Base for z/OS, Version 1 Release 5 or later (5655-V93)</td>
</tr>
<tr>
<td>IMS Tools Knowledge Base,</td>
<td>IBM IMS Tools Base for z/OS, Version 1 Release 5 or later: IMS Tools Knowledge Base (5655-V93)</td>
</tr>
<tr>
<td>IMS Tools KB, ITKB</td>
<td></td>
</tr>
<tr>
<td>IMS Tools Online System</td>
<td>IBM IMS Tools Base for z/OS, Version 1 Release 5 or later: IMS Tools Online System Interface (5655-V93)</td>
</tr>
<tr>
<td>Interface, TOSI</td>
<td></td>
</tr>
<tr>
<td>Short name</td>
<td>Product and function name</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------</td>
</tr>
</tbody>
</table>
| IPR        | The generic name for one of the following functions:  
|            | • The IPR Driver of IMS Parallel Reorganization  
|            | • The Smart Reorg utility of IMS Database Reorganization Expert |
IMS Index Builder components

IMS Index Builder is composed of the driver and the common services group. The driver is invoked by the job-step executable, which initiates the IMS Index Builder master address space (MAS), analyzes the input, processes the database control tables, and attaches common services. When the driver receives the messages that indicate the processes are complete, it terminates.

The common services group includes:

**Planner**
The planner service builds the table for the process and passes it to the creator.

**Creator**
The creator service scans the build plan table and creates or attaches processing tasks, which are typically scans and sorts.

**Sort driver**
The sort driver coordinates subordinate address space (SAS) scans and sorts.

**Messages**
The messages service processes informational messages from the components and builds the report.

**Activity monitor**
The activity monitor service monitors the build plan table and stops IMS Index Builder if no activity is indicated.

**Index mergers**
The index mergers service attaches a merger for every index if a striped sort is planned.

**Index loaders**
The index loaders service attaches a loader to every index and loads the index.

**Initializer**
The initializer service processes index initializations.

**Capture merger**
The capture merger service merges the WTO capture files from the subordinate address space and master address space.

**DBRC processor**
The DBRC processor service writes DBRC commands to the DBRC input data set that is to be processed when IMS Index Builder stops.

**Registrar**
The registrar service registers the indexes through the IMS region during the build process.
IMS Index Builder architecture

The IMS Index Builder code is object-oriented. It is based on the message flow between servers that are attached by the driver after a preliminary analysis of the input control statements.

The following figure shows the IMS Index Builder architecture:

![Figure 1. IMS Index Builder architecture](image)

exec iustart

**Driver**

**Registrar**

**Planner**

**Monitor**

**Initializer**

**Sort Driver**

**Image Copy Processor**

**DBRC Processor**

**Capture Merger**

**CREATOR**

**Build Plan Table**

**Indexes**

**Index Loader**

**Index Merger**

**REPORT**

**SCAN SAS**

**SORT SAS**

**Database**
IMS Index Builder business scenarios

You can use IMS Index Builder to address many of your day-to-day business problems.

The following scenarios illustrate how you can use IMS Index Builder to address typical business problems.

**Non-HALDB initial load or reorganization load**

After loading or reloading the prime database, use IMS Index Builder to process the work file (DFSURWF1 or HPSRSIDX) to load the secondary indexes. You can also use IMS Index Builder with a dummy DFSURIDX file to initialize indexes that are not to be loaded yet.

**HALDB reload**

After reloading a HALDB, use IMS Index Builder to reload the PSINDEX partitions and optionally, the primary indexes and ILDS.

**HALDB online reorganization**

After ending online reorganization for certain partitions, use IMS Index Builder to reload PSINDEXes and/or partition primary indexes. Rebuilding ILDS for terminated partitions would create entries for the last reorganization only.

**Individual index reload**

During regular production, an index might be updated extensively, which will decrease performance. In some cases, an index will become corrupted. Therefore, reallocating the index is necessary. Use IMS Index Builder to rebuild the selected index by scanning the database.
IMS tools product integration

IMS Index Builder, fully compatible with standard IMS utilities, must be enabled to share common services for storing and viewing reports. IMS Index Builder also interfaces with other IMS products.

Compatibility with IMS utilities

IMS Index Builder is fully compatible with standard IMS utilities.

It is also compatible with any vendor utility that uses the DFSURWF1 or DFSURIDX files (in their standard format) for input or output.

Dependencies

IMS Index Builder can build PSINDEXes by using the WFP output files that are generated by IMS High Performance Load.

Support for IMS Tools Knowledge Base

IMS Tools Knowledge Base is an IBM IMS Tools Base for z/OS component that provides common services for storing and viewing reports that are generated by other participating IMS Tools products.

To fully participate in the IMS Tools Knowledge Base information management environment, the code for each IMS tool must be enabled to communicate with the IMS Tools Knowledge Base server. An enabled IMS tool can automatically send its generated reports to the IMS Tools Knowledge Base repository. This version of IMS Index Builder is enabled to participate in the IMS Tools Knowledge Base environment.

IMS Index Builder uses the ITKBSRVR statement to interface with IMS Tools Knowledge Base.

For more information about activating IMS Tools Knowledge Base, see "Configuring the IMS Tools Knowledge Base server" on page 31.

Support for IMS Tools Online System Interface

IMS Index Builder supports the IMS Tools Online System Interface of IBM IMS Tools Base for z/OS. By using the interface, IMS Index Builder can automatically issue IMS commands (/DBD, /DBR, and /STA) so that you do not need to issue these IMS commands manually. See "Configuring IMS Tools Online System Interface" on page 32 to configure the IMS Tools Online System Interface.
IMS Index Builder restrictions

Certain restrictions apply to using IMS Index Builder.

IMS Index Builder does not support the following types of indexes:
- Secondary or primary indexes that are DOS compatible
- Shared secondary indexes

Shared secondary indexes

IMS Index Builder does not support shared secondary indexes.

When multiple secondary indexes are in the same database, that database is a shared index database. Although using a shared index database can save some main storage, the disadvantages of doing so generally outweigh the small amount of space that is saved.

Some of the disadvantages of using a shared index database are:
- HALDB does not support shared secondary indexes.
- Performance can decrease when multiple application programs simultaneously use a shared index database.
- Search time is increased.
- Using shared secondary indexes restricts the overall size of a database.
- Maintenance, recovery, and reorganization of a shared index database can decrease performance across all secondary indexes.
- When you reorganize a database that is accessed using a secondary index, IMS automatically builds a new secondary index. All of the other indexes in the shared index database must be copied to the new shared index database.

For more information about shared secondary indexes, see IMS Database Administration.

IMS exit routines

This restriction applies only to HALDBs.

IMS Index Builder calls the following IMS exit routines:
- Secondary Index Database Maintenance exit routine
- Segment edit/compression exit routine

If these routines are called in subordinate address spaces while processing HALDBs, values in some fields in the following IMS control blocks, which are passed by IMS Index Builder, might be incompatible with the IMS control region.
- Partition Specification Table (PST)
- Field Description Blocks (FDB) pointed to by the Physical Segment Description Block (PSDB)

If either of the IMS exit routines refers to the fields in these IMS control blocks, specify a value of 2 for the MAXTASKS control statement so that the routines are called in the master address space.
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.
• 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 Index Builder enable users to:

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

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

Before you can use IMS Index Builder for the first time, you must configure the operating environment. After the configuration and installation are complete, you can activate IMS Index Builder and verify that the installation was successful.

**Prerequisite:** Before you can configure IMS Index Builder, you must complete the SMP/E installation and create an executable SIIULMOD. Instructions are documented in the program directory that is provided with the product.

**Topics:**
- “Hardware and software requirements” on page 26
- “Configuring the operating environment” on page 27
- “Activating and verifying IMS Index Builder installation” on page 28
- “Configuring for DBRC command authorization” on page 30
- “Configuring the IMS Tools Knowledge Base server” on page 31
- “Configuring IMS Tools Online System Interface” on page 32
Hardware and software requirements

Before you configure IMS Index Builder, verify that your environment meets the following minimum hardware and software requirements. IMS Index Builder is installed by using SMP/E and standard RECEIVE, APPLY, and ACCEPT processing.

Hardware requirements

IMS Index Builder operates on any hardware configuration that supports the required versions of IMS.

Software requirements

IMS Index Builder operates in a z/OS environment.

Operating system requirement

IBM z/OS, Version 2 Release 1 (5650-zOS) or later

Installation requirements

IBM IMS Tools Base for z/OS, Version 1 Release 5 (5655-V93) or later

Mandatory operational requirements

IMS Index Builder requires one of the following IMS releases:

• IBM IMS Version 13 Release 1 (5635-A04)
• IBM IMS Database Value Unit Edition Version 13 Release 1 (5655-DSM)
• IBM IMS Version 14 Release 1 (5635-A05)
• IBM IMS Database Value Unit Edition Version 14 Release 1 (5655-DSE)
• IBM IMS Version 15 Release 1 (5635-A06)
• IBM IMS Database Value Unit Edition Version 15 Release 1 (5655-DS5)

Conditional operational requirements

DFSORT that is included in IBM z/OS, Version 2 Release 1 (5650-zOS) or later, or a functionally equivalent sort program

IMS Index Builder conditionally uses the following components of IBM IMS Tools Base for z/OS:

• IMS Tools Knowledge Base, to store reports in a central repository
• IMS Tools Common Services (IMS Tools Online System Interface), to automatically issue IMS commands
  • IMS Tools Common Services (IMS Catalog Interface) in IBM IMS Tools Base for z/OS Version 1 Release 6 or later, to retrieve database information from the IMS catalog
Configuring the operating environment

If you are a first-time user, you must configure the operating environment to run IMS Index Builder. Your system programmer can assist with this configuration after you complete the SMP/E installation and create an executable SIIULMOD.

The configuration that is described in this topic is needed to correctly configure your operating environment to run IMS Index Builder.

Main storage requirements

IMS Index Builder invokes the sort product internally and more than one sort can be attached in the address space. Because fast I/O is used, which requires large buffers, and sort work file requirements can become unexpectedly high, a region size of 250 megabytes is recommended.

Enabling inter-region communication (APF authorization)

The master address space creates subordinate address space environments to scan primary databases, sort the index records that are built, and eventually, write the indexes. The process is coordinated by inter-region communication and requires the master address space to be non-swappable during the entire IMS Index Builder run.

To enable inter-region communication, APF authorizes the STEPLIB concatenation of the main IMS Index Builder job.

If the STEPLIB concatenation for IMS Index Builder is not correctly APF authorized, the IMS Index Builder MAS issues the following message, and processing stops with abend code S047:

IIUB104E APF AUTHORIZATION CHECK FAILED
Activating and verifying IMS Index Builder installation

After you install IMS Index Builder, you must do some tasks to activate it before using it. After activation, you can run an optional installation verification program to ensure that the installation procedure completed successfully.

Activating IMS Index Builder

After you complete the SMP/E installation, activate IMS Index Builder.

About this task

To activate IMS Index Builder, you must configure the environment so that IMS Index Builder can perform the following actions:

- Start some subordinate address spaces for scans and sorts.
- Allocate stripe data sets to pass data between its address spaces; the data includes the WTO messages, the sort messages, the index records, and the duplicate keys. IMS Index Builder allocates one or more stripe data sets in each address space.

Procedure

1. Configure the environment so that IMS Index Builder can start some subordinate address spaces.
   a. Ensure that the target load library SIIULMOD is APF authorized.
      For more information, see “Enabling inter-region communication (APF authorization)” on page 27.
   b. Copy the procedure, IIUBSRT, from the target library, SIIUPROC, to your started task procedure library.
   c. Optional: Specify the job names for the subordinate address spaces.
      For more information, see Chapter 6, “Defining the subordinate address space procedure,” on page 99.
   d. Define RACF authorization to establish security authorization for subordinate address spaces.
      For more information, see “RACF considerations” on page 101.

2. Configure the environment so that IMS Index Builder can allocate stripe data sets.
   a. Optional: Specify the high-level qualifier of stripe data sets. The default is IIU.STRIPE.
      For more information, see the explanation of the IIURDFLT CLASS entry in Chapter 7, “Defining runtime parameters,” on page 103.
   b. Optional: Set up the SMS environment to associate the high-level qualifier of stripe data sets with the SMS-managed volumes.
   c. Define RACF authorization to establish proper access to stripe data sets.
      For more information, see “RACF considerations” on page 101.

3. Configure the environment so that IMS Index Builder can issue DBRC commands and send DBRC API requests.
   For more information, see “Configuring for DBRC command authorization” on page 30.
Verifying IMS Index Builder installation

Before you use IMS Index Builder, it is recommended that you run the installation verification program to verify that the installation and the configurations were successful.

Procedure

1. Run the sample members of the target library SIIUBASE to verify that IMS Index Builder is installed correctly.

   To verify the installation, modify the JCL to apply to your installation before running the programs. Comments in the JCL describe the changes that you must make.

   There are two installation verification programs:

   **IIUIVPF**
   - Verifies full-function database installation

   **IIUIVPH**
   - Verifies HALDB database installation

2. Review the job log. If the job ended with some errors, review the reports and the MVS™ system log.

   It is important to review the MVS system log for messages that are issued before the started task main module receives control to ensure that no first-time run failures have occurred.

   Subordinate address spaces are created by using a started task procedure, which is located in the PROCLIB concatenation. The WTO messages for subordinate address spaces are captured by IMS Index Builder and included in the WTO capture summary list on IIUSNAP. IMS Index Builder includes as much data as possible in the report. However, messages that are issued by the operating system before the started task main module receives control cannot be intercepted, and remain in the MVS system log. In most cases, these are first-time run failures that are caused by improper started task procedures. After successfully running once, procedures must remain stable. Therefore, review the MVS system log while IMS Index Builder is running for the first time.
Configuring for DBRC command authorization

If IMS DBRC command and API request authorization support with RACF is used, the user ID of the master address space must have authorization to access resources so that IMS Index Builder can issue DBRC commands and DBRC API requests.

Procedure

Authorize the user ID for the following resources:
For DBRC commands:
• hlq.CHANGE.DB.dbname
• hlq.CHANGE.DBDS.dbname
• hlq.NOTIFY.REORG.dbname
• hlq.NOTIFY.UIC.dbname (if VIC YES is specified)
For DBRC API requests:
• hlq.STDBRC
• hlq.LIST.RECON
• hlq.LIST.DB.dbname

Where:

hlq  The high-level qualifier of the resource name.
dbname  The name of the prime database, partition, or index.

For more information about authorizing user IDs, see the topic "Security for DBRC commands and API requests" in IMS System Administration.
Configuring the IMS Tools Knowledge Base server

You can use IMS Tools Knowledge Base to store IMS Index Builder reports in the output repository.

Procedure

To write reports to the output repository of IMS Tools Knowledge Base, complete the following steps:

1. Install the IMS Tools Base IMS Tools Knowledge Base product.
2. Set up an IMS Tools Knowledge Base server.
3. Register IMS Index Builder to the registry of the IMS Tools Knowledge Base server by following the instructions in the topic "Registering IMS Tools products by using the definition table of the product" in the Tools Base Configuration Guide for IMS. The product definition table for registering IMS Index Builder is IIUTITKB.
4. Follow the instructions in the topic "Listing registered products and reports" in the Tools Base Configuration Guide for IMS to list the registered products and reports, and ensure that IMS Index Builder (product ID is DX) has been added.
5. If needed, add the definitions for the RECON data sets.

What to do next

After you complete these steps, IMS Index Builder reports can be stored in the Output repository of IMS Tools Knowledge Base. To store reports in the repository, you must also change the IMS Index Builder JCL as follows:

- Include the IMS Tools Knowledge Base load library by one of the following ways:
  - Add the IMS Tools Knowledge Base load library to the STEPLIB DD concatenation.
  - Specify the IMS Tools Knowledge Base load library by coding the ITKBLOAD control statement or specifying it as the default runtime parameter.
- Specify the IMS Tools Knowledge Base server name by coding the ITKBSRVR control statement or specifying it as the default runtime parameter.

Before you run IMS Index Builder jobs, always confirm that the IMS Tools Knowledge Base server is initialized.
Configuring IMS Tools Online System Interface

IMS Index Builder can internally call IMS Tools Online System Interface to issue IMS commands (/DBD, /DBR, and /STA). The /DBD and /DBR commands prevent transactions or programs from updating the databases while the indexes are built and the /STA command restarts the databases after the indexes are built.

Procedure

For instructions for configuring IMS Tools Online System Interface, see the Tools Base IMS Tools Common Services User’s Guide.

What to do next

When the IMS Tools Online System Interface is configured, add libraries and control statements for enabling the IMS Tools Online System Interface to IMS Index Builder JCL. See “Issuing the IMS commands automatically” on page 61.
Chapter 3. Migrating from IMS Index Builder Version 2.3 to Version 3.1

IMS Index Builder Version 3.1 can execute IMS Index Builder Version 2.3 JCL.

**About this task**

When migrating from IMS Index Builder Version 2.3, several actions are required to best utilize IMS Index Builder Version 3.1 functionality.

To effectively migrate from IMS Index Builder Version 2.3 to Version 3.1, complete the following steps:

**Procedure**

1. Modify the IMS Index Builder JCL.
   a. Concatenate the load libraries for related products to STEPLIB.
      If applicable, concatenate IMS Tools Knowledge Base load library to STEPLIB.
   b. Remove the SORTFSZ control statements.

      **Important:** The function of the SORTFSZ statement has changed. In IMS Index Builder Version 3.1, SORTFSZ is used to support heavy sort loads. SORTFSZ will also trigger striped sort mode.
   c. Optional: If you want to provide the sort file size for index sorts, use the INDEX statement.

      **Important:** For PSINDEX, the values for sort file sizes are valid for the whole index and not for the index partition as in IMS Index Builder Version 2.3.
   d. Optional: If you want to add the sort parameters for the sort program that runs in the master address space and in the subordinate address spaces, add the SORTOPT control statement.
   e. Optional: If you want IMS Index Builder Version 3.1 to issue the DBRC NOTIFY.UIC command to turn off the IC NEEDED flag, add the VIC YES control statement.
      For more information, see "VIC control statement” on page 94.
   f. Remove the ALTER, EXEC, NDXIOBF, SORTE15, SORTID, SORTOUT, and SORTSTAT control statements.
      These statements are provided in the IMS Index Builder JCL Version 3.1 by default.
   g. Optional: If you want to store IMS Index Builder reports in the output repository of IMS Tools Knowledge Base, add the ITKBSRVR control statement.
      The server name can be specified in the JCL or in the runtime parameters module.
      For more information, see Chapter 5, “JCL and input control statements,” on page 65.

2. Define the subordinate address space procedure.
   Include the IIUBSRT started task procedure in PROCLIB.
Notes:

a. Do not use the IMS Index Builder Version 2.3 started task procedures because Version 2.3 procedures do not work with IMS Index Builder Version 3.1.

b. In IMS Index Builder Version 3.1, the started tasks are activated by internally assigned job names. The started tasks procedure name is not used.

c. If you want to add the sort parameters for the sort program that runs in the subordinate address space, specify the DFSPARM DD statement in the IIUBSRT started task procedure or specify the SORTOPT control statement in the IMS Index Builder JCL.

For more information, see Chapter 6, “Defining the subordinate address space procedure,” on page 99.

3. Optional: Define the runtime parameters.

   By defining the runtime parameters, you can override the default procedure name and the default job names that are used when starting subordinate address spaces, and the default high-level qualifier for stripe data sets.

   For more information about runtime parameters, see Chapter 7, “Defining runtime parameters,” on page 103.

4. Optional: Specify the scan methods.

   IMS Index Builder Version 3.1 has adopted a new scan technology that runs several additional scan tasks and, by default, this scan technology is used. You can specify to use the scan methods that were used in IMS Index Builder Version 2.3. For more information, see "Selecting the scan method" on page 59.

5. Define RACF authorization.

   a. Establish security authorization for subordinate address spaces.

   b. Establish proper access to stripe data sets.

   See "RACF considerations" on page 101 for more information.

6. Optional: Configure the IMS Tools Knowledge Base server.

   For more information, see "Configuring the IMS Tools Knowledge Base server" on page 31.
Part 2. Using IMS Index Builder

IMS Index Builder enables you to build (or rebuild) IMS secondary indexes, Hierarchical Indexed Direct Access Method (HIDAM) primary indexes, and Indirect List Data Sets (ILDS).

Topics:

- Chapter 4, “IMS Index Builder functions,” on page 37
- Chapter 5, “JCL and input control statements,” on page 65
- Chapter 6, “Defining the subordinate address space procedure,” on page 99
- Chapter 7, “Defining runtime parameters,” on page 103
- Chapter 8, “Index records with duplicate keys,” on page 111
Chapter 4. IMS Index Builder functions

For any supported hierarchical direct (HD) and HISAM databases, IMS Index Builder scans the existing physical database and rebuilds IMS secondary indexes or creates one or more new secondary index databases. This process removes the need for a full database reorganization.

Topics:

- “Creating secondary indexes” on page 38
- “Building a HIDAM primary index” on page 52
- “Building PHIDAM partition primary indexes” on page 56
- “Building HALDB partition ILDS” on page 57
- “Building indexes and ILDSs for IMS catalog databases” on page 58
- “Selecting the scan method” on page 59
- “Issuing the IMS commands automatically” on page 61
- “Using the IMS catalog” on page 63
Creating secondary indexes

Use IMS Index Builder to create new secondary indexes without a full reorganization of the physical database.

Build secondary indexes of a physical IMS database for one of the following purposes:

- To recover one or more damaged secondary indexes and avoid a full database reorganization.
- To avoid taking image copies of secondary indexes. Instead, you can use IMS Index Builder to build indexes after recovering the physical IMS database. In this case, the full recovery of an IMS database with secondary indexes becomes a two-step process as follows:
  - Recover the physical IMS database
  - Use IMS Index Builder to build the secondary index databases
- After reloading a HALDB or in the course of online reorganization.

The indexes to be built must be either reusable or deleted/defined by IDCAMS. Because IMS Index Builder rebuilds indexes in VSAM load mode by the control interval (CI), the free space value in the DEFINE CLUSTER command must be low. IMS Index Builder will correctly process indexes with non-unique keys.

IMS Index Builder performs the following functions to build the secondary indexes:

- Analyzes the control statements, the DBD library, the RECONs, and the physical database, and plans the index build process; initiates WTO capture on the master address space and initiates IMS Tools Knowledge Base.
- Starts tasks to scan the prime database. Initiates WTO capture for every scan address space that is created.
- Starts tasks to sort the index records that are built. Initiates WTO capture for every sort address space that is created.
- When scans are finished, signals the end of input to all sort tasks. Stops WTO capture for scan address spaces.
- When sorts are finished, initiates the index load process. Stops WTO capture for sort address spaces.
- Initializes empty indexes and indexes that contain errors. ILDS initialization is included if HALDB.
- Sends messages to IMS Index Builder output and IMS Tools Knowledge Base and reports the progress of the process and critical control events.
- Ensures that all scan and sort tasks that are started by IMS Index Builder end properly.
- Notifies DBRC RECON for indexes that are successfully built.
- Stops WTO capture for the master address space and creates the WTO capture report.
- Ends the IMS Index Builder main task with the relevant return code.
The following figure shows the functions that IMS Index Builder performs when building secondary indexes:

![Diagram showing IMS Index Builder functions]

*Figure 2. Building secondary indexes*
Creating new secondary indexes

Use IMS Index Builder to create new secondary indexes without a full reorganization of the physical database.

About this task

You can create a new secondary index only when the segments in the physical database are not changed and only indexes are added.

Restriction:
The create secondary index function is available only when DBRC is off. HALDB are not supported.

Requirement:
If you are creating a new secondary index with a target segment that is not a root segment and is not the target of an already existing secondary index, you must use the IMS Reorganization Reload utility to create the physical parent pointer.

Procedure

1. Define new DBDs for the physical database and for the new indexes.
2. Put the resulting DBDs into a separate library because the old DBD for the physical database is needed for the scan.
3. Run IMS Index Builder against the database using the IBSCAN keyword on the INPUT statement. Use DD name IMS for the library containing the old DBD for the physical database. Use DD name IMSALT for the library containing the new DBDs for the physical database and the new indexes. IMS Index Builder always checks whether DD name IMSALT was defined. If so, it reads all the DBDs needed to build the secondary indexes from DD name IMSALT instead of using DD name IMS. If IMSALT was not specified, it uses the IMS DD name statement, which must point to the new DBDs.
4. After IMS Index Builder has successfully completed, copy the new DBDs into the regular DBD library.

Example

In the example that is shown in the following figure, a new index called TSTNDX5 is added to database TESTDBD1. Dynamic allocation is used for the database DSGs and the primary index. The TSTNDX5 index is allocated using DD name TSTNDX5.

Before executing IMS Index Builder, the DBD for index TSTNDX5 and the new DBD for database TESTDBD1 must be generated and stored in the first library of the concatenation for DD name IMSALT. The old DBD for database TESTDBD1 must not be deleted until IMS Index Builder has successfully completed.
Creating a New Secondary Index to an Existing Database

With IMS IB V3 R1 (UG FIGURE 03) IIUSMP02

Provide values for the following variables:

- SET SIIULMOD= Index Builder load library
- SET SHKTLOAD= ITKB load library
- SET RESLIB= IMS RESLIB
- SET DBDLIB= DBD library
- SET ADDBLIB= ALT DBD library

NOTE: The Index Builder load library and all other datasets concatenated to STEPLIB must be APF authorized.

Figure 3. Example JCL for creating a new secondary index to an existing database
DFSURWF1 and HPSRSIDX output files

Use IMS Index Builder to build secondary indexes as part of the standard IMS reorganization process. IMS Index Builder uses either output from the IMS reorganization reload utility or output from the initial database load as input. This output file is normally named DFSURWF1. HPSURIDX is the output file that is created by High Performance Load and contains the same content as DFSURWF1.

These files contain information to generate secondary indexes and information to perform prefix resolution. IMS Index Builder uses only the information that is needed to generate secondary indexes.

When generating secondary indexes, instead of scanning the prime database, IMS Index Builder reads records from DFSURWF1 and passes the records to the appropriate sort task as shown in the following figure:

DFSURWF1 is not applicable when building a partitioned secondary index (PSINDEX).
Creating input for fast prefix resolution (split function)

When IMS Index Builder builds secondary indexes using the DFSURWF1 file as input, the split function feature can be used to improve the performance of prefix resolution.

About this task

When IMS Index Builder reaches the end of DFSURWF1, it closes the output file where the prefix resolution records were written and deallocates it so that it is immediately ready for prefix resolution processing.

Because IMS Index Builder reaches the end of the DFSURWF1 file in the early stage of processing, data that is needed for prefix resolution is quickly available for processing. IMS Index Builder issues message IIUB001I to notify the operator that data for prefix resolution is now available (see Figure 5 on page 44).

DFSURWF1 contains records to build secondary indexes and to perform prefix resolution. To reduce the size of the sort that is needed to do the prefix resolution, specify an optional output file to which all DFSURWF1 records (needed by prefix resolution only) will be written as they are read in.

This step is not necessary if you do not need prefix resolution.

IMS Index Builder uses the split function to improve prefix resolution performance by reading data from the DFSURWF1 or HPSRSIDX file, as shown in the following figure:
DFSURWF1 is not applicable when building a partitioned secondary index (PSINDEX).

**Procedure**

To use the split function, include an OUTPUT statement in the IIUIN input control file and an output DD statement in the JCL. See the following figure for an example of how to build a secondary index by using DFSURWF1 and HPSRSIDX as input:
You can use IMS Index Builder to build secondary indexes using the output (normally referred to as the DFSURIDX file) from prefix resolution as input. This feature is provided for compatibility only and should not be used to improve performance. When prefix resolution is needed, IMS Index Builder should be used with the DFSURWF1 input file combined with the split function. A sort that is performed by prefix resolution is significantly slower than a parallel sort that is performed by IMS Index Builder.

**About this task**

When building secondary indexes, IMS Index Builder reads data from the DFSURIDX file and loads the records into the secondary index databases in sequence, as shown in the following figure.

---

**Building secondary indexes using DFSURIDX as input**

You can use IMS Index Builder to build secondary indexes using the output (normally referred to as the DFSURIDX file) from prefix resolution as input. This feature is provided for compatibility only and should not be used to improve performance. When prefix resolution is needed, IMS Index Builder should be used with the DFSURWF1 input file combined with the split function. A sort that is performed by prefix resolution is significantly slower than a parallel sort that is performed by IMS Index Builder.

**Figure 6. Example JCL for building a secondary by index using DFSURWF1 as input**

```plaintext
//IIUBLD JOB &SYSUID,MSGCLASS=H,REGION=233M,TIME=1339,
// RESTART=*,CLASS=A,NOTIFY=&SYSUID
//******************************************************************************
// Building Secondary Indexes Using DFSURWF1/HPREFIXIDX
//******************************************************************************
// Provide values for the following variables:
// SET SIIULMOD= Index Builder load library
// SET SHKTLOAD= ITKB load library
// SET RESLIB= IMS RESLIB
// SET DBDLIB= DBD library
// SET RECON1= Recon1 data set name
// SET RECON2= Recon2 data set name
// SET RECON3= Recon3 data set name
// NOTE: The Index Builder load library and all other datasets
// concatenated to STEPLIB must be APF authorized.
//******************************************************************************
//IIUBSCN EXEC PGM=IIUSTART
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD
// DD DISP=SHR,DSN=&RESLIB
// DD DISP=SHR,DSN=&SHKTLOAD
// IMS DD DISP=SHR,DSN=&DBDLIB
// RECON1 DD DISP=SHR,DSN=&RECON1
// RECON2 DD DISP=SHR,DSN=&RECON2
// RECON3 DD DISP=SHR,DSN=&RECON3
//IIUPRINT DD SYSOUT=*//IIUCAPT DD SYSOUT=*//IIUSNAP DD SYSOUT=*//IIUSOUT DD SYSOUT=*//IIUIN DD * PROC BLD_ALL,TESTDBD1 INPUT DFSURWF1 /* DFSURWF1 DD DISP=SHR,DSN=MYTEST.TESTDBD1.DFSURWF1 TESTDB1P DD DISP=SHR,DSN=MYTEST.TESTDB1.HIDAM.OSAM TESTDB1X DD DISP=SHR,DSN=MYTEST.TESTDB1.HIDAM.INDEX TESTDB1Y DD DISP=SHR,DSN=MYTEST.TSTNDX1.SECINDEX TESTDB2Y DD DISP=SHR,DSN=MYTEST.TSTNDX2.SECINDEX TESTDB3Y DD DISP=SHR,DSN=MYTEST.TSTNDX3.SECINDEX TESTDB4Y DD DISP=SHR,DSN=MYTEST.TSTNDX4.SECINDEX TESTDB5Y DD DISP=SHR,DSN=MYTEST.TSTNDX5.SECINDEX
```
**Procedure**

To build a secondary index by using DFSURIDX as input, define the JCL using the following example:
Figure 8. Example JCL for building a secondary index using DFSURIDX as input
Building partitioned secondary indexes using HPSRWFP as input

You can use IMS Index Builder to build partitioned secondary indexes (PSINDEXes) using the output from IMS High Performance Load as input. This feature improves performance by eliminating scanning of the entire target HALDB in the Index Builder job step.

About this task

When building PSINDEXes, IMS Index Builder reads data from the WFP files that were generated by IMS High Performance Load.

Procedure

To build partitioned secondary indexes by using HPSRWFP as input, define the JCL using the following example:

```
//IIUUBLD JOB &SYSUID,MSGCLASS=H, // RESTART=*,CLASS=A,NOTIFY=&SYSUID  
//***************************************************************************  
//* BUILD PSINDEX USING HPSRWFP FILE  
***************************************************************************  
//* Provide values for the following variables:  
// SET SIIULMOD= Index Builder load library  
// SET RESLIB= IMS RESLIB  
// SET MDALIB= IMS MDALIB  
// SET DBDLIB= DBD library  
// SET DDEFPDS= IDCAMS commands library  
//* NOTE: The Index Builder load library and all other datasets  
//* concatenated to STEPLIB must be APF authorized.  
//***************************************************************************  
//IDXBUILD EXEC PGM=IIUSTART  
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD  
// IMS DD DISP=SHR,DSN=&RESLIB  
// IMSDDALIB DD DISP=SHR,DSN=&MDALIB  
// DDEFPDS DD DISP=SHR,DSN=&DDEFPDS  
//SYSUDUMP DD SYSDUMP=A  
//IIUIN DD *  
// PROC BLD_SECONDARY,HMPORDP1,ALL  
// INPUT HPSRWFP,DBRC=Y  
// WFPHELQ IMSTOOLS.WORK.HMPORDP1  
// WFPDEL YES  
// MAXTASKS 2,PSORT  
//  
```

Figure 9. Example JCL for building PSINDEXes using HPSRWFP as input

Tips:

- The DD statements for the WFP files are not needed. The data sets are allocated dynamically.
- The optional WFPDEL YES statement specifies to delete the WFP files after the indexes are loaded successfully.
- The optional MAXTASKS 2,PSORT statement improves the performance of index sort processing.
This example job requires WFP files for all HALDB partitions. If IMS High Performance Load processes some HALDB partitions and generates WFP files only for those partitions, you must add the UPDATE option for the PROC statement as follows:

PROC BLD_SECONDARY,HMPORDP1,ALL,UPDATE
Initializing secondary indexes

You can use IMS Index Builder to initialize the primary index, one or more secondary indexes, and the ILDS by using an empty or dummy DFSURIDX file as input.

About this task

When you use an empty or dummy DFSURIDX file as input, the load is skipped and all data sets will be initialized and marked ready for update processing. Note that IMS Index Builder will automatically initialize the indexes if the result of rebuilding is an empty data set.

For ILDSs, you can use the control statement ILDS INITONLY. See “ILDS control statement” on page 80.

Procedure

To initialize secondary indexes, use an empty or dummy DFSURIDX file as input, and define the JCL by using the following example:

```plaintext
//IIUBBLD JOB &SYSUID,MSGCLASS=H,REGION=33M,TIME=1339,
//   RESTART=*,CLASS=A,NOTIFY=&SYSUID
polator*********************************************************************
//INITIALIZING INDEX DATA SETS WITH IMS IB V3 R1
** IIUSMP05 *
��lor**************************************************************************
** Provide values for the following variables:
** SAR SIULMOD= Index Builder load library
** SET SHKTLOAD= IKB load library
** SET RESLIB= IMS RESLIB
** SET MDALIB= IMS MDALIB
** SET DBOLIB= DBD library
** SET RECON1= Recon1 data set name
** SET RECON2= Recon2 data set name
** SET RECON3= Recon3 data set name
** NOTE: The Index Builder load library and all other datasets
** concatenated to STEPLIB must be APF authorized.
polator*********************************************************************
//IIUBSCN EXEC PGM=IIUSTART
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD
//   DD DISP=SHR,DSN=&SHKTLOAD
//   DD DISP=SHR,DSN=&RESLIB
//   DD DISP=SHR,DSN=&MDALIB
//   DD DISP=SHR,DSN=&DBOLIB
//RECON1 DD DISP=SHR,DSN=&RECON1
//RECON2 DD DISP=SHR,DSN=&RECON2
//RECON3 DD DISP=SHR,DSN=&RECON3
//IIUPRINT DD SYSOUT=
//IIUCAPT DD SYSOUT=
//IIUSNAP DD SYSOUT=
//IIUSOUT DD SYSOUT=
//IIUIUN DD *
   PROC BLD_ALL,TESTDBH1
   INPUT DFSURIDX,DBRC=N
/*
//DFSURIDX DD DUMMY
��lor**************************************************************************
```

Figure 10. Example JCL for initializing data sets
Reorganizing and recovering secondary indexes

If you are planning to use IMS Index Builder for standard recovery of secondary indexes, consider changing the skeletal JCL used by DBRC for recovery. Skeletal JCL execution members are stored in IMS PROCLIB, and they are generated as part of IMS generation.

Procedure

To create JCL to recover indexes, issue the DBRC GENJCL.RECOV command. DBRC generates the JCL based on the skeletal JCL that is stored in IMS PROCLIB and the current data in the RECON data sets.

You can also use the following IMS database tools for recovering and rebuilding indexes:

- IMS Database Control Suite creates skeletal JCL for rebuilding indexes for recovery
- IMS Parallel Reorganization and IMS Database Reorganization Expert integrate with IMS Index Builder to rebuild non-HALDB secondary indexes
- IMS Database Recovery Facility integrates with IMS Index Builder to rebuild indexes during recovery
Building a HIDAM primary index

When the primary index of a HIDAM database has been broken, you can use IMS Index Builder to rebuild it by scanning the prime database or partition, if HALDB.

About this task

If sequential scan is applicable, the primary index build is included in the secondary index step. If hierarchical scan must be applied to rebuild the secondary indexes, preliminary sequential scan of the prime database is needed to rebuild the primary index.

The indexes to be built must be either reusable or deleted/defined by IDCAMS. Because IMS Index Builder rebuilds indexes in VSAM load mode by the control interval (CI), the free space value in the DEFINE CLUSTER command must be low.

Procedure

To build a HIDAM primary index, define the JCL using the following example:
IIUBBLD  JOB &SYSUID,MSGCLASS=H,REGION=233M,TIME=1339,
//  RESTART=*,CLASS=A,NOTIFY=&SYSUID
//*********************************************************************
//* Building a HIDAM Primary Index with IMS IB V3 R1  IIUSMP06 *
//*********************************************************************
//* Provide values for the following variables:
// * SET SIIULMOD= Index Builder load library
// * SET SHKTLOAD= ITKB load library
// * SET RESLIB= IMS RESLIB
// * SET MDALIB= IMS MDALIB
// * SET DBDLIB= DBD library
// * SET RECON1= Recon1 data set name
// * SET RECON2= Recon2 data set name
// * SET RECON3= Recon3 data set name
//* NOTE: The Index Builder load library and all other datasets
//* concatenated to STEPLIB must be APF authorized.
//*********************************************************************
IIUBSCN  EXEC PGM=IIUSTART
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD
// DD DISP=SHR,DSN=&SHKTLOAD
// DD DISP=SHR,DSN=&RESLIB
// DD DISP=SHR,DSN=&MDALIB
// IMS DD DISP=SHR,DSN=&DBDLIB
// RECON1 DD DISP=SHR,DSN=&RECON1
// RECON2 DD DISP=SHR,DSN=&RECON2
// RECON3 DD DISP=SHR,DSN=&RECON3
// IIUPRINT DD SYSOUT=* 
// IIOCPT DD SYSOUT=* 
// IIUSNAP DD SYSOUT=* 
// IIUSOUT DD SYSOUT=* 
//IIUIN DD * 
//  PROC BLD_PRIMARY,TSTDATA 
// INPUT IBSCAN
/*

Figure 12. Example JCL for building a HIDAM primary index
Building a HIDAM primary index on an alternate primary index data set

When the primary index of a HIDAM database is intact, it can be used to build a new primary index. The new index is written to the alternate index data set, which has been previously defined.

About this task

Because the alternate primary index and the existing index must coexist while IMS Index Builder is running, the job contains the following three steps:

- Using IDCAMS, defines the new primary index data set with the same attributes as the existing primary index but with a different name.
- Builds the new primary index using IMS Index Builder.
- Uses IDCAMS to delete the old primary index data set and invoke the ALTER command to change the data set name of the new index to the data set name of the old index.

Procedure

1. Make sure that the alternate (new) and existing primary indexes are both cataloged in the same catalog. Otherwise, problems will be encountered with the ALTER command.
2. Define the JCL using the following example:

```plaintext
//IIUBBLD JOB &SYSUID,MSGCLASS=H,REGION=233M,TIME=1339,
// RESTART=*,CLASS=A,NOTIFY=&SYSUID
//*********************************************************************
//**** Building a HIDAM Primary Index on an Alternate Primary Index  *
// Data Set with IMS IB V3 R1 IIUSMP07 *
//*********************************************************************
//**** Provide values for the following variables:                *
// SET SIIULMOD= Index Builder load library
// SET SHKTLOAD= ITKB load library
// SET RESLIB= IMS RESLIB
// SET MDALIB= IMS MDALIB
// SET DBLIB= DBD library
// SET RECON1= Recon1 data set name
// SET RECON2= Recon2 data set name
// SET RECON3= Recon3 data set name
//* NOTE: The Index Builder load library and all other datasets
//* concatenated to STEPLIB must be APF authorized.
//*********************************************************************
```

Figure 13. Example JCL for building a HIDAM primary index on an alternate primary index data set (Part 1 of 2)
Invoke IDCAMS to Create Temporary Index and Copy First Record of Old Index

----

//DEFINE EXEC PGM=IDCAMS,COND=(0,LT)
//SYSPRINT DD SYSOUT=
//SYSIN DD *

  DEFINE CLUSTER (NAME(MYTEST.TEST.NEWNDX0) - MODEL(MYTEST.TESTDB1.HIDAM.INDEX))

----

Invoke IMS IB V3 R1 to Build the Primary Index

----

//IIUBSCN EXEC PGM=IIUSTART,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD
// DD DISP=SHR,DSN=&SHKTLOAD
// DD DISP=SHR,DSN=&RESLIB
// DD DISP=SHR,DSN=&MDALIB
// IMS DD DISP=SHR,DSN=&DBDLIB
// RECON1 DD DISP=SHR,DSN=&RECON1
// RECON2 DD DISP=SHR,DSN=&RECON2
// RECON3 DD DISP=SHR,DSN=&RECON3
// IIUPRINT DD SYSOUT=
// IIUCAPT DD SYSOUT=
// IIUSNAP DD SYSOUT=
// IIUSOUT DD SYSOUT=
// IIUIN DD *

PROC BLD_PRIMARY,TESTDBD1
 INPUT IBSCAN,DBRC=N
 OUTPUT DATOUT
/*
//TESTDB1P DD DISP=SHR,DSN=MYTEST.TESTDB1.HIDAM.OSAM
//TESTDB1X DD DISP=SHR,DSN=MYTEST.TESTDB1.HIDAM_INDEX
//DATOUT DD DISP=OLD,DSN=MYTEST.TEST.NEWNDX0

----

Invoke IDCAMS to Replace The Old Index with the New One

----

//REPLACE EXEC PGM=IDCAMS,COND=(0,LT)
//SYSPRINT DD SYSOUT=
//SYSIN DD *

  DELETE MYTEST.TESTDB1.HIDAM.INDEX
  IF MAXCC ^= 0 THEN -
    CANCEL
    ALTER MYTEST.TEST.NEWNDX0 -
     NEWNAME(MYTEST.TESTDB1.HIDAM.INDEX)
/*

----

Figure 14. Example JCL for building a HIDAM primary index on an alternate primary index data set (Part 2 of 2)
Building PHIDAM partition primary indexes

PHIDAM HALDB partition primary indexes are rebuilt the same way a HIDAM index is built for all partitions or for selected partitions.

About this task

The DATAOUT DD statement is not applicable for HALDB, alternate primary indexes cannot be built. Sequential scan is performed by partition, and the primary indexes are loaded as soon as the partition scan completes.

Procedure

To build a PHIDAM partition primary index, define the JCL, using the following example:

```plaintext
//IIUBBLD JOB &SYSUID,MSGCLASS=H,REGION=233M,TIME=1339, // RESTART=*,CLASS=A,NOTIFY=&SYSUID //***************************************************************************** //* Building PHIDAM Partition Primary Indexes * //* with IMS IB V3 R1 IIUSMP08 * //***************************************************************************** //** Provide values for the following variables: //** SET SIIULMOD= Index Builder load library //** SET SHKTLOAD= ITKB load library //** SET RESLIB= IMS RESLIB //** SET MDALIB= IMS MDALIB //** SET DBDLIB= DBD library //** SET RECON1= Recon1 data set name //** SET RECON2= Recon2 data set name //** SET RECON3= Recon3 data set name //** NOTE: The Index Builder load library and all other datasets //** concatenated to STEPLIB must be APF authorized. //***************************************************************************** //IIUBSCN EXEC PGM=IIUSTART //STEPLIB DD DISP=SHR,DSN=&SIIULMOD // DD DISP=SHR,DSN=&SHKTLOAD // DD DISP=SHR,DSN=&RESLIB // DD DISP=SHR,DSN=&MDALIB // IMS DD DISP=SHR,DSN=&DBDLIB //RECON1 DD DISP=SHR,DSN=&RECON1 //RECON2 DD DISP=SHR,DSN=&RECON2 //RECON3 DD DISP=SHR,DSN=&RECON3 //IIUPRINT DD SYSOUT=* //IIUCAPT DD SYSOUT=* //IIUSNAP DD SYSOUT=* //IIUSOUT DD SYSOUT=* //IIUIN DD * // PROC BLD_PRIMARY,TESTDBH1 INPUT IBSCAN,DBRC=Y PARTNAME BH1PTNA BH1PTND /* /*
```

Figure 15. Building a PHIDAM partition primary index
Building HALDB partition ILDS

PHIDAM HALDB partition ILDSs are rebuilt in an analogous way as building primary indexes. ILDS of all or selected partitions can be rebuilt.

About this task

Sequential scan is performed by partition and the ILDSs are loaded as soon as the partition scan completes.

The indexes to be built must be either reusable or deleted/defined by IDCAMS. Because IMS Index Builder rebuilds indexes in VSAM load mode by the control interval (CI), the free space value in the DEFINE CLUSTER command must be low.

Procedure

To build a HALDB partition ILDS, define the JCL using the following example.

```plaintext
//IIUBLD  JOB &SYSUID,MSGCLASS=H,REGION=233M,TIME=1339,
 //       RESTART=*,CLASS=A,NOTIFY=&SYSUID
//****************************************************************************
// Building Partition ILDS with IMS IB V3 R1 IIUSMP09 *
//****************************************************************************
// Provide values for the following variables:
// SET SIIULMOD= Index Builder load library
// SET SHKTLOAD= ITKB load library
// SET RESLIB= IMS RESLIB
// SET MDALIB= IMS MDALIB
// SET DBDLIB= DBD library
// SET RECON1= Recon1 data set name
// SET RECON2= Recon2 data set name
// SET RECON3= Recon3 data set name
//* NOTE: The Index Builder load library and all other datasets
//* concatenated to STEPLIB must be APF authorized.
//****************************************************************************
//IIUBSCN EXEC PGM=IIUSTART
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD
// DD DISP=SHR,DSN=&SHKTLOAD
// DD DISP=SHR,DSN=&RESLIB
// DD DISP=SHR,DSN=&MDALIB
// DD DISP=SHR,DSN=&DBDLIB
// DD DISP=SHR,DSN=&RECON1
// DD DISP=SHR,DSN=&RECON2
// DD DISP=SHR,DSN=&RECON3
//IIUPRINT DD SYSOUT**
//IIUCAPT DD SYSOUT**
//IIUSNAP DD SYSOUT**
//IIUSOUT DD SYSOUT**
//IIUIN DD *
// PROC BLD_ILDS,TESTDBH2
// INPUT IBSCAN,DBRC=Y
//*
//*

Figure 16. Building HALDB partition ILDS
```
Building indexes and ILDSs for IMS catalog databases

IMS Index Builder treats an IMS catalog database as a PHIDAM database. Primary indexes, secondary indexes, and indirect list data sets (ILDSs) for IMS catalog databases are rebuilt the same way PHIDAM indexes are built.

Procedure

When building IMS catalog indexes and ILDSs, the following JCL and control statement requirements might apply depending on the usage scenario:

Place the IMS Catalog Definition exit routine (DFS3CDX0) in the STEPLIB DD concatenation in the following cases:

- If the IMS catalog database partitions are defined by using the IMS Catalog Partition Definition Data Set utility (DFS3UCD0) instead of by using DBRC.
- If an IMS catalog alias name other than ALIAS=DFSC is defined.

If the IMS Catalog Partition Definition Data Set utility (DFS3UCD0) is used, specify the statements as follows:

- Specify DBRC=N on the INPUT control statement.
- If the MDA member for the HALDB partition definition data set (DFSHDBSC) is used, specify the library on the STEPLIB DD statement, not on the IMSDALIB DD statement.
Selecting the scan method

IMS Index Builder provides a sequential scan technology for building primary, secondary, and ILDS indexes, and this sequential scan technology is used as the default scan method. However, you can change the scan method to apply the scan method that was used in IMS Index Builder Version 2.3.

About this task

When you build a primary index and secondary indexes together in IMS Index Builder Version 2.3, the primary database was scanned twice, once by using the sequential scan method and once by using the hierarchical HPU scan method. However, in this version of IMS Index Builder, the primary database is scanned only once by using the sequential scan method, which reduces the elapsed time.

When the sequential scan method is used to build one or more secondary indexes, several sort tasks are started depending on the segment level of the index sources. The segment occurrences of the primary database are read in physical order, then the index pointers and the index keys are collected and sorted in hierarchical order. For example, when segment level of an index source is the second level of dependent segments, two sort tasks (IIUSRTV1 and IIUSRRTV2) are started, and when the index target or the index source is a variable-length segment, an additional sort task (IIUSRDTD0) is started.

The messages from the sort tasks are printed in the IIUSOUT data set. The following figure shows an example of the sort messages.

```
IIUSRRTV1 PPP9PPP1 SEQSCAN SORT MESSAGES:
----------------------------------------------------------84--
ICE200I 0 IDENTIFIER FROM CALLING PROGRAM IS CA12
ICE201I A RECORD TYPE IS F - DATA STARTS IN POSITION 1
ICE751I 0 C5-140658 C6-135397 C7-135397 C8-140658 E4-140658 C9-135397 E5-138877 E6-131999 C4-131999 E7-140658
ICE143I 0 BLOCKSET SORT TECHNIQUE SELECTED
ICE250I 0 VISIT http://www.ibm.com/storage/dfsorf FOR DFSORT PAPERS, EXAMPLES AND MORE
ICE000I 0 - CONTROL STATEMENTS FOR 5650-ZOS, Z/OS DFSORT V2R2 - 08:03 ON TUE MAY 29, 2018 -
   SORT FIELDS=(001,010,CH,A),DYNALLOC,FILSZ=E000001000000
   RECORD TYPE=F,LENGTH=00195
   OPTION MSGDDN=IIUSRRTV1,SORTDD=T009
   DEBUG ABEND
ICE193I 0 ICEAM2 INVOCATION ENVIRONMENT IN EFFECT - ICEAM2 ENVIRONMENT SELECTED
```

Figure 17. Sort messages in the IIUSOUT data set

These sort tasks require additional sort workspace. The sort file size of IIUSRRTVx is estimated based on the sum of index records, and the estimated value is given to the SORT FILSZ option. When the VSAM data set for the index is not reused, the number of index records is estimated based on the allocation characteristics of the VSAM data set. You can override the estimated value by using the SORTFSZ control statement.

When additional sort workspace is not available, you can use the scan method that was used in IMS Index Builder Version 2.3.
Procedure

Specify the MAXTASKS 2 control statement. The MAXTASKS 2 control statement specifies IMS Index Builder to use the hierarchical HPU scan method for building secondary indexes, such as in IMS Index Builder Version 2.3. When the MAXTASKS 2 control statement is specified, additional sort workspace is not needed.

When the IMS Index Builder job ends, you can identify the scan method that was used in the event log that is printed in the IIUSNAP data set. The following figure shows an example of the event log.

---

IIUE096M A0000777 008C0BF8 IB CONTROL FLOW EVENT 603 AT IUISCA01 12.00.00.00  
SEQUENTIAL SERIAL SCANNER ACTIVATED

IIUE096M A0000777 006D5D0B IB CONTROL FLOW EVENT 609 AT IUISCA12 12.00.00.00  
SEQUENTIAL SAS SCANNER ACTIVATED

IIUE096M A0000777 007A1088 IB CONTROL FLOW EVENT 623 AT IUISCA55 12.00.00.00  
HIERARCHICAL HPU SCANNER ACTIVATED

---

*Figure 18. Event log in the IIUSNAP data set*
Issuing the IMS commands automatically

By using the IMS Tools Online System Interface, IMS Index Builder can issue IMS commands (/DBD, /DBR, and /STA) automatically. The /DBD and /DBR commands prevent transactions or programs from updating the databases while the indexes are built. The /STA command restarts the databases after the indexes are built.

About this task

When building secondary indexes, IMS Index Builder issues /DBD commands for the prime database to be scanned and /DBR commands for secondary indexes to be built. After building the indexes, IMS Index Builder issues /STA commands for the prime database and the secondary indexes.

Procedure

To request IMS Index Builder to issue the /DBD, /DBR, and /STA commands of IMS, define the JCL using the following example:

```plaintext
//IIUUBLD JOB &SYSUID,MSGCLASS=H,REGION=33M,TIME=1339,
//  RESTART**,CLASS=A,NOTIFY=&SYSUID
//******************************************************************************
//** BUILDING SECONDARY INDEXES AND **
//** ISSUING IMS COMMANDS AUTOMATICALLY USING DFSURIDX **
//******************************************************************************
//** Provide values for the following variables:
// SET SIIULMOD= INDEX BUILDER LOAD LIBRARY |
// SET SFOILOAD= TOSI LOAD LIBRARY
// SET SHKTLOAD= ITKB LOAD LIBRARY
// SET RESLIB= IMS RESLIB
// SET MDALIB= IMS MDALIB
// SET DBDLIB= DBD LIBRARY
// NOTE: The Index Builder load library and all other datasets
// concatenated to STEPLIB must be APF authorized.
//******************************************************************************
//IIUUBSCN EXEC PGM=IIUSTART
//STEPLIB DD DISP=SHR,DSN=&SIIULMOD
// DD DISP=SHR,DSN=&SFOILOAD
// DD DISP=SHR,DSN=&SHKTLOAD
// DD DISP=SHR,DSN=&RESLIB
// DD DISP=SHR,DSN=&MDALIB
// DD DISP=SHR,DSN=&DBDLIB
// IMS DD DISP=SHR,DSN=&MDALIB
// IMSDALIB DD DISP=SHR,DSN=&MDALIB
//*
//IIUIN DD *
// PROC BLD_SECONDARY,TSTDATA
// INPUT IBSCAN
// VIC YES
// TOSIXCFGRP GRP01
// TOSIDBD YES
// TOSIDBR YES
// TOSISTA YES,ALL
//*
```

- **TOSIXCFGRP GRP01** specifies the XCF group name of the IMS Tools Online System Interface. For more information, see "TOSIXCFGRP control statement" on page 93.
- **TOSIDBD YES** specifies to issue /DBD DB commands to the prime database to be scanned. For more information, see "TOSIDBD control statement" on page 89.
• TOSIDBR YES specifies to issue /DBR DB commands to the indexes to be built. For more information, see “TOSIDBR control statement” on page 91.

• TOSISTA YES, ALL specifies to issue /STA DB commands to both the prime database and indexes after building secondary indexes. For more information, see “TOSISTA control statement” on page 92.

• VIC YES specifies to issue the NOTIFY.UIC command for each index that is built. If TOSISTA YES is specified, this control statement must be specified to turn off the DBRC IC NEEDED flags. For more information, see “VIC control statement” on page 94.
Using the IMS catalog

IMS Index Builder can build IMS catalog databases. It can also retrieve database information from IMS catalog instead of DBD libraries.

Building IMS catalog databases

IMS Index Builder can build primary indexes and ILDSs of catalog PHIDAM databases and partitions of catalog PSINDEX databases. Information of the IMS catalog databases can be retrieved from either the DBD library or the IMS catalog itself.

Retrieving database information from the IMS catalog

If you enable the IMS management of ACBs, IMS Index Builder can retrieve database information from the IMS catalog. When IMS Index Builder retrieves database information from the IMS catalog, IMS Index Builder does not refer to DBD libraries.

To run IMS Index Builder in an IMS managed ACBs environment, complete the following steps:

1. Register all databases that will be processed by IMS Index Builder to the IMS catalog.
2. Enable the IMS management of ACBs by completing one of the following steps:
   - Create an IMS Catalog Definition exit routine (DFS3CDX0) which enables IMS management of ACBs.
   - Create a DFSDFXXX member in IMS.PROCLIB. This member must contain the ACBMGMT=CATALOG parameter in the <CATALOG> section.
3. Concatenate the following program libraries to the STEPLIB DD statement of IMS Index Builder JCL:
   - IMS Tools Catalog Interface of IMS Tools Base for z/OS
   - IMS Catalog Definition exit routine, if the exit is used to enable IMS managed ACBs
4. If the DFSDFXXX member is used to enable IMS managed ACBs, complete the following steps:
   a. Specify the PROCLIB DD statement, which contains the IMS.PROCLIB data set with the DFSDFXXX member, in the IMS Index Builder JCL.
   b. Specify the DFSDF control statement in the IIUIN data set to identify the DFSDFXXX member that IMS Index Builder uses.
5. If you want to retrieve database information from the IMS catalog staging data set, specify the DFSACBPDPD DD statement with the STAGING keyword.

Restrictions for using IMS managed ACBs

IMS Index Builder does not support IMS managed ACBs if either of the following conditions is met. Use the DBD library to retrieve database information in these cases:

- Retrieving database information from an IMS catalog database that is not registered to the DBRC RECON data set.
- Building an IMS catalog database with a DBD name other than the standard names (DFSCD000 and DFSCX000).
Chapter 5. JCL and input control statements

The JCL and control statements that are described in this topic are needed to run IMS Index Builder.

Topics:
- “IMS Index Builder JCL” on page 66
- “IMS Index Builder input control statements” on page 72
IMS Index Builder runs as an authorized program facility (APF) program and typically runs as a single job step.

The job step requires an EXEC statement and several DD statements. You must issue the statements in the following order:
1. EXEC statement
2. STEPLIB statement
3. Other DD statements

**EXEC statement**
The EXEC statement marks the beginning of a step within a job, and specifies the name of a program or cataloged procedure to be run.

The EXEC statement must specify PGM=IIUSTART. No parameters are passed to program IIUSTART.

**DD statements**
DD statements are used to identify the source of input and the placement of output information. The DD statements that are listed in this section are supported by IMS Index Builder.

**STEPLIB DD statement**
This DD statement can contain the following libraries:
- IMS Index Builder load library (Required). It must be at the top of the STEPLIB DD concatenation.
- IMS RESLIB (Required).
- IMS Tools Online System Interface load library, if you want the IMS commands issued automatically.
- IMS Tools Knowledge Base load library, if storing reports to the knowledge base is enabled.
- IMS Tools Catalog Interface load library, if you want to enable the IMS management of ACBs.
- The SGLXLOAD library of IMS Tools Base, if you want to offload eligible workloads to zIIP processors.
- Libraries that contain exit routines and IMS dynamic allocation modules (MDAs). These libraries must be present if there are any exits or dynamic allocation modules.

If the MDA library is not APF-authorized, specify the library on the IMSDALIB DD statement.

All the libraries that are specified by this statement must be authorized through the Authorized Program Facility (APF). For more information, see “Enabling inter-region communication (APF authorization)” on page 27.

This statement is required.
**DDEFPDS DD statement**

This DD statement provides IMS Index Builder delete and define statements for the indexes it processes. It is optional. You might want to use it to change the allocation parameters (such as the volume or space allocations) of the new index. IMS Index Builder looks for the delete and define statements for each index data set in a member with the same name as the index DD name. The member contents must be only delete and define statements.

Members for PSINDEX partitions are ignored if the UPDATE option is specified on the PROC control statement.

Define this data set as a PDS, fixed blocked with LRECL=80 and fixed-block format.

In parallel IMS Index Builder job runs, the DDEFPDS functionality might result in concurrent IDCAMS delete and define operations, and one operation typically fails.

This statement is optional.

**DFSACBPD DD statement**

This DD statement points to the control statement data set that defines the type of the IMS catalog that IMS Index Builder refers to in an IMS managed ACBs environment. This DD statement is optional. STAGING is the only valid keyword.

When the IMS management of ACBs is enabled, by default, IMS Index Builder obtains the active database from the IMS catalog directory. To override the default behavior and obtain pending ACBs from the staging data sets in the IMS catalog, specify the keyword STAGING on this DD statement.

When the IMS management of ACBs is not enabled, this DD statement is ignored.

**DFSCTL DD statement**

This DD statement defines a file that is used by IMS OSAM sequential buffering to define its environment.

If IMS Index Builder plans a DL/I hierarchical scan for an OSAM database and a DFSCTL DD statement is not specified, the statement is generated by IMS Index Builder.

**DFSPARM DD statement**

The DFSPARM DD statement provides sort control options. Because you can use any compatible sort product with IMS Index Builder, you must refer to the sort product documentation to understand the sort control options that are provided by the DFSPARM DD statement.

This statement is optional.

**DFSURIDX DD statement**

This DD statement contains output from the prefix resolution process. IMS Index Builder uses the content of the file to build one or more secondary index databases. When this file is empty or has the DUMMY attribute, all of the selected data sets that are being built will be initialized.
If the INPUT control statement specifies DFSURIDX, this statement is required.

**DFSURWF1 DD statement**

This DD statement contains the output from the reorganization process. This file is used as input by IMS Index Builder to build one or more secondary index databases.

To use two or more files as input to the same job step, concatenate the work files that are generated by each of the reload utility job steps.

If the INPUT control statement specifies DFSURWF1, this statement is required.

**DFSVSAMP DD statement**

This DD statement is used by IMS. If this DD statement is specified, it defines the file that contains definitions for the database I/O buffer subpools for VSAM and, if required, for OSAM. If you do not specify this DD statement, IMS Index Builder dynamically generates sufficient subpool definitions for the hierarchical scan process.

This statement is optional.

**HPSRSIDX DD statement**

This DD statement contains the output from High Performance Load that is needed to build one or more secondary indexes. It has the same format at DFSURWF1.

This statement is required if the control statement specifies HPSRSIDX.

**HSSROPT DD statement**

Specify this statement to use High Performance Unload for hierarchical scan. High Performance Unload commands are supplied by the user. For more information about HSSROPT options and control statements, see the *IMS High Performance Unload User's Guide*.

This statement is optional.

**IIUCABP DD statement**

This DD statement defines the file that is created by IMS Index Builder to specify the CAB buffering parameters that are used during the scan. IMS Index Builder creates the appropriate control statement options for CAB buffering parameters; however, you can change them by using this DD statement.

This statement is optional.

**Attention:** Explicitly specifying CAB buffering parameters can impact the optimal performance of the scan, so taking the program-generated values is strongly recommended.
IIUCAPT DD statement

This DD statement contains the combined WTO messages from all of the address spaces that are involved in the process.

This statement replaces SYSPRINT.

This statement is optional. If not specified, this file is dynamically allocated as //IIUCAPT DD SYSOUT=*.

IIUDUMP DD statement

This DD statement contains a dump that is generated as the result of an error. You can use this dump to diagnose the error.

This statement is optional. If not specified, this file is allocated dynamically as //IIUDUMP DD SYSOUT=*.

IIUIN DD statement

This DD statement defines the main IMS Index Builder input file. It contains the control statements that specify which actions will occur and which options to use.

This statement is required.

IIUOPT DD statement

This DD statement authorizes IMS Index Builder to select appropriate options for the IMS HP Unload feature available for hierarchical scans. The options are identical to the options for the HSSROPT DD statement, which is used in IMS HP Unload.

This DD is reserved for system use.

IIUSTAT DD statement

This file contains the performance statistics resulting from the hierarchical HPU scan.

This statement is optional. If not specified, this file is dynamically allocated by IMS Index Builder as //IIUSTAT DD SYSOUT=*.

IIUPRINT DD statement

This DD statement defines the main IMS Index Builder output file. It contains the index build report and error messages, if any.

This statement is optional. If not specified, this file is allocated dynamically as //IIUPRINT DD SYSOUT=*.

IIUSNAP DD statement

This DD statement contains the critical events report and the following output:

- High Performance Image Copy
- High Performance Unload
- DBRC notify commands

This statement is optional. If not specified, this file is dynamically allocated as
//IIUSNAP DD SYSOUT=*.

**IIUSOUT DD statement**

This DD statement defines a file for the sort messages and the IDCAMS messages
for the delete and define statements.

This statement is optional. If not specified, this file is dynamically allocated as
//IIUSOUT DD SYSOUT=*.

**IIUWFPnn DD statement**

This DD statement is allocated by IMS Index Builder dynamically when the INPUT
HPSRWFP statement is specified.

*nn* is in the range of 00 - 99. Do not specify the DD statements in JCL.

**IMS DD statement**

This DD statement is used to access the involved databases. The file points to one
or more libraries that contain the DBDs of the physical database and its indexes.

This statement is required unless the IMS management of ACBs is enabled.

**IMSDALIB DD statement**

This DD statement contains the IMS dynamic allocation modules (MDAs).

**OUTPUT DD statement**

If you defined an OUTPUT statement as one of your input control statements, you
must specify this DD statement. The DD name that you use must match the name
that you specify in the OUTPUT statement.

Use the following guidelines when you specify the OUTPUT DD statement:

- When you use the split function to read records from DFSURWF1, this DD
  statement points to a flat file that contains only those records in DFSURWF1 that
  are needed for prefix resolution. The output file must be defined with
  DCB=(RECFM=VB,LRECL=900).
- When you build the HIDAM primary index of a non-HALDB database, this DD
  statement points to the new primary index database. The output DD name must
  be different from the DD name that is defined in the DBD for the primary index.
**Important**: Do not use the OUTPUT DD statement when building a HALDB primary index database. Partitioned primary index databases are built by using the original index databases.

**PROCLIB DD statement**

This DD statement points to the IMS.PROCLIB data set that contains the DFSDF\_\_\_ member to enable the IMS management of ACBs. The DFSDF\_\_\_ member must contain the ACBMGMT=CATALOG parameter in the <CATALOG> section. You must also specify the DFSDF control statement in the IIUIN data set so that IMS Index Builder refers to the DFSDF\_\_\_ member.

This DD statement is required only when IMS Index Builder uses IMS managed ACBs through the DFSDF\_\_\_ member. This DD statement is not required if IMS Index Builder refers to DBD libraries or if you enable the IMS management of ACBs with the IMS Catalog Definition exit routine (DFS3CDX0).

**RECON1, RECON2, RECON3 DD statements**

These DD statements point to the RECON1, RECON2, and RECON3 data sets.

If you want to allocate RECON data sets dynamically, do not specify these DD statements. If no RECON\_\_ DD statement is specified, RECON\_\_ data sets are dynamically allocated from the MDA members in STEPLIB or IMSDALIB.

**Secondary index DD statements**

The DD statement for each IMS full-function secondary index or PSINDEX partition that is being built can be specified.

If a secondary index DD statement is not specified, IMS Index Builder tries to allocate the secondary index dynamically as follows:

- For a PSINDEX, IMS Index Builder extracts the information from DBRC RECON.
- For a non-partitioned index database, an IMS dynamic allocation member is required in the dynamic allocation library. If this member is not found, IMS Index Builder will stop.

**Data set group DD statements**

When IMS Index Builder scans the non-partitioned primary database, the JCL DD statements for all the DSGs and the primary index (when HIDAM) can be specified.

If the DD statements are not specified, IMS Index Builder tries to allocate them dynamically. If this happens, IMS dynamic allocation members for these databases must exist in the IMS dynamic allocation library. Otherwise, IMS Index Builder will stop.

Do not specify DD statements for a partitioned primary database; allocation is done by IMS Index Builder by using the appropriate information from the DBRC RECON data set.
IMS Index Builder input control statements

Input control statements are used to specify which functions IMS Index Builder will do and which options it must use.

Input control statement rules

The input control statements override the runtime defaults that are specified in the IIURDFLT parameters of the runtime parameters module.

The input control statements are located in the IIUIN file and can be specified in any order.

Adhere to the following guidelines when coding control statements:
• Only characters in positions 1 through position 72 are read by IMS Index Builder. Characters after position 72 are ignored.
• Control statements must be entered on a single line.
• An asterisk (*) in position 1 indicates a comment line.
• Statements are composed of the following two strings, which are separated by one or more blank spaces:
  – The statement type
  – The statement parameters (a blank terminates the parameter string)
• Anything after the parameter string is considered a comment and is ignored.
• Control statements can be indented. Any leading blank spaces are ignored.
• When coding the parameter strings, no blank spaces should be embedded.
Required control statements

IMS Index Builder has two required input control statements: the PROC control statement and the INPUT control statement. You must specify these required input control statements for each IMS Index Builder job.

PROC control statement

The PROC control statement specifies the type of processing to be done by IMS Index Builder.

This statement is required. If you specify PROC statements to build both a primary and a secondary index database in the same job, the statements must apply to the same database.

The following diagrams show the format of the PROC statement for specifying secondary, primary, ILDS, or all index databases (respectively).

PROC  Specifies that this is a PROC statement.

BLD_PRIMARY

Specifies that the primary index of the HIDAM and PHIDAM databases must be built. You can specify only one PROC BLD_PRIMARY statement per job.

BLD_SECONDARY

Specifies that secondary index databases must be built. If the prime database does not have secondary indexes, an error is indicated, and IMS Index Builder stops at the initialization phase.

If you specify INPUT IBSCAN, you can specify only one PROC BLD_SECONDARY statement per job.
If you specify INPUT DFSURWF1 or INPUT DFSURIDX, you can specify one or more PROC BLD_SECONDARY statements per job.

**ALL**  Builds all secondary indexes of the database. If you specify **ALL**, any INDEX statements that are associated with the database named in the PROC statement are ignored. **ALL** is used as the default.

**SELECTED**  Builds the secondary indexes that are selected by the INDEX statement.

**LOAD**  Builds the secondary index in load mode. IMS Index Builder builds and loads all the secondary index records. **LOAD** is used as the default.

**UPDATE**  Updates the PSINDEX in update mode. IMS Index Builder builds and loads only the PSINDEX records that point to certain HALDB partitions. Use this mode when HALDB partitions are selectively reorganized. The VSAM data sets of the PSINDEX must fully contain index records for all the HALDB partitions before the reorganization. IMS Index Builder determines the target HALDB partitions from the following statements:

- If the INPUT IBSCAN statement is specified, IMS Index Builder refers to the PARTNAME statement. If no PARTNAME statement is specified, IMS Index Builder processes the PSINDEX records for all the HALDB partitions.

```plaintext
PROC   BLD_SECONDARY,HMPORDP1,ALL,UPDATE
INPUT  IBSCAN
PARTNAME HORDP1B,HORDP1C
```

- If the INPUT HPSRWFP statement is specified, IMS Index Builder dynamically allocates all the WFP files for the HALDB partitions that IMS High Performance Load reloaded and processes the records found in the WFP files.

```plaintext
PROC   BLD_SECONDARY,HMPORDP1,ALL,UPDATE
INPUT  HPSRWFP
WFPHLQ  IMSTOOLS.WORK.HMPORDP1
```

**Usage notes for update mode:**

- If the DDEFPDS DD statement is specified, the IDCAMS commands for PSINDEX are not performed.
- CI or CA splits in VSAM data sets remain.
- If IMS Index Builder cannot find an index record that should be updated, IMS Index Builder inserts a new index record.

**BLD_ILDS**  Specifies that the indirect list data sets be built. You can specify only one PROC BLD_ILDS statement per job.

**BLD_ALL**  Specifies that the primary index, secondary indexes, and ILDS be built. You can specify only one PROC BLD_ALL statement per job. It will build only indexes appropriate to the database you specified.
The UPDATE option can be specified. This option is applied only when building PSINDEX.

database
Is the DBD name of the IMS physical database whose primary, secondary, or ILDS index must be built.

**INPUT control statement**
The INPUT control statement specifies the type of input that is used by IMS Index Builder to build indexes.

The following diagram shows the format of the INPUT statement.

![INPUT Diagram](image)

**INPUT**
Specifies that this is an INPUT statement.

**IBSCAN**
Specifies that a scan of the physical database is used as input. This option applies to primary and secondary indexes, and ILDS.

**DFSURWF1**
Specifies that data generated by initial database load or the IMS reload utility is used as input. Input is to be read from the DFSURWF1 file.

If you specify DFSURWF1, you must also specify the PROC BLD_SECONDARY statement.

DFSURWF1 is not applicable when building a partitioned secondary index (PSINDEX).

**HPSRSIDX**
Specifies that data generated by initial database load or the IMS reload utility is used as input. Input is to be read from the HPSRSIDX file.

If you specify HPSRSIDX, you must also specify the PROC BLD_SECONDARY statement.

HPSRSIDX is not applicable when building a partitioned secondary index (PSINDEX).

**DFSURIDX**
Specifies that data generated by the IMS prefix resolution utility is used as input. Input is to be read from the DFSURIDX file.

If you specify DFSURIDX, you must also specify the PROC BLD_SECONDARY statement.

You cannot use DFSURIDX when building a partitioned secondary index (PSINDEX) or a secondary index with non-unique keys.
HPSRWFP
Specifies that data generated by IMS High Performance Load is used as input for building partitioned secondary indexes (PSINDEXes).

If you specify HPSRWFP, you must also specify the PROC BLD_SECONDARY statement and the WFPHLQ control statement. If the WFP files have been produced only for selected HALDB partitions, the UPDATE option must be specified on the PROC control statement. See “PROC control statement” on page 73.

DBRC=
Specifies whether DBRC NOTIFY commands will be issued for the indexes loaded. Valid values are Y and N.

If you do not specify this parameter, the IMS default value for DBRC is used for non-HALDB. For HALDB, DBRC=Y is used. If you specify DBRC=N, no notification is issued, regardless of the database type.

Notes:
1. IMS Index Builder will always use DBRC for HALDB databases. If DBRC=N is specified, no notification is issued, regardless of the database type.
2. If you issue the CHANGE.DB NONRECOV NOICREQ command to each nonrecoverable index in advance, the IC NEEDED flag is not turned on even if DBRC NOTIFY command is issued by IMS Index Builder. Alternatively, you can use the VIC YES statement to turn off the IC NEEDED flag. This statement specifies whether IMS Index Builder issues the DBRC NOTIFY.UIC command.

If INPUT is not specified, the default value is IBSCAN.
Optional control statements

Use the optional control statements of IMS Index Builder to control the behavior of an IMS Index Builder job.

CLASS control statement
IMS Index Builder uses data sets to pass data between the sort, scan, and master address spaces. These data sets exist only for the duration of the run. By default, they are allocated with the high-level qualifier IIU.STRIPE. The CLASS control statement can be used to specify a different high-level qualifier.

CLASS

CLASS specifies that this is a CLASS statement.

qualifier

Specifies a high-level qualifier of the stripe data sets. If %TIME is not specified, the maximum length is 17 characters.

.%TIME

Specifies that IMS Index Builder replaces %TIME with the time when the job started in Thimmsst format. For example, the high-level qualifier might be IIU.STRPT2359599. The maximum length is 17 characters in total.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT CLASS entry in Chapter 7, “Defining runtime parameters,” on page 103.

COMPAUTH control statement
The COMPAUTH control statement specifies whether the segment compression routine is called in supervisor state.

If you use an encryption exit of InfoSphere® Guardium® Data Encryption for Db2® and IMS Databases as the segment compression exit, specifying COMPAUTH YES reduces performance degradation.

COMPAUTH

COMPAUTH specifies that this is a COMPAUTH statement.

NO The segment compression exit is called in problem state.

YES The segment compression exit is called in supervisor state.

If the COMPAUTH statement is not specified, NO is the default.
**Tip:** You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT COMP AUTH entry in Chapter 7, “Defining runtime parameters,” on page 103.

**DBAUTH control statement**

The DBAUTH control statement specifies whether IMS Index Builder issues the IMS DBRC authorization request for the prime database and the indexes to prevent transactions and programs from updating the database.

The following tables summarize the access intent that is requested through IMS DBRC authorization. EX indicates exclusive access, and RD indicates read access.

### Table 1. Access intent for prime database and indexes (non-HALDB)

<table>
<thead>
<tr>
<th>PROC control statement</th>
<th>Prime database</th>
<th>HIDAM primary index</th>
<th>Secondary index</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLD_SECONDARY</td>
<td>RD (see Note)</td>
<td>N/A</td>
<td>EX</td>
</tr>
<tr>
<td>BLD_PRIMARY</td>
<td>RD</td>
<td>EX</td>
<td>N/A</td>
</tr>
<tr>
<td>BLD_ALL</td>
<td>RD</td>
<td>EX</td>
<td>EX</td>
</tr>
</tbody>
</table>

**Note:** If DFSURWF1 or HPSRSIDX is used as the input, no request is issued for the prime database.

### Table 2. Access intent for prime database and indexes (HALDB)

<table>
<thead>
<tr>
<th>PROC control statement</th>
<th>Prime HALDB partition</th>
<th>PSINDEX partition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLD_SECONDARY</td>
<td>RD (see Notes)</td>
<td>EX</td>
</tr>
<tr>
<td>BLD_PRIMARY</td>
<td>EX</td>
<td>N/A</td>
</tr>
<tr>
<td>BLD_ILDS</td>
<td>EX</td>
<td>N/A</td>
</tr>
<tr>
<td>BLD_ALL</td>
<td>EX</td>
<td>EX</td>
</tr>
</tbody>
</table>

**Notes:**
- If HPSRWFP is used as the input, no request is issued for the prime HALDB partitions.
- If the RECOVERY NEEDED flag of the primary index or ILDS is turned on, exclusive access (EX) is requested.

**DBAUTH**

Specifies that this is a DBAUTH statement.

**YES**

IMS Index Builder issues the DBRC authorization request.

**RECOV**

IMS Index Builder issues the CHANGE.DBDS RECOV command to turn on the RECOVERY NEEDED flag to prevent access from other IMS subsystems throughout the entire process.

The DBRC authorization for the resource is released when the IMS cleanup processing ends for the IMS region. Message DFS627I indicates the end of this processing. Without the RECOV option, other IMS subsystems might access the resource after the DBRC authorization is released until the
DBRC NOTIFY commands are issued. The RECOV option prevents any access during this period by turning on the RECOVERY NEEDED flag.

Restriction: IMS Index Builder does not issue the CHANGE.DBDS RECOV command for PSINDEX partitions whose PARTITION INIT NEEDED flag is set to YES.

NO IMS Index Builder does not issue the DBRC authorization request.

If the DBAUTH statement is not specified, YES is the default.

If DBRC=N is specified on the INPUT control statement, no DBRC authorization request is issued even if DBAUTH YES is specified.

Note: The longest possible time that IMS Index Builder can wait for a response from DBRC is 30 minutes.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT DBAUTH entry in Chapter 7, "Defining runtime parameters," on page 103.

DFSDF control statement
The DFSDF control statement specifies the suffix of the DFSDFxxx member in the IMS.PROCLIB data set. The member with this suffix must exist in a data set specified in the PROCLIB DD statement. If the member contains the ACBMGMT=CATALOG parameter in the <CATALOG> section, IMS managed ACBs is enabled.

This control statement is required only when IMS Index Builder uses IMS managed ACBs through the DFSDFxxx member. This DD statement is not required if IMS Index Builder refers to DBD libraries or if you enable the IMS management of ACBs with the IMS Catalog Definition exit routine (DFS3CDX0).

DFSDF
  Specifies that this is a DFSDF statement.

*NO IMS Index Builder does not refer to the DFSDFxxx member in the PROCLIB data set.

suf Specifies the 3-character suffix of the DFSDFxxx member in the PROCLIB data set.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT DFSDF entry in Chapter 7, "Defining runtime parameters," on page 103.

IIURSORT control statement
The IIURSORT control statement specifies the procedure name to start SORT and SCAN subordinate address spaces. By default, the IIUBSRT procedure is used.
For more information about the procedure, see Chapter 6, “Defining the subordinate address space procedure,” on page 99.

IIURSORT

Specifies that this is an IIURSORT statement.

procname

Specifies the 1- to 8-byte procedure name.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT IIURSORT entry in Chapter 7, “Defining runtime parameters,” on page 103.

ILDS control statement

The ILDS control statement specifies whether to build ILE records or to initialize the data set without building any ILE records in the ILDS build process.

This control statement can be specified with the PROC BLD_ALL or PROC BLD_ILDS statement for a HALDB.

ILDS

Specifies that this is an ILDS statement.

BUILDILE

Builds and loads the ILE records.

INITONLY

Initializes the data set without building any ILE records.

The DBRC command to turn off the RECOVERY NEEDED flag is issued in either case.

Use the ILDS INITONLY control statement only when all the PSINDEXes are to be built after each reorganization of the target HALDB. In these cases, ILE records are not read in the HALDB self-healing pointer process.

If the ILDS statement is not specified, BUILDILE is the default.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT ILDS entry in Chapter 7, “Defining runtime parameters,” on page 103.

INDEX control statement

The INDEX statement specifies the name of one or more secondary index databases to be recovered or built.
Use this statement if you specify the SELECTED keyword on the PROC
BLD_SECONDARY input control statement. If ALL is specified in a PROC
BLD_SECONDARY statement, any INDEX statement that is associated with the
database that is specified in the PROC statement is ignored.

INDEX
Specifies that this is an INDEX statement. You can specify one or more
INDEX statements.

index, psindex
Specifies the DBD name of the secondary index database to be built.

pspartname
Specifies one or more PSINDEX partitions to be rebuilt. PSINDEX DBD
names must precede the names of selected PSINDEX partitions.

indexsize
Specifies the number to be used as SORTFSZ for the index database
relative to the position (set by the INDEX control statements sequence) of
the number if the index is deleted by IDCAMS. If the index is being
reused, the current index size takes precedence. Specify 0 to indicate that
the sort file size will be estimated based on IDCAMS define allocation
values. It is recommended to either use indexsize for all selected databases
or not use it at all. The number follows each database and can be specified
as a parallel list to databases. The data that is entered by the INDEX
statement is separated into two arrays, words and numbers. Words are
matched to numbers, no match is zero.

An example of using indexsize is:
INDEX TSTINDX1,TSTINDX2,TSTINDX3
INDEX 500000,0,2000000

This input means:
• TSTINDX1 will be sorted using an estimated number of records based
  on its current size, unless it has been deleted, in which case an estimate
  of half a million records will be used.
• TSTINDX2 will be sorted using an estimated number of records based
  on its current size, unless it has been deleted, in which case the estimate
  will be based on the allocated space.
• TSTINDX3 will be sorted using an estimated number of records based
  on its current size, unless it has been deleted, in which case an estimate
  of 2 million records will be used.

ITKBLOAD control statement
The ITKBLOAD control statement specifies the name of the load module library
for IMS Tools Knowledge Base. This control statement is used only when
ITKBRSRVR control statement is specified.
ITKBLOAD
Specifies that this is an ITKBLOAD statement.

*NO The modules of IMS Tools Knowledge Base are loaded from the private library or the system library of the job. If ITKBLOAD NO is specified, IMS Index Builder applies ITKBLOAD *NO.

libName
Specifies the name of the load module library for IMS Tools Knowledge Base. This library must be APF-authorized.

If the ITKBLOAD statement is not specified, *NO is the default.

Tip: You can specify the library name in the runtime parameters module. See the explanation of the IIURDFLT ITKBLOAD entry in Chapter 7, “Defining runtime parameters,” on page 103.

ITKBSRVR control statement
The ITKBSRVR control statement specifies the name of the IMS Tools Knowledge Base server where the IMS Index Builder reports are to be stored and viewed.

ITKBSRVR
Specifies that this is an ITKBSRVR statement.

*NO IMS Index Builder does not store reports in the IMS Tools Knowledge Base Output repository. If ITKBSRVR N0 is specified, IMS Index Builder applies ITKBSRVR *NO.

server name
Is the 1- to 8-byte IMS Tools Knowledge Base server name where the IMS Index Builder reports are to be stored and viewed.

If the ITKBSRVR statement is not specified, *NO is the default.

Tip: You can specify the library name in the runtime parameters module. See the explanation of the IIURDFLT ITKBSRVR entry in Chapter 7, “Defining runtime parameters,” on page 103.

JOBNSCAN control statement
The JOBNSCAN control statement specifies the job name of each SCAN subordinate address space, which is started when the sequential scan method is used for HALDB partitions. By default, the IIUSCANS is used for the job name.
JOBNSCAN

Specifies that this is a JOBNSCAN statement.

jobname

Specifies the 1- to 8-byte job name of each SCAN subordinate address space.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT SCAN entry in Chapter 7, “Defining runtime parameters,” on page 103.

JOBNSORT control statement

The JOBNSORT control statement specifies the job name of each SORT subordinate address space. By default, IIUSORTS is used for the job name.

JOBNSORT

Specifies that this is a JOBNSORT statement.

jobname

Specifies the 1- to 8-byte job name of each SORT subordinate address space.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT SORTP entry in Chapter 7, “Defining runtime parameters,” on page 103.

MAXTASKS control statement

The MAXTASKS control statement is used to customize the IMS Index Builder run.

This parameter allows you to control the number of subordinate address spaces (SAS) created. When using the low values range (0, 1, 2), MAXTASKS allows you to pre-determine the scan and sort modes. In default mode, without MAXTASKS specified, IMS Index Builder will select the scan and sort modes, and create the subordinate address spaces that are needed to efficiently process your request. While default mode is generally recommended, you might want to use MAXTASKS instead of the default mode to improve efficiency, or use MAXTASKS after a failure as an alternative approach. MAXTASKS takes effect only when secondary indexes are rebuilt.

When the MAXTASKS statement is specified, IMS Index Builder creates less than or equal to the number of subordinate address spaces that are specified. Two thirds of the subordinate address spaces that are created are allocated to scan. Valid values are 0 - 36, with leading zeros. The maximum field length is 15 characters.
If MAXTASKS 0 is specified, all processing occurs in the master address space using striped sorts. If MAXTASKS 2 is specified, the hierarchical HPU scan is selected and subordinate address spaces are created for buffered sorts.

```
  MAXTASKS nn PSORT
```

**MAXTASKS**

Specifies that this is a MAXTASKS statement.

*nn* Valid values are 0 - 36, with leading zeros. Do not specify MAXTASKS 1.

**PSORT**

Enables the parallel sort for a PSINDEX so that the partitions are sorted in parallel. This sort type can improve the performance of building a PSINDEX by starting a buffered sort for each PSINDEX partition and by loading the index records in the E35 sort exit.

For example, if a PSINDEX has 12 partitions, and the PSORT keyword is specified, 12 buffered sorts are started for the PSINDEX.

If you specify MAXTASKS 2 PSORT, four subordinate address spaces for sort are created and three buffered sorts are started in each address space.

If you specify MAXTASKS 36 PSORT, 12 subordinate address spaces are created and a buffered sort is started in each address space.

**Restrictions:** The following restrictions apply to the parallel index sort:

- For *nn*, specify an integer value that is greater than or equal to 2.
- Do not specify partition names on the INDEX control statement.
- The SORTFSZ control statement is ignored.
- When IMS Index Builder is invoked by another utility, PSORT is ignored.

PSORT is an optional keyword.

**OUTPUT control statement**

The OUTPUT control statement specifies the DD name that is used for output.

Use an OUTPUT statement when:

- Using the split function to read input from DFSURWF1. For more information about the split function, see “Creating input for fast prefix resolution (split function)” on page 43.

- Building a HIDAM (non-partitioned) primary index. This statement is optional. If specified, the DD statement that is referred to specifies a new data set where the primary index is loaded. If not specified, the primary index is loaded into the existing data set.

**Tip:** Do not use the OUTPUT statement when building a PHIDAM (partitioned) primary index database.
OUTPUT

Specifies that this is an OUTPUT statement.

\textit{dd\_name}

Is the name of the DD statement that is used for output.

To build a HIDAM primary index (INPUT IBSCAN), the DD statement must point to a VSAM KSDS data set.

To use the split function (INPUT DFSURWF1), the DD statement must point to a sequential file with DCB=(RECFM=VB,LRECL=900).

\textbf{PARTNAME control statement}

The PARTNAME control statement specifies one or more HALDB (PHDAM or PHIDAM) partitions to be processed.

IMS Index Builder scans the selected partitions and builds the index records of primary index, ILDS, and PSINDEX.

To build a PSINDEX, the UPDATE option must be specified on the PROC control statement.

This statement is optional. If this statement is omitted, IMS Index Builder scans all partitions of the HALDB.

\textbf{PARTNAME}

Specifies that this is a PARTNAME statement. You can specify one or more PARTNAME statements. If PARTNAME is not specified, all HALDB partitions are processed.

\textit{partname}

Is the 1- to 7-character name of a PHIDAM partition for which a primary index is being built. You can specify one or more names separated by a comma or a blank.

\textbf{PROGMON control statement}

The PROGMON control statement specifies the interval for issuing WTO IIUB119I messages, which are messages to indicate the progress of the load process for each secondary index.

If this statement is not specified, the default value 0 is used.
PROGMON

  Specifies that this is a PROGMON statement.

  nnnnnn
  Specifies the interval for issuing IIUB119I messages. The value can be in
  the range of 0 - 999999. For example, if you specify 1000, an IIUB119I
  message is issued for every 1000 loaded records. If you specify 0, IIUB119I
  messages are suppressed.

  Tip: You can override the default value in the runtime parameters module. See the
  explanation of the IIURDFLT PROGMON entry in Chapter 7, “Defining runtime
  parameters,” on page 103.

SIDXBUF control statement

The SIDXBUF control statement specifies the number of buffers that are to be used
in the load process for each secondary index.

  SIDXBUF
    d_buf, i_buf, s_buf

SIDXBUF

  Specifies that this is an SIDXBUF statement.

  d_buf  Specifies the number of VSAM buffers that are to be used for the data
          component of the KSDS data set. The value must be an integer in the
          range of 1 - 255. If omitted, the default value applied is as follows:
          • In load mode, the default value of 16 is used for each partition of
            PSINDEX.
          • In update mode, a value that is equivalent to five tracks of disk space is
            used for each partition of PSINDEX.
          • For each non-HALDB secondary index, the default value of 64 is used.

  i_buf  Specifies the number of VSAM buffers that are to be used for the index
          component of the KSDS data set. The value must be an integer in the
          range of 1 - 255. For each partition of PSINDEX, the default value is 16 in
          load mode and 2 in update mode. For each non-HALDB secondary index,
          the default value is 64.

  s_buf  Specifies the number of QSAM buffers for reading the SORTOUT data set
          of striped sort. The value must be an integer in the range of 1 - 255. The
          default value is 64.

  Tip: You can override the default value in the runtime parameters module. See the
  explanation of the IIURDFLT SIDXBUF entry in Chapter 7, “Defining runtime
  parameters,” on page 103.
**SORTFSZ control statement**

The SORTFSZ control statement selects striped sort mode or specifies large sort volumes for sequential scan. SORTFSZ will trigger striped sort mode, the value that is entered will be used to determine the size of the sort stripes.

When writing index record to a stripe to be used as sort input file, reaching this number would cause the stripe to be closed, and a new stripe is opened to continue the process.

The SORTFSZ value is also used for stripe allocation. Values larger than 500,000 are accepted, else the default of 2000000 is used.

SORTFSZ is used in the internal sorts of the scan.

SORTFSZ also selects striped sorts when the indexes have been deleted and defined, even if they are defined as REUSE.

The number \textit{nnnnnnnn} is never passed to sort as an estimate of the number of records in the index.

- For striped sorts, the number \textit{nnnnnnnn} is used only to decide the size of the stripes if striped sort is selected.
- For sequential scan, the number \textit{nnnnnnnn} is passed to sort as an estimate of the number of records in internal sort associated with the scan.

\begin{verbatim}
  SORTFSZ nnnnnnnnn
\end{verbatim}

**SORTFSZ**

Specifies that this is a SORTFSZ statement

\textit{nnnnnnnn}

An integer in the range of 1 - 999999999.

**SORTOPT control statement**

The SOROPT statement specifies SORT parameters to be appended to the SORT OPTION statement of all the sorts that are started by an IMS Index Builder job.

The SOROPT statement is optional.

\textbf{Tip:} To specify parameters that are longer than 33 characters, consider using the DFSPARM data set instead of using the SOROPT statement. The DFSPARM DD statement can be specified in the IMS Index Builder JCL and the subordinate address space procedure.
SORTOPT
Is a keyword specifying that this is SORTOPT statement. You can specify only one SORTOPT statement.

sort_parms
Is a string of sort option parameters that are separated by commas or blanks. The string is appended, unchanged, to the SORT OPTION statement that is generated by IMS Index Builder. The maximum length of the string is 33 characters, and the entire string must be on one line. You can use this string to specify performance options or to override installation settings. For example:

MAINSIZE=MAX,MOSIZE=MAX,DYNSPC=512

Do not specify any parameter that is related to record type, sort fields, or sort order.

STRIPE control statement
The STRIPE control statement specifies the UNIT value and the volume count for allocating stripe data sets.

| STRIPE | SYSALLDA | UNIT=unit, VOLCNT=nnn |

STRIPE
Specifies that this is a STRIPE statement.

UNIT=unit
Specifies the device type or the device group name. The default is SYSALLDA.

VOLCNT=nnn
Specifies the maximum number of volumes in the range of 1 - 255. The value is used to allocate the following stripe data sets:

- Stripe data sets that contain index records when the SORTFSZ control statement is specified
- Stripe data sets that contain index records when duplicate key errors are detected

The default value is 10.

VOLCNT=1 is always used for the following stripe data sets:

- Stripe data sets that contain WTO messages of each address space
- Stripe data sets that contain SORT messages

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT STRIPE entry in Chapter 7, “Defining runtime parameters,” on page 103.

SVCDUMP control statement
The SVCDUMP control statement specifies whether IMS Index Builder requests an SVC dump file after a system abend or a user abend.
The SVC dump file is required to analyze the inter-region communication between
the master address space and subordinate address spaces.

---

**SVCDUMP**

Specifies that this is an SVCDUMP statement.

- **YES**  IMS Index Builder requests an SVC dump file.
- **NO**  IMS Index Builder does not request an SVC dump file.

If the SVCDUMP statement is not specified, YES is the default.

**Tip:** You can override the default value in the runtime parameters module. See the
explanation of the IIURDFTL SVCDUMP entry in Chapter 7, “Defining runtime
parameters,” on page 103.

---

**TMRWAIT control statement**

The TMRWAIT control statement specifies the IIUTMRXT timeout value in
seconds.

The IIUTMRXT module of IMS Index Builder tracks the progress of the process
and issues a 618 event log message every 15 seconds. When no progress is
observed for a certain amount of time, a 619 event log message is issued and each
process is terminated forcibly.

If this statement is not specified, the default value (300 seconds) is used.

---

**TMRWAIT**

Specifies that this is a TMRWAIT statement.

- **nnnn**  Is the timeout value in seconds, in the range of 0 - 9999. If you specify 0,
  timeout will not occur.

**Tips:**

- If the completion code for an index is 722, IMS Index Builder might have ended
due to a timeout. In this case, you can retry the job after adding this control
statement to make the wait time longer.
- You can override the default value in the runtime parameters module. See the
  explanation of the IIURDFTL TMRWAIT entry in Chapter 7, “Defining runtime
  parameters,” on page 103.

---

**TOSIDBD control statement**

The TOSIDBD control statement issues the `/DBDUMP (/DBD) command of IMS
for the prime database before the scan process starts.
The /DBD command is issued only when the IBSCAN keyword is specified on the INPUT control statement. IBSCAN is the default keyword for the INPUT control statement.

**Requirement:** The command is issued by using the IMS Tools Online System Interface. To specify the TOSIDBD control statement, you must also specify the TOSIXCFGRP control statement.

If the BLD_PRIMARY, BLD_ILDS, or BLD_ALL keyword is specified on the PROC control statement for a HALDB, /DBRECOVERY (/DBR) commands must be issued instead of /DBD commands. In such a case, specify the TOSIDBR control statement instead of the TOSIDBD control statement.

**TOSIDBD**
- **NO** IMS Index Builder does not issue /DBD commands.
- **YES** IMS Index Builder issues /DBD commands.

The following keywords specify the parameters that are issued with the /DBD commands.

**LOCAL**
The /DBD command is issued to each IMS subsystem that belongs to the XCF group of the IMS Tools Online System Interface.

**GLOBAL**
The /DBD command is issued with the GLOBAL keyword to one of the IMS subsystems that belongs to the XCF group.

**NOPFA**
The NOPFA parameter is specified on all /DBD commands, which means that the Read Only flag in the RECON data set is not set.

**NOFEOV**
The NOFEOV parameter is specified on all /DBD commands.

**LASTFEOV**
The NOFEOV parameter is specified on all /DBD commands except for the last one.

**FEOV** The NOFEOV parameter is not specified on the /DBD commands.

For more information about the /DBD command and its parameters, see the topic "/DBDUMP command" in *IMS Commands*.

If this statement is not specified, NO is the default. If only TOSIDBD YES is specified, TOSIDBD YES, LOCAL, NOFEOV is used.

For a HALDB, the /DBD command is issued only for the HALDB master database.
Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFT TOSIDBD entry in Chapter 7, “Defining runtime parameters,” on page 103.

TOSIDBR control statement
The TOSIDBR control statement issues the /DBRECOVERY (/DBR) command of IMS for each index to be built before the build process starts.

Requirement: The command is issued by using the IMS Tools Online System Interface. To specify the TOSIDBR control statement, you must also specify the TOSIXCFGR0 control statement.

TOSIDBR
   Specifies that this is a TOSIDBR statement.

NO
   IMS Index Builder does not issue /DBR commands.

YES
   IMS Index Builder issues /DBR commands.

The following keywords specify the parameters that are issued with the /DBR commands.

LOCAL
   The /DBR command is issued to each IMS subsystem that belongs to the XCF group of the IMS Tools Online System Interface.

GLOBAL
   The /DBR command is issued with the GLOBAL keyword to one of the IMS subsystems that belongs to the XCF group.

NOPFA
   The NOPFA parameter is specified on all /DBR commands, which means that the Prohibit Authorization flag in the RECON data set is not set.

NOFE0V
   The NOFE0V parameter is specified on all /DBR commands.

LASTFE0V
   The NOFE0V parameter is specified on all /DBR commands except for the last one.

FEOV
   The NOFE0V parameter is not specified on the /DBR commands.

For more information about the /DBR command and its parameters, see the topic “/DBRECOVERY command” in IMS Commands.

If this statement is not specified, NO is the default. If only TOSIDBR YES is specified, TOSIDBR YES, LOCAL, NOFE0V is used.

For a HALDB, if the BLD_PRIMARY, BLD_ILDS, or BLD_ALL keyword is specified on the PROC control statement and TOSIDBR YES is specified, the /DBR
command is issued for the HALDB master database. If a list of partition names is provided on the PARTNAME control statement, the /DBR command is issued for each HALDB partition in the list instead of the HALDB master database.

For each PSINDEX to be built, the /DBR command is issued for the PSINDEX master database. If a list of partition names is provided on the INDEX control statement, the /DBR command is issued for each PSINDEX partition in the list instead of the PSINDEX master database.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFTL TOSIDBR entry in Chapter 7, “Defining runtime parameters,” on page 103.

TOSISTA control statement

The TOSISTA control statement issues the /START (/STA) command of IMS for the prime database and each index after the build process.

Requirement: The command is issued by using the IMS Tools Online System Interface. To specify the TOSISTA control statement, you must also specify the TOSIXCFGGRP control statement.

TOSISTA

Specifies that this is a TOSISTA statement.

NO IMS Index Builder does not issue /STA commands.

YES IMS Index Builder issues /STA commands.

INDEX IMS Index Builder issues /STA commands only for the built indexes.

ALL IMS Index Builder issues /STA commands for the prime database and the built indexes.

The following keywords specify the parameters that are issued with the /STA commands.

LOCAL

The /STA command is issued to each IMS subsystem that belongs to the XCF group of the IMS Tools Online System Interface.

GLOBAL

The /STA command is issued with the GLOBAL keyword to one of the IMS subsystems that belongs to the XCF group.

If this statement is not specified, NO is the default. If only TOSISTA YES is specified, TOSISTA YES, INDEX, LOCAL is used.
If a list of partition names is specified on the PARTNAME control statement or the INDEX control statement, the /STA commands are issued for the partitions in the list. Otherwise, the /STA command is issued for the HALDB master database or PSINDEX master database.

**Tip:** You can override the default value in the runtime parameters module. See the explanation of the IIURDFT L TOSISTA entry in Chapter 7, “Defining runtime parameters,” on page 103.

**TOSIWAIT control statement**
The TOSIWAIT control statement specifies the IMS Tools Online System Interface response timeout value.

```
TOSIWAIT nn
```

**TOSIWAIT**
Specifies that this is a TOSIWAIT statement.

**nn**
Specifies the timeout value in seconds in the range of 0 - 9999.

If this statement is not specified, 300 is the default.

**Tip:** You can override the default value in the runtime parameters module. See the explanation of the IIURDFT L TOSIWAIT entry in Chapter 7, “Defining runtime parameters,” on page 103.

**TOSIXCFGRP control statement**
The TOSIXCFGRP control statement specifies the XCF group name for the IMS Tools Online System Interface.

```
TOSIXCFGRP group_name
```

**TOSIXCFGRP**
Specifies that this is a TOSIXCFGRP statement.

**NO**
The IMS Tools Online System Interface is not used.

**group_name**
The XCF group name that is defined by the XCFGROUP parameter in the IMS Tools Online System Interface PROCLIB member for each IMS subsystem. The name is a 1 - 5 alphanumeric character string.

If this statement is not specified, NO is the default.

**Tip:** You can override the default value in the runtime parameters module. See the explanation of the IIURDFT L TOSIXCFGRP entry in Chapter 7, “Defining runtime parameters,” on page 103.
VIC control statement
The VIC control statement specifies whether IMS Index Builder issues the DBRC NOTIFY.UIC command to turn off the IC NEEDED flag.

The VIC control statement works with the following types of indexes:
- Primary index of HIDAM
- Secondary index of non-HALDB
- PSINDEX

VIC Specifies that this is a VIC statement.
NO IMS Index Builder does not issue the DBRC NOTIFY.UIC command.
YES IMS Index Builder issues the DBRC NOTIFY.UIC command.
udata Specifies the UDATA parameter of the NOTIFY.UIC command. The maximum length of the string is 64 characters.

For example, you can specify this statement in the following form:
VIC YES,IIU.UIC

If the VIC statement is not specified, NO is the default.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT VIC entry in Chapter 7, “Defining runtime parameters,” on page 103.

WFPDEL control statement
The WFPDEL control statement specifies whether IMS Index Builder deletes the WFP data sets after loading indexes.

This statement is in effect when the INPUT HPSRWFP statement is specified.

WFPDEL Specifies that this is a WFPDEL statement.
NO IMS Index Builder does not delete the WFP data sets.
YES IMS Index Builder deletes the WFP data sets after indexes are built successfully.
FORCE

IMS Index Builder deletes the WFP data sets regardless of whether indexes are built successfully.

If the WFPDEL statement is not specified, NO is the default.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT WFPDEL entry in Chapter 7, “Defining runtime parameters,” on page 103.

WFPHLQ control statement

The WFPHLQ control statement specifies the high-level qualifier of the WFP data sets that were generated by IMS High Performance Load.

This control statement is required when the INPUT HPSRWFP statement is specified.

```
WFPHLQ hlq
```

WFPHLQ

Specifies that this is a WFPHLQ statement.

hlq

Specifies the high-level qualifier of the WFP data sets for input. The maximum length is 32 characters.

The data set hlq.partname.WFP for each HALDB partition must be generated by IMS High Performance Load in advance. IMS Index Builder allocates each data set dynamically.

Tip: You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT WFPHLQ entry in Chapter 7, “Defining runtime parameters,” on page 103.

ZIIPMODE control statement

The ZIIPMODE control statement specifies whether IMS Index Builder offloads eligible workloads to zIIP processors.

```
ZIIPMODE COND
```

ZIIPMODE

Specifies that this is a ZIIPMODE statement.

COND

Offloads eligible IBSCAN VSAM I/O workloads to zIIP processors. The following conditions must be satisfied. Otherwise, the job is run on the main CPs.

• ZIIPMODE COND statement is specified.
• The SGLXLOAD library of IMS Tools Base is specified to the STEPLIB DD.
• zIIP processors are available.
• The data sets of the prime database that IMS Index Builder scans are VSAM ESDS.

**NEVER**

Does not offload any workload to zIIP processors.

If the ZIIPMODE statement is not specified, NEVER is the default.

**Tip:** You can override the default value in the runtime parameters module. See the explanation of the IIURDFLT ZIIPMODE entry in Chapter 7, “Defining runtime parameters,” on page 103.
Control statements incompatible with Version 3.1

Some control statements that are supported in Version 2.3 are incompatible with Version 3.1.

The following statements are not supported by IMS Index Builder Version 3.1

- ALTER statement
- CATLG statement
- EXEC statement
- NDXIOBUF statement
- SORTE15 statement
- SORTE35 statement
- SORTID statement
- SORTOUT statement
- SORTSTAT statement
The JCL and input statements by task table summarizes the main input and output requirements for several tasks that you can do with IMS Index Builder.

### Table 3. JCL and input statements by task

<table>
<thead>
<tr>
<th>Task</th>
<th>DFSURWF1 DD (input)</th>
<th>DFSURIDX DD (input)</th>
<th>IMSALT DD (input)</th>
<th>OUTPUT DD (output)</th>
<th>PROC control statement value</th>
<th>INPUT control statement value</th>
<th>OUTPUT control statement value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Build secondary indexes by using output from a reorganization</td>
<td>required</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>BLD_SECONDARY (one or more)</td>
<td>DFSURWF1</td>
<td>N/A</td>
</tr>
<tr>
<td>Extract data needed for prefix resolution (split function)</td>
<td>required</td>
<td>N/A</td>
<td>N/A</td>
<td>required</td>
<td>BLD_SECONDARY (one or more)</td>
<td>DFSURWF1</td>
<td>required</td>
</tr>
<tr>
<td>Build secondary indexes by using output from prefix resolution</td>
<td>N/A</td>
<td>required</td>
<td>N/A</td>
<td>N/A</td>
<td>BLD_SECONDARY (one or more)</td>
<td>DFSURIDX</td>
<td>N/A</td>
</tr>
<tr>
<td>Build partitioned secondary indexes (PSINDEXes) by using output from</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>BLD_SECONDARY (one or more)</td>
<td>HPSRWFP</td>
<td>N/A</td>
</tr>
<tr>
<td>the IMS HP Load utility (HPSRWFP) as input</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initialize empty primary and secondary indexes and ILDS</td>
<td>N/A</td>
<td>required</td>
<td>N/A</td>
<td>N/A</td>
<td>BLD_SECONDARY BLD_PRIMARY BLD_ILDS BLD_ALL</td>
<td>DFSURIDX</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Chapter 6. Defining the subordinate address space procedure

IMS Index Builder can initiate multiple subordinate address spaces to perform scans, to perform sorts, and to support the API for IMS Index Builder integration with IMS Database Recovery Facility (DRF). Each subordinate address space is a started task that by default executes the subordinate address space procedure named IIUBSRT.

About this task

Follow the procedure to define the IIUBSRT started task.

Procedure

1. Ensure that the JCL that runs IMS Index Builder has the IMS Index Builder load library in the STEPLIB concatenation.

   The master address space determines its load library’s data set name from its STEPLIB. This data set name is passed in the IIUSTEPL parameter when the master address space starts the IIUBSRT procedure. This process ensures that the master and subordinate address spaces use the same software level.

2. Define the IIUBSRT started task JCL, as in the following example. This example is also placed into SIIUPROC during the SMP/E installation.

```sas
//****************************************************************************
/* Standard SAS Procedure for IMS IB V3 R1                                    */
/*****************************************************************************/
//IIUBSRT PROC IIUSTEPL=
//IEFPROC EXEC PGM=IIUBSCTL,REGION=233M,TIME=1439,PERFORM=60
//STEPLIB DD DISP=SHR,DSN=&IIUSTEPL
//IIUPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*  
//SYSUDUMP DD DUMMY
/*****************************************************************************/
```

Figure 19. Example of subordinate address space procedure (IIUBSRT)

You can add the DFSPARM DD statement to provide sort control options. Because you can use any compatible sort product with IMS Index Builder, you must refer to the sort product documentation to understand the sort control options that are provided by the DFSPARM DD statement.

The following table shows the default procedure name and the default job names for subordinate address spaces. You can override the default names by using the runtime parameters module (IIURPRMS) or by providing the names with the control statements in each IMS Index Builder job.

<table>
<thead>
<tr>
<th>Procedure name</th>
<th>Default name</th>
<th>Override with the runtime parameters module (IIURPRMS)</th>
<th>Override with a control statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Procedure name</td>
<td>IIUBSRT</td>
<td>&quot;IIURSORT entry&quot; on page 106</td>
<td>&quot;IIURSORT control statement&quot; on page 79</td>
</tr>
</tbody>
</table>
Table 4. Procedure name and job names for subordinate address spaces (continued)

<table>
<thead>
<tr>
<th>Job name for the sequential scan (only for HALDBs)</th>
<th>Default name</th>
<th>Override with the runtime parameters module (IIURPRMS)</th>
<th>Override with a control statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIUSCANS</td>
<td></td>
<td>“IIURDFLT SCAN entry” on page 108</td>
<td>“JOBNSCAN control statement” on page 82</td>
</tr>
<tr>
<td>Job name for the index sort</td>
<td>IIUSORTS</td>
<td>“IIURDFLT SORTP entry” on page 108</td>
<td>“JOBNSORT control statement” on page 83</td>
</tr>
<tr>
<td>Job name for IMS Database Recovery Facility integration</td>
<td>IIUAPIFC</td>
<td>“IIURDFLT APIP entry” on page 106</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**Related concepts:**
Chapter 7, “Defining runtime parameters,” on page 103
RACF considerations

Before using the subordinate address space or stripe data sets, make sure that you have established proper security authorizations and access.

For subordinate address spaces

The subordinate address space's user ID is obtained in two phases, as described in the following list:

1. Before the IMS Index Builder code is dispatched in the subordinate address space and has the opportunity to set its security identity, the operating system attempts to use the user ID from the profile definitions in the security system's STARTED class as the user ID for these started tasks.

   If you use the IMS Index Builder default job names, you can use the following RDEFINE to specify the user ID. The STARTED class is RACLIST'ed and must be refreshed after you make updates.

   RDEFINE STARTED IIU*.* STDATA(USER(userid))
   SETROPTS RACLIST(STARTED) REFRESH

   If you use the IIURPRMS module to override the default job names, you must define the STARTED class profiles that correspond to the installation names.

   Defining the IMS Index Builder load library to the security system with UACC(READ) is recommended so that all user IDs have access to the STEPLIB. If you do not want to define UACC(READ), you must set READ access to the load library for the user ID or for the group associated with the STARTED class profile.

   Improper security system definitions result in errors when opening STEPLIB during subordinate address space initialization. The subordinate address space can fail with an ABENDS913, or the master address space can terminate by issuing an IIUB050E error message with REASON=799.

2. After the IMS Index Builder code is dispatched in the subordinate address space, IMS Index Builder processing determines the user ID under which the master address space runs. The subordinate address space changes its own initial user ID to match that of the master address space. Once this change occurs, the subordinate address space runs with the same security authorities as the master.

For stripe data sets

IMS Index Builder uses stripe data sets to pass data between its address spaces; the data includes the WTO messages, the sort messages, the index records, and the duplicate keys. The stripe data sets exist only for the duration of that particular IMS Index Builder run.

The user ID associated with the master address space requires ALTER access to these data sets. You associate ALTER access to the stripe data sets by defining an IIU.STRIPE.* DATASET profile with UACC(ALTER). If your installation security policies do not allow UACC(ALTER), you must add each user that executes IMS Index Builder to this DATASET profile's access list with ALTER access.

By default, stripe data sets are allocated with the high-level qualifiers IIU.STRIPE. You can override this default by using the IIURDFLT CLASS setting in the IIURPRMS parameter module. However, if you override the default high-level qualifiers, you must define a DATASET profile corresponding to the installation's qualifiers.
Related concepts:

Chapter 7, “Defining runtime parameters,” on page 103
Chapter 7. Defining runtime parameters

The IIURPRMS module can be used to specify IMS Index Builder runtime parameters and to override defaults.

The IIURPRMS module allows the following:
- Override the default procedure name that is used to start subordinate address spaces
- Override the default job names that are used when starting subordinate address spaces
- Override the default high-level qualifier that is used for creating data sets
- Specify the name of the IMS Tools Knowledge Base server

You can use the JCL stream IUI.UICNTL(IUISMP10) as a sample when you replace the IIURPRMS module. It assembles the user-specified code and links it into the IMS Index Builder load library to replace IIURPRMS or into a separate load library to override it.

The following figure shows an example for coding the IIURPRMS module. Here, the procedure name to start subordinate address spaces is overridden, and the job names of the scan address spaces and the sort address space are overridden.
The IIURPRMS module is a table.
Each entry in this table is 52 bytes or 104 bytes
The first entry must be IIURPRMS
The last entry must be IIUREND
IIURPRMS can optionally contain an IIURSORT entry
IIURPRMS can optionally contain multiple IIURDFLT entries
A complete definition minimally contains IIURPRMS and IIUREND entries
The keywords IIURPRMS, IIURSORT, IIURDFLT, and IIUREND must be coded in the entry's first byte
Blank spaces are optional following the IIURSORT and IIURDFLT keywords

The default IIURPRMS module contains only the IIURPRMS and IIUREND entries. You can add the IIURSORT entry and the IIURDFLT entries to override each default.

Subsections:
- “IIURSORT entry” on page 106
- “IIURDFLT APIP entry” on page 106
- “IIURDFLT CLASS entry” on page 106
- “IIURDFLT COMPAUTH entry” on page 106
- “IIURDFLT DBAUTH entry” on page 106
- “IIURDFLT DFSDF entry” on page 107
- “IIURDFLT ILDS entry” on page 107
- “IIURDFLT ITKBLOAD entry” on page 107
- “IIURDFLT ITKBSRVR entry” on page 107
- “IIURDFLT MAXTASKS entry” on page 107
- “IIURDFLT PROGMON entry” on page 107
- “IIURDFLT SCAN entry” on page 108
- “IIURDFLT SIDXBUF entry” on page 108
- “IIURDFLT SORTP entry” on page 108
- “IIURDFLT SORTFSZ entry” on page 108
- “IIURDFLT STRIPE entry” on page 108
- “IIURDFLT SVCDUMP entry” on page 108
- “IIURDFLT TMRWAIT entry” on page 109
- “IIURDFLT TOSIDBD entry” on page 109
- “IIURDFLT TOSIDBR entry” on page 109
- “IIURDFLT TOSISTA entry” on page 109
- “IIURDFLT TOSIWAIT entry” on page 109
- “IIURDFLT TOSIXCFGRP entry” on page 109
- “IIURDFLT VIC entry” on page 109
- “IIURDFLT WFPDEL entry” on page 110
- “IIURDFLT WFPHLQ entry” on page 110
- “IIURDFLT ZIIPMODE entry” on page 110
- “IIURDFLT entries not supported by Version 3.1” on page 110
**IIURSORT entry**

By default, IMS Index Builder starts subordinate address spaces using the IIUBSRT procedure. The IIURSORT entry can be used to specify a different procedure name.

The following IIURPRMS example entry can be used to define IIUSAS as the procedure to start SORT and SCAN subordinate address spaces:

```
DC CL52'IIURSORT IIUSAS'
```

**IIURDFTL APIP entry**

By default, IMS Index Builder starts a subordinate address space for DRF interface using IIUAPIFC as job name. The IIURDFTL APIP entry can be used to specify a different job name.

The following IIURPRMS example entry can be used to define IIUAPI as the job name:

```
DC CL52'IIURDFTL APIP IIUAPI'
```

**IIURDFTL CLASS entry**

IMS Index Builder uses data sets to pass data between the sort, scan, and master address spaces. These data sets exist only for the duration of the run. By default, they are allocated with the high-level qualifier IIU.STRIPE. The IIURDFTL CLASS entry can be used to specify a different high-level qualifier. The maximum length is 17 characters.

The following IIURPRMS example entry can be used to define UTILITY.IB.STRIPE as the high-level qualifier for these data sets:

```
DC CL52'IIURDFTL CLASS UTILITY.IB.STRIPE'
```

Here is another example:

```
DC CL52'IIURDFTL CLASS IIU.STRP.%TIME'
```

IMS Index Builder replaces %TIME with a time stamp (Thhmmssst). The high-level qualifier is IIU.STRP:T2359599 and the maximum length is 17 characters. By specifying in this format, you can avoid duplicate data set names.

Notes that the user ID processing the IMS Index Builder requires RACF ALTER access for these data sets.

**IIURDFTL COMPAUTH entry**

Use the IIURDFTL COMPAUTH entry to specify whether the segment compression exit is called in supervisor state.

For details, see “COMPAUTH control statement” on page 77.

**IIURDFTL DBAUTH entry**

Use the IIURDFTL DBAUTH entry to specify whether to issue the DBRC authorization request.
For details, see “DBAUTH control statement” on page 78.

**IIURD FLT DFS DF entry**

Use the IIURD FLT DFS DF entry to specify the 3-character suffix for the DFSDFxxx member in the PROCLIB data set.

For details, see “DFSDF control statement” on page 79.

**IIURD FLT ILDS entry**

Use the IIURD FLT ILDS entry to specify whether to build ILE records or to initialize the data set without building any ILE records.

For details, see “ILDS control statement” on page 80.

**IIURD FLT ITKBLOAD entry**

Use the IIURD FLT ITKBLOAD entry to specify the name of the load module library for IMS Tools Knowledge Base.

This entry is 104 bytes to contain a library name up to 44 characters. The following IIURPRMS example entry can be used to specify a library name.

```
DC CL104'IIURDFLT ITKBLOAD IMSTOOLS.SHKTLOAD'
```

For details, see “ITKBLOAD control statement” on page 81.

**IIURD FLT ITKB SRVR entry**

Use the IIURD FLT ITKBSRVR entry to specify the name of the IMS Tools Knowledge Base server where the IMS Index Builder reports are to be stored and viewed. The server name is a 1- to 8-byte value.

The following IIURPRMS example entry can be used to specify UTILITKB as the server name:

```
DC CL52'IIURDFLT ITKBSRVR UTILITKB'
```

For details, see “ITKBSRVR control statement” on page 82.

**IIURD FLT MAXT ASKS entry**

Use the IIURD FLT MAXT ASKS entry to customize the IMS Index Builder run.

For details, see “MAXT ASKS control statement” on page 83.

**IIURD FLT PROGMON entry**

Use the IIURD FLT PROGMON entry to specify the interval for issuing WTO IIUB119I messages, which are messages to indicate the progress of the load process for each secondary index.

For details, see “PROGMON control statement” on page 85.
**IIURDFLT SCAN entry**

When processing HALDB databases, IMS Index Builder starts scanning subordinate address spaces by using the default job name IIUSCANS. You can use the IIURDFLT SCAN entry to specify a different job name.

The following IIURPRMS example entry can be used to define IIUSCNAS as the job name for non-HALDB scan address spaces:

```
DC CL52 'IIURDFLT SCAN IIUSCNAS'
```

**IIURDFLT SIDXBUF entry**

Use the IIURDFLT SIDXBUF entry to specify the number of buffers that are to be used in the load process of secondary indexes.

For details, see “SIDXBUF control statement” on page 86.

You can also change this value by adding the SIDXBUF control statement to each job.

**IIURDFLT SORTP entry**

By default, IMS Index Builder starts sort subordinate address spaces using IIUSORTS as a job name. The IIURDFLT SORTP entry can be used to specify a different job name.

The following IIURPRMS example entry can be used to define IIUSRAS as the sort address space job name:

```
DC CL52 'IIURDFLT SORTP IIUSRAS'
```

**IIURDFLT SORTFSZ entry**

Use the IIURDFLT SORTFSZ entry to specify striped sort mode and the number of records in each sort stripe.

For details, see “SORTFSZ control statement” on page 87.

You can also change this value by adding the SORTFSZ control statement to each job.

**IIURDFLT STRIPE entry**

Use the IIURDFLT STRIPE entry to specify the UNIT value and the volume count for allocating stripe data sets.

For details, see “STRIPE control statement” on page 88.

**IIURDFLT SVCDUMP entry**

Use the IIURDFLT SVCDUMP entry to specify whether an SVC dump file is needed.

For details, see “SVCDUMP control statement” on page 88.
IIURDFLT TMRWAIT entry

Use the IIURDFLT TMRWAIT entry to specify the IIUTMRXT timeout value in seconds. The default value is 300.

You can also change this value by adding the TMRWAIT control statement to each job.

For details, see “TMRWAIT control statement” on page 89.

IIURDFLT TOSIDBD entry

Use the IIURDFLT TOSIDBD entry to specify whether the IMS /DBDUMP commands are issued automatically.

For details, see “TOSIDBD control statement” on page 89.

IIURDFLT TOSIDBR entry

Use the IIURDFLT TOSIDBR entry to specify whether the IMS /DBRECOVERY commands are issued automatically.

For details, see “TOSIDBR control statement” on page 91.

IIURDFLT TOSISTA entry

Use the IIURDFLT TOSISTA entry to specify whether the IMS /START DB commands are issued automatically.

For details, see “TOSISTA control statement” on page 92.

IIURDFLT TOSIWAIT entry

Use the IIURDFLT TOSIWAIT entry to specify the response timeout value.

For details, see “TOSIWAIT control statement” on page 93.

IIURDFLT TOSIXCFGGRP entry

Use the IIURDFLT TOSIXCFGGRP entry to specify the XCF group name for the IMS Tools Online System Interface.

For details, see “TOSIXCFGGRP control statement” on page 93.

IIURDFLT VIC entry

Use the IIURDFLT VIC entry to specify whether the DBRC NOTIFY.UIC command is needed.

For details, see “VIC control statement” on page 94.

You can also change this value by adding the VIC control statement to each job.
IIURDFTL WFPDEL entry

Use the IIURDFTL WFPDEL entry to specify whether the WFP data sets are deleted after loading indexes.

For details, see “WFPDEL control statement” on page 94.

IIURDFTL WFPHLQ entry

Use the IIURDFTL WFPHLQ entry to specify the high-level qualifier of the WFP data sets.

For details, see “WFPHLQ control statement” on page 95.

IIURDFTL ZIIPMODE entry

Use the IIURDFTL ZIIPMODE entry to specify whether IMS Index Builder offloads eligible workloads to zIIP processors.

For details, see “ZIIPMODE control statement” on page 95.

IIURDFTL entries not supported by Version 3.1

Some IIURDFTL entries that are supported in Version 2.3 are incompatible with Version 3.1. The following are not supported by IMS Index Builder Version 3.1:

- IIURDFTL ALTER
- IIURDFTL CATLG
- IIURDFTL DUMP
- IIURDFTL EXEC
- IIURDFTL NDXIOBUF
- IIURDFTL NOFIX
- IIURDFTL PRTROOT
- IIURDFTL SCANH
- IIURDFTL SORTE15
- IIURDFTL SORTE35
- IIURDFTL SORTOUT
- IIURDFTL SORTSTAT
- IIURDFTL STEPLIB
- IIURDFTL TRACE
Chapter 8. Index records with duplicate keys

IMS Index Builder does not acknowledge key uniqueness until all records of a target index are sorted and until index load is initiated. Before loading a record, IMS Index Builder checks for key duplication.

If the index is defined with unique keys and one or more duplicate keys are encountered, the completion code for the index is A30 and the completion messages show the count of the loaded records and the count of the records that are not loaded due to key duplication. The records with duplicate keys are stored in the stripes with LLQ "DUPLKEYS". If DBRC is active, a DBRC NOTIFY REORG is not done.

The DUPLKEYS stripe can be used for control purposes and subsequent record processing, and will be retained after the job ends.
Part 3. Troubleshooting

IMS Index Builder provides resources that can be used to help you troubleshoot and diagnose IMS Index Builder problems.

Topics:
- Chapter 9, “Gathering diagnostic information,” on page 115
- Chapter 10, “IMS Index Builder output and control flow,” on page 117
- Chapter 11, “IMS Index Builder reason codes,” on page 121
- Chapter 12, “IMS Index Builder messages,” on page 145
- Chapter 13, “How to look up message explanations,” on page 161
- Chapter 14, “IMS Index Builder abend codes,” on page 163
- Chapter 15, “Diagnostics Aid,” on page 165
Chapter 9. Gathering diagnostic information

You can use the diagnostic documentation guidelines for gathering proper diagnostic documentation when reporting a problem with IMS Index Builder to IBM Software Support.

Procedure

Provide the following information for every IMS Index Builder problem:

- Problem description
- Product release number and the number of the last PTF (program temporary fix) that was installed
- The IMS release number

Additional documentation is also required for various incident types. In general, gather the suggested documentation for the following incident types:

- When IMS Index Builder ran successfully, but appears to have processed incorrectly
  - The entire job output
  - The IIUDUMP dump output
  - Any documentation supporting the incorrect processing
  - The DBD source
  - The Diagnostics Aid (IIUUDIAG) report

- When an IMS Index Builder job abends
  - The entire failing job output
  - The SYSUDUMP dump output
  - The SVC dump output to analyze the inter-region communication between the master address space and subordinate address spaces
  - The DBD source
  - The Diagnostics Aid (IIUUDIAG) report

Related information:

[Chapter 15, “Diagnostics Aid,” on page 165]
Chapter 10. IMS Index Builder output and control flow

This section describes the IMS Index Builder output file and control flow events.

Topics:

- “IMS Index Builder output files” on page 118
- “Control flow events reported on IIUSNAP” on page 119
IMS Index Builder output files

IMS Index Builder has a main output file and, depending on the type of processing, up to three additional output files.

Main report file

File IIUPRINT is the main output file. It is formatted so as to be easily browsed on a 3278 type terminal.

In the file, every message starts with an ID in the form:

IIUBnnnt

Where:
- nnn is a 3-digit decimal number
- t is a character that can be I for information, W for warning, or E for error

IMS Index Builder prints the following information in file IIUPRINT:
- Processing options that you selected
- Layout of the indexes to be built
- Content of file DFSVSAMP when scan is used on the physical database and DFSVSAMP is not provided by you
- The number of scanned segments that are in the prime database
- The number of records for indexes that were successfully built
- Diagnostic information for indexes that could not be built

Sort output file

The output from each sort is printed in the file that is specified by the IIUSOUT DD statement.

When sort problems are encountered, inspect the IIUSOUT data set.

Process events output file

File IIUSNAP is used to snap out time-sequenced Index Builder starting and stopping messages.

Under IMS Database Recovery Facility, IIUCA PT is the process events output file.
Control flow events reported on IIUSNAP

IMS Index Builder records major processing events. You can monitor the progress of processing so that you are aware of the type of processing that IMS Index Builder has selected. In some cases, information about processing events can be used to perform diagnostic functions.

The processing events that IMS Index Builder records are shown in the following list:

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Event Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>601</td>
<td>SCAN PROCESS INITIALIZATION IN PROGRESS</td>
</tr>
<tr>
<td>602</td>
<td>SORT PROCESS INITIALIZATION IN PROGRESS</td>
</tr>
<tr>
<td>603</td>
<td>SEQUENTIAL SERIAL SCANNER READY</td>
</tr>
<tr>
<td>604</td>
<td>SAS SORT SERVER READY</td>
</tr>
<tr>
<td>605</td>
<td>SAS SORT BROKER READY</td>
</tr>
<tr>
<td>606</td>
<td>MESSAGE RELEASER TERMINATED</td>
</tr>
<tr>
<td>607</td>
<td>MESSENGER LISTENING</td>
</tr>
<tr>
<td>608</td>
<td>SORT DRIVER READY</td>
</tr>
<tr>
<td>609</td>
<td>SAS PRIME DB SCANNER READY</td>
</tr>
<tr>
<td>610</td>
<td>INITIALIZER LISTENING</td>
</tr>
<tr>
<td>611</td>
<td>SAS SORT BROKER PROCESSING INDEX STRIPE</td>
</tr>
<tr>
<td>612</td>
<td>EXTENDER LISTENING</td>
</tr>
<tr>
<td>613</td>
<td>HIERARCHICAL DLI SCANNER READY</td>
</tr>
<tr>
<td>614</td>
<td>INDEX BUILD PLANNER STARTED</td>
</tr>
<tr>
<td>615</td>
<td>PLANNING PROCESS INITIATED</td>
</tr>
<tr>
<td>616</td>
<td>BUILD PLAN ENQUEUE TO CREATE</td>
</tr>
<tr>
<td>617</td>
<td>INDEX BUILD PLANNER ENDED</td>
</tr>
<tr>
<td>618</td>
<td>SCAN IN PROGRESS</td>
</tr>
<tr>
<td>619</td>
<td>SORT IN PROGRESS</td>
</tr>
<tr>
<td>620</td>
<td>SORT DRIVER STRIPING</td>
</tr>
<tr>
<td>621</td>
<td>INDEX BUILD PLAN PROCESSOR LISTENING</td>
</tr>
<tr>
<td>622</td>
<td>SAS TERMINATION CHECK OUT</td>
</tr>
<tr>
<td>623</td>
<td>HIERARCHICAL HPU SCANNER READY</td>
</tr>
<tr>
<td>624</td>
<td>ALL SAS TERMINATED</td>
</tr>
<tr>
<td>625</td>
<td>DFSURWFI/HPSRSDX MAS READER READY</td>
</tr>
<tr>
<td>626</td>
<td>DFSURIDX DIRECT INDEX LOADER READY</td>
</tr>
<tr>
<td>627</td>
<td>BUILD PLAN NOT COMPLETE</td>
</tr>
<tr>
<td>628</td>
<td>DFSURWFI/HPSRSDX SAS TWIN READY</td>
</tr>
<tr>
<td>629</td>
<td>PARA STRIPED SORT SERVER READY</td>
</tr>
<tr>
<td>633</td>
<td>SAS TERMINATION COMPLETED</td>
</tr>
<tr>
<td>635</td>
<td>INDEX STRIPE CREATOR READY</td>
</tr>
<tr>
<td>636</td>
<td>SEQUENTIAL SERIAL SCANNER TWINNING</td>
</tr>
<tr>
<td>637</td>
<td>INDEX BUFFER RELEASER TERMINATED</td>
</tr>
<tr>
<td>638</td>
<td>MAS DATA BUFFER RELEASER TERMINATED</td>
</tr>
</tbody>
</table>
SAS DATA BUFFER RELEASER
TERMINATED
Chapter 11. IMS Index Builder reason codes

A reason code is a return code that describes the reason for the failure or partial success of an attempted operation. IMS Index Builder generates reason codes to help you troubleshoot and diagnose IMS Index Builder problems.

Topics:

- “IMS Index Builder return codes and error handling” on page 122
- “IMS Index Builder initialization phase primary reason codes” on page 124
- “IMS Index Builder initialization phase secondary reason codes” on page 128
- “IMS Index Builder control flow event codes” on page 130
- “IMS Index Builder processing phase failure reason codes” on page 133
- “IMS Index Builder verification phase failure reason codes” on page 144
IMS Index Builder return codes and error handling

IMS Index Builder generates return codes to indicate the success or failure of a routine. You can use these return codes to influence the execution of succeeding instructions or programs.

IMS Index Builder issues the following return codes.

Table 5. Return codes issued by IMS Index Builder

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>IMS Index Builder completed normally, and all indexes were built successfully.</td>
</tr>
<tr>
<td>4</td>
<td>IMS Index Builder built all indexes that were requested and found nothing to rebuild.</td>
</tr>
<tr>
<td>8</td>
<td>IMS Index Builder built some indexes successfully, but one or more indexes could not be built.</td>
</tr>
<tr>
<td>12</td>
<td>IMS Index Builder could not build any indexes.</td>
</tr>
<tr>
<td>16</td>
<td>IMS Index Builder could not build any indexes. Either an input, JCL, system, or feature error was encountered.</td>
</tr>
<tr>
<td>20</td>
<td>The prime database is not available for scan. No indexes were built.</td>
</tr>
</tbody>
</table>

In general, a return code higher than 4 indicates that IMS Index Builder encountered an error for at least one index that is being built.

Two types of errors can be encountered. The first type prevents all indexes from being built. The second type prevents a particular index from being built.

Errors that prevent building all indexes

Errors that prevent all indexes from being built include:

- Input errors, for example, an invalid control statement in the input file IIUIN
- DBD errors, for example, if an index is not defined in the physical database DBD
- JCL errors, for example, if a required DD name is missing
- System errors, for example, an I/O error when reading input control statements
- IBSCAN errors, for example, if the scan returned a bad status code when scanning the physical database
- Any index-related error that was detected when the index databases are being opened

Errors that prevent building one or more indexes

When IMS Index Builder encounters an index-related error after successfully opening all indexes for output, it stops processing that index.

Some indexes might already be successfully loaded when the error occurs. Check the output in IIUPRINT to determine which, if any, indexes were built.

The following errors are index-related errors:

- Reading an invalid index-related record from file DFSURWF1 or DFSURIDX
- Failing to load an index record
• Receiving a nonzero return code from the sort task that is associated with an index database

External product error messages

IMS Index Builder calls IMS Tools Knowledge Base and High Performance Unload. These applications place their error messages on the SYSOUT data set. The WTOs for these applications are captured on the IIUCA PT data set.
IMS Index Builder initialization phase primary reason codes

The following primary reason codes are issued by the Index Builder initialization subsystem and are reported in message IIUB100E.

064
Explanation: Unsupported IMS version.
User response: IMS V9 must be the lowest IMS version installed.

100
Explanation: Unsupported IMS version.
User response: IMS V9 must be the lowest IMS version installed.

101
Explanation: The initialization module IIUBINIT or IIUSINxx could not load a load module while processing the pre-load list.
User response: Check the STEPLIB concatenation for load module data sets containing all Index Builder modules.

102
Explanation: Control statements that are provided by IIUIN or by the runtime parameters module are in error.
User response: The statement that caused the error is displayed in the Index Builder output report. Correct the error and rerun the job. If the statement that caused the error is not identified, contact IBM Software Support.

103
Explanation: Invalid parameters were entered for control statements provided by IIUIN or the runtime parameters module.
User response: The statement that caused the error is displayed in the Index Builder output report. Correct the error and rerun the job. If the statement that caused the error is not identified, contact IBM Software Support.

104
Explanation: Required data sets could not be allocated during initialization.
User response: Check for MVS messages. If the problem persists, contact IBM Software Support.

105
Explanation: The DBRC parameter value in the INPUT control statement (Y/N) cannot override DBRC system settings.
User response: Check the DBRC system settings, change the Index Builder control statement accordingly, and rerun the job. If the DBRC requirements cannot be satisfied, contact IBM Software Support.

106
Explanation: The RECON1 data set is not provided through the DD statement.
User response: Ensure that the RECON1 DD statement is specified correctly.

107
Explanation: IPR or DRF integration required storage could not be obtained during SAS initiation.
User response: Contact IBM Software Support.

109
Explanation: IDCAMS processing fails for one or more data sets to be rebuilt. IIUBE053E lines denote the data sets in error. Initialization is terminated with RC=16 (X'10').
User response: Analyze error messages for the data set causing the error. This data set might require IDCAMS processing to correct the situation.

110
Explanation: Unable to open IIUIN data set.
User response: Check the IIUIN DD statement in the Index Builder JCL deck. Look for MVS messages. If the IIUIN data set is defined correctly, contact IBM Software Support.

111
Explanation: Unable to open IIUPRINT data set.
User response: Check the IIUPRINT DD statement in the Index Builder JCL deck. Look for accompanying MVS messages. If the IIUPRINT data set was defined correctly, contact IBM Software Support.
112 "OUTPUT" DD STATEMENT MISSING
Explanation: The OUTPUT control statement was used with PROC BLD_PRIMARY and no matching DD statement was provided.
User response: Provide a DD statement to match the OUTPUT statement parameter or remove the OUTPUT control statement and rerun the job. If the problem persists, contact IBM Software Support.

113 UNABLE TO OPEN IIUSOUT DATA SET
Explanation: The SORTOUT DD statement that was specified or that was allocated by Index Builder could not be opened.
User response: Check the IIUSOUT definition, look for accompanying MVS messages, or remove the SORTOUT control statement. If the problem persists, contact IBM Software Support.

114 UNABLE TO ALLOCATE HPSRWFP DATA SET
Explanation: Failed to allocate the HPSRWFP data set.
User response: Locate the IIUB062I message, which shows the return code and the reason code.

115 UNABLE TO ALLOCATE INDEX DBD NAME LIST
Explanation: Storage allocation failed.
User response: Look for accompanying MVS messages. If the problem persists, contact IBM Software Support.

116 UNABLE TO ALLOCATE PART NAME LIST
Explanation: Storage allocation failed.
User response: Look for accompanying MVS messages. If the problem persists, contact IBM Software Support.

117 UNABLE TO ALLOCATE DB ANCHOR LIST
Explanation: Storage allocation failed.
User response: Look for accompanying MVS messages. If the problem persists, contact IBM Software Support.

118 PRIME DATABASE SCAN NOT REQUESTED
Explanation: According to the input control statement provided, no database scan is required. Thus, MAXTASKS values 1 and 2 are meaningless.
User response: The MAXTASKS control statement is ignored.

119 BUILD SECONDARY NOT REQUESTED
Explanation: According to the procedure control statement provided, build secondary is not required. Thus, MAXTASKS is meaningless.
User response: The MAXTASKS control statement is ignored.

120 PRIME DATABASE DBD PROCESSING ERROR
Explanation: A prime database DBD processing error occurred.
User response: Verify that the DBD entered in the PROC control statement is the correct prime DBD, and that it, and all index DBDs referred, are contained in the DBD library that is defined by the IMS DD statement. Check that no DBD names are overridden by concatenation. If the problem persists, contact IBM Software Support.

121 TARGET DATABASE DBD PROCESSING ERROR
Explanation: A target database DBD processing error occurred.
User response: Verify that the DBD entered in the PROC control statement is the correct prime DBD, and that it, and all index DBDs referred, are contained in the DBD library that is defined by the IMS DD statement. Check that no DBD names are overridden by concatenation. If the problem persists, contact IBM Software Support.

122 PRIME DATABASE HAS NO SECONDARY INDEXES
Explanation: A secondary build was requested for a database without secondary indexes. Thus, MAXTASKS is meaningless.
User response: The MAXTASKS control statement is ignored.
123  UNABLE TO OPEN HPSRWFP DATA SET
Explanation: Failed to open the HPSRWFP data set.
User response: Ensure that the data set was generated correctly.

124  HPU OPTION DATA SET PROCESS FAILURE
Explanation: IUOPT or HSSROPT data sets are in error.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

125  DFSVSAMP DATA SET GENERATION FAILURE
Explanation: If DFSVSAMP is not provided by the Index Builder JCL deck or by MDA, Index Builder generates it.
User response: Look for accompanying MVS messages or provide DFSVSAMP. If the problem persists, contact IBM Software Support.

126  DFSCTL DATA SET GENERATION FAILURE
Explanation: The data set was not provided by the user and IMS Index Builder failed to generate it.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

127  HPSRWFP DATA SET PROCESS FAILURE
Explanation: Failed to read the HPSRWFP data set.
User response: Ensure that the data set was generated correctly.

129  PROCESSING MODE INCOMPATIBLE WITH OLR
Explanation: IMS Index Builder is not initiated from STEPLIB. This processing mode is not applicable when OLR is ended.
User response: Complete OLR, or run IMS Index Builder from STEPLIB.

130  PRIME DATABASE AUTHORIZED
Explanation: DBRC processing of the database shows that there are active subsystems.
User response: Run LIST,SUBSYS to obtain the subsystem names, then stop the active subsystems.

131  DBRC API FAILURE
Explanation: An error occurred in the IMS DBRC API call.
User response: See message “IIUB063E” on page 152.

135  DATABASE ALLOCATION FAILURE
Explanation: Required database could not be allocated.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

137  UNABLE TO OBTAIN HALDB PARTITION DEFINITIONS
Explanation: IMS Index Builder failed to obtain HALDB partition definitions form the DBRC RECON data sets.
User response: Check if the database and partitions are registered to the RECON data sets and that the DD statements for the RECON data sets are specified in the IMS Index Builder JCL.

139  DFSVSAMP/DFSCTL DEFINE FAILURE
Explanation: Standard IMS data sets were not provided by the user and could not be defined by IMS Index Builder.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

140  DISPLAY DATABASE ATTRIBUTES FAILURE
Explanation: Processed database attributes are displayed in the Index Builder report.
User response: Look for previous messages that might explain the error and take the appropriate actions. Contact IBM Software Support with appropriate diagnostic documentation.

141  DATABASE DATA SETS ALLOCATION FAILURE
Explanation: Required data sets could not be allocated.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

142  UNABLE TO ALLOCATE SEGMENT STAT ARRAY
Explanation: Storage allocation failed.
User response: Look for accompanying MVS messages. If the problem persists, contact IBM Software Support.
<table>
<thead>
<tr>
<th>Code</th>
<th>Reason Code Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>143</td>
<td>UNABLE TO ALLOCATE AMODE 24 AREA</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> Storage allocation failed.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Look for accompanying MVS messages. If the problem persists, contact IBM Software Support.</td>
</tr>
<tr>
<td>150</td>
<td>ALESERV STOKEN FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> Inter-region communication could not be established.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>152</td>
<td>LOCASCB STOKEN FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> Inter-region communication could not be established.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>153</td>
<td>IMS BATCH REGION FAILED</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The attach task failed without processing.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Review the preceding messages to determine the cause. If the cause cannot be determined, contact IBM Software Support.</td>
</tr>
<tr>
<td>154</td>
<td>SYNC WITH IMS BATCH REGION FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> Timeout occurred while waiting for the completion of the initialization process in the IMS Batch region. The maximum wait time is 30 minutes.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Performance might degrade. Wait a while and rerun the job.</td>
</tr>
<tr>
<td>190</td>
<td>IB BUFFER MANAGER INIT FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The cause of this failure was most likely an IMS Index Builder internal error.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>191</td>
<td>SERVICE TASK ATTACH FAILED</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> Parallel scan subordinate address space files initialization service module attach failed.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> Check the STEPLIB concatenation for complete Index Builder load module data sets, look for additional messages, review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>192</td>
<td>INDEX BUILDER DATA BUFFER ENQUEUE FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The buffer pool might be corrupted, or the receive queue might be not initiated.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> The code assists with the problem determination.</td>
</tr>
<tr>
<td>193</td>
<td>SCANNER QUEUE INITIATION FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> Sequential scan send queue could not be initiated.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> The code assists with the problem determination.</td>
</tr>
<tr>
<td>194</td>
<td>SCANNER INITIATION FAILURE</td>
</tr>
<tr>
<td></td>
<td><strong>Explanation:</strong> The scan initiation module returned non-zero code.</td>
</tr>
<tr>
<td></td>
<td><strong>User response:</strong> The code assists with the problem determination.</td>
</tr>
</tbody>
</table>
## IMS Index Builder initialization phase secondary reason codes

The following secondary reason codes are issued by the IMS Index Builder initialization subsystem and are reported in message IIUB100E.

<table>
<thead>
<tr>
<th>Code</th>
<th>Reason Code</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>INVALID DBD PROCESS FUNCTION CODE</td>
<td>The code assists with the problem determination. Internal error. IIUBDBD was called with an invalid function code.</td>
</tr>
<tr>
<td>302</td>
<td>DYNAMIC ALLOCATION FAILURE</td>
<td>The code assists with the problem determination. Internal error. DBD library could not be dynamically allocated.</td>
</tr>
<tr>
<td>303</td>
<td>OPEN DBD LIBRARY FAILURE</td>
<td>The code assists with the problem determination. Internal error. DBD library could not be opened. MVS log might contain additional messages.</td>
</tr>
<tr>
<td>304</td>
<td>BUILD DYNAMIC PSB FAILURE</td>
<td>The code assists with the problem determination. Internal error. The dynamic PSB could not be built.</td>
</tr>
<tr>
<td>305</td>
<td>UNABLE TO LOAD TARGET DBD</td>
<td>The code assists with the problem determination. Internal error. DBD load failed from DBD library.</td>
</tr>
</tbody>
</table>
| 306  | LOADED DBD HAS INCORRECT NAME | The loaded member is not in the correct DBD format. It might be an MDA member with the DBD name. The possible reasons are:  
  - The MDA library contains a reusable member with the DBD name.  
  - The MDA member with the DBD name is placed in the IMS Index Builder load library.  
User response: Ensure that the DBD library and the MDA library are correct. |
| 307  | INVALID DBD ORGANIZATION | The code assists with the problem determination. Internal error. The DBD organization flags in the DBD do not show HISAM, HDAM, HIDAM, or index organization. |
| 308  | INVALID DBD STRUCTURE | The code assists with the problem determination. Internal error. The DBD named by the BLD_ statement does not have external DBD. |
| 309  | EXT DB REFERENCE FAILURE | The code assists with the problem determination. An externally referenced DBD could not be correctly processed. A secondary index build was requested, but no secondary index is defined in the DBD of the prime database. |
| 310  | DB CHARACTERISTICS FAILURE | The code assists with the problem determination. Internal error. DBD characteristics such as segments and fields could not be correctly processed for the prime database or for some of the externally referenced databases. |
| 311  | OBTAIN DB DSNAME FAILURE | The code assists with the problem determination. Internal error. The DBD data set name(s) of either the prime database or some of the externally referenced databases could not be obtained through dynamic allocation by DDN obtained from the DBD. |
| 312  | BUILD SEC INDEX STR FAILURE | The code assists with the problem determination. Internal error. The control block for a secondary index could not be correctly built due to controversial database characteristics obtained by DBD analysis. |
| 313  | UNSUPPORTED INDEX TYPE | One or more indexes to be built are of an unsupported type. See "IMS Index Builder restrictions" on page 19 for index types that are not supported. |
| 314  | IMS TOOLS CATALOG INTERFACE ERROR | The IMS Tools Catalog Interface returned an error. See message IIUB125E for details. |
| 315  | UNREFERENCED INDEXES | The code assists with the problem determination. Index DBD entered by the INDEX or PSINDEX partition entered by the PARTNAME statement could not be located by the DBD analysis. |
| 318  | BUILD DYNAMIC PSB FAILURE | The code assists with the problem determination. Internal error. The dynamic PSB could not be built. |
**Explanation**: This error was most likely caused by an IMS Index Builder internal error. The code assists with the problem determination.

319  **RANDOMIZER LOAD FAILURE**

**Explanation**: The randomizer module that is needed for HDAM/PHDAM could not be loaded. It might not have been provided in the STEPLIB concatenation.

320  **PIPE TABLE INITIATION FAILED**

**Explanation**: This is most likely the result of an IMS Index Builder internal error. The code assists with the problem determination.

321  **SEGMENT DATA AREAS OBTAIN FAILED**

**Explanation**: This is most likely the result of an IMS Index Builder internal error. The code assists with the problem determination.

322  **COMPACT ROUTINE LOAD FAILURE**

**Explanation**: This is the result of either a compaction routine not found in the STEPLIB concatenation or the standby SAS ULU IMS not launched. This is most likely a DBRC sign-on problem.

330  **STORAGE ALLOCATION ERROR**

**Explanation**: The code assists with the problem determination Internal error. Indicates non-specific storage allocation failure in IIUBDBD.

331  **HIDAM ROOT HAS NO LOGICAL CHILDREN**

**Explanation**: The code assists with the problem determination. DBD analysis could not find logical children of HIDAM root segment.

332  **HIDAM ROOT HAS NO FIELDS**

**Explanation**: The code assists with the problem determination. DBD analysis could not find fields in HIDAM root segment.

333  **HIDAM NO ROOT OBTAINED**

**Explanation**: The code assists with the problem determination. DBD analysis could not obtain HIDAM root segment.

334  **HIDAM ROOT HAS NO SEQUENCE FIELD**

**Explanation**: The code assists with the problem determination. DBD analysis could not obtain sequence field in HIDAM root.

335  **PRIMARY INDEX HAS MANY LOGICAL CHILDREN**

**Explanation**: The code assists with the problem determination. DBD analysis found more than one logical children in the DBD supposed to be the primary index of a HIDAM database.

336  **PRIMARY INDEX NOT OBTAINED**

**Explanation**: The code assists with the problem determination. DBD analysis could not find the primary index of a HIDAM database.
IMS Index Builder control flow event codes

The IMS Index Builder processing subsystem issues the Index build control flow event codes.

The IMS Index Builder control flow events are reported on IIUSNAP by messages IIUE096M

The first variable parameter of the message is the job key of the issuer as reported at the MVS log. The second variable parameter is normally the TCB of the issuer. Events 618, 619, and 622 are exceptions.

601 SCAN PROCESS INITIALIZATION IN PROGRESS
Explanation: Scan object is being initialized.
User response: None.

602 SORT PROCESS INITIALIZATION IN PROGRESS
Explanation: Sort object is activated and initialized.
User response: None.

603 SEQUENTIAL SERIAL SCANNER ACTIVATED
Explanation: Single sequential scan object is activated in MAS. The prime database will be sequentially scanned.
User response: None.

604 PARA BUFFERED SORT SERVER READY
Explanation: The indicated sort object is activated in SAS. Up to three sort tasks can be attached to run parallel using E15 and E35 exits.
User response: None.

605 SAS SORT BROKER READY
Explanation: The indicated sort object is activated.
User response: None.

606 MESSAGE RELEASER TERMINATED
Explanation: Messages are no longer released.
User response: System stall may be expected.

607 MESSENGER LISTENING
Explanation: The Messenger object is to serialize and edit the IIUPRINT, IIUCAPT, and IIUSOUT report components.
User response: None.

608 SORT DRIVER BUFFERING
Explanation: The sort driver object is activated in buffered mode. Index buffers will be sent to a composite queue served by multiple sort objects using E15 and E35 exits.
User response: None.

609 SEQUENTIAL SAS SCANNER ACTIVATED
Explanation: Sequential scan object is activated in SAS. The prime database will be sequentially scanned. For HALDB, multiple scan objects can be used for parallel partitions scan.
User response: None.

610 INITIALIZER LISTENING
Explanation: The Initializer object is activated to initialize empty indexes or PSINDEX partitions asynchronously.
User response: None.

611 SAS SORT BROKER PROCESSING INDEX STRIPE
Explanation: The indicated sort object is activated. No exits will be used, an input sort stripe will be sorted into an output sort stripe.
User response: None.

612 EXTENDER LISTENING
Explanation: The initialization phase completed, the driver extension module takes over to initialize and control the resources that are needed for index load.
User response: None.

613 HIERARCHICAL DLI SCAN ACTIVATED
Explanation: The indicated scan object is activated. The prime database will be scanned by DLI GN calls.
User response: None.
INDEX BUILD PLANNER STARTED
Explanation: The Planner object is to create the build plan table.
User response: None.

PLANNING PROCESS INITIATED
Explanation: The Planner initialization completed successfully.
User response: None.

BUILD PLAN ENQUEUE TO CREATE
Explanation: The build plan is ready and is being passed to the Creator object.
User response: None.

INDEX BUILD PLANNER ENDED
Explanation: The Planner ended after successful processing.
User response: None.

Explanation: The first parameter shows the number of segments scanned. The second parameter shows the number of the index buffers released. No second line reciting the event code is displayed.
User response: None.

Explanation: The last progress log line, first parameter shows the number of segments scanned. The second parameter shows the number of the index buffers released.
User response: None.

SORT DRIVER STRIPING
Explanation: The sort driver object is activated in striping mode. The index buffers will be written on sort input stripes.
User response: None.

INDEX BUILD PLAN PROCESSOR LISTENING
Explanation: The Creator object is activated to process the build plan table.
User response: None.

SAS TERMINATION CHECK OUT
Explanation: Termination of sort or scan object. The second parameter is SORT or SCAN in hex.
User response: None.

HIERARCHICAL HPU SCANNER ACTIVATED
Explanation: The indicated scan object is activated. The prime database will be scanned by HPU GN calls.
User response: None.

ALL SAS TERMINATED
Explanation: No outstanding subtasks.
User response: None.

DFSURWFI/HPSRSIDX MAS READER ACTIVATED
Explanation: The indicated scan object is activated.
User response: None.

DFSURIDX DIRECT INDEX LOADER ACTIVATED
Explanation: The indicated scan object is activated.
Note: DFSURIDX records are pre-sorted and directly loaded to indexes.
User response: None.

BUILD PLAN NOT COMPLETE
Explanation: Abnormal termination initiated due to errors found in the planning process.
User response: None.

PROCESS INITIATION FAILURE
Explanation: Abnormal termination initiated due to errors detected by the initialization of a scan or sort object.
User response: None.

PARA STRIPED SORT SERVER READY
Explanation: Striped parallel SAS sort object up, no exits.
User response: None.
MAS SORT STRIPER TWINNING
Explanation: Sort object of above class is up parallel to scan.
User response: None.

MAS SORT COMPOSER TWINNING
Explanation: Sort object of above class is up parallel to base scan which could be either serial or hierarchical.
User response: None.

MONO BUFFERED SORT SERVER READY
Explanation: Sort object of the indicated class is activated in SAS to sort and load a single index or PSINDEX partition through E15 and E35 exits. IPR HPIC API can be enabled.
User response: None.

SAS TERMINATION COMPLETED
Explanation: SAS reported termination.
User response: None.

MONO API BROKER READY
Explanation: DRF API object is activated in SAS.
User response: None.

INDEX STRIPE CREATOR READY
Explanation: A separate object is activated to compose stripes.
User response: None.

SEQUENTIAL SERIAL SCANNER TWINNING
Explanation: Scan object of above class is up parallel to a base scan object, typically of hierarchical class.
User response: None.

INDEX BUFFER RELEASER TERMINATED
Explanation: Index buffers will be no more released.
User response: None.

MAS DATA BUFFER RELEASER TERMINATED
Explanation: Data buffers will no longer be released.
User response: Stall can be expected due to the data buffer pool being exhausted.

SAS DATA BUFFER RELEASER TERMINATED
Explanation: For sequential scan in SAS, the data buffers will no longer be released.
User response: Stalls can be expected due to the data buffer pool being exhausted.

HPSRWFP MAS READER ACTIVATED
Explanation: The indicated scan object is activated in MAS.
User response: None.

HALDB CASCADING IS ACTIVE
Explanation: The internal HALDB cascading process is activated for a PSINDEX. The second variable parameter in message IIUE096E shows the PSINDEX name.
User response: None.

HALDB CASCADING IS INACTIVE
Explanation: The internal HALDB cascading process is not activated for a PSINDEX. The second variable parameter in message IIUE096E shows the PSINDEX name.
User response: None.

INDEX COUNT CHECK IN PROGRESS
Explanation: The index count check process is activated. The second variable parameter in message IIUE096M shows the name of the index database.
User response: None.

INDEX COUNT CHECK TIMEOUT
Explanation: The index count check process terminated due to a timeout. Processing continues without checking the index count. The second variable parameter in message IIUE096M shows the name of the index database.
User response: None.
IMS Index Builder processing phase failure reason codes

The IMS Index Builder processing subsystem issues the following processing phase failure reason codes.

The processing phase failure reason codes are reported by the following messages:

- IIUB040E
- IIUB048E
- IIUB049E
- IIUB050E
- IIUB051E
- IIUB052E

<table>
<thead>
<tr>
<th>Reason Code</th>
<th>Message Description</th>
<th>Explanation</th>
<th>User Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>701</td>
<td>TIMER EXIT LOAD FAILURE</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>702</td>
<td>PCB STATUS IS BAD</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>703</td>
<td>PCB STATUS IS UNKNOWN</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>704</td>
<td>PRIME DB DATA SET OPEN FAILURE</td>
<td>Database data set could not be opened for scan.</td>
<td>Check for additional MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>705</td>
<td>MESSAGE DEQUEUE FAILURE</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>706</td>
<td>GB STATUS WITH NO DATA PROCESSED</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>707</td>
<td>STRIPE DATA SET ALLOCATION FAILURE</td>
<td>An error occurred during stripe data set allocation.</td>
<td>Check whether a stripe data set with the name that is indicated by the IIUB062I message exists. If the file exists, delete the file. If no stripe file is found, contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>708</td>
<td>ENQUEUE ON DRIVER FAILED</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>709</td>
<td>ENQUEUE ON BUFFER RELEASER FAILED</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>710</td>
<td>PARTITION NAME NOT FOUND</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>711</td>
<td>STORAGE ALLOCATION FAILURE</td>
<td>This error was most likely caused by an IMS Index Builder internal error.</td>
<td>Look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
</tbody>
</table>
Software Support with appropriate diagnostic documentation.

712  NULL VALUE STRING ALLOCATION FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.

713  SPARSE EXIT LOAD FAILURE
Explanation: This error might have occurred because the user exits library was not provided.
User response: Check the Index Builder JCL for exits library concatenated to STEPLIB. If the problem persists, contact IBM Software Support.

714  INDEX EXCLUDED FROM PROCESSING
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

715  INVALID WRITE INDEX FUNCTION CODE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

716  GET BUFFER MANAGER FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

717  ENQUEUE BUFFER MANAGER BUFFER FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

718  TARGET SEGMENT NOT FOUND IN LEVEL TABLE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

719  INDEX SORT STRIPE DCB OPEN FAILURE
Explanation: Sort failed due to control file open error, possibly internal error.
User response: Check for additional MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.

720  VSAM OPEN ERROR
Explanation: An error occurred in the VSAM OPEN macro.
User response: Locate the IIUB048E message, which shows the return code and the reason code that are stored in the ACBERFLG field.

721  VSAM PUT ERROR
Explanation: An error occurred in the VSAM PUT macro.
User response: Locate the IIUB048E message, which shows the return code and the reason code. If the problem persists, contact IBM Software Support.

722  INDEX PROCESS TERMINATED FORCELY
Explanation: The index process was terminated for some reason. If no other error messages are printed, the IMS Index Builder job terminated due to a timeout.
User response:
1. Provide a TMRWAIT control statement with the number that is greater than the default value to make the wait time longer.
2. If the error is not resolved by providing a TMRWAIT control statement with a larger number, check the last 618/619 event log line. If a large number of data segments were scanned, provide a SORTFSZ control statement (numbers in the range of millions are typically thought as large numbers). Specify the number of data segments that were scanned as a parameter and rerun the job. The sequential scan internal sort might have abended with a B37 due to the estimated large number of segments that would need to be scanned.

723  ROOT LOCATE FAILED
Explanation: This error likely occurred because the prime database is corrupted.
User response: Check the prime database integrity. If the problem persists, contact IBM Software Support.
724  **VSAM DATA MANAGEMENT ERROR**

**Explanation:**  This error likely occurred because the Index was incorrectly defined.

**User response:**  Check the index IDCAMS and look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.

725  **IDCAMS LISTCAT FAILURE**

**Explanation:**  An IDCAMS listcat failure occurred.

**User response:**  Run the index IDCAMS and look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.

726  **SEPARATED DATA SEGMENT CODE MISMATCH**

**Explanation:**  This error likely occurred because the prime database is corrupted.

**User response:**  Check the prime database integrity. If the problem persists, contact IBM Software Support.

727  **SEGMENT DECOMPRESS FAILED**

**Explanation:**  The segment compression exit routine returned an error.

**User response:**  See message IIUB117E.

728  **PRIMARY INDEX ALLOCATION FAILED**

**Explanation:**  The primary index allocation failed.

**User response:**  Look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.

729  **SORT EXIT ROUTINE LOAD FAILED**

**Explanation:**  The SORT exit routine load failed.

**User response:**  Review the MVS log and check the STEPLIB concatenation. If a serial scan was performed but no STEPLIB TRAN, check for STEPLIB in the started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

730  **VSAM ATTRIBUTES NOT CONSISTENT WITH DBD**

**Explanation:**  One or more attributes of the VSAM data set are not consistent with the segment definitions in the DBD.

- KEY LENGTH
- KEY OFFSET
- RECORD LENGTH

**User response:**  Correct the RECORDSIZE or the KEYS parameter of the IDCAMS DEFINE CLUSTER command. The recommended values are printed in the DBDGEN output.

731  **INDEX DATA SET NOT EMPTY**

**Explanation:**  The index data set is not empty and is not reusable.

**User response:**  Run index IDCAMS LISTCAT and rerun the job. If the problem persists, contact IBM Software Support.

732  **TWINNING OPTION NOT SUPPORTED**

**Explanation:**  Object activation requested with processing option not supported. Possibly internal error.

**User response:**  If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

733  **DFSURWF1 DATA SET ALLOCATION ERROR**

**Explanation:**  No DD statement or MDA was provided for DFSURWF1, or DUMMY DD statement was provided.

**User response:**  Provide valid DFSURWF1 input.

734  **INSUFFICIENT ACCESS AUTHORIZATION**

**Explanation:**  The index data set is not empty and is not reusable.

**User response:**  Run index IDCAMS LISTCAT and rerun the job. If the problem persists, contact IBM Software Support.

735  **VSAM EXTENDED ADDRESSABILITY IS NOT SUPPORTED**

**Explanation:**  A VSAM KSDS was allocated with the extended addressability attribute, which is not supported.

**User response:**  Remove the attribute from the data set and rerun the job.

736  **TASK CREATE FAILURE**

**Explanation:**  Either a scan entry was not found in the build plan table or the STEPLIB concatenation scan failed.

**User response:**  Contact IBM Software Support with appropriate diagnostic documentation.
737  BLOCK SCAN ERROR
Explanation: The database might be corrupted.
User response: Contact the installation DBA, check DB integrity.

738  DATA BUFFER GET FAILURE
Explanation: Internal error.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

739  INDEX BUFFER ENQUEUE FAILED
Explanation: Internal error.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

740  ADDRESS SPACE RESOURCE FAILURE
Explanation: This error occurred because of an IMS Index Builder internal error or a started task procedure error.
User response: Review the MVS log and check the started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

741  CREATE ADDRESS SPACE FAILURE
Explanation: This error occurred because of an IMS Index Builder internal error or a started task procedure error.
User response: Review the MVS log and check the started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

742  ADDRESS SPACE TERMINATION FAILURE
Explanation: This error occurred because of an IMS Index Builder internal error or a started task procedure error.
User response: Review the MVS log and check the started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

743  PRIME DB DSG CLOSE FAILURE
Explanation: Data set could not be closed after scan.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

744  PRIME DB DATA SET POINT FAILURE
Explanation: The POINT macro failed and scan could not continue.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

745  PRIME DB DATA SET GET FAILURE
Explanation: The GET macro failed and scan could not continue.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

746  INDEX DATA SET ALLOCATION FAILURE
Explanation: Data set could not be allocated when loading index.
User response: Check for additional MVS messages. If the problem persists, contact IBM Software Support.

747  RECURRING INDEX DATA SET ALLOCATION FAILURE
Explanation: Possible internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

748  COMMON SCAN FAILURE ABEND
Explanation: Scan could not continue.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

749  COMMON SORT FAILURE ABEND
Explanation: An IIUSRITxx sort processing failed during sequential scan.
User response: See the explanation of the IIUB059W message.

750  DESURWF1, HPSRSDIX OR DFSURIDX OPEN ERROR
Explanation: This error occurred because of a work file definition or allocation error. Parallel scan disposition must not be OLD or NEW.
**User response:** Review the MVS log and check the JCL and started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

---

### 751 OUTPUT DATA SET OPEN ERROR

**Explanation:** This error occurred because of a work file definition or allocation error. Parallel scan disposition must not be OLD or NEW.

**User response:** Review the MVS log and check the JCL and started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

---

### 752 DFSURIDX ALLOCATION ERROR

**Explanation:** A work file definition or allocation error occurred. Parallel scan disposition must not be OLD or NEW.

**User response:** Review the MVS log and check the JCL and started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

---

### 753 INDEX DATA SET ALLOCATION ERROR

**Explanation:** An index data set allocation error occurred.

**User response:** Look for MVS messages. Contact IBM Software Support with appropriate diagnostic documentation.

---

### 754 SORT FAILURE - RETRY INITIATED

**Explanation:** A striped parallel scan failure occurred during SORT. The SORT procedure stopped and a retry was initiated.

**User response:** None.

---

### 755 SORT FAILURE - NO MORE RETRIES

**Explanation:** A striped parallel scan failure occurred during SORT. The SORT procedure stopped and the last retry was initiated.

**User response:** None.

---

### 756 SEQUENTIAL BUFFER SCAN FAILURE

**Explanation:** The database might be corrupted.

**User response:** Check database integrity. If the problem persists, contact the installation DBA.

If the cause is an IMS Index Builder internal error, a snap dump is generated in the IIUDUMP data set. This snap dump is required for analysis.

The general-purpose registers at taking the snap dump contain the following information:

<table>
<thead>
<tr>
<th>Register</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>R2</td>
<td>RBA of the database block or CI</td>
</tr>
<tr>
<td>R3</td>
<td>Storage address of the database block or CI</td>
</tr>
<tr>
<td>R4</td>
<td>Storage address of the segment currently processed</td>
</tr>
</tbody>
</table>

---

### 757 STRIPE ALLOCATION FAILED

**Explanation:** IMS Index Builder Version 3 Release 1 standard work file (stripe) could not be allocated. Possible internal error.

**User response:** Check for additional MVS messages. If the problem persists, contact IBM Software Support.

---

### 758 BUILD PLAN INITIATION FAILED

**Explanation:** A build plan table was not provided. Possible internal error.

**User response:** If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

---

### 759 INDEX INITIALIZATION FAILED

**Explanation:** The initialization of an empty index failed.

**User response:** Proceed according to symptoms.

---

### 760 MESSAGE BUFFER OBTAIN FAILURE

**Explanation:** This error was most likely caused by an IMS Index Builder internal error.

**User response:** Contact IBM Software Support with appropriate diagnostic documentation.

---

### 761 MESSAGE BUFFER ENQUE FAILURE

**Explanation:** This error was most likely caused by an IMS Index Builder internal error.

**User response:** Contact IBM Software Support with appropriate diagnostic documentation.

---

### 762 STORAGE ALLOCATION FAILURE

**Explanation:** This error was most likely caused by an IMS Index Builder internal error.

**User response:** Check the MVS log and look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.

---

### 763 NAME TOKEN SERVICE FAILURE

**Explanation:** This error was most likely caused by an IMS Index Builder internal error.

**User response:** Contact IBM Software Support with appropriate diagnostic documentation.
INV  ALID CALL  VECTOR
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

765  CREATOR BUFFER ENQUEUE  FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

766  DATA BUFFER RELEASE  FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

767  MESSAGE BUFFER RELEASE  FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

768  END OF DRIVE NOTIFY  FAILED
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

769  STRIPING NOTIFY  FAILED
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

770  BUFFER/MESSAGE DEQUEUE  FAILED
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

771  MESSAGE FUNCTION INVALID
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

772  VSAM OPEN ERROR - OVERFLOW  DS
Explanation: An error occurred in the VSAM OPEN macro.
User response: Locate the IIUB048E message, which shows the return code and the reason code that are stored in the ACBERFLG field. If the problem persists, contact IBM Software Support.

773  VSAM PUT LOGICAL ERROR - OVERFLOW  DS
Explanation: An error occurred in the VSAM PUT macro.
User response: Locate the IIUB048E message, which shows the return code and the reason code. If the problem persists, contact IBM Software Support.

774  VSAM PUT PHYSICAL ERROR - OVERFLOW  DS
Explanation: An error occurred in the VSAM PUT macro.
User response: Locate the IIUB048E message, which shows the return code and the reason code. If the problem persists, contact IBM Software Support.

775  LOADER TASK ATTACH  FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

776  MERGER TASK ATTACH  FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

777  UNABLE TO ALLOCATE HPSRWFP  DATA SET
Explanation: Failed to allocate the HPSRWFP data set.
User response: Locate the IIUB062I message, which shows the return code and the reason code.

778  UNABLE TO OPEN HPSRWFP  DATA SET
Explanation: Failed to open the HPSRWFP data set.
User response: Ensure that the data set was generated correctly.
HPSRWFP DATA SET PROCESS FAILURE
Explanation: Failed to read the HPSRWFP data set.
User response: Ensure that the data set was generated correctly.

VSAM GET ERROR
Explanation: An error occurred in the VSAM GET macro.
User response: Locate the IIU048E message, which shows the return code and the reason code. If the problem persists, contact IBM Software Support.

VSAM POINT ERROR
Explanation: An error occurred in the VSAM POINT macro.
User response: Locate the IIU048E message, which shows the return code and the reason code. If the problem persists, contact IBM Software Support.

VSAM ERASE ERROR
Explanation: An error occurred in the VSAM ERASE macro.
User response: Locate the IIU048E message, which shows the return code and the reason code. If the problem persists, contact IBM Software Support.

INDEX RECORD LAYOUT ERROR
Explanation: An index record layout error occurred.
User response: Check the index layout in the Index Builder report, correct if possible, and rerun the job. If the problem persists, contact IBM Software Support.

PHIDAM PRIMAR INDEX INITIALIZATION FAILURE
Explanation: Initialization of PHIDAM primary index failed. Initialization is needed for a PHIDAM partition in which the HALDB Online Reorganization (OLR) cursor is active and the output side contains no root segment occurrence yet.
User response: Locate the IIU048E message, which shows the return code and the reason code.

ILDS INITIALIZATION FAILURE
Explanation: ILDS initialization failed.
User response: Locate the IIU048E message, which shows the return code and the reason code.

NO INDEX RECORD CREATED
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.

ALL INDEXES EXCLUDED FROM PROCESSING
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.

PRIME DB/PARTITION SCAN WAS UNEXPECTEDLY TERMINATED
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Look for additional messages. Contact IBM Software Support with appropriate diagnostic documentation.
795  SORT ADDRESS SPACE NOT ACTIVE
   - EOD

Explanation: The started task procedure might be in error.

User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

796  SORT ADDRESS SPACE NOT ACTIVE
   - SCAN

Explanation: The started task procedure might be in error.

User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

797  PRIME DATA BASE SCAN ABENDED

Explanation: This error was most likely caused by an IMS Index Builder internal error.

User response: Contact IBM Software Support with appropriate diagnostic documentation.

798  PROCEDURE ERROR

Explanation: Severe procedural error detected, type 2 emergency termination.

User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

799  PROCEDURE ERROR

Explanation: Severe procedural error or security error was detected during subordinate address space initialization. One of the following errors might have occurred:

- JCL error in a subordinate address space (SAS)
- RACF error while creating a SAS
- Allocation error of a stripe data set for the WTO capture in a SAS

User response: Review the MVS log. If setup is incomplete, see Chapter 6, "Defining the subordinate address space procedure," on page 99.

If the failure is caused by an allocation error, an IIUB062I WTO message is printed in the SAS job log. If the message shows FUNC=ALLOC RC=04, check whether any stripe data sets exist and if a stripe data set exists, delete the data set and rerun the job. Otherwise, contact IBM Software Support with appropriate diagnostic documentation.

901  DSPURXRT ATTACH FAILURE

Explanation: This error was most likely caused by an IMS Index Builder internal error.

User response: Contact IBM Software Support with the appropriate diagnostic documentation.

902  DSPURXRT SYSIN OPEN FAILURE

Explanation: This error was most likely caused by an IMS Index Builder internal error.

User response: Contact IBM Software Support with the appropriate diagnostic documentation.

903  DBRC UTILITY ERROR

Explanation: The IMS DBRC utility program was invoked to issue the DBRC NOTIFY commands; however, the program ended with an error code.

User response: See the IMS DBRC message.

A01  SAS NAME/TOKEN FAILURE

Explanation: Storage obtain failed, most probably Index Builder internal error.


A02  BUILD PLAN TABLE ENTRY OBTAIN FAILURE

Explanation: A storage obtain failure, most likely caused by an IMS Index Builder internal error, occurred.

User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A03  UNABLE TO ALLOCATE ERROR

Explanation: A storage obtain failure occurred, most likely caused by an IMS Index Builder internal error.

User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A04  UNABLE TO LOAD IIUBE15 EXIT

Explanation: The SAS STEPLIB might be incomplete.

User response: Review the MVS log and check the STEPLIB started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.
A05 UNABLE TO LOAD IIUBE35 EXIT
Explanation: The SAS STEPLIB might be incomplete.
User response: Review the MVS log and check the STEPLIB started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

A06 UNABLE TO OPEN INDEX DATA SET
Explanation: The index data set could not be opened.
User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A07 VSAM PUT DATA RECORD ERROR
Explanation: A VSAM put data record error occurred.
User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A08 UNABLE TO LOAD IIUBLERR
Explanation: The SAS STEPLIB might be incomplete.
User response: Review the MVS log and check the STEPLIB started task procedure. Contact IBM Software Support with appropriate diagnostic documentation.

A09 INDEX BUILD PHASE 0 FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

A10 ASEXTR SERVICE FAILURE
Explanation: The SAS cross memory access failed.
User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A11 ALESERV SERVICE FAILURE
Explanation: The SAS cross memory access failed.
User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A12 EXTRACT FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

A13 LOCASCB STOKEN FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

A14 TRANSIT AREA OBTAIN FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

A15 DYNAMIC ALLOCATION ERROR
Explanation: A dynamic allocation error occurred.
User response: Review the MVS log. If running a parallel scan, look for IIU.STRIPE.* files, delete them, and rerun the job. Contact IBM Software Support with appropriate diagnostic documentation.

A16 RACROUTE CREATE FAILURE
Explanation: Internal error.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

A23 SAS SORT ATTACH FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

A24 ATTACHED SAS SORT FAILURE
Explanation: This error was most likely caused by an IMS Index Builder internal error.
User response: Contact IBM Software Support with appropriate diagnostic documentation.

A25 LINK SORT FAILURE
Explanation: A link sort failure occurred.
User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

A26 SORT PROCESSING FAILURE
Explanation: An index sort processing error occurred.
User response: See the explanation of the IIUB059W message.
INDEX BUILDER SIGNALLED SHUTDOWN
Explanation: Index Builder signaled shutdown.
User response: Review the MVS log. Contact IBM Software Support with appropriate diagnostic documentation.

DRF API PROCESSING FAILED
Explanation: Internal error.
User response: If the problem persists, contact IBM Software Support with appropriate diagnostic documentation.

DUPLICATE KEY ENCOUNTERED
Explanation: IMS Index Builder detected duplicate keys for an index that does not support duplicate keys.
User response: Investigate the database definition and the source of the index data for inconsistency.
IMS Index Builder creates a data set job_key DUPLKEYS that contains the duplicate records to aid in analysis.
hlq is the user specified dataset class.
dbdname is the index dbdname being processed.
job_key is generated for uniqueness.

COPY SORT SYSOUT ERROR
Explanation: A copy sort SYSOUT error occurred.
User response: Review the MVS log, correct any errors, and rerun the job. Contact IBM Software Support with appropriate diagnostic documentation.

MONO SORT Index Schedule Failure
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

HPIC/IPR INIT call failed.
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

HPIC/DRF INIT call failed.
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

HPIC/IPR TERM call failed.
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

HPIC/DRF TERM call failed.
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

HPIC/DRF TERM call failed.
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

RAUX INIT call failed
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

RAUX TERM call failed
Explanation: Internal error.
User response: Contact IBM support with appropriate diagnostic documentation.

HPPC/DRF OPEN call failed
Explanation: Internal error.
User response: Contact IBM Softwar e Support with appropriate diagnostic documentation.
<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Explanation</th>
<th>User response</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0C</td>
<td>HPIC/IPR OPEN call failed</td>
<td>Internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>B0D</td>
<td>HPIC/DRF OPEN call failed</td>
<td>Internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>B0E</td>
<td>DRF API TERM call failed</td>
<td>Internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>B13</td>
<td>HPIC/IPR PUT call error</td>
<td>Internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
<tr>
<td>B14</td>
<td>HPIC/IPR CLOSE call failed</td>
<td>Internal error.</td>
<td>Contact IBM Software Support with appropriate diagnostic documentation.</td>
</tr>
</tbody>
</table>
IMS Index Builder verification phase failure reason codes

The IMS Index Builder verification subsystem issues the following verification phase failure reason codes.

In the verification phase, IMS Index Builder verifies whether the number of records that were loaded to each index is equal to the number of the scanned occurrences of the prime database. If the numbers are not equal, the IIUB060E message and one of the following reason codes are issued.

792  NUM OF RECORDS MISMATCH: PRIMARY INDEX

Explanation: The number of the records that were loaded to the primary index must be equal to the sum of the following values:

1. The number of scanned occurrences of the root segment. (This number does not include the following two types of segment occurrences.)
2. The number of scanned occurrences of the root segment whose delete byte indicates that the segment is physically deleted.
3. The number of scanned occurrences of the root segment that has a high key value of all X'FF's. This number is 1 for each HIDAM or PHIDAM partition.

This error was most likely caused by an IMS Index Builder internal error.

User response: Contact IBM Software Support with appropriate diagnostic documentation.

793  NUM OF RECORDS MISMATCH: ILDS

Explanation: The number of the records that were loaded to the ILDS must be equal to the number of the scanned occurrences of the following segments in the prime database:

- Target segments of secondary indexes
- Physically paired logical child segments
- Logical parent segments of unidirectional logical children

This error was most likely caused by an IMS Index Builder internal error.

User response: Contact IBM Software Support with appropriate diagnostic documentation.

794  NUM OF RECORDS MISMATCH: SECONDARY INDEX

Explanation: The sum of the number of the records that were loaded to the secondary index and the number of the rejected records must be equal to the number of the scanned occurrences of the index source segment in the prime database.

If a DFSURWF1 file, a DFSURIDX file, or HPSRWFP files were used as input, the number of the records that were read from that file is verified instead of the number of scanned occurrences.

If update mode is specified for building the PSINDEX, the number of the records that were updated or inserted is used instead of the number of the loaded records.

This error was most likely caused by an IMS Index Builder internal error.

User response: Contact IBM Software Support with appropriate diagnostic documentation.
Chapter 12. IMS Index Builder messages

IMS Index Builder issues messages that can be used to help you troubleshoot and diagnose IMS Index Builder problems.

For messages that are issued by IMS or DFS, see IMS Messages and Codes Volumes 1 and 2.

IMS Index Builder messages adhere to the following format:
IIUxyyz

Where:

IIU Indicates that the message was issued by IMS Index Builder.

x Indicates the program that issued the message:

B Indicates that IMS Index Builder issued the message.

D Indicates that IMS Index Builder Diagnostics Aid issued the message. For Diagnostics Aid messages, see [“Messages” on page 168].

H Indicates that the hierarchical HPU scan issued the message. See the explanation for the corresponding FABH:xxx message in the IMS High Performance Unload User’s Guide.

yyy Indicates the message identification number.

z Indicates the severity of the message:

A Indicates that operator intervention is required before processing can continue.

E Indicates that an error occurred and that a requested function did not complete successfully. The condition 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 variable entry fields are (if any).

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.

IIUB001I  FILE DD = ddname HAS BEEN CLOSED AND DEALLOCATED
Explanation: The data for prefix resolution is now available in the file that is specified by ddname.

System action: Processing continues

User response: None. This message is informational.

IIUB002E  PROC CONTROL STATEMENT IS MISSING

Explanation: No PROC control statement was found.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB003E  DBD NAME MISSING IN xxxxxxxx CONTROL STATEMENT or DBD NAME INVALID OR MISSING IN xxxxxxxx CONTROL STATEMENT or DBD NAME dbdname INVALID IN xxxxxxxx CONTROL STATEMENT

Explanation: A database name must be specified in the xxxxxxxx control statement.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB004E  DBD NAME SPECIFIED IN BLD_PRIMARY AND BLD_SECONDARY CONTROL STATEMENTS DO NOT MATCH

Explanation: If you want to build both a primary and a secondary index in the same job, the database names in the PROC statements must be identical.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB005E  INVALID OR MISSING KEYWORD IN xxxxxxxx STATEMENT

Explanation: An incorrect keyword is specified for the indicated statement.

System action: This statement is ignored and processing continues.

User response: Correct the statement and, if necessary, rerun the job.

IIUB006W  INDEX STATEMENT IGNORED, SECONDARY INDEX REBUILD NOT REQUESTED

Explanation: An INDEX statement is required only if BLD_SECONDARY and SELECTED are specified in the PROC statement.

System action: Processing continues.

User response: None.

IIUB007E  DBRC OPTION xxx INCONSISTENT WITH IMS SCD

Explanation: The DBRC variable xxx is not Y or N, or is inconsistent with the IMS generation DBRC default.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB008E  INVALID SYNTAX OR PARAMETER IN xxxxxxxx STATEMENT

Explanation: A syntax error has been detected in the xxxxxxxx statement.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB009I  RECALLING: dsn

Explanation: The data set dsn was migrated and is being recalled.

System action: Processing resumes after the recall completes. Completion of the recall will be indicated by message IIUB021I.

User response: None. This message is informational.

IIUB010E  PRIMARY INDEX BUILD NOT APPLICABLE FOR HALDB

Explanation: ILDS exists for HALDB only.

System action: Processing stops.

User response: Do not specify PROC_ILDS for non-HALDB. Correct the error and rerun the job.

IIUB011E  DD NAME MISSING IN OUTPUT STATEMENT

Explanation: A DD name must be specified in the OUTPUT control statement.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB012E  REQUIRED xxxxxxxx CONTROL STATEMENT MISSING or REQUIRED xxxxxxxx CONTROL STATEMENT MISSING FOR SELECTED DBD dbdname

Explanation: An xxxxxxxx control statement is required if IBSCAN is specified in the INPUT control statement.

System action: Processing stops.

User response: Correct the error and rerun the job.
**IIUB013E** NO ddname DD STATEMENT OR DYNAMIC ALLOCATION MEMBER FOUND FOR DBD dbdname

**Explanation:** Neither a DD statement for ddname nor a dynamic allocation library member was found for this dbdname. If a DD statement is not present, IMS Index Builder attempts to perform dynamic allocation, which requires a dbdname member in the IMS dynamic allocation library.

**System action:** Processing stops.

**User response:** Create the appropriate dynamic allocation member or specify a DD statement, and rerun the job.

**IIUB014E** DBD dbdname NOT FOUND IN DBD LIBRARY

**Explanation:** The IMS DBD library must contain member dbdname.

**System action:** Processing stops.

**User response:** A DBDGEN is required for the dbdname database. Ensure that you specify the correct dbdname and rerun the job.

**IIUB015E** DATABASE ORGANIZATION OF DBD dbdname IS NOT HIDAM xxxxxxx

**Explanation:** Primary indexes are supported only for HIDAM databases.

**System action:** Processing stops.

**User response:** Specify a supported database and rerun the job.

**IIUB016E** IMS RELEASE nnn IS NOT SUPPORTED

**Explanation:** IMS Index Builder does not support IMS release nnn.

**System action:** Processing stops.

**User response:** Specify an IMS RESLIB that meets the minimum requirements for IMS Index Builder.

**IIUB017E** IMSALTL IMPROPERLY USED FOR ALTERNATE DBD

**Explanation:** IMSALTL can be used only when DBRC is not involved.

**System action:** Processing stops.

**User response:** Remove the IMSALTL DD and resubmit the job.

**IIUB018E** VSAM vsam ERROR FOR DBD dbdname.

**RC =** rc  **FDBK =** fdbk

**Explanation:** A VSAM vsam error has been detected while loading the index database dbdname.

**System action:** Processing stops.

**User response:** Use the return code (RC) and VSAM feedback code (FDBK) to determine the cause of the problem. Contact IBM for support, if necessary.

**IIUB019E** KEYWORD xxxxxxx INVALID FOR HALDB

**Explanation:** The function that is requested by the INPUT keyword xxxxxxx is not applicable for a partitioned database.

**System action:** Processing stops.

**User response:** Correct the error and rerun the job.

**IIUB020E** INVALID COMBINATION OF INPUT AND PROC KEYWORDS

**Explanation:** The function that is requested by the INPUT statement is inconsistent with the keyword specified in the PROC statement.

**System action:** Processing stops.

**User response:** Correct the error and rerun the job.

**IIUB021I** IIUB021I RECALL COMPLETE: dsn

**Explanation:** The recall of data set dsn completed.

**System action:** Processing resumes.

**User response:** None. This message is informational.

**IIUB022E** BLD_SECONDARY REQUIRED FOR 'INPUT' STATEMENT OPTION DFSURWF1, HPSRSIDX OR DFSURIDX

**Explanation:** If you select DFSURWF1, HPSRSIDX, or DFSURIDX in the INPUT statement, you must specify PROC BLD_SECONDARY or BLD_ALL.

**System action:** Processing stops.

**User response:** Correct the error and rerun the job.

**IIUB023E** 'INPUT' STATEMENT OPTIONS DFSURWF1, HPSRSIDX OR DFSURIDX ARE NOT APPLICABLE FOR HALDB

**Explanation:** You tried to build a partitioned secondary index using work files that are not created during the reorganization of a partitioned database.

**System action:** Processing stops.

**User response:** To build a partitioned secondary
index, you must use IBSCAN. Correct the error and rerun the job.

IIUB024E  PROC CONTROL STATEMENT ASSOCIATED WITH SELECTED INDEX IS MISSING INDEX xxxxxxxx

Explanation: No PROC control statement was found for the primary database containing the secondary index xxxxxxx specified in the INDEX statement.

System action: Processing stops.

User response: Ensure that you have a PROC statement for the selected index, and that the name of the secondary index is spelled correctly. Rerun the job.

IIUB025E  ONLY ONE xxxxxxx CAN BE SPECIFIED PER JOB

Explanation: More than one xxxxxxx control statement was specified. Only one is valid in a single job.

System action: Processing stops.

User response: Correct the control statements. Rerun the job.

IIUB026E  PARTITION [MASTER] DB 'dbname' NOT REGISTERED IN RECON

Explanation: A partitioned database or partitioned master database named dbname is not registered in RECON.

System action: Processing stops.

User response: Define the partitioned database or partitioned master database in RECON and rerun the job.

IIUB027I  MAXTASKS STATEMENT OUT OF CONTEXT —REASON=XXX

Explanation: The MAXTASKS control statement was provided in a context where user control of created tasks cannot be exercised.

System action: Processing continues.

User response: None. This message is informational. The reason codes explain the IMS Index Builder processing mode entered.

IIUB028E  RECORD LENGTH IN VSAM DEFINE SMALLER THAN RECOMMENDED LENGTH IN DBDGEN FOR DBD=dbname, DDN=ddname

Explanation: The index record length that is specified in the VSAM DEFINE is less than the length computed by IMS.

System action: Processing stops.

User response: One possible cause for this error is that the DSN in the VSAM define is not the same as the DSN in the RECON. In this case, correct the DSN and rerun the job.

Otherwise, list the VSAM catalog and check the record length of the index. If it is less than the computed length in the record layout, increase it and rerun the job.

IIUB029E  IDCAMS PROCESSING FAILED FOR DBD=dbname. INDEX DATA SET(S) MAY BE NOT EMPTY AND NOT REUSABLE.

Explanation: IDCAMS processing failed for one or more of the data sets to be rebuilt in the current IMS Index Builder run. IIUB053E lines denote the data sets in error.

System action: Processing stops, the return code is 16 (X'10').

User response: Run IDCAMS for the data sets in error.

IIUB030E  DBRC NOTIFY FAILED

Explanation: DBRC is specified, but the function fails. The database might not be defined in RECON.

System action: Processing stops.

User response: Look for additional MVS messages explaining the cause of the failure. Correct the error and rerun the job.

IIUB031E  UNABLE TO LOAD PSB=psbname

Explanation: The generic PSB psbname was not found in any of the IMS data sets.

System action: Processing stops.

User response: Ensure that all the dynamic PSBs (IIUPSB01, IIUPSB04, IIUPSB16, and IIUPSB64) exist in the IMS concatenation. In particular check that the Index Builder LOADLIB data set is the first library in this concatenation.

IIUB032E  psbname NOT FOUND IN IMS CONCATENATION

Explanation: The generic PSB psbname was not found in any of the IMS data sets concatenated under the IMS DD.

System action: Processing stops.

User response: Ensure that all the dynamic PSBs (IIUPSB01, IIUPSB04, IIUPSB16, and IIUPSB64) exist in the STEPLIB concatenation. The library that contains the dynamic PSBs must be the first concatenation. For more information, see the definition for the IMS DD statement in “IMS Index Builder JCL,” on page 66.
"INPUT IBSCAN" NOT COMPATIBLE WITH "OUTPUT"

**Explanation:** Control statements INPUT SCAN and OUTPUT cannot be specified in the same job.

**System action:** Processing stops.

**User response:** Correct the input control statements and rerun the job.

INVALID INPUT MODE FOR BUILD_SECONDARY

**Explanation:** Controls statements INPUT SCAN and PROC BLD_SECONDARY cannot be specified in the same job.

**System action:** Processing stops.

**User response:** Correct the input control statements and rerun the job.

**IIUB035W** \[xxxxxxx\] STATEMENT HAS INVALID PARAMETER \[parameter\]

**Explanation:** The input control statement \[xxxxxxx\] contains an invalid parameter.

**System action:** The invalid parameter is ignored and processing continues.

**User response:** Correct the invalid parameter and rerun the job, if necessary.

MAXIMUM SUPPORTED NUMBER OF \[xxxxxxx\] NAMES EXCEEDED

**Explanation:** The maximum number of names was exceeded while parsing input control statements from the IIUN DD statement.

**System action:** Processing stops.

**User response:** Correct the problem and rerun the job.

**IIUB037I** \[xxxxxxx\] KEYWORD IGNORED — NOT SUPPORTED

**Explanation:** The keyword that is specified in the IIUN input control statement is not supported.

**System action:** The keyword on the input control statement is ignored. Processing continues.

**User response:** Correct the input control statement.

**IIUB038I** \[xxxxxxx\] STATEMENT IGNORED — NOT RELEVANT

**Explanation:** During the initialization phase, IB31 parsed an input statement that is not relevant to current version of IMS Index Builder.

**System action:** The statement is ignored. Processing continues.

**IIUB039I** HALDB haldb REQUIRES "PARTNAME" STATEMENT.

**Explanation:** Partition name statements are provided when building the ILDS, primary, or secondary indexes. The PARTNAME statement identifies the partition to be processed.

**System action:** Processing continues.

**User response:** Check that secondary indexes will not be affected.

**IIUB040E** \[xxxxxxx\] INDEX PROCESSING ERROR — REASON=reason

**Explanation:** An error was encountered while processing an index data record for the named index. Reason code reason defines the error.

**System action:** Processing stops for the secondary index being processed.

**User response:** Correct the failure and rerun the job for the failed index. See “IMS Index Builder processing phase failure reason codes” on page 133 for information about the reason code that this message issued and the appropriate action that you should take.

**IIUB041E** SEGMENT EXCEPTION — sssssss, cc, ll, rba

**Explanation:** While building a secondary index, the current segment could not be identified. Either the segment name (sssss), segment code (cc), or segment level (ll) is invalid. The RBA of the segment in error (rba) also displays.

**System action:** Processing stops for the secondary index being processed.

**User response:** Check the prime database integrity.

**IIUB042W** WARNING: INDEX(ES) IN ERROR—PLEASE CHECK INDEX LOAD REPORT

**Explanation:** While creating the index record layout or while building the index, logical errors were encountered. Some of the indexes might not have been created or are in error. The corresponding index load reports contain the specific reason code. This message is issued only when at least one index has been built.

**System action:** Index load completes with a return code of 8.

**User response:** None. Either a user or internal IMS Index Builder error occurred.
IIUB043I  DATA BASE dbname IS USING DATA SETS NOT REGISTERED WITH DBRC

Explanation:  The DDNAME or DSNAME that is indicated by dbname did not match the names that are registered to DBRC.

System action:  Processing continues without issuing DBRC authorization requests or DBRC notifications for dbname.

User response:  This message is issued for compatibility with DFS3341I (IMS message). For more information, see the explanation of the DFS3341I message in IMS Messages and Codes.

IIUB044E  ENVIRONMENT DOES NOT ALLOW OLR SPLIT PROCESSING

Explanation:  OLR has been stopped for at least one HALDB partition and sequential scan cannot be applied.

System action:  Processing stops.

User response:  Complete OLR and resubmit the job.

IIUB045E  USER NOT AUTHORIZED TO UPDATE DBD=dbname, DDN=ddname INDEX BUILDER FAILURE

Explanation:  User ID of IMS Index Builder main job not allowed CONTROL access to the index.

System action:  Processing stops.

User response:  None. User error.

IIUB046E  ITKB XXXXXX FAILED RC=XXX, REASON=XXX


User response:
1. Make sure that the IMS Tools Knowledge Base server is operational.
2. Make sure that you correctly specified the server name in the ITKBSRVR statement.
3. Make sure that the ITKB communication subsystem is installed on the z/OS system that you are running on.

IIUB047E  INVALID SEGMENT SCANNED, NAME=name, CODE=code — INDEX BUILDER FAILURE

Explanation:  An invalid segment was obtained from the scan.

System action:  Processing stops.

User response:  Check the database integrity and contact IBM Software Support.

IIUB048E  ddname xxxxxxxx VSAM ERROR WHILE LOADING INDEX — REASON=reason

Explanation:  A VSAM error occurred while loading INDEX.

System action:  Processing stops.

User response:  See “IMS Index Builder processing phase failure reason codes” on page 133 for information about the reason code that this message and the appropriate action to take.

xxxxxxx shows the VSAM macro return code and reason code in the RPL feedback area or in the ACB error field. For the meaning of the codes, see DFSMS Macro Instructions for Data Sets.

If the cause is an IMS Index Builder internal error, a snap dump is generated in the IIUDUMP data set. This snap dump is required for analysis.

IIUB049E  xxxxxxxx ERROR WHILE PROCESSING INDEX - REASON=reason

Explanation:  An error occurred during processing and the index record was not created.

System action:  Processing stops.

User response:  See “IMS Index Builder processing phase failure reason codes” on page 133 for information about the reason code that this message issued and the appropriate action that you should take.

IIUB050E  [index module] SUBORDINATE PROCESS FAILURE — REASON=reason

Explanation:  The subordinate address space encountered an error. Reason code reason defines the error. For some reason codes, the name of the index sorted in the subordinate address space is printed.

System action:  Processing stops for the subordinate address space.

User response:  See “IMS Index Builder processing phase failure reason codes” on page 133 for information about the reason code that this message issued and the appropriate action that you should take. Correct the error and rerun the job for the failed index.
**IIUB051E**  
**INDEX LOAD FAILURE — REASON=XXX**

**Explanation:**  
Index data set allocation failure

**System action:**  
Processing stops

**User response:**  
Correct the error and rerun the job

---

**IIUB052E**  
**ERROR IN XXXXXXXX — REASON=XXX**

**Explanation:**  
Processing error was detected by the module that is shown in the message.

**System action:**  
Processing stops

**User response:**  
Contact IBM Software Support with appropriate diagnostic documentation.

---

**IIUB053E**  
**DATA SET IDCAMS PROCESSING FAILED**

**Explanation:**  
This error line is placed in the data set description block of the IMS Index Builder report, below the DSN line to denote IDCAMS processing failure which could be:
- Data set allocation failed, DSN in the above line is missing
- DEL/DEF failed if any
- Data set not defined in catalog
- Data set not empty and not reusable
- Data set LRECL less than DBD LRECL
- Data set with the extended addressability attribute

**System action:**  
Processing stops

**User response:**  
List IDCAMS for data set and correct error.

This message is issued when the IDCAMS delete and define statements are coded in members of the DDEFPDS data set and one or more of the statements fail. See the IDCAMS messages in the IIUSOUT data set and correct the error.

---

**IIUB054I**  
**DRF/IB INTEGRATION IN EFFECT**

**Explanation:**  
The IMS Index Builder run was initiated by IMS Database Recovery Facility.

**System action:**  
Processing continues.

**User response:**  
None. This message is informational.

---

**IIUB056I**  
**HPIC INDEX INITIATION**

**Explanation:**  
The image copy was requested by IMS Database Recovery Facility for IMS Index Builder.

**System action:**  
Processing continues.

**User response:**  
None. This message is informational.

---

**IIUB057E**  
**DUPLICATE XXXXXXXX KEYWORD CODED ON XXXXXXXX STATEMENT(S)**

**Explanation:**  
The coded keyword that is identified in this message can be specified only once.

**System action:**  
IMS Index Builder stops during initialization.

**User response:**  
If the error was detected in the IIUIN control statements, remove the duplicate specification and resubmit the job.

If the error was detected in the runtime default specification, remove duplicate specification from the IIURDFLT source member. Assemble the updated source and link it into the program library of IMS Index Builder.

---

**IIUB058E**  
**INVALID VALUE CODED FOR KEYWORD XXXXXXXX ON XXXXXXXX STATEMENT**

**Explanation:**  
The keyword is assigned a value that either contains a syntax error or is an unacceptable value for the parameter.

**System action:**  
Index Builder stops during initialization.

**User response:**  
If the error was detected in the IIUIN control statements, correct the parameter’s value and resubmit.

If the error was detected in the runtime default specification, correct the parameter in the IIURDFLT source member. Assemble the updated source and link it into the program library of IMS Index Builder.

---

**IIUB059W**  
**WARNING: SORT ABEND EVIDENCED — PLEASE CHECK IIUSOUT**

**Explanation:**  
A sort processing ended with an error. The IIUSOUT data set contains the messages that were issued by the sort program.

**System action:**  
If the scan process has not completed, IMS Index Builder processing stops. Otherwise, IMS Index Builder continues to process other indexes.

**User response:**  
Check and correct the error. When the work space capacity for sort has exceeded, consider increasing the value of the following options:
- The JCL REGION option of the main and subordinate address spaces
- The sort DYNSPC option
- The primary allocation size of the VSAM data set, which is used to estimate the value of the sort FILSZ option when the sort SORTIN option is not specified.

When an internal program error has occurred, contact IBM Software Support with appropriate diagnostic documentation.
IIUB060E  index NUMBER OF RECORDS MISMATCH

Explanation: The number of the records that were loaded to this index is not equal to the number of the occurrences that were scanned in the prime database or the number of the records that were read from a DFSURWF1 file.

System action: Processing for this index stops. DBRC notification for this index is not issued.

User response: See “IMS Index Builder verification phase failure reason codes” on page 144 for information about the reason code that this message issued.

IIUB061I  HPIC INDEX PROCESSING REQUESTED

Explanation: The Image Copy processing for secondary indexes was requested in the IMS Parallel Reorganization job or in the IMS Database Reorganization Expert job.

System action: Processing continues.

User response: None. This message is informational.

IIUB062I  DYNALLOC FAILURE; DD=ddname
            FUNC=function RC=rc RSN=yyyyyyyy
            S99EERR=xxxxx S99EINFO=xxxxx
            S99ERSN=xxxxx DSN=stripe_dsnn

Explanation: Dynamic allocation for ddname failed. function is one of ALLOC, UNALLOC, CONCAT, and INFO. rc is the return code of the DYNALLOC macro (SVC 99) in decimal. yyyy is the error reason code (S99ERROR) and zzzz is the informational reason code (S99INFO) in hexadecimal. S99EERR, S99EINFO, and S99ERSN codes are also shown in hexadecimal.

System action: Processing might or might not continue.

User response: Check whether a stripe file with the name stripe_dsnn exists, and if the file exists, delete the file. If no stripe file with the indicated name is found, check the cause of the error. The meanings of the codes are described in the MVS Authorized Assembler Services Guide.

IIUB063E  DBRC API ERROR: FUNC=function
            RC=xxxxxx RSN=yyyyyyyy

Explanation: An error is detected in an IMS DBRC API call. xxxxxxxx and yyyyyyyyy are return and reason codes of each function of the IMS DBRC API in hexadecimal. The meanings of the codes are described in IMS System Programming APIs. DBRC messages (DSPxxxxx) might be printed in the IIUSNAP data set or the IIUSTAT data set.

System action: Processing stops.

User response: Correct the error and rerun the job.

IIUB064I  DBD=[dbname1 partname] RSN=xxxxxxxx

Explanation: This message follows the IIUB063E message and shows the reason code for each database or partition that is stored in the AUTH output block (DSPAPAUB) in hexadecimal.

When RSN=C1003xx is displayed in the message, xx indicates the IMS authorization reason code in hexadecimal. For the meaning of the code, see message DFS047A in IMS Messages and Codes.

This message is issued only when DBAUTH YES,API is specified.

System action: None.

User response: None. This message is informational.

IIUB065W  THE ACCESS TO OUTPUT REPOSITORY WAS CANCELED.
            REASON: reason

Explanation: IMS Index Builder canceled its access to the IMS Tools Knowledge Base output repository, because the initialization process failed. reason shows one of the following texts:

Reason Meaning
ITKBLOAD DYNALLOC FAILURE
   Dynamic allocation for the library that is specified in the ITKBLOAD control statement failed.

HKTXXLI LOAD FAILURE
   Failed to load the HKTXXLI module.

ITKBLOAD OPEN FAILURE
   Failed to open the library that is specified in the ITKBLOAD control statement.

ITKBLOAD NOT AUTHORIZED
   The library that is specified in the ITKBLOAD control statement is not APF-authorized.

System action: Processing continues, but no reports are stored to the IMS Tools Knowledge Base output repository.

User response: If you want to store reports to the IMS Tools Knowledge Base output repository, specify the IMS Tools Knowledge Base product load module library correctly. If the reason is ITKBLOAD NOT AUTHORIZED, the library must be APF-authorized.

IIUB066I  THE ITKBLOAD DD NAME WAS SPECIFIED

Explanation: An IMS Tools Knowledge Base load module data set name was specified for the ITKBLOAD parameter, but the specification was ignored because
HKTLOAD DD statement was specified in the IMS Index Builder JCL.

**System action:** Processing continues. IMS Index Builder loads the IMS Tools Knowledge Base load module from the HKTLOAD DD statement.

**User response:** None. This message is informational.

---

**IIUB067I APF AUTHORIZATION CHECK FAILED (DD: ddname, REASON: nn)**

**Explanation:** The APF-authorization check module (IIUBAPF0) returned an error for the indicated DD statement. The reason code *nn* is used by the module internally.

**System action:** IMS Index Builder utility continues processing.

**User response:** This error was most likely caused by an IMS Index Builder internal error. Contact IBM Software Support.

---

**IIUB068E TIMEOUT OCCURRED IN DBRC API PROCESS**

**Explanation:** Timeout occurred while waiting for the response from IMS DBRC API. The maximum wait time is 30 minutes.

**System action:** Processing stops.

**User response:** There might be a performance degradation in the DBRC function. Wait a while and rerun the job.

---

**IIUB069E SUBSYS RECORD AND AUTHORIZATION REMAINED. SSID=xxxxxxxx**

**Explanation:** A DBRC SUBSYS record and the authorization state for databases remained due to an error. *xxxxxxxx* is the subsystem identification name.

This message is issued only when DBAUTH YES,API is specified.

**System action:** None.

**User response:** You must issue the following DBRC commands to remove the SUBSYS record and correct the authorization state of the RECON data sets:

- `CHANGE.SUBSYS SSID(xxxxxxxx) STARTRCV`
- `CHANGE.SUBSYS SSID(xxxxxxxx) ENDRCOV`
- `DELETE.SUBSYS SSID(xxxxxxxx)`

For more information, see *IMS Commands.*

---

**IIUB070I DB AUTHORIZATION IS REQUESTED. SSID=xxxxxxxx**

**Explanation:** The IMS DBRC authorization for the prime database and indexes is requested. *xxxxxxxx* is the subsystem identification name that is used for DBRC sign-on.

**System action:** Processing continues.

**User response:** None. This message is informational.

---

**IIUB071I DB UNAUTHORIZATION IS REQUESTED. SSID=xxxxxxxx**

**Explanation:** The IMS DBRC unauthorization for the prime database and indexes are requested through IMS DBRC API.

This message is issued only when DBAUTH YES,API is specified.

**System action:** Processing continues.

**User response:** None. This message is informational.

---

**IIUB072I SVC DUMP SUPPRESSED BY SVCDUMP CONTROL STATEMENT**

**Explanation:** An SVC dump was requested in the error recovery routines, but the request was suppressed by the SVCDUMP control statement specification (SVCDUMP NO).

**System action:** Processing continues.

**User response:** None. This message is informational.

If you want to obtain the SVC dump, specify SVCDUMP YES.

---

**IIUB073E module_name TASK IS TERMINATED FORCEDLY**

**Explanation:** The indicated service task was terminated due to a timeout. The maximum wait time is 5 minutes.

If the indicated module is IIUDINTL, this message indicates that the VSAM initialization process has not yet ended for an empty index.

**System action:** Processing stops.

**User response:** Rerun the job. If the situation persists, contact IBM Software Support.

---

**IIUB074I task_name TASK IS TERMINATED FORCEDLY**

**Explanation:** The indicated task was terminated due to a timeout. The maximum wait time is 5 minutes. *task_name* shows the service task name or the job name of a subordinate address space. This message might be issued when the system is busy.

**System action:** Processing continues.
User response: None if no error message is issued. If this message continues to be issued, contact IBM Software Support.

IIUB075I  PSINDEX PARALLEL SORTING IN EFFECT
Explanation: The parallel sort for a PSINDEX is enabled because the PSORT keyword was found on the MAXTASKS control statement.
System action: Processing continues.
User response: None. This message is informational.

IIUB076W  NO INDEX TO BE BUILT
Explanation: The BLD_SECONDARY ALL keyword or the BLD_ALL keyword is specified on the PROC statement, but there is no index to be built.
System action: Processing terminates without building any indexes.
User response: None.

IIUB077I  DATA BASE dbname IS NOT REGISTERED WITH DBRC
Explanation: The named database is not registered with DBRC.
System action: Processing continues without issuing DBRC authorization requests or DBRC notifications for dbname.
User response: Correct the error and rerun the job.

IIUB078I  DATA BASE dbname IS NOT REGISTERED WITH DBRC
Explanation: The named database is not registered with DBRC. All databases must be registered to the RECON data sets because the RECON data sets were initialized with the FORCER option.
System action: Processing stops.
User response: For a non-HALDB, if you do not want to register this database, specify DBRC=N on the INPUT control statement.

For a HALDB, if DD statements for the database data sets are specified in the JCL, consider removing the statements.

IIUB080E  DB DATA SET OPEN FAILURE - REASON=yy, DSN=dsname
Explanation: The database data set could not be opened for scan. yy shows one of the following reason codes:

- 01 The block size or the CI size is zero. Check whether this data set had been loaded or initialized.
- 99 Unknown reason.

System action: Processing stops.
User response: Correct the error and rerun the job.

IIUB081I  ILDS INITIALIZED FOR DATABASE WITH LOGICAL RELATIONSHIPS
Explanation: The control statement ILDS INITONLY is specified for a HALDB that has logical relationships.
System action: Processing continues.
User response: If ILE records are needed for logical relationships, remove the ILDS INITONLY statement and rerun the job.

IIUB082E  DBRC AUTHORIZATION ERROR: DBD=dbname DDN=ddname CODE=xx RC=yyyy
Explanation: The IMS Database Authorization module (DFSDBAU0) returned an error. dbname, ddname, and the code (xx) indicate the resource information that is passed to the DFSDBAU0 module, and RC=yyyy indicates the return code from the module. xx indicates the utility type code of the caller:

- 05 Database Surveyor utility
- 06 Database Scan utility
- 11 Database Recovery utility

System action: Processing stops.
User response: See the IMS DFS047A WTO message to determine the cause of the problem. Correct the error, and rerun the job.
IIUB083E  HPSRWFP FILE IS NOT FOUND

Explanation:  No HPSRWFP file is found.

System action:  Processing stops.

User response:  Ensure that the HPSRWFP data sets were generated correctly by IMS High Performance Load.

IIUB084I  HPSRWFP FILE IS NOT FOUND:  PART=xxxxxxxx

Explanation:  The HPSRWFP file for the indicated HALDB partition is not found.

System action:  Processing continues.

User response:  If the indicated partition is empty and if the partition was not processed by IMS High Performance Load, you can ignore this message. In other cases, rerun the IMS High Performance Load job to generate the HPSRWFP data sets.

IIUB085E  INCORRECT HPSRWFP FILE:  RSN=xx

Explanation:  The HPSRWFP file is incorrect.  xx shows one of the following reason codes:

01  The name of the partition data set is not found in the RECON data sets.
02  The reorganization number of the partition is not found in the RECON data sets.
03  The records for the indicated PSINDEX are missing.
04  The header record is incorrect.
05  The HPSRWFP format level is unrecognized.

System action:  Processing stops.

User response:  Ensure that the HPSRWFP data sets were generated correctly by IMS High Performance Load.

IIUB086E  'INPUT STATEMENT OPTION HPSRWFP IS NOT APPLICABLE FOR NON-HALDB

Explanation:  The INPUT HPSRWFP statement is specified. However, the database that is specified for the PROC statement is not a HALDB.

System action:  Processing stops.

User response:  Correct the error and rerun the job.

IIUB087E  'ILDS INITONLY' IS REQUIRED FOR HPSRWFP INPUT

Explanation:  The ILDS INITONLY statement is required for the PROC BLD_ILDS statement or the PROC BLD_ALL statement when the INPUT HPSRWFP statement is specified.

System action:  Processing stops.

User response:  Correct the error and rerun the job.

IIUB088E  'WFPHLQ' IS REQUIRED FOR HPSRWFP INPUT

Explanation:  The INPUT HPSRWFP statement is specified, but the WFPHLQ statement is not specified.

System action:  Processing stops.

User response:  Correct the error and rerun the job.

IIUB091I  SNAP DUMP NOT PRINTED BECAUSE DUMMY SPECIFIED FOR IIUDUMP DD.  JOBID=job_id

Explanation:  A snap dump is not printed because IIUDUMP DD DUMMY is coded in the JCL of the indicated job.

System action:  Processing continues.

User response:  If the snap dump is needed for analysis, remove the IIUDUMP DD statement and rerun the job.

IIUB092I  DBD=[dbdname] (RECON)

Explanation:  This message follows message IIUB063E and shows the database for which the IMS DBRC API call failed.

System action:  None.

User response:  None. This message is informational.

IIUB093I  COMMAND=command

Explanation:  This message follows message IIUB063E and shows the DBRC command for which the IMS DBRC API call failed.

System action:  None.

User response:  None. This message is informational.
IIUB094I  THE NUMBER OF INDEX SOURCE
SEGMENTS (segment_name) EXCEEDS
4,294,967,296

Explanation:  The number of scanned index source
segments exceeds 4,294,967,296.  The count of scanned
segments and the count of built index records that are
printed in the report are inaccurate.

System action:  Processing continues.
User response:  None. This message is informational.

IIUB095M  xxxxxxxx xxxxxxxx AN UNEXPECTED
MESSAGE xxx OF xxxxxxxx
HH.MM.SS.TH

Explanation:  Potential problems with the messaging
system.

System action:  Processing continues.
User response:  Supports failure analysis.

IIUB096M  xxxxxxxx xxxxxxxx IMS INDEX
BUILDER CONTROL FLOW EVENT
xxx A xxxxxxxx HH.MM.SS.TH

Explanation:  Reports critical events by event code xxx.

System action:  None.
User response:  See IMS Index Builder control flow
event codes for information about the events being
reported.

IIUB097I  THE NUMBER OF HPSRWFP
RECORDS EXCEEDS 4,294,967,296

Explanation:  The number of HPSRWFP records
exceeds 4,294,967,296.  The count of the total HPSRWFP
records that is printed in the report is inaccurate.

System action:  Processing continues.
User response:  None. This message is informational.

IIUB098I  THE NUMBER OF HPSRWFP
RECORDS (index) EXCEEDS
4,294,967,296

Explanation:  The number of HPSRWFP records for the
indicated index exceeds 4,294,967,296.  The count of
HPSRWFP records and the count of built index records
that are printed in the report are inaccurate.

System action:  Processing continues.
User response:  None. This message is informational.

IIUB099E  PROCESSING ERROR
ENCOUNTERED, MODULE
module_name

Explanation:  A processing error occurred in the IMS
Index Builder module module_name.  One of the
following reason messages will accompany this
message:

(01)  DBD DBDNAME NOT FOUND IN
SECONDARY INDEX TABLE

(02)  ERROR WHILE PROCESSING function

(06)  DYNAMIC ALLOCATION ERROR FOR
DBD=dbdname, DDN=ddname, REASON=reason

System action:  Processing might or might not
continue, depending on the error.
User response:  Correct the error condition and rerun
the job.

IIUB100E  INDEX BUILDER INIT FAILURE —
REASON=reason1  ERROR=reason2

Explanation:  An error was encountered during
initialization of the IMS Index Builder environment.
The reason1 and reason2 codes are explained in “IMS
Index Builder initialization phase primary reason
codes” on page 124 and “IMS Index Builder
initialization phase secondary reason codes” on page
128.

System action:  Processing stops.
User response:  See the explanation for the reason code
that this message issued for information about the
reason code and the appropriate action that you should
take.  Correct the error condition and rerun the IMS
Index Builder job.

IIUB101I  SVC DUMP SUPPRESSED BY DAE

Explanation:  An SVC dump was requested but was
suppressed by DAE.

User response:  None. This message is informational.
If a dump is needed, set a SLIP or reset DAE.

IIUB102E  INDEX PROCESS TERMINATED
FORCEDLY DUE TO A TIMEOUT

Explanation:  The index process terminated because of a
timeout.

System action:  Processing stops.
User response:  This error might be caused by a
system slow down.  The timeout value can be changed
by the TMRWAIT control statement.  Increase the value
for the TMRWAIT control statement and then rerun the
job.  For more information, see “TMRWAIT control
statement” on page 89.
IIUB103I  statement IS IGNORED UNLESS STAND-ALONE INDEX BUILDER

Explanation:  The indicated statement is ignored because the statement is supported only in stand-alone Index Builder jobs.

System action:  Processing continues.

User response:  None. This message is informational.

IIUB104E  APF AUTHORIZATION CHECK FAILED

Explanation:  One or more libraries in the STEPLIB concatenation are not correctly APF authorized.

System action:  Processing stops.

User response:  APF authorize all the libraries that are concatenated to the STEPLIB and rerun the job.

IIUB105W  TOSI ERROR: FUNC=function RC=rc RSN=rst

Explanation:  The IMS Tools Online System Interface returned an error. The codes are shown in hexadecimal.

System action:  Processing continues but IMS commands are not issued.

User response:  Locate the preceding messages that were issued by the IMS Tools Online System Interface to identify the cause of the error. Messages that are issued by the IMS Tools Online System Interface are prefixed with FOI.

IIUB106I  TOSI ERROR: FUNC=function RC=rc RSN=rst

Explanation:  The IMS Tools Online System Interface returned a warning. The codes are shown in hexadecimal.

System action:  Processing continues.

User response:  Locate the preceding messages that were issued by the IMS Tools Online System Interface to identify the cause of the error. Messages that are issued by the IMS Tools Online System Interface are prefixed with FOI.

IIUB107E  DB=database IS IN USE. IMSID=imsid

Explanation:  The indicated database is in use in the indicated IMS subsystem. The IMS /DBD command cannot be processed.

System action:  Processing stops.

User response:  Terminate the program that is using the database or wait until the program ends. Then, rerun the IMS Index Builder job.

IIUB108I  IMS [/DBD1[/DBR1[/STA] COMMAND PROCESSED: IMSID=imsid DB=dbdname RC=rc

Explanation:  The IMS /DBD, /DBR, or /STA command for the indicated database is processed in the indicated IMS subsystem.

System action:  Processing continues.

User response:  None. This message is informational.

IIUB109W  IMS [/DBD1[/DBR1[/STA] COMMAND PROCESSED: IMSID=imsid DB=dbdname RC=rc

Explanation:  The IMS /DBD, /DBR, or /STA command for the indicated database failed in the indicated IMS subsystem.

System action:  Processing continues but no more /DBD, /DBR, and /STA commands are issued.

User response:  For the meaning of the return code, see the explanation of the IMS DFS0488I message.

IIUB110W  IMS [/DBD1[/DBR1[/STA] COMMAND NOT RETURNED: IMSID=imsid DB=dbdname

Explanation:  The IMS /DBD, /DBR, or /STA command for the indicated database is requested for the indicated IMS subsystem, but no response is returned.

System action:  Processing continues.

User response:  If the system is busy, increase the IMS Tools Online System Interface response timeout value on the TOSIWAIT control statement. For more information, see TOSIWAIT control statement on page 93.

IIUB111W  IMS /STA COMMAND CANCELED FOR DB=dbdname BECAUSE IMAGE COPY NEEDED

Explanation:  The IMS /STA command cannot be issued because the IC NEEDED flag is still turned on for a data set of the indicated database.

System action:  Processing continues.

User response:  Create image copies of the index that was built. If you do not need an image copy of the index, specify the VIC YES control statement and rerun the job. For more information, see VIC control statement on page 94.
IIUB112E  ERROR IN DFSPSEL REQUEST=request
PART=partname  RC=rc  RSN=rsn

Explanation: An error occurred when the IMS DFSPSEL macro was used. rc is the return code in decimal and rsn is the reason code in hexadecimal. For the meaning of the reason code, see the explanation of message DFS0832I in IMS Messages and Codes.

System action: Processing stops.

User response: Correct the error condition and rerun the job.

IIUB113E  ERROR IN DFSLOC0

Explanation: An error occurred when the IMS DFSLOC macro was used to open or close a database. See the DFS0730I message issued before this message for details on the error.

System action: Processing stops.

User response: Correct the error condition and rerun the job.

IIUB114E  VSAM EXTENDED ADDRESSABILITY IS NOT SUPPORTED

Explanation: The data set was allocated by using a data class that specifies the extended addressability attribute. VSAM data sets that are defined with the extended addressability attribute are not supported.

System action: Processing stops.

User response: In the index attributes report, locate message IIUB053E, which identifies the data set in error. Remove the extended addressability attribute from the data set and rerun the job.

IIUB116I  IEANTCR SERVICE FOR name: RC=rc

Explanation: The IEANTCR service returned the indicated return code. The code is shown in hexadecimal format.

System action: Processing continues.

User response: None. This message is informational.

IIUB117E  COMPRESSION EXIT ROUTINE

nnnnnnnn INITIALIZATION ERROR - Uaaaa REASON rrrrrrrrr

Explanation: An initialization error was detected by a Segment Edit/Compression exit routine. In the message text:

nnnnnnnn
Name of the exit routine.

Uaaaa The IMS user abend code generated by the exit routine.

rrrrrrrr The unique label at which the error was detected. This label corresponds to the error reason code. Find the meaning of the reason code in the user’s guide of the Segment Edit/Compression exit routine or contact the supplier of the Segment Edit/Compression exit routine.

System action: Processing stops.

User response: Determine the cause of the error and correct the problem. See “IMS Index Builder restrictions” on page 19 for consideration about exit routines.

IIUB118E  VSAM ATTRIBUTES NOT CONSISTENT WITH DBD

Explanation: One or more attributes of the VSAM data set are not consistent with the segment definitions in the DBD.

- KEY LENGTH
- KEY OFFSET
- RECORD LENGTH

System action: Processing stops.

User response: In the index attributes report, locate message IIUB053E, which identifies the data set in error. Correct the RECORDSIZE or the KEYS parameter of the IDCAMS DEFINE CLUSTER command for the VSAM data set. The recommended values are printed in the DBDGEN output.

IIUB119I  INDEX [DBD | PART]=xxxxxxxx - n,mn,mn,mn RECORDS LOADED ["COMPLETED"]

Explanation: This WTO message shows the number of loaded index records for an index or a partition for monitoring the process.

System action: Processing continues.

User response: None. This message is informational.

IIUB120I  PSINDEX UPDATE MODE IN EFFECT

Explanation: The PSINDEX is being built in update mode because the UPDATE option is specified on the PROC control statement.

System action: Processing continues.

User response: None. This message is informational.

IIUB121E  HPSRWF FILE IS NOT FOUND:
PART=partname

Explanation: Load mode is selected for building the PSINDEX database. Load mode requires that HPSRWF files exist for all the partitions, but the HPSRWF file for the indicated HALDB partition was not found.
**System action:** Processing stops.

**User response:** Rerun the IMS High Performance Load job to generate all the HPSRWF.P files, or rerun the IMS Index Builder job in update mode. See "[PROC control statement](#)" on page 73 for more information about load and update modes.

---

**IIUB122I**  
'declaration' STATEMENT OPTION option
IS NOT APPLICABLE FOR NON-HALDB

**Explanation:** The indicated option is not supported for non-HALDBs.

**System action:** Processing continues.

**User response:** None. This message is informational.

---

**IIUB123E**  
VSAM EMPTY DATA SET IS NOT SUPPORTED IN UPDATE MODE

**Explanation:** The UPDATE option is specified on the PROC control statement but one or more PSINDEX partitions are empty.

**System action:** Processing stops.

**User response:** Rerun the job in load mode.

---

**IIUB124I**  
IMS MANAGED ACBS ENABLED.
[ACTIVE | PENDING] DATABASES ARE REFERRED

**Explanation:** The IMS management of ACBs is enabled. IMS Index Builder obtains database information from the IMS catalog.

ACTIVE indicates that the database information is retrieved from the IMS catalog directory data sets. PENDING indicates that the database information is retrieved from the IMS catalog staging data sets.

**System action:** Processing continues.

**User response:** None. This message is informational.

---

**IIUB125E**  
IMS TOOLS CATALOG INTERFACE ERROR: FUNCTION=close RC=rc
RSN=rsn

**Explanation:** The IMS Tools Catalog Interface returned an error. The codes are shown in hexadecimal. A preceding message from the IMS Tools Catalog Interface describes the detail of the error. The typical causes are:

• The IMS Tools Catalog Interface library is not found in the STEPLIB concatenation.
• The input or output database is not registered to the IMS catalog.

**System action:** Processing stops.

**User response:** Locate the error message issued by the IMS Tools Catalog Interface, correct the error, and rerun the job.

---

**IIUB126I**  
IMS TOOLS CATALOG INTERFACE ERROR: FUNCTION=CLOSE RC=rc
RSN=rsn

**Explanation:** The IMS Tools Catalog Interface CLOSE function returned an error. The codes are shown in hexadecimal.

**System action:** Processing continues.

**User response:** None. This message is informational.

---

**IIUB127E**  
FAILED TO ACCESS PROCLIB MEMBER: DFSDFxxx

**Explanation:** An error occurred while processing the DFSDFxxx member in the PROCLIB data set. The possible causes are:

• The PROCLIB DD statement is not specified.
• The DFSDFxxx member is not found in the PROCLIB data set.
• An open error occurred in the PROCLIB data set.
• A statement error is found in the DFSDFxxx member.

**System action:** Processing continues.

**User response:** Identify the cause of the error. Correct the error condition and rerun the job.

---

**IIUB128E**  
IMS MANAGEMENT OF ACB WITH UNREGISTERED CATALOG IS NOT SUPPORTED

**Explanation:** IMS managed ACBs is enabled but the IMS catalog database is not registered to the DBRC RECON data sets. IMS Index Builder does not support IMS catalogs that are not registered to the DBRC RECON data sets.

**System action:** Processing stops.

**User response:** Register the IMS catalog databases to DBRC or disable IMS management of ACBs, then rerun the IMS Index Builder job.

---

**IIUB130I**  
HPQZIIP USED (HH:MM:SS.THMIJU)
WAS: xxx:xx:xxxxx

**Explanation:** This informational message shows the CPU time consumed by zIIP processors for the IMS Index Builder job.

**System action:** Processing continues.

**User response:** None. This message is informational.
IIUB798E • IIUB999E

IIUB798E  SUBORDINATE ADDRESS SPACE IS BEING ABENDED: JOBID=job_id

Explanation: The master address space determined that the subordinate address did not end as expected. The subordinate address space to be terminated is identified by job_id. The job name is shown in the address spaces report.

System action: An abend U0798 request is made in the subordinate address space.

User response: Locate preceding messages from the subordinate on the console log and the job output from the master address space and contact IBM Support.

IIUB799E  SUBORDINATE ADDRESS SPACE IS BEING TERMINATED: JOBID=job_id

Explanation: The master address space determined that the subordinate address did not end as expected. Either an abend could not be requested or the abend was not successful. The subordinate address space to be terminated is identified by the job_id. The job name is shown in the address spaces report.

System action: The address space is ended.

User response: Locate preceding messages from the subordinate on the console log and the job output from the master address space and contact IBM Support.

IIUB999E  INTERNAL ERROR RSN=rsn

Explanation: An internal error occurred. The hexadecimal reason code identifies the type of the error. Additional messages showing the error detail may follow this message.

System action: Processing stops.

User response: Contact IBM Software support with appropriate diagnostic documentation.
Chapter 13. How to look up message explanations

You can use several methods to search for messages and codes.

Searching for messages on the web

You can use any of the popular search engines that are available on the web to search for message explanations. When you type the specific message number or code into the search engine, you are presented with links to the message information in IBM information centers.
Chapter 14. IMS Index Builder abend codes

This reference section provides detailed information about IMS Index Builder abend codes.

IMS Index Builder issues user abends when fast ending of the process is more efficient, and diagnostic information would aid IBM Software Support.

U100  An initialization phase abend. It can follow an IIUB100E error message if the abend flag is set by the initialization modules. IB31 initialization does not complete, and documentation is provided for IBM Software Support.

U199  A diagnostic abend triggered by the control statement TRACE D4. The process is stopped at a diagnostic check point, and documentation is provided for IMS Software Support.

U701 - U799  These abends can be issued instead of reason codes R701 - R799. They have the same meaning as the corresponding reason codes. The process is ended immediately, and documentation is provided to aid failure analysis.
Chapter 15. Diagnostics Aid

If you have a problem that you think is not a user error, use the Diagnostics Aid to collect the necessary information before you contact IBM Software Support.

1. Run IMS Index Builder Diagnostics Aid (IIUUDIAG) and obtain the IMS Index Builder Load Module APAR Status report.
2. Attach the report to the other diagnostic documents (such as job dump list or I/O of the utility).
3. Report the error to IBM.

IMS Index Builder Diagnostics Aid (IIUUDIAG) generates the IMS Index Builder Load Module APAR Status report for the IMS Index Builder maintenance by IBM. This report shows the latest APAR fixes that are applied to each module of IMS Index Builder components.

Topics:
- “How to run Diagnostics Aid with JCL stream” on page 166
- “Diagnostics Aid report” on page 167
- “Diagnostic messages and codes” on page 168
How to run Diagnostics Aid with JCL stream

To run IMS Index Builder Diagnostics Aid (IIUUDIAG), supply an EXEC statement and a DD statement that define the output data set.

**EXEC**

This statement must be in the following form:

```
//stepname EXEC PGM=IIUUDIAG
```

**STEPLIB DD**

This statement defines the library containing the IIUUDIAG program (typically IIU.SIIULMOD).

**SIIULMOD DD**

This statement defines the library containing the IMS Index Builder load modules (typically IIU.SIIULMOD) for which you have a problem.

If this DD statement is not provided, or if DD DUMMY is specified, the IMS Index Builder Load Module APAR Status report is not generated.

It is always recommended that you specify this DD statement.

**SYSPRINT DD**

This output data set contains the IMS Index Builder Load Module APAR Status report. The data set contains 133-byte, fixed-length records. It can reside on a tape, direct-access device, or printer; or it can be routed through the output stream. If BLKSIZE is coded in the DD statement, it must be a multiple of 133. However, it is recommended that you use the following:

```
//SYSPRINT DD SYSOUT=A
```

The following figure shows example JCL statements to run IMS Index Builder Diagnostics Aid.

```
//DIAG EXEC PGM=IIUUDIAG
//STEPLIB DD DISP=SHR,DSN=IIU.SIIULMOD
//SIIULMOD DD DISP=SHR,DSN=IIU.SIIULMOD
//SYSPRINT DD SYSOUT=A
```

*Figure 21. Example JCL statements to run IMS Index Builder Diagnostics Aid*
Diagnostics Aid report

Diagnostics Aid (IIUUDIAG) generates the IMS Index Builder Load Module APAR Status report for the IMS Index Builder maintenance by IBM.

**IMS Index Builder Load Module APAR Status report**

The IMS Index Builder Load Module APAR Status report contains information about the modules and their applied APARs.

This report contains the following information:

**MODULE LIBRARY**
This field includes the data set names that are specified in the SIIULMOD DD statement. If more than 30 data sets are concatenated, only the first 30 are listed.

**MODULE NAME**
This field shows the name of either the load module member or the alias that belongs to the IMS Index Builder feature.

**ALIAS-OF**
This field shows the name of the original member of the alias. If the module name is not an alias, this field is left blank.

**CSECT NAME**
This field shows the name of the CSECT included in the module. The CSECT names are listed in the order in which they are included in the module.

**APAR NUMBER**
This field shows the latest APAR number that is applied to the module that is represented by the CSECT name. If no APAR is applied, NONE is shown.

**APAR FIX-DATE**
This field shows the date on which the modification for the module represented by the CSECT name was prepared. If no APAR is applied, N/A is shown.

**Notes:**
1. If the CSECT name does not start with III, or if the program structure of the CSECT does not identify the APAR number and the APAR fixed date as specified by the IMS Index Builder module standard, the fields APAR NUMBER and APAR FIX-DATE are filled with asterisks (*).
2. If the load module is a member of the PDSE library, the following statement is shown on the report line and the job completes with a return code of 4.
   
   ** IT CAN NOT BE ANALYZED DUE TO PDSE LIBRARY MEMBER **

3. If the load macro for an IMS Index Builder member fails, the following statement is shown on the report line and the job completes with a return code of 8.

   ** IT CAN NOT BE ANALYZED DUE TO LOAD FAILED MEMBER **
Diagnostic messages and codes

The following topics describe the return codes, abend codes, and messages that are issued by IMS Index Builder Diagnostics Aid.

Return codes

IIUUDIAG generates return codes to indicate the success or failure of a job.

IIUUDIAG issues the following return codes.

<table>
<thead>
<tr>
<th>Return code</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Successful completion of the program.</td>
</tr>
<tr>
<td>4</td>
<td>Warning messages were issued, but the requested operation was completed.</td>
</tr>
<tr>
<td>8</td>
<td>Error messages were issued, but the request operation was completed.</td>
</tr>
</tbody>
</table>

Abend codes

All 36xx abend codes are accompanied by an IIUUD36xx message. See the appropriate message for problem determination.

Messages

The IIUD messages are issued by the Diagnostic Aid.

**IIUD1001I**  
**IIUUDIAG ENDED NORMALLY**

**Explanation:** This message is generated when IIUUDIAG has been completed successfully.

**System action:** IIUUDIAG completes the job with a return code of 0.

**User response:** None. This message is informational.

**IIUD1002W**  
**IIUUDIAG ENDED WITH WARNINGS**

**Explanation:** This message is generated when trivial error conditions are encountered by IIUUDIAG.

**System action:** IIUUDIAG ends with a return code of 4.

**User response:** See other messages that are generated by Diagnostic Aid to determine the nature and the cause of the detected errors. Correct the problem, and rerun the job.

**IIUD1003E**  
**IIUUDIAG ENDED WITH ERRORS**

**Explanation:** This message is generated when severe error conditions are encountered by IIUUDIAG.

**System action:** IIUUDIAG ends with a return code of 8.

**User response:** See other messages that are generated by IIUUDIAG to determine the nature and the cause of the detected errors. Correct the problem, and rerun the job.

**IIUD1005W**  
**SIIULMOD DD STATEMENT NOT FOUND**

**Explanation:** IIUUDIAG could not find the SIIULMOD DD statement.

**System action:** IIUUDIAG sets an end-of-job return code of 4 and continues processing. IIUUDIAG does not generate a report for the load module.

**User response:** If you intended to specify the indicated DD statement, correct the error, and rerun the job.

**IIUD1006W**  
**DUPLICATE member name IN LIBRARY DDNAME ddname**

**Explanation:** IIUUDIAG found a duplicated member in the concatenated libraries.

**System action:** IIUUDIAG uses the member that is first found in the concatenated libraries. IIUUDIAG sets an end-of-job return code of 4 and continues processing.

**User response:** Make sure which libraries have correct module/macro libraries. Correct the error, and rerun the job if necessary.
IIUD1007W  DUMMY SPECIFIED FOR SIIULMOD DD STATEMENT
Explanation: DUMMY was specified for the SIIULMOD DD statement.
System action: IIUUDIAG sets an end-of-job return code of 4 and continues processing. IIUUDIAG does not generate a report for the load module.
User response: If you did not intend to specify the dummy DD statement, correct the error, and rerun the job.

IIUD1008W  NO IMS INDEX BUILDER MODULE MEMBERS FOUND IN DDNAME SIIULMOD
Explanation: IIUUDIAG could not find any IMS Index Builder module members from the DD ddname data set.
System action: IIUUDIAG sets an end-of-job return code of 4 and continues processing.
User response: Ensure that the libraries have correct IMS Index Builder module libraries. Correct the error and rerun the job.

IIUD2001E  LOAD FAILED FOR DDNAME ddname MODULE member
Explanation: IIUUDIAG could not load a member name from ddname.
System action: IIUUDIAG sets an end-of-job return code of 8 and continues processing.
User response: Make sure that the member indicated exists in the data set specified for the indicated ddname. Correct the error, and rerun the job.

IIUD3600E  OPEN FAILED FOR DDNAME ddname
Explanation: The named DCB could not be opened.
System action: IIUUDIAG ends with an abend code of U3600.
User response: Make sure that a ddname DD statement exists, and that it specifies the correct DD parameter. Correct any errors, and rerun the job.

IIUD3601E  GET FAILED FOR DDNAME ddname
Explanation: The GET failed for a directory from the DD ddname data set.
System action: IIUUDIAG ends with an abend code of U3601.
User response: See the MVS system message and its programmer response. Correct the error, and rerun IIUUDIAG. If the error persists, report it to IBM.

IIUD3602E  READ FAILED FOR DDNAME ddname MEMBER member
Explanation: The READ failed for a member from the DD ddname data set.
System action: IIUUDIAG ends with an abend code of U3602.
User response: See the MVS system message and its programmer response. Correct the error, and rerun IIUUDIAG. If the error persists, report it to IBM.

IIUD3603E  BDLI FAILED FOR DDNAME ddname MEMBER member
Explanation: The member was not found when the BDLI macro searched the PDS directory for the ddname.
System action: IIUUDIAG ends with an abend code of U3603.
User response: Make sure that the member indicated exists in the data set specified for the indicated ddname. Correct the error, and rerun the job. If the error persists, report it to IBM.

IIUD3604E  LOAD FAILED FOR DDNAME ddname MODULE member
Explanation: IIUUDIAG could not load the member name from the ddname.
System action: IIUUDIAG ends with an abend code of U3604.
User response: See the MVS system message and its programmer response. Correct the error, and rerun IIUUDIAG. If the error persists, report it to IBM.

IIUD3605E  DELETE FAILED FOR MODULE member
Explanation: IIUUDIAG could not delete a member name.
System action: IIUUDIAG ends with an abend code of U3605.
User response: Report it to IBM.

IIUD3606E  PUT FAILED FOR SYSPRINT
Explanation: IIUUDIAG could not put report data in SYSPRINT.
System action: IIUUDIAG ends with an abend code of U3606.
User response: See the MVS system message and its programmer response. Correct the error, and rerun IIUUDIAG. If the error persists, report it to IBM.
IIUUD3607E • IIUUD3607E

IIUUD3607E  OPEN FAILED FOR SYSPRINT
Explanation:  SYSPRINT DCB could not be opened.
System action:  IIUUDIAG ends with an abend code of U3607.
User response:  Make sure that a ddname SYSPRINT DD statement exists, and that it specifies the correct DD parameter. Correct any errors, and rerun the job.

IIUUD3608E  FIND FAILED FOR DDNAME ddname MEMBER member
Explanation:  The FIND failed for a member from DDNAME ddname data set.
System action:  IIUUDIAG ends with an abend code of U3608.
User response:  Make sure that the member indicated exists in the data set specified for the indicated ddname. Correct the error, and rerun the job. If the error persists, report it to IBM.

IIUUD3609E  DEVTYPE FAILED FOR DDNAME ddname
Explanation:  The DEVTYPE failed for a DDNAME ddname data set.
System action:  IIUUDIAG ends with an abend code of U3609.
User response:  Report it to IBM.

IIUUD3610E  RDJFCB FAILED FOR DDNAME ddname
Explanation:  The READJFCB failed for a DDNAME ddname data set.
System action:  IIUUDIAG ends with an abend code of U3610.
User response:  Report it to IBM.

IIUUD3611E  GETMAIN FAILED. INSUFFICIENT STORAGE TO RUN THE JOB
Explanation:  Workspace for IIUUDIAG could not be obtained.
System action:  IIUUDIAG ends with an abend code of U3611.
User response:  Increase the region size, and rerun the job.

IIUUD3612E  TOO MANY IMS INDEX BUILDER MODULE MEMBERS DETECTED IN DDNAME SIIULMOD
Explanation:  Too many IMS Index Builder members are in the SIIULMOD DD data set.
Use the following information to interpret syntax diagrams included in the IMS Index Builder topics.

Topics:
• Chapter 16, “How to read syntax diagrams,” on page 173
Chapter 16. How to read syntax diagrams

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

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

  - Required items appear on the horizontal line (the main path).

  ![Required item diagram]

  - Optional items appear below the main path.

  ![Optional item with required item diagram]

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

  ![Optional item above required item diagram]

  - If you can choose from two or more items, they appear vertically, in a stack.
    If you **must** choose one of the items, one item of the stack appears on the main path.

  ![Two required choices diagram]

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

  ![Optional choices diagram]

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

  ![Default choice diagram]

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

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

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown. Variables appear in all lowercase italic letters (for example, column-name). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses; for example, (1).
Part 5. 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.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM’s future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to
IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided “AS IS”, without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

**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.html](http://www.ibm.com/legal/copytrade.html).

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the Unites States, other countries, or both.

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 119
IMS Index Builder 163
accessibility overview 23
activating installation 28
activation 25
IMS Tools Knowledge Base 31
ITKB 31
adding new secondary index example 40
APF authorization 66
architecture diagram 16
authorized program facility 66

B
blanks (none) in parameter syntax 72
building 37
HALDB partition ILDS 57
HIDAM on an alternate primary index 54
IMS catalog databases 63
IMS catalog indexes 58
partitioned secondary index HPSRWFP 48
PHIDAM partition primary index 56
secondary index DFSURIDX 45
functions 38
IBSCAN 38
reasons 38
split function 43
using DFSURWF1 42
using HPSRSIDX 42
building HIDAM primary index IBSCAN 52

C
changes summary 4
CLASS input control statement 77
codes return 122
coding rules control statements 72
common services group 15
compatibility 18
compatibility with other utilities 18
COMPAUTH input control statement 77
components 15
configuration 25
IMS Tools Knowledge Base 31
IMS Tools Online System Interface 32
ITKB 31
operating environment 27
prerequisites 26
control flow events 117
control flow events reported on IIUSNAP 119
control statements CLASS 77
COMPAUTH 77
DBAUTH 78
described 72
DFSDF 79
IIURSORT 80
ILDS 80
INDEX 81
INPUT 75
ITKBLOAD 82
ITKBSVRD 82
JOBNSCAN 82
JOBNSORT 83
MAXTASKS 83
OUTPUT 84
PARTNAME 85
PROC 73, 98
PROGMON 85
rules for coding 72
SIDXBUF 86
SORTFFSZ 87
SORTOPT 87
STRIPE 88
summary 98
SVCDUMP 89
TMRRWAIT 89
TOSIDBD 90
TOSIDBR 91
TOSISTA 92
TOSIWTAI 93
TOSIWTAI 93
TOSIXCFG 93
VIC 94
WFPDEL 94
WFPHELQ 95
ZIIIPMODE 95
cookie policy 177, 180
creating secondary index functions 38
IBSCAN 38
reasons 38
creating secondary indexes IBSCAN 40
tasks 40
critical events report 69

data set group DD statements 71
data sets initializing example 50
database reorganization 42
database scan 75
DBAUTH input control statement 78
DBDs specifying new 70
DBRC 75
DBRC command authorization 30
DBRC recovery changing JCL 51
DD statements 66
data set group 71
DDEFPPDS 67
DFSACBPBD 67
DFSDCL 67
DFSPARM 66, 67
DFSURIDX 67, 98
DFSURWF1 68, 98
DFSUVSAMP 68
HPSRSDIX 68
HSSROPT 68
IICAPB 68
IIUCAPT 69
IIUDUMP 69
IIuin 69
IIUOPT 69
IIUUPRINT 69
IIUSNAP 69
IIUSOUT 70
IIUWFn 70
IMS 70
IMSALT 40, 70, 98
IMSDALIB 70
OUTPUT 70, 98
PROCLIB 71
RECONN 71
secondary index 71
STEPFLIB 66
DDEFPPDS DD statement 67
defining runtime defaults 103
dependencies 18
DFSACBPBD DD statement 67
DFSDCL DD statement 67
DFSDF input control statement 79
DFSMPAR DD statement 67
DFSURIDX 37, 75
building secondary index 45
DFSURIDX DD statement 67, 98
DFSURWF1 75
building secondary index 42, 43
prefix resolution 43
DFSURWF1 DD statement 68, 98
DFSUVSAMP DD statement 68
diagnostic documentation 115
diagnostic messages and codes 168
Diagnostics Aid 165
distribution library IIUFBLDPR listing 54
IIUFBLDNS, listing 40
documentation accessing 21
sending feedback 21
driver 15
duplicate keys 111
DUPLKEY stripe 111
dynamic allocation 68, 70, 71
dynamic allocation modules 70
E
error handling 122
error messages 145
error records
   snapshots of 66
extamples
   adding secondary index to existing database 40
   initializing data sets 50
   rebuilding alternate primary index 54
EXEC DD statement 66
EXEC statements 66

F
features 3, 9
FILSZ parameter 87
functions 9, 37, 51

G
generic PSBs 70

H
HALDB partition ILDS
   rebuilding 57
hardware requirements 26
HIDAM primary index
   building 52
   rebuilding on an alternate primary index data set 54
   hierarchical scan
      using High Performance Unload 68
   High Performance Unload
      for hierarchical scan 68
      interface 68
highlights 9
HPSRSIDX 75
   building secondary index 42
HPSRSIDX DD statement 68
HPSRFWP 75
   building secondary index 48
HSSROPT DD statement 68
IIUIN DD statement 69
IIUOPT DD statement 69
IIUPRINT DD statement 69
IIURDFT
   adding parameters 103
IIUSORT input control statement 80
IIUSNAP 119
IIUSNAP DD statement 69
IIUSNAP output file 118
IIUSORT 99
IIUSOUT DD 118
IIUSOUT DD statement 70
IIUSTART 66
IIUUDIAG 165
IIUWFFn DD statement 70
ILDS input control statement 80
IMS catalog 63
IMS catalog ILDS
   rebuilding 58
IMS catalog indexes
   rebuilding 58
IMS DD statement 70
IMS Index Builder
   abend codes 163
   support 18
IMS managed ACBs 63
IMS reorganization reload utility
   output 66
IMS reorganization utility 40, 42
IMS Tools Knowledge Base 31
IMS Tools Knowledge Base interface 82
IMS Tools Knowledge Base support 18
IMS Tools Online System Interface 61
IMS utilities 18
IMSDLT DD statement 40, 70, 98
IMSDLIB DD statement 70
input control statement (continued)
   building phase failure reason codes 130
   INDEX input control statement 81
   index records with duplicate keys 111
   initialization phase failure reason codes 124, 128
   initializing empty secondary index 50
   input
   DFSURWF1 37
   IBSCAN 40
input control statement
   ITKBLOAD 82
input control statements 65, 72
   CLASS 77
   COMPAUTH 77
   DBAUTH 78
   DFSDF 79
   IIUSORT 80
   ILDS 80
   INDEX 81
   INPUT 75
   ITKBLOAD 82
   ITKBRSRV 82
   JOBCS 91
   JOBCS input control statement 86
   JOIN 82
   JOINSCOR 83
   JOBSCOR 83
   MAXTASKS 83
   OUTPUT 84
   PARTNAME 85
   PROC 73, 98
   PROGMON 85
   SIDXBUF 86
input control statements (continued)
   SORTFSZ 87
   SORTOPT 87
   STRIPE 88
summary 98
SYCDUMP 89
TMWAI 89
TODBD 90
TOSDBD 91
TOSISTA 92
TOSIWAIT 93
TOSIXCFGRP 93
VIC 94
WFPDEL 94
WFPHLQ 95
ZIIPMODE 95
INPUT input control statement 75
INPUT statement keyword
   DBRC 75
   DFSURIDX 75
   DFSURWF1 75
   HPSRIDX 75
   IBSCAN 75
input type
   specifying 75
installation
   activating 28
   verifying 28
integration 18
inter-region communication 27
issuing IMS commands automatically 61
ITKB 31
ITKB support 18
ITKBRSRV input control statement 82

J
JCL 65, 66
   DD statements 66
      data set group 71
      DDFPDS 67
      DFSACBP 67
      DFSCTL 67
      DFSARM 67
      DFSURIDX 67, 98
      DFSURWF1 68, 98
      DFSUSAMP 68
      EXEC 66
      HPSRIDX 68
      HSSROPT 68
      IIUCABP 68
      IIUCAP 69
      IIUDUMP 69
      IIUIN 69
      IIUOPT 69
      IIUPRT 69
      IIUSORT 80
      IIUSNAP 69
      IIUSOUT 70
      IIUWFPPn 70
      IMS 70
      IMSDLT 70, 98
      IMSDLIB 70
      OUTPUT 70, 98
      PROCLIB 71
      RECON 71
   secondary index 71
   STEPLIB 66

I
I/O buffer subpools 68
IBM Software Support
   diagnostic documentation 115
IBSCAN 40, 75
   building HIDAM primary index 52
   building secondary index 38
   creating new secondary index 40
IIU.STRIPER 99
   defining in RACF 99
IIUBLDFR
   listing 54
IIUBLDSN, listing 40
IIUBR 99
IIUCAP DD 118
IIUCAP DD statement 69
IIUDKES 111
IIUDUMP DD statement 69
PARTNAME input control statement 85
performance benefits 9
PHIDAM partition primary index rebuilding 56
prefix resolution
input to build index 45
output from 67
split function 43
primary index
building 52
PHIDAM 52
PROC control statement 98
PROC input control statement 73
PROC statement keywords
ALL 73
BLD_ALL 73
BLD_PRIMARY 73
BLD_SECONDARY 73
SELECTED 73
process events output file 118
processing phase reason codes 133
processing type 73
PROCCTRL 99
PROCCTRL DD statement 71
product features 9
product overview 3
PROCCTRL input control statement 85
programming interface information 177
progress log creation
TRACE 72
PSIs generic 70
PSINDEX 75
requirements (continued)
hardware 26
inter-region communication 27
operating environment 27
region size 27
software 26
started task procedures 29
restrictions 19
return codes 122
runtime parameters
defining SORT procedure name 103
runtime parameters module 103

S
scan method 59
scan physical database 75
scanning the database rebuilding with alternate primary index data set 54
scenarios 17
screen readers and magnifiers 23
secondary index building
DFSURIDX 45
HPSRWFF 48
IBSCAN 38
split function 43
using DFSURWFF 42
using HPSRWFF 42
creating new
IBSCAN 40
tasks 40
example of adding 40
initializing 50
rebuilding 51
recovering 51
reorganizing 51
secondary index DD statements 71
sequential scan technology 59
service information 20
shared secondary index 19
SIDXXBUF control statement 86
snapshots of error records 66
software requirements 26
SORT output file 118
SORT parameters
FILSZ 87
MAINSIZE 87
SORT procedure
passing options to
SORTFSZ 87
sort work file requirements 27
SORTFSZ control statement 87
SORTOPT control statement 87
specify processing type 73
specify subordinate address space 83
split function 37, 43, 70, 84
STEPCTRL DD statement 66
storage requirements 27
STRIPE control statement 88
subordinate address space 87
specify amount 87
subordinate address space procedure
IIUBSRT 99

K
knowledge-base support 18

L
legal notices
cookie policy 177, 180
notices 177
programming interface information 177
trademarks 177, 179
listings
IIUBLDPR 54
IIUBLDSN 40

M
main output file 69
main output file (SYSPRINT) 118
MAXTASKS input control statement 83
messages 145
methods for accessing 161
migration from version 2.3 to 3.1 33
MVS log 29

N
notices 177

O
operating environment requirements 27
APF authorization
inter-region communication 27
started task procedures 29
OSAM files 66
output
DFSURIDX 37
OUTPUT control statement 70
OUTPUT DD statement 70, 98
output file 117
main 69
output files
main output file (SYSPRINT) 118
process events file (IIUSNAP) 118
SORT output file 118
OUTPUT input control statement 84
overview 3

P
parallel scan 99
partitioned secondary index 75

R
RACF authorization 99
reader comment form 21
reason codes 121
index build phase failure 130
initialization phase failure 124, 128
processing phase 133
verification phase 144
rebuilding 37
HALDB partition ILDS 57
HIDAM on an alternate primary index 54
IMS catalog indexes 58
PHIDAM partition primary index 56
RECONn DD statement 71
recovering
secondary index 51
reference
abend codes 163
control flow events 117
error messages 145
IBM Software Support
diagnostic documentation 115
messages 145
output file 117
reason codes 121
region size requirements 27
reload utility 42
reorganization reload utility output 66
reorganization utility 40, 42
requirements
APF authorization 27
configuration 27
subpools
 allocation of 68
summary of changes 4
support information 20
SVCDUMP control statement 89
syntax
 control statements 72
syntax diagrams
  how to read 173
SYSPRINT output file 118

T
 technotes 21
terminology 11
TMRWAIT control statement 89
TOSIDBD control statement 90
TOSIDBR control statement 91
TOSISTA control statement 92
TOSIWAIT control statement 93
TOSIXCFGRP control statement 93
TRACE dx 72
TRACE mode 72
trademarks 177, 179

U
 unsupported control statements 97
utilities
  compatibility 18
  IMS reorganization 40
  prefix resolution and split
  function 43
UUICABP DD statement 68

V
 verification phase reason codes 144
verifying installation 28
version 2.3 to 3.1 migration 33
VIC control statement 94
VSAM files 66

W
 WFPDEL control statement 94
WFPHELQ control statement 95
work class file
  defining in RACF 99

Z
 ZIIPMODE control statement 95