IBM IMS Fast Path Solution Pack for z/OS
2.1

IMS Fast Path Basic Tools User's Guide

IBM
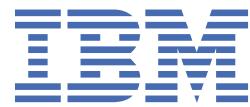
Note:

Before using this information and the product it supports, read the information in "Notices" on page 349.

Contents

About this information

IBM IMS Fast Path Solution Pack for z/OS® IMS High Performance Fast Path Utilities (also referred to as IMS HP Fast Path Utilities) improves performance and availability by streamlining database administrator (DBA) tasks.

To use the procedures in this information, you must first install IMS HP Fast Path Utilities as described in the *Program Directory for IMS Fast Path Solution Pack for z/OS 2.1, GI13-5905*, and then perform the post-installation steps as described in the *IMS Fast Path Solution Pack: Overview and Customization, GC27-9596*.

These topics are designed for database administrators and technical support personnel who are involved in database management, maintenance, and performance tuning, and require a knowledge of how to operate IMS Fast Path Basic Tools of IMS HP Fast Path Utilities, and are specifically for those who manage the IMS Data Entry Databases (DEDBs). These topics help database administrators and technical support personnel to perform these tasks:

- Understand the functions of IMS HP Fast Path Utilities IMS Fast Path Basic Tools
- Run and use IMS HP Fast Path Utilities IMS Fast Path Basic Tools after it is installed
- Use DD statements to control how you use IMS HP Fast Path Utilities IMS Fast Path Basic Tools

IMS HP Fast Path Utilities includes all the features you need to manage your IMS Fast Path databases.

For information about other utilities and tools of IMS Fast Path Solution Pack IMS HP Fast Path Utilities, see the following information:

- *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*
- *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*

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:

https://www.ibm.com/support/pages/node/712955

The IMS Tools Product Documentation page includes:

- Links to IBM Documentation for the user guides ("HTML")
- Links to the PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Technical articles from IBM Software Support
- White papers that describe product business scenarios and solutions

Chapter 1. Overview of IMS Fast Path Basic Tools

IBM IMS Fast Path Solution Pack for z/OS IMS High Performance Fast Path Utilities IMS Fast Path Basic Tools (also referred to as IMS Fast Path Basic Tools or FPB) can help you analyze, maintain, tune and migrate DEDBs without bringing up an IMS online system environment. These powerful tools, which offer numerous utilities and functions, are DEDB Pointer Checker, DEDB Tuning Aid, and DEDB Unload/Reload.

DEDB Pointer Checker
> Provides a fast and efficient mechanism for verifying the integrity of all IMS pointer values, free space element chains, VSAM control fields, and space management in IMS DEDBs. DEDB Pointer Checker provides the comprehensive analysis information that is required to effectively manage the space utilization, performance characteristics, and physical attributes of an IMS DEDBs.

DEDB Tuning Aid
> An extension of DEDB Pointer Checker. This feature enables you to evaluate the DEDB definition in reorganization, or to reduce the impact of changes to an existing DEDB, without consuming time to unload and reload the database.

DEDB Unload and Reload
> A set of high-performance utilities for efficient Unload and Reload process, or to reorganize IMS DEDBs.

IMS High Performance Fast Path Utilities provides many other tools and functions. For an overview of other IMS High Performance Fast Path Utilities tools and functions, see the topic "IMS HP Fast Path Utilities features and benefits" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Topics:

- "What's new in IMS Fast Path Basic Tools" on page 1
- "Features and benefits of FPB" on page 2
- "IMS Fast Path Basic Tools compatibility" on page 3
- "Service updates and support information" on page 5
- "Product documentation and updates" on page 5
- "Accessibility features" on page 6

What's new in IMS Fast Path Basic Tools

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.

Revision markers follow these general conventions:

- Only technical changes are marked; style and grammatical changes are not marked.

- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.

- If a topic is changed by more than 50%, the entire topic is marked with revision markers (so it might seem to be a new topic, even though it is not).

Revision markers do not necessarily indicate all the changes made to the information because deleted text and graphics cannot be marked with revision markers.

SC27-9597-02 - February 2025

Description	Related APARs
Updated the description of JOBLIB/STEPLIB DD in the following topics: • "FABADA1 JCL" on page 21 • "DD statements for DEDB Unload" on page 123 • "FABCUR3 JCL" on page 165	N/A

SC27-9597-01 - July 2023

Description	Related APARs
Minor documentation updates made to Chapter 8, "Reference: Return code user exit routine (FABARCEX)," on page 195.	N/A

Features and benefits of FPB

IMS Fast Path Basic Tools consists of the following tools; DEDB Pointer Checker, DEDB Tuning Aid, DEDB Unload, and DEDB Reload.

DEDB Pointer Checker is a tool for database administrators and technical support personnel. It provides complete database integrity verification, and summary and detail analysis reports for those directly involved in database management, maintenance, and performance tuning.

When used in an appropriate manner, on a regular basis, DEDB Pointer Checker can help you to:

- Verify the complete integrity of all IMS pointer chains, FSE chains, VSAM control fields, and space management
- Improve performance and transaction throughput
- Determine the optimal time for database reorganization
- Determine the optimal physical attributes for the database (that is, optimal values for the ROOT and UOW parameters on the AREA statement in your DBD)
- Eliminate application failures caused by out-of-space conditions
- Increase productivity of database support personnel
- Make more efficient use of IMS regions and buffer pools
- Evaluate the effectiveness of the randomizing module
- Reduce maintenance frequency and the associated costs (for example, logging activity or log record volumes)
- Reduce DASD requirements

DEDB Tuning Aid is a comprehensive tuning utility for DBAs and technical support staff who are directly involved in database management, maintenance and performance management. DEDB Tuning Aid tests necessary database changes before beginning the time-consuming unload/reload process and shares a common report generation module to help simplify tuning evaluation and comparison. None of these functions influences or impacts IMS system performance or database availability.

DEDB Unload and DEDB Reload are tools for database administrators and technical support personnel involved in database management, maintenance, and performance tuning. The facilities provided by DEDB Unload and DEDB Reload can:

- Increase application availability
- Significantly reduce the time and resources required for database maintenance
- Significantly reduce the time and resources required for offline sequential reporting runs
- Increase the productivity of database support personnel

- Significantly reduce maintenance and database conversion costs.

Support for IMS managed ACBs environment

In an IMS managed ACBs environment, IMS can manage the runtime application control blocks (ACBs) for databases and program views for you. When IMS manages ACBs, IMS no longer requires DBD, PSB, and ACB libraries.

The following tools of FPB support IMS managed ACBs environment.

- DEDB Pointer Checker
- DEDB Unload
- DEDB Reload

For more information, see the following topics:

- Topic "IMS Tools Catalog Interface" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*
- For DEDB Pointer Checker, IMSCATHLQ and IMSCATACB_INPUT keywords in "FABADA1 SYSIN DD data set" on page 39
- For DEDB Unload, IMSCATHLQ, IMSCATACB_INPUT, and IMSCATACB_OUTPUT keywords in "DEDB Unload SYSIN DD data set control statements" on page 130
- For DEDB Reload, IMSCATHLQ and IMSCATACB_INPUT keywords in "DEDB Reload SYSIN DD data set control statements" on page 172

Support for DEDB encryption

All the tools of FPB support encrypted DEDB area data sets (ADSs).

To enable the tools to process DEDB ADSs that are encrypted with z/OS data set encryption, APAR PI83756 must be applied to IMS 15.

You do not need to change the JCL streams to process encrypted DEDB ADSs.

IMS Fast Path Basic Tools compatibility

The following topics describe the compatibilities of the tools within IMS HP Fast Path Utilities.

Compatibilities between IMS Fast Path Basic Tools and former releases of IMS Fast Path Basic Tools

There are considerations that apply when you migrate from IMS DBT 2.3 or FPB 1.x to this release of IMS Fast Path Basic Tools (FPB).

The JCL streams and the input control statements that are used in the former releases of FPB are compatible with this release of FPB.

Compatibilities for DEDB Unload and Reload utilities

- These items are compatible with later versions:
  - JCL in which DCB LRECL parameter for output unloaded data set is not specified explicitly
  - control statements
  - DURDBDFN data set
  - DEDB unloaded data set
- A user program of IMS DBT 2.3 that includes FABCRMIF must be re-link edited with the FABCRMIF module that is provided with FPB of IMS HP Fast Path Utilities.

DD statements that specify LRECL=121 explicitly are supported because FPB of IMS High Performance Fast Path Utilities converts them to LRECL=133 internally.

Macros intended for customer use

FPB provides macros that can be used by customers. This section summarizes the relationship of macro names between FPB and IMS DBT 2.x.

Macro names that are provided in FPB are the same as those provided in FPB 1.x. Macro names that are provided in IMS DBT 2.x are also valid in FPB. The following tables show the relationship between FPB macro names and IMS DBT 2.x macro names.

DEDB Pointer Checker and DEDB Tuning Aid provide only the product-sensitive macros listed in the following table.

Table 1. Relationship of macro names between FPB and IMS DBT 2.x (Pointer Checker and Tuning Aid)

FPB	IMS DBT 2.x	Description
FABAMDAR	DA#MDAR0	DSECT generation macro for DEDB analyzer records.
FABAMHST	DA#MHST0	DSECT generation macro for history data set record.
FABAMHS2	N/A	DSECT generation macro for history2 data set record.
FABAMDDT	UT#MDDT0	DSECT generation macro for DDT, ADT, and SDT tables.
FABAMGDD	N/A	DSECT generation macro for "Get DEDB DMB" subroutine interface internal control block.
FABALREC	N/A	DSECT generation macro for LARGEREC data set record.

The DEDB Unload and Reload utilities provide only the product-sensitive macros listed in the following table.

Table 2. Relationship of macro names between FPB and IMS DBT 2.x (DEDB Unload and Reload)

FPB	IMS DBT 2.x	Description
FABCMUSR	UR#MUSR0	DSECT generation macro for fast DEDB Unload/Reload.
FABCMDDT	UT#MDDT0	DSECT generation macro for DDT, ADT, and SDT tables.
FABCMGDD	N/A	DSECT generation macro for "GET DEDB DMB" subroutine interface internal control block.
FABC1TAL	N/A	DSECT generation macro for the UR1TOTAL data set record.

Compatibility between IMS Fast Path Basic Tools and IMS Fast Path Advanced Tool

There are compatibilities and incompatibilities between the IMS Fast Path Basic Tools (FPB) utilities and the IMS Fast Path Advanced Tool (FPA) functions.

Product material
  The following items are compatible:

  - The unloaded segment records data set that is created by the FPA Unload function and the FPB DEDB Unload utility. You can use the unloaded segment records data set that is created by FPB DEDB Unload utility for FPA Reload function.
  - The total segments records file that is created by the FPA Unload, Reload, Change functions, and the FPB DEDB Unload utility.
  - The History file records that are created by the FPA Analyze function and the FPB DEDB Pointer Checker utility.

The following items are not compatible:

- The RBA values for segment allocation differ due to design difference when the LOADCTL subcommand is specified for the FPA Reload function and when the LOADCTL control statement is specified for the FPB Unload utility.

- The History2 file records that are created by the FPA Analyze function and the FPB DEDB Pointer Checker utility. The logic to round off the values has changed. You might see a slight difference in some fields.

- The tuning aid record data sets, which are used by the FPA Tune function, are not compatible with the FPB DEDB Tuning Aid utility.

DD statements
DD statements are not compatible.

Utility control statements
The utility control statements, keywords, and parameters are not compatible.

Size of intermediate storage data sets
The size of the following intermediate storage data sets that are used by DFSORT are compatible:

- In FPA, the size of *ittt*WK*nn* DD data set that is used when PTRCHKLVL=FULL is specified, is similar to the size of SORTWK*nn* DD data set that is used in the FPB DEDBPC SORT3CS step.

- In FPA, the size of *ittt*WK*nn* DD data set that is used when CISNAP=YES is specified, is similar to the size of SORTWK*nn* DD data set that is used in the FPB DEDBPC SORT4 step.

- In FPA, the size of *ittt*WK*nn* DD data set that is used when the REPORT subcommand is specified, is similar to the total size of SORTWK*nn* DD data sets that are used in the FPB DEDBPC SORT12 step and in the FPB DEDBPC SORTR13 step.

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:

IBM Support: IMS Fast Path Solution Pack for z/OS

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:

https://www.ibm.com/support/pages/node/712955

The IMS Tools Product Documentation web page includes:

- Links to IBM Documentation for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- Technical articles from IBM Software Support
- 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 emails that notify you when new technical articles 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 https://www.ibm.com/support/mynotifications
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 us provide the most accurate and highest quality information. If you have any comments about this information, see How to provide feedback in IBM Documentation.

When you provide feedback, include as much information as you can about the content you are commenting on, where we can find it, and what your suggestions for improvement might be.

Prerequisite knowledge and publications

Before using this information, you should understand basic IMS concepts, the IMS environment, and your installation's IMS system.

The IMS publications are prerequisite for all IMS HP Fast Path Utilities components.

Related publications

This information describes IMS Fast Path Basic Tools. For information about other utilities and tools of IMS Fast Path Solution Pack IMS HP Fast Path Utilities, see the following information:

- *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide* SC27-9536
- *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* SC27-9598

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.

Accessibility features

The major accessibility feature in IMS HP Fast Path Utilities is the keyboard-only operation for ISPF editors. It uses the standard TSO/ISPF interface.

Keyboard navigation

You can access the information center and IMS ISPF panel functions by using a keyboard or keyboard shortcut keys.

For information about navigating the IMS ISPF panels using TSO/E or ISPF, refer to the following publications

- *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.

IBM and accessibility

See the IBM Human Ability and Accessibility Center at www.ibm.com/able for more information about the commitment that IBM has to accessibility.

Chapter 2. Analyzing a DEDB offline with FPB

You can analyze a DEDB by using the DEDB Pointer Checker utility of FPB.

The DEDB was introduced as part of the Fast Path feature of IMS. It is the preferred access method for databases in high performance, random processing environments. The use of DEDBs, even in MPPs, can significantly increase transaction throughput and reduce response time. Besides having the potential for the most random performance, poorly tuned DEDBs have the potential to be inefficient because excessive physical I/O can occur when accessing the database. Also, integrity verification and space/performance management utilities for DEDBs were not provided with IMS.

FPB DEDB Pointer Checker provides a fast, efficient mechanism for verifying the integrity of all IMS pointer values, free space element chains, VSAM control fields, and space utilization in IMS DEDBs. DEDB Pointer Checker provides the comprehensive analysis information required to effectively manage the space utilization, performance characteristics, and physical attributes of IMS DEDBs. DEDB Pointer Checker uses as input either an image copy data set or the VSAM cluster. DEDB Pointer Checker can be an aid for database administrators and technical support personnel in database management, maintenance, and performance tuning.

The DEDB Pointer Checker product requires *no* source code modifications to any user or IMS module.

The three Pointer Checkers of IMS HP Fast Path Utilities

There are three distinct pointer checker products. Therefore, before getting started, it is important to read this topic in order to avoid confusion later on.

This topic describes the relationship among the following three Pointer Checkers:

- Analyze function of FPA
- DEDB Pointer Checker of FPB
- Online Pointer Checker of FPO

The Analyze function of FPA and DEDB Pointer Checker of FPB run in the offline environment and Online Pointer Checker of FPO runs in the online environment.

The Analyze function of FPA is part of the IMS HP Fast Path Utilities package. It has the function similar to that of DEDB Pointer Checker, which runs while the database area is offline. FPA verifies the integrity of all IMS pointer values and verifies the integrity of the pointer segments in secondary index databases. It also produces complete analysis reports in one step, whereas DEDB Pointer Checker runs the same functions in multiple steps.

DEDB Pointer Checker of FPB is also part of the IMS HP Fast Path Utilities package. It uses program FABADA1 to collect pointer verification data. This program must be run while the database area is offline. The collected data can then be passed to the pointer integrity checking programs and various report generating programs in this package.

Online Pointer Checker of FPO is also part of the IMS HP Fast Path Utilities package. It runs as a Fast Path utility to verify the integrity of all IMS pointer values and produce complete analysis reports in one step while the database area is online. However, you can use OPC to run a fast scan of database integrity and collect pointer verification data from an online DEDB area, then pass it to DEDB Pointer Checker of FPB to check cross-reference of pointers and produce complete analysis reports. This type reduces the online run time but must be run in multiple steps.

Topics:

Functions of DEDB Pointer Checker

The functions of DEDB Pointer Checker provide complete database integrity verification, and summary and detail analysis reports for those directly involved in database management, maintenance, and performance tuning.

Some of the highlights of DEDB Pointer Checker are as follows:

- Provides complete verification of the integrity of all IMS and VSAM control information.
- Provides a special operating mode to detect pointer integrity errors in a fast and efficient manner.
- Provides an option with which you can produce segment information records with expanded segment data from compressed segments.
- Uses as input either an image copy of the database or the VSAM cluster.
- May specify by external sources that the EXCP count for a database has increased. If the increase is not volume-related, DEDB Pointer Checker provides the information required to determine the cause of the problem.
- Produces a comprehensive set of reports, which are organized by DEDB area, to give multiple views of space utilization, record and segment profiles and placement, and physical I/O requirements.
- Produces the optional CI Map/CI Dump report and the Pointer Chain Reconstruction report as a result of processing, which enables you to analyze the CIs containing errors and to determine the best way to repair them without performing another run to obtain such information.
- Provides optional automated threshold reporting, which causes warning messages to be generated, and a special return code to be set when conditions within a database do not meet user-specified criteria. This feature can be used to show when reorganization, expansion, or analysis is required, or when database performance falls below acceptable levels.
- Provides optional generation of a report identifying the UOWs that exceed user-specified reorganization criteria. Identifying and reorganizing specific UOWs can significantly reduce the time and resources required for database maintenance.
- Provides optional generation of a History file containing the key space utilization and performance values for each area. This file lets you carry out statistical and trend analysis with the use of user-written programs.
- Provides optional generation of a History2 file that contains the history data that is totaled by UOW range. This file enables you to do statistical and trend analysis with the use of user-written programs.
- Does not affect IMS system performance or availability when DEDB Pointer Checker is run using image copy input.
- Allows no source code modifications to any user-written or IMS program or control block. Because IMS program integrity is maintained, new exposures are not introduced.

Related reference

FABADA1 SYSIN DD data set

The SYSIN DD data set contains the FPB user's description of the processing to be done by module FABADA1. It describes the database and area to be analyzed, and it contains optional user's requests that further define the operating mode.

DEDB integrity verification

DEDB Pointer Checker provides complete DEDB integrity verification.

Specifically, the following areas of integrity are addressed:

- All the following pointer values are verified to ensure that they refer to valid segments:
  - Root Anchor Point (RAP)
  - Physical Twin Forward (PTF)
  - Physical Child First (PCF)
  - Physical Child Last (PCL)
  - Subset (SSP)
  - Sequential Dependent (SDEP)

  All orphan segments, invalid pointer values, and broken pointer chains are detected and reported. PCF/PCL/SSP pointer interdependency errors are also detected and reported.
- All free space element (FSE) chains are checked for validity.
- Space utilization is verified by totalizing the lengths of all segments, FSEs, and scraps within a CI, and ensuring that the calculated total equals the usable CI size (CI size minus VSAM and IMS overhead). This process will detect FSEs within segments, or segments within FSEs.
- The VSAM CIDF and RDF fields are also verified to ensure that their values are correct. DEDB Pointer Checker will report the RBA of any invalid CIs encountered and continue processing.

  DEDB Pointer Checker provides the following reports that can be used to determine the best way to repair invalid CIs:
  - CI Map/CI Dump report for invalid CIs
  - Pointer Chain Reconstruction report

DEDB Pointer Checker (program FABADA1) reads and serially deblocks each database CI. Serially deblocking a CI is carried out by starting at offset +8 and advancing through the CI based on the lengths of the items encountered. Valid items are segments, scraps, and FSEs. During the deblocking process, DEDB Pointer Checker accumulates the number and lengths of all segments, scraps, and FSEs. At end-of-CI, DEDB Pointer Checker chases the FSE chain and tallies the number and length of all FSEs. DEDB Pointer Checker then ensures that the number and length of FSEs encountered during the serial deblocking process matches the number found by following the FSE chain. This test finds segments within FSEs and FSEs within segments. DEDB Pointer Checker verifies CI integrity by ensuring that the sum of all segment, FSE, and scrap-lengths plus IMS and VSAM overhead equals CI size.

If the input is the area data set (VSAM ESDS), DEDB Pointer Checker also verifies that the VSAM control fields (CIDF, RDF) are correct. This provides a rapid way of identifying the exact location of corrupted values when the Image Copy utility abnormally terminates because of incorrect VSAM control fields.

Quick mode

TYPRUN = PTRSCAN provides a fast scan of database integrity. This mode is a single job step that creates no output data sets.

During execution in this mode, DEDB Pointer Checker can detect and report the exact RBA of the following integrity exposures:

- Invalid block type (DBLKBTID).
- Invalid segment codes (that is, greater than the maximum defined segment code and not an FSE or scrap).

- Invalid FSE in the FSE chain (that is, not X'80').
- Number or length of FSEs found during serial deblocking does not equal that tallied by chasing the FSE chain.
- CI space utilization discrepancies (that is, the total length of all segments, FSEs, and scraps plus IMS and VSAM overhead does not equal CI size).
- The following pointer interdependency problems:
  - PCF pointer nonzero and PCL pointer zero.
  - PCF pointer zero and PCL or Subset pointer nonzero.
- SDEP pointers that contain a pointer value that is outside the sequential dependent part of the area.
- Logical end (LE) in the DMAC is less than the logical beginning (LB).
- Invalid VSAM control fields (CIDF, RDF) when the input data set is the area ESDS.

DEDB Pointer Checker also does a checksum verification of the following pointers versus the appropriate segment RBAs:

- PCF/PTF versus appropriate segment RBAs
- PCL versus appropriate segment RBAs
- SDEP pointer values and SDEP segments (between LB and LE).

The checksum verification process can detect the presence of invalid pointers but cannot report the exact RBA of the problem. Checksum verification cannot be done for subset pointers.

Full mode

TYPRUN = PTRALL provides in-depth pointer verification. Besides the PTRSCAN integrity verification functions, DEDB Pointer Checker writes enough pointer and segment information to carry out in-depth pointer analysis through a matching process.

This process reports the exact RBAs of the following pointer problems:

- Orphan segments (that is, segments that are not pointed to by any valid pointer).
- Pointers that reference nonexistent segments.
- Pointers to wrong segment types (that is, DEDB Pointer Checker verifies that the pointer points to the correct segment type).
- Segments pointed to by multiple parents.
- Segments that cannot be "resolved" back to a valid RAP RBA.
- Subset pointer validation can only be done in this mode.

Related reference
FABADA1 SYSIN DD data set
The SYSIN DD data set contains the FPB user's description of the processing to be done by module FABADA1. It describes the database and area to be analyzed, and it contains optional user's requests that further define the operating mode.

DEDB analysis and report

The complete analysis reports are provided to effectively manage the space utilization, performance characteristics, and physical attributes of IMS DEDBs. It provides integrity verification reports, and summary and detailed analysis reports for database administrators and technical support personnel in database management, maintenance, and performance tuning.

The statistical information includes:

- Free space analysis, including statistics on the number and size of the free space elements and inter-database section references.
- Database record and segment occurrence profiles.

- Database record and segment placement analysis.
- Complete overflow usage analysis of both dependent overflow (DOVF) and independent overflow (IOVF).
- Physical I/O statistics for each segment type and for the typical database record.
- Root distribution and synonym chain analysis.
- Optional automated threshold reporting, which generates warning messages when conditions within the database do not meet user-specified criteria. This feature can be used to show when reorganization or expansion is required, or if database performance falls below acceptable standards.
- Optional generation of:
  - A report that describes the largest database records.
  - A report that identifies the units of work (UOWs) that exceed a user-specified reorganization criteria.
  - The control statements that are required to run the IMS High-Speed DEDB Direct Reorganization utility (DBFUMDR0).

Reorganizing specific UOWs can significantly reduce the time and resources that are required for database maintenance.

Restrictions of DEDB Pointer Checker

There are some restrictions you must follow to use DEDB Pointer Checker.

- DEDB Pointer Checker should not be used against an actual DEDB area that will be updated concurrently by the IMS online system. In such a case, pointer errors may occur during the DEDB Pointer Checker processing.
- DEDB Pointer Checker should not be used against a Concurrent Image Copy (CIC) obtained from an area that could be updated by the IMS online system. In such a case, pointer errors may occur during the DEDB Pointer Checker processing.
- When the parent ROOT segment is deleted, DEDB Pointer Checker views its child SDEP segment as an orphan. It is recommended that you delete all reported orphan SDEP segments using the IMS DEDB Sequential Dependent Delete Utility.
- DEDB Pointer Checker (FABADA1) does not support Automatic RECON Loss Notification.
- If you concatenate image copy data sets, DEDB Pointer Checker can process multiple areas. However, if the CI size is different among the concatenated areas, DEDB Pointer Checker cannot process multiple areas.

Data and system flow of DEDB Pointer Checker

This topic describes the data and system flow of the DEDB Pointer Checker.

The following figure shows the general data flow for the DEDB Pointer Checker utility. Input consists of the DEDB area, the SYSIN data set, and the location of ACB definitions.

ACB definitions are retrieved from the ACB library (ACBLIB DD) or, if MODSTAT and MODSTAT2 DD statements are specified, from the libraries specified by IMSACBA and IMSACBB DD statements. However, if the IMS catalog and the IMS management of ACBs are enabled, ACB definitions can be retrieved from the IMS directory.

Output consists of the validation error messages, the analysis report, and other analysis reports.
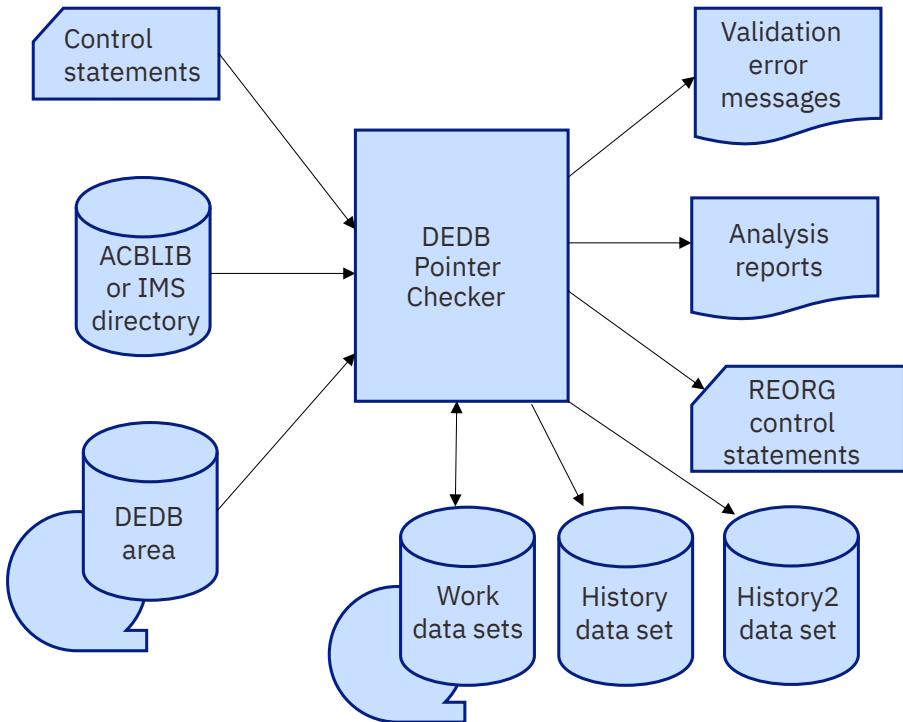
Figure 1. Flow of DEDB Pointer Checker

Modules FABADA1, FABADA3, FABADA4, and FABADA5 generate detailed analysis reports for the root addressable part (RAA BASE and DOVF), the independent overflow part (IOVF), and the sequential dependent part (SDEP) of a DEDB. They do not produce a report for the SDEP space utilization data.

Data for all detailed analysis reports is extracted from ACBLIB and either the current image copy data set or directly from the VSAM cluster. Reports for all or selected areas of a DEDB can be generated by concatenating the appropriate image copy data sets.

Since all data is extracted and processed by a series of z/OS batch programs, using an image copy of the database in running DEDB Pointer Checker does not affect IMS system performance. Using the VSAM cluster as input provides a fast, convenient mechanism for resolving problems.

Subsections:

- "Processing flow of DEDB Pointer Checker" on page 14
- "Load modules of DEDB Pointer Checker" on page 17

Processing flow of DEDB Pointer Checker

DEDB Pointer Checker programs run in a virtual storage region as z/OS job steps. Region size varies with the size of a UOW and the number of segments and free space elements in a UOW. FABADA1 and FABADA3 run in 31-bit addressing mode.

The following figure shows the flow of the DEDB Pointer Checker procedures.
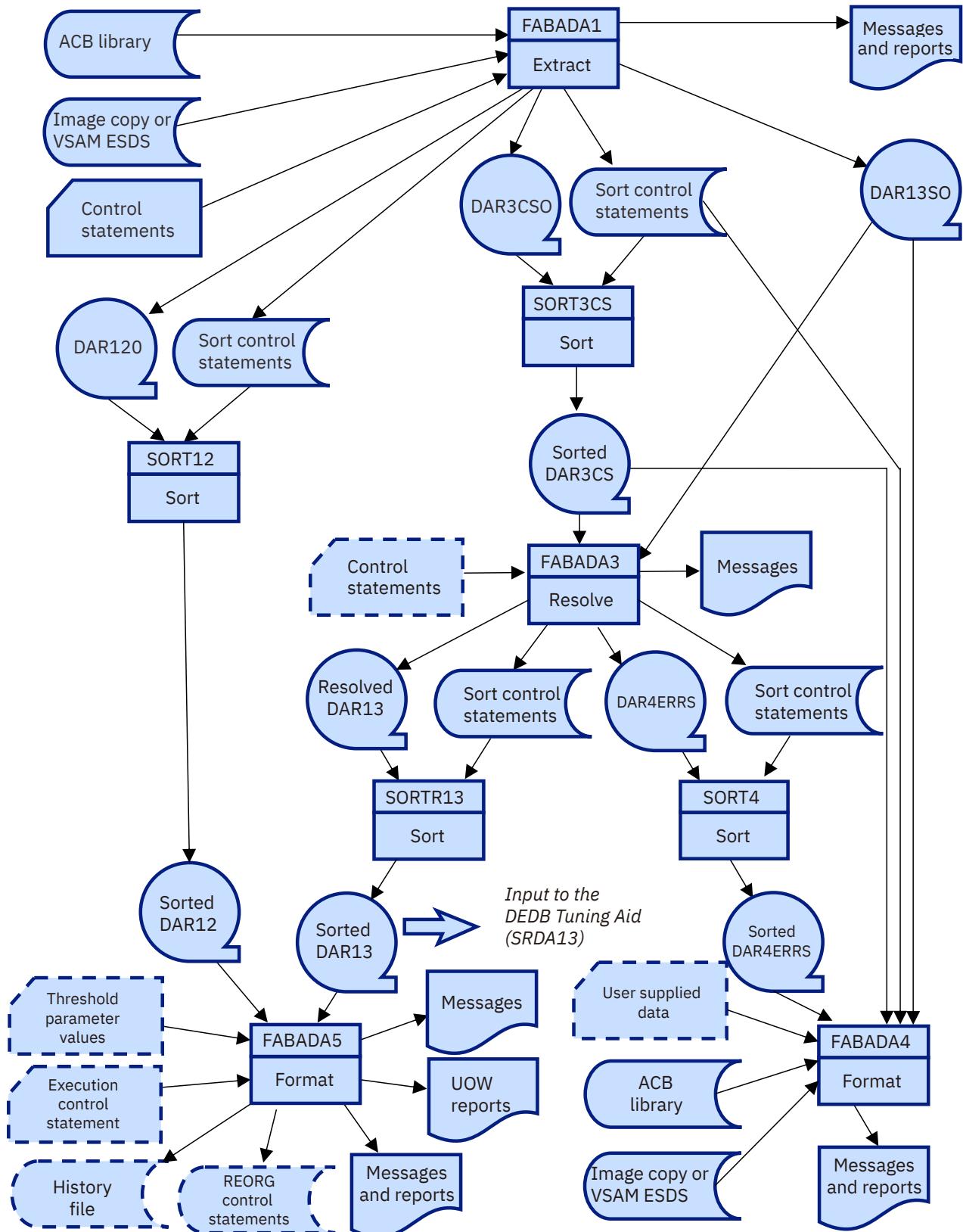
Figure 2. Processing flow of pointer checking for DEDB area data set

The root addressable and IOVF processing steps include:

FABADA1

This FPB program strips all required data from the database DMB (in ACBLIB) and from either the image copy data sets, or directly from the VSAM cluster.

Basically, FABADA1 deblocks each CI into its component parts (that is, segments, scraps, and FSEs). It does this by starting at offset 8 and continue until the end of CI, by incrementing the offset by the length of the item. During the deblocking process, the total lengths of all FSEs, scraps, and segments are accumulated. The number of FSEs found is also totaled.

After a CI is deblocked, the program chases the FSE chain and totals the number and length of FSEs. If this value differs from the value calculated during the deblocking process, a DEDB integrity problem has been detected. This test finds FSEs within segments, or segments within FSEs. FABADA1 also verifies that total segment lengths, plus total scrap and FSE lengths, plus VSAM and IMS overhead, equals CI size.

Three output files are created:

1. Area and free space information (used by FABADA4 and FABADA5)
2. Segment information for all segments that reside in the Root Addressable part of an area (used by FABADA3 and FABADA4)
3. Segment information for segments in IOVF and SDEP, and pointer information (used by FABADA3 and FABADA4).

The record types generated are as follows:

Type
   Purpose

1,0
   Contains area definition information from the DMB (one record per area)

1,1
   Contains segment definition information from the DMB (four segments per record)

2,T
   Contains total free space in each of the component parts of the area (one record per area)

2,R
   Contains a count of the number of references to other CIs in the area (one per CI that has external references)

2,F
   Contains the length and location of each FSE or scrap encountered (one per FSE, no record generated for empty DOVF or IOVF CIs)

3,C
   Contains the RBA of all nonzero pointers: subtype 0 = RAP pointer, 1 = PTF pointer, 2 = PCF pointer, 3 = PCL pointer, 4 = SSP pointer (one per nonzero pointer)

3,S
   Contains the RBA and length of all segments (one per segment)

4,P
   Contains the RBA of all nonzero SDEP pointers (one per nonzero SDEP pointer)

4,S
   Contains the RBA of all SDEP segments (one per SDEP segment).

FABADA1 terminates if integrity errors or invalid VSAM CIDF and RDF fields are encountered in more than 10 CIs for the area being analyzed.

DFSORT

IBM Data Facility Sort sorts the DAR3CSO data set that is created by FABADA1. Include this step only if FABADA3 is run.

FABADA3

This FPB program uses the segment and pointer information generated by FABADA1 (that is, record types 3,C, 3,S, 4,P and 4,S) to resolve all pointer references and build the records required for report generation. IMS pointer discrepancies are detected and reported by this program.

Basically, FABADA3 builds an in-storage table of pointer RBAs for each UOW (using the type 3,C records). It matches the segment records (type 3,S) against this table by a balance-line process to resolve the pointer values and to determine the appropriate segment sequence number. A table entry with no corresponding segment record, or a segment record with no corresponding table entry, signifies a pointer problem.

Optionally FABADA3 creates DAR4ERRS data set that will be used by FABADA4 to generate CI Map/CI Dump report and Pointer Chain Reconstruction report.

DFSORT

The IBM Data Facility Sort sorts the DAR4ERRS data set that is created by FABADA3. Include this step if FABADA3 creates DAR4ERRS data set.

FABADA4

This FPB program uses the pointer error information generated by FABADA3 and generates and prints the analysis reports.

DFSORT

The IBM Data Facility Sort sorts the DAR12O data set that is created by FABADA1. Include this step only if FABADA5 is run.

DFSORT

The IBM Data Facility Sort sorts the DAR13O data set that is created by FABADA3. Include this step whenever both FABADA3 and FABADA5 are run.

FABADA5

This FPB program uses the area and free space information generated by FABADA1 and the segment and pointer information resolved by FABADA3 to generate and print the analysis reports. Optionally, FABADA5 performs threshold analysis, generates control statements for the IMS High-Speed DEDB Direct Reorganization utility, and writes a History and History2 file records.

The steps in a DEDB Pointer Checker job stream vary, depending on your particular functional options. A typical job contains some or all of the processing steps described here.

Load modules of DEDB Pointer Checker

The DEDB Pointer Checker contains four load modules. These modules are run one after another for checking and analyzing the pointers in IMS databases. The following table lists these modules.

Table 3. Load modules of DEDB Pointer Checker

Load module name	Function
FABADA1	Extract required information
FABADA3	Resolve pointer values
FABADA4	Generate, format, and print reports
FABADA5	Generate, format, and print reports

The following table presents a list of the TYPRUN parameter values, the job steps required, and the output data sets created in the DEDB Pointer Checker process.

Table 4. Job steps and output data sets (DEDB Pointer Checker)

TYPRUN=	Job steps	Required output data sets (*: SYSOUT and SORTOUT)	Optional output data sets
PTRSCAN	FABADA1	SYSPRINT	SNAPPIT, REPORTS

Table 4. Job steps and output data sets (DEDB Pointer Checker) (continued)

TYPRUN=	Job steps	Required output data sets (*: SYSOUT and SORTOUT)	Optional output data sets
PTRALL	FABADA1	SYSPRINT, DAR13SO, DAR3CSO, SORT3CS	SNAPPIT, REPORTS
	SORT3CS	(*)	
	FABADA3	SYSPRINT	DAR4ERRS, SORT4CTL
	SORT4	(*)	
	FABADA4	SYSPRINT, SNAPPIT, REPORTS	
FS	FABADA1	SYSPRINT, DAR12O, SORT12,	SNAPPIT
	SORT12		REPORTS
	FABADA5	(*) SYSPRINT, REPORTS	(Other files are optional)
RPT	FABADA1	SYSPRINT, DAR13SO, DAR3CSO, SORT3CS, DAR12O, SORT12	SNAPPIT, REPORTS
	SORT3CS	(*)	
	FABADA3	SYSPRINT, DAR13O, SORTR13	DAR4ERRS, SORT4CTL
	SORT4	(*)	
	FABADA4	SYSPRINT, SNAPPIT, REPORTS	
	SORT12	(*)	
	SORTR13	(*)	
	FABADA5	SYSPRINT, REPORTS	(Other files are optional)
MODEL	FABADA1	SYSPRINT, DAR13SO, DAR3CSO, SORT3CS, DAR12O, SORT12	SNAPPIT, REPORTS
	SORT3CS	(*)	
	FABADA3	SYSPRINT, DAR13O, SORTR13	DAR4ERRS, SORT4CTL
	SORT4	(*)	
	FABADA4	SYSPRINT, SNAPPIT, REPORTS	
	SORT12	(*)	
	SORTR13	(*)	
	FABADA5	SYSPRINT, REPORTS	(Other files are optional)

Running DEDB Pointer Checker

You can run the DEDB Pointer Checker utility to perform complete DEDB integrity verification and generate the complete analysis reports. You can also use this function to correct integrity problems that were detected by a DEDB integrity verification process. The reports generated by the DEDB Pointer Checker utility can be used to effectively manage IMS DEDBs.

There are four main steps in running the DEDB Pointer Checker.

1. Allocate and initialize all permanent data sets required by the DEDB Pointer Checker program. If you want to change the return codes of DEDB Pointer Checker, you need to create the return code user exit routines. This is a preprocess to run the DEDB Pointer Checker program, and is done only once.

2. Run the DEDB Pointer Checker program.

3. Correct integrity problems, if there are any.

4. If necessary, analyze the analysis reports and tune your environment.

Related reference

Reference: Return code user exit routine (FABARCEX)

Fast Path DEDB Pointer Checker supports a return code user exit routine that can be used to change the return codes of FABADA1 and FABADA3.

Preprocess for DEDB Pointer Checker: Initializing permanent data sets

To use the DEDB Pointer Checker function, you must have the Statistics History data set and the Statistics History2 data set initialized.

Table 5. Statistics History data set and Statistics History2 data set

Statistics History data set	Statistics History2 data set
This data set is used by module FABADA5. It is defined by the HISTORY DD statement. You must use DISP=(MOD,KEEP,KEEP).	This data set is used by module FABADA5. It is defined by the HISTORY2 DD statement. You must use DISP=(MOD,KEEP,KEEP).
Whenever you run FABADA5 with a HISTORY DD statement present, the Statistics History data set is updated. FABADA5 adds one record for each DEDB area that it processes. The record contains the key space utilization and performance information for that area.	Whenever you run FABADA5 with a HISTORY2 DD statement specified, the Statistics History2 data set will be updated.
	FABADA5 adds one record for each UOW range that is specified by UOWCTL. The record contains the information that is totaled by UOW range for that area.
Note: Use of this data set in your FABADA5 JCL is optional. If you do not intend to maintain a history of your DEDB Pointer Checker results, you do not have to allocate the Statistics History data set.	**Note:** Use of this data set in your FABADA5 JCL is optional. If you do not intend to maintain a History2 data set of your DEDB Pointer Checker results, you do not need to allocate the Statistics History2 data set.

The following figure shows the JCL you should use to allocate the Statistics History data set and the Statistics History2 data set.

```
//****************************************************************
//** ALLOCATE AND INITIALIZE HISTORY/HISTORY2 DATA SET
//****************************************************************
//IDCAMS   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=A
//SYSIN     DD *
   DELETE  ( HPFP.STAT.HIST  ) NONVSAM
   DELETE  ( HPFP.STAT.HIST2 ) NONVSAM
   SET MAXCC = 0
/*
//****************************************************************
//**  ALLOCATE STATISTICS HISTORY DATA SET
//**  ALLOCATE STATISTICS HISTORY2 DATA SET
//****************************************************************
//IEFBR14 EXEC PGM=IEFBR14
//DASDHIO   DD DSN=HPFP.STAT.HIST,DISP=(NEW,CATLG,CATLG),
//             DCB=(DSORG=PS,RECFM=FB,LRECL=100,BLKSIZE=800,
//             UNIT=SYSDA,SPACE=(CYL,(2,2),RLSE)
//DASDHIO2  DD DSN=HPFP.STAT.HIST2,DISP=(NEW,CATLG,CATLG),
//             DCB=(DSORG=PS,RECFM=FB,LRECL=200,BLKSIZE=1600,
//             UNIT=SYSDA,SPACE=(CYL,(4,4),RLSE)
//*
```

Figure 3. Sample JCL to initialize permanent data sets: History and History2 data sets

Running the DEDB Pointer Checker process

To use the DEDB Pointer Checker, you must run several programs. To run the programs, you must code JCL statements for those programs. The steps in a DEDB Pointer Checker job stream vary, depending on your particular functional options. But there are series of steps you must follow to run DEDB Pointer Checker.

Procedure

To run DEDB Pointer Checker, you must complete the following tasks:

1. Determine the particular DEDB Pointer Checker functions that you want to run.
2. Decide which of the various DEDB Pointer Checker and DFSORT steps should be run.
3. Code the JCL for DEDB Pointer Checker (FABADA1, FABADA3, FABADA4, and FABADA5) and DFSORT job steps that you need to run.

   FPB allows you to specify site default parameters for DEDB Pointer Checker (FABADA1). Macros and sample JCL streams are provided to generate the site default table.
4. Code the control statements needed for the programs (DEDB Pointer Checker creates all required DFSORT statements).
5. Make a test run.
6. Interpret the output reports to verify that process completed successfully.
7. Put the resulting JCL and control-statement into production use.

Related reference
Site default support for FPB
This topic describes the site default support for FPB.

Correcting integrity problems in DEDB areas

A broken DEDB can be found by running DEDB Pointer Checker. In this case, you can repair the broken DEDB area.

Procedure

To repair a broken DEDB area, do the following tasks:

1. Create an image copy backup of the DEDB area.
2. Run DEDB Pointer Checker on the broken DEDB area.
3. Obtain the CI Map/CI Dump report generated by the FABADA1 and FABADA4 steps and the Pointer Chain Reconstruction report generated by the FABADA4 step.
4. Analyze the results. Verify the CI content and determine the appropriate corrections.
5. Repair the DEDB area. The IMS Database Repair Facility provides a convenient tool for printing and changing the contents of a CI without modifying the Format 1 DSCB in the VTOC.
6. Create an image copy backup of the repaired DEDB area.
7. Rerun DEDB Pointer Checker with image copy input to verify database integrity.

Analyzing the complete analysis reports

The complete analysis reports can be used to effectively manage the space utilization, performance characteristics, and physical attributes of IMS DEDBs.

About this task
For more information about using DEDB Pointer Checker reports for DEDB monitoring and tuning, see the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Region size requirements for DEDB Pointer Checker

The DEDB Pointer Checker utility operates in a virtual storage region as z/OS batch job steps.

Region size for the FABADA1 program

The region size required for the FABADA1 program of the DEDB Pointer Checker utility is a function of the size of a UOW and the TBLENTRY keyword that is specified for the SDEP table.

The length of an entry in the SDEP table is 34 bytes. The table resides above 16M by using GETMAIN. So if you need to define many entries for a deleted SDEP check, increase your region size as well.

The specification on the EXEC statement for FABADA1 of a region that is too small, is detected during the initialization processing. This causes processing to be terminated at that point.

Region size for the FABADA3 program

The region size required for the FABADA3 program of the DEDB Pointer Checker utility is a function of the size of a UOW and the TBLENTRY keyword that is specified for two internal tables. The sizes of these tables vary based on the size of a UOW and on the number of segments within a UOW (including IOVF CIs that are logically owned by a UOW).

Use the following formula to calculate the buffer size that is obtained by GETMAIN:

```
TBLENTRY * (30 + Hierarchical level * 4)
```

The tables reside above 16M by using GETMAIN. You can increase your region size depending on the TBLENTRY value.

The specification on the EXEC statement for FABADA3 of a region that is too small, is detected during the initialization processing. This causes processing to be terminated at that point.

DD statements for DEDB Pointer Checker

DD statements for DEDB Pointer Checker determine the input and output data sets and how DEDB Pointer Checker is run.

You must specify DD statements for the job control language (JCL) for each of the DEDB Pointer Checker programs.

FABADA1 JCL

The FABADA1 program extracts all required data from the database DMB (in ACBLIB) and from either the image copy data sets or the VSAM cluster.

1. The output for DAR12O, DAR13SO, and DAR3CSO is blocked to the block size of the output device. Since the blocking factor is determined at run time, standard labels must be used on all output data sets.

2. Optionally, the block size may be specified on the DD statement. RECFM and LRECL should not be specified. FABADA1 uses the following block size, if block size is not specified on the DD statement:

   3380
   : The default block size is 23476 bytes. If LRECL is larger than 23476 bytes, this LRECL is used.

   3390
   : The default block size is 27998 bytes. If LRECL is larger than 27998 bytes, this LRECL is used.

   9345
   : The default block size is 22928 bytes. If LRECL is larger than 22928 bytes, this LRECL is used.

   Other device
   : Defaults to the maximum block size of the output device.

Space requirements for the various work data sets created by FABADA1 are summarized in the following table. Most output data sets do not require DCB information to be specified in your JCL.

Table 6. DEDB Pointer Checker FABADA1 data set sizes

DDNAME	Space required
DAR12O	# CI/area x 1.5 x 20 bytes
DAR13SO	# segments x 40 bytes
DAR3CSO	# segments x 1.5 x 26 bytes
SORT12	< 1 track
SORT3CS	< 1 track

FABADA1 is run as a standard z/OS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

The EXEC statement must be in the following form:

```
//      EXEC PGM=FABADA1,PARM='DBRC=x,FORCE=y,IMSPLEX=imsplex,DBRCGRP=xxx',
//      REGION=rrrrM
```

DBRC=Y | <u>N</u>
This keyword parameter determines whether to connect to the DBRC interface.

DBRC=Y instructs FABADA1 to build a communication interface with DBRC. DBRC=N instructs FABADA1 not to build a communication interface with DBRC. DBRC=N is the default value.

FORCE=Y | <u>N</u>
This keyword parameter selects continuation of FABADA1 when AREA status is 'RECOVERY NEEDED' or when the database is accessed by other IMS online programs. This parameter works with DBRC=Y.

FORCE=Y instructs FABADA1 to continue its process when selected area data set status in DBRC is 'RECOVERY NEEDED,' or the selected DEDB is accessed by other IMS online programs with update mode.

FORCE=N instructs FABADA1 to terminate its process with an error message when the area status is 'RECOVERY NEEDED,' or the selected DEDB is accessed by other IMS online programs update mode. This is the default setting.

IMSPLEX=
IMSPLEX is a one- to five-character optional parameter that specifies which IMSplex DBRC should join in.

DBRCGRP=
DBRCGRP is a one- to three-character optional parameter, which is used by DBRC to identify which sharing group the notification belongs to.

DD statements

The following table summarizes the DD statements.

For the required output data sets as determined by each TYPRUN= value, see Table 4 on page 17.

Table 7. FABADA1 DD statements

DDNAME	Use	Format	Required or optional
JOBLIB or STEPLIB	Input	PDS	Required
SYSIN	Input	LRECL=80	Required

Table 7. FABADA1 DD statements (continued)

DDNAME	Use	Format	Required or optional
ACBLIB	Input	PDS	Optional
IMSACBA	Input	PDS	Optional
IMSACBB	Input	PDS	Optional
MODSTAT	Input		Optional
MODSTAT2	Input		Optional
RECON1	Input	KSDS	Optional
RECON2	Input	KSDS	Optional
RECON3	Input	KSDS	Optional
DFSUDUMP	Input	Image copy, or Image copy 2	Optional
DARVSAM	Input	VSAM ESDS cluster	Optional
RMODLIB	Input	PDS	Optional
IMSDALIB	Input	PDS	Optional
SYSPRINT	Output	LRECL=133	Required
REPORTS	Output	LRECL=133	Optional
DAR12O or DAR12	Output		Optional
DAR13SO or DAR13S	Output		Optional
SORT12	Output	LRECL=80	Optional
SORT3CS	Output	LRECL=80	Optional
SNAPPIT	Output	LRECL=133	Optional
DAR3CSO or DAR3CS	Output		Optional

JOBLIB/STEPLIB DD
>   When the DBRC=Y parameter of the EXEC statement is specified, the IMS load module library
>   (IMSVS.SDFSRESL) must be concatenated with the DEDB Pointer Checker library on the JOBLIB or
>   STEPLIB DD.
>
>   If you specify the IMSCATHLQ=*bsdshlq* keyword, the IMS load module library (IMSVS.SDFSRESL) and
>   the IMS Tools Base library (SGLXLOAD) must be concatenated to JOBLIB or STEPLIB DD.
>
>   To dynamically allocate an area data set, the IMS library that contains DFSMDA members must be
>   concatenated with the DEDB Pointer Checker library on JOBLIB or STEPLIB DD.

SYSIN DD
>   This statement defines the input control statement data set. This data set can reside on a direct-
>   access device or be routed through the input stream.

ACBLIB DD
>   This statement defines the library that contains the DBD member that describes the database to be
>   analyzed. For example:

```
//ACBLIB DD DSN=IMSVS.ACBLIB,DISP=SHR
```

>   The location of ACB definitions must be specified with one of the following methods:
>
>   • Specify the ACB library with the ACBLIB DD statement.

- Specify the high-level qualifier of the bootstrap data set of the IMS directory with the IMSCATHLQ=*bsdshlq* keyword. In this case, ACB definitions are obtained from the IMS directory instead of from the ACB library.
- Specify the ACB library with IMSACBA and IMSACBB DD statements. MODSTAT and MODSTAT2 DD statements must also be specified.

If the IMSCATHLQ=*bsdshlq* keyword is specified, ACBLIB, IMSACBA, IMSACBB, MODSTAT, and MODSTAT2 DD statements are ignored.

IMSACBA DD

Defines the library that contains the DMB for the database.

If you do not specify the IMSCATHLQ=*bsdshlq* keyword or the ACBLIB DD statement, you must specify this DD statement together with IMSACBB, MODSTAT, and MODSTAT2 DD statements.

IMSACBB DD

Defines the library that contains the DMB for the database.

If you do not specify the IMSCATHLQ=*bsdshlq* keyword or the ACBLIB DD statement, you must specify this DD statement together with IMSACBA, MODSTAT, and MODSTAT2 DD statements.

MODSTAT DD

Defines the MODSTAT data set.

If you do not specify the IMSCATHLQ=*bsdshlq* keyword or the ACBLIB DD statement, you must specify this DD statement together with IMSACBA, IMSACBB, and MODSTAT2 DD statements.

MODSTAT2 DD

Defines the MODSTAT2 data set.

If you do not specify the IMSCATHLQ=*bsdshlq* keyword or the ACBLIB DD statement, you must specify this DD statement together with IMSACBA, IMSACBB, and MODSTAT DD statements.

RECON1 DD

This statement defines the first DBRC RECON data set. This RECON1 data set must be the RECON1 data set used in the control region.

RECON2 DD

This statement defines the second DBRC RECON data set. This RECON2 data set must be the RECON2 data set used in the control region.

RECON3 DD

This statement defines the third DBRC RECON data set. This RECON3 data set must be the RECON3 data set used in the control region.

Omit the RECON data set DD statements when the DBRC=N parameter on the EXEC statement is specified or if you have created MDA members for them using the DFSMDA macro and want to dynamically allocate the RECONs.

You need to specify the IMSPLEX and DBRCGRP parameters on the EXEC statement for parallel RECON access to make sure that all DBRC instances in a sharing group have access to the same RECON data sets.

DFSUDUMP DD

This statement defines the image copy data sets of the areas to be analyzed. The data sets must have been previously generated using the IMS Image Copy utility (DFSUDMP0), the image copy function of IMS Image Copy Extensions, or the Image Copy 2 utility (DFSUDMT0). Concatenating the data sets from different utilities is prohibited. The data sets can reside on a tape or a direct-access device. You do not need to specify this DD statement if input is a VSAM data set.

Notes:

1. When you use compressed data sets generated by IMS Image Copy Extensions, you must specify RMODLIB DD, which defines the library in which the IMS Image Copy Extensions compression routine resides.

2. When you use data sets generated by the Image Copy 2 utility, DCB is required if the input data set is not labeled.
3. When you use compressed data sets generated by the Image Copy 2 utility, you must apply DFSMSdss APAR 0W50226.
4. LBI (Large Block Interface) image copy can be specified on this DD statement, but under the following conditions, LBI image copy cannot be processed:
   - DEDB Pointer Checker runs on an operating system that does not support LBI.
   - An LBI compressed image copy that is produced by IMS Image Copy Extensions.

DARVSAM DD
This statement defines the VSAM cluster (area data set) to be analyzed. This DD statement is omitted if input is an image copy data set.

Instead of DARVSAM DD, *areaname* DD (following) can be used to define the VSAM cluster.

***areaname* DD**
This statement defines the VSAM cluster (area data set) to be analyzed. This DD statement is omitted if input is an image copy data set, or DARVSAM DD is specified, or a DFSMDA member for the area data set with the *areaname* DD is provided.

If DARVSAM DD is specified, the *areaname* DD is ignored.

Note: When input is a VSAM data set, FABADA1 calculates and uses enough buffers for efficient processing. The buffer calculation rule used is as follows:

```
BUFND = ( n X CIs per track ) + 1
```

where n is 12 for a 3350 and 4 for a 3380.

RMODLIB DD
This statement defines the library in which the IMS Image Copy Extensions compression routine and the segment edit/compression routine reside. This DD statement is needed if the input data set is compressed by IMS Image Copy Extensions or if the segment edit/compression routine is defined in the DBDGEN for the DEDB to be processed with the operating mode TYPRUN=DATA.

IMSDALIB DD
This statement defines the IMS library that contains DFSMDA members to allocate an area data set and/or RECON data sets dynamically. If this statement is provided, program FABADA1 loads DFSMDA members from the IMSDALIB DD prior to the JOBLIB DD or the STEPLIB DD.

SYSPRINT DD
This statement defines the output message and statistics data set. The data set can reside on a direct-access device, printer, or be routed through the output stream. You should code:

```
//SYSPRINT DD SYSOUT=A
```

REPORTS DD
Defines the output data set for the area information report produced by FABADA1.

The area information report is not generated if the REPORTS DD statement is not provided.

The data set can reside on tape, a direct-access device or printer, or be routed through the output stream.

It is recommended that you code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

DAR12O DD
This statement defines the output data set that contains the free space and area information records. The data set can reside on either a direct-access device or a tape.

Space requirements vary depending on the size of the areas and the number of free space elements. (Approximation: *number of CIs in area* x 1.5 x 20 bytes.) Do not specify DISP=MOD for this DD statement.

If this DD statement is not specified in the JCL statement when it should be specified, the DAR12 DD statement is used as an alternative DD.

DAR12 DD
If DAR12O DD is not specified in the JCL statement, this statement is used instead of the DAR12O DD statement. This statement defines the output data set that contains the free space and area information records. The data set can reside on either a direct-access device or a tape.

Space requirements vary depending on the size of the areas and the number of free space elements. (Approximation: *number of CIs in area* x 1.5 x 20 bytes.) Do not specify DISP=MOD for this DD statement.

DAR13SO DD
This statement defines the output data set that contains the segment information records for all segments that reside in the root addressable part of the database. The data set can reside on either a direct-access device or a tape.

Space requirements vary depending on the size of the areas and the number of segments in the areas. (Approximation: *number of segments* x 40 bytes.) Do not specify DISP=MOD for this DD statement.

If this DD statement is not specified in the JCL statement when it should be specified, the DAR13S DD statement is used as an alternative DD.

DAR13S DD
If DAR13SO DD is not specified in the JCL statement, this statement is used instead of the DAR13SO DD statement. This statement defines the output data set that contains the segment information records for all segments that reside in the root addressable part of the database. The data set can reside on either a direct-access device or a tape.

Space requirements vary depending on the size of the areas and the number of segments in the areas. (Approximation: *number of segments* x 40 bytes.) Do not specify DISP=MOD for this DD statement.

SORT12 DD
This statement defines the output data set for the SORT control statements for the SORT12 step. The data set must reside on a direct-access device.

Space requirements are very small (1 track suffices). DCB information is hard-coded in FABADA1. Do *not* code DCB information in your JCL. Do not specify DISP=MOD for this DD statement.

SORT3CS DD
This statement defines the output data set for the SORT control statements for the sort before pointer resolution. The data set must reside on a direct-access device.

Space requirements are very small (1 track suffices). DCB information is hard-coded in FABADA1. Do *not* code DCB information in your JCL. Do not specify DISP=MOD for this DD statement.

SNAPPIT DD
This statement defines the output data set for the CI maps/CI dumps produced by FABADA1.

The CI map/CI dump option is ignored if the DD statement is not provided.

The data set can reside on direct-access device, printer, or be routed through the output stream.

You should code your DD statement as follows:

```
//SNAPPIT DD SYSOUT=A
```

DAR3CSO DD
This statement defines the output data set that contains all pointer information records and the segment information records for all segments that reside in IOVF and SDEP. The data set can reside on either a direct-access device or a tape.

Space requirements vary depending on the size of the areas and the number of segments in the areas. (Approximation: *number of segments* x 1.5 x 26 bytes.) Do not specify DISP=MOD for this DD statement.

If this DD statement is not specified in the JCL statement when it should be specified, the DAR3CS DD statement is used as an alternative DD.

DAR3CS DD

If DAR3CSO DD is not specified in the JCL statement, this statement is used instead of the DAR3CSO DD statement. This statement defines the output data set that contains all pointer information records and the segment information records for all segments that reside in IOVF and SDEP. The data set can reside on either a direct-access device or a tape.

Space requirements vary depending on the size of the areas and the number of segments in the areas. (Approximation: *number of segments* x 1.5 x 26 bytes.) Do not specify DISP=MOD for this DD statement.

DFSORT JCL (STEP SORT3CS)

The SORT3CS program sorts the DAR3CSO data set that is created by FABADA1. The sorted data set is used by FABADA3 and FABADA4.

Space requirements for the various work data sets created by SORT3CS are summarized in the following table. Most output data sets do not require DCB information to be specified in your JCL.

Table 8. DEDB Pointer Checker SORT3CS data set sizes

DDNAME	Space required
SORTOUT	# segments x 1.5 x 26 bytes

EXEC statement

The EXEC statement must be in the following form:

```
//      EXEC PGM=SORT
```

DD statements

To run DFSORT, you have to supply the appropriate DD statements. The following table summarizes the DD statements needed to run DFSORT. All statements in this table are required.

Table 9. DFSORT DD statements (Step SORT3CS)

DDNAME	Use	Format	Required or optional
SORTIN	Input		Required
SYSIN	Input		Required
SORTOUT	Output		Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

SORTIN DD

This input data set is the DAR3CSO data set from FABADA1.

SYSIN DD
    This input data set contains DFSORT control statements. It is the SORT3CS data set from FABADA1.

SORTOUT DD
    This output data set contains the sorted records. It is used by FABADA3 and FABADA4. Required space is the same size as the SORTIN data set. Do not specify DISP=MOD for this DD statement.

SYSOUT DD
    This output data set contains the message produced by DFSORT.

SORTWK*nn* DD
    These are intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about coding the SORTWK*nn* DD statements.

    Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

FABADA3 JCL

The FABADA3 program uses the segment and pointer information records generated by FABADA1 (sorted) to resolve all pointer references and builds the records required for report generation. IMS pointer discrepancies are detected and reported by this program.

1. The output for DAR13O is blocked to the block size of the output device. Since the blocking factor is determined at run time, standard labels must be used.

2. Optionally, the block size may be specified on the DD statement. RECFM and LRECL should not be specified. FABADA3 uses the following block size, if block size is not specified on the DD statement:

   3380
       The default block size is 23476 bytes. If LRECL is larger than 23476 bytes, this LRECL is used.

   3390
       The default block size is 27998 bytes. If LRECL is larger than 27998 bytes, this LRECL is used.

   9345
       The default block size is 22928 bytes. If LRECL is larger than 22928 bytes, this LRECL is used.

   Other device
       Defaults to the maximum block size of the output device.

3. Space requirements for DAR13O are one-half of the space used for the input files.

Space requirements for the various work data sets created by FABADA3 are summarized in the following table. Most output data sets do not require DCB information to be specified in your JCL.

Table 10. DEDB Pointer Checker FABADA3 data set sizes

DDNAME	Space required
DAR13O	(DAR13SO + DAR3CSO)/2
SORTR13	< 1 track
DAR4ERRS	< 1 cylinder
SOR4CTL	< 1 track

FABADA3 is run as a standard z/OS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

This statement must be in the form:

```
//      EXEC PGM=FABADA3,REGION=rrrrM
```

DD statements

The following table summarizes the DD statements. See the table for the required output data sets as determined by the FABADA1 TYPRUN= value.

Table 11. FABADA3 DD statements

DDNAME	Use	Format	Required or optional
SYSIN	Input	LRECL=80	Optional
DAR13SI or DAR13S	Input		Required
DAR3CSI or DAR3CS	Input		Required
SYSPRINT	Output	LRECL=133	Required
DAR13O or DAR13R	Output		Optional
SORTR13 or SORT13R	Output	LRECL=80	Optional
DAR4ERRS	Output		Optional
SORT4CTL	Output	LRECL=80	Optional

SYSIN DD
> This statement defines the input control statement data set. This data set can reside on a direct-access device or be routed through the input stream.

DAR13SI DD
> This statement defines the input data set that contains the segment information records. It is the DAR13SO data set from FABADA1.
>
> If this DD statement is not specified in the JCL statement, the DAR13S DD statement is used as an alternative DD.

DAR13S DD
> If DAR13SI DD is not specified in the JCL statement, this statement is used instead of the DAR13SI DD statement.

DAR3CSI DD
> This statement defines the input data set that contains the sorted segment and pointer information records. It is the SORTOUT data set from SORT3CS.
>
> If this DD statement is not specified in the JCL statement, the DAR3CS DD statement is used as an alternative DD.

DAR3CS DD
> If DAR3CSI DD is not specified in the JCL statement, this statement is used instead of the DAR3CSI DD statement.

SYSPRINT DD
> This statement defines the output message and statistics data set. The data set can reside on a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

DAR13O DD
> This statement defines the output data set that contains the information required for report generation. The data set can reside on either a direct-access device or a tape.
>
> Space requirements vary depending on the size of the areas and the number of segments in the areas. (Approximation: one-half of the sum of the size of the DAR13SO and DAR3CSO data sets from FABADA1.) Do not specify DISP=MOD for this DD statement.

If this DD statement is not specified in the JCL statement, the DAR13R DD statement is used as an alternative DD.

Note: This data set is not required when TYPRUN=PTRALL is specified in FABADA1.

DAR13R DD

If DAR13O DD is not specified in the JCL statement, this statement is used instead of the DAR13O DD statement. Do not specify DISP=MOD for this DD statement.

SORTR13 DD

This statement defines the output data set that contains the SORT control statement for the file associated with the DAR13O DD statement. This data set must reside on a direct-access device. Do not specify DISP=MOD for this DD statement.

If this DD statement is not specified in the JCL statement, the SORT13R DD statement is used as an alternative DD.

SORT13R DD

If SORTR13 DD is not specified in the JCL statement, this statement is used instead of the SORTR13 DD statement. Do not specify DISP=MOD for this DD statement.

DAR4ERRS DD

This statement defines the output data set that contains the segment information records for all segments with invalid pointers, which will be used by FABADA4. The data set can reside on either a direct-access device or a tape. Do not specify DISP=MOD for this DD statement.

Note: This data set is not required if you do not run FABADA4.

SORT4CTL DD

This statement defines the output data set for the SORT control statements for the files associated with the DAR4ERRS DD statement. This must reside on a direct-access device. Do not specify DISP=MOD for this DD statement.

Note: This data set is not required if you do not run FABADA4.

DFSORT JCL (STEP SORT4)

The SORT4 program sorts the DAR4ERRS data set that is created by FABADA3. The sorted data set is used by FABADA4.

Space requirements for the various work data sets created by SORT4 are summarized in the following table. Most output data sets do not require DCB information to be specified in your JCL.

Table 12. DEDB Pointer Checker SORT4 data set sizes

DDNAME	Space required
SORTOUT	< 1 cylinder

EXEC statement

This statement must be in the following form:

```
//       EXEC PGM=SORT
```

DD statements

To run DFSORT, you have to supply the appropriate DD statements. The following table summarizes the DD statements needed to run DFSORT. All statements in this table are required.

Table 13. DFSORT DD statements (Step SORT4)

DDNAME	Use	Format	Required or optional
SORTIN	Input		Required
SYSIN	Input		Required
SORTOUT	Output		Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

SORTIN DD
    This input data set is the DAR4ERRS file from FABADA3.

SYSIN DD
    This input data set contains DFSORT control statements. It is the SORT4CTL file from FABADA3.

SORTOUT DD
    This output data set contains the sorted records. It is used by FABADA4. Required space is the same size as the SORTIN data set. Do not specify DISP=MOD for this DD statement.

SYSOUT DD
    This output data set contains the message produced by DFSORT.

SORTWK*nn* DD
    These are intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about coding SORTWK*nn* DD statements.

    Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

FABADA4 JCL

The FABADA4 program uses the information generated by FABADA3 (sorted) for segments with invalid pointers or user-provided segment information to generate CI Map/CI Dump report and Pointer Chain Reconstruction report.

FABADA4 is run as a standard z/OS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

This statement must be in the following form:

```
//      EXEC PGM=FABADA4,REGION=1200K
```

The format of the optional PARM parameter is described as follows.

The PARM parameter of the EXEC statement for module FABADA4 can be used to determine which data set is to be used for input. The PARM parameter for module FABADA4 can be supplied on the EXEC statement in one of the following four formats:

PARM=
    If the PARM parameter is not selected, the default is the null. Supplying a null PARM (or not supplying a PARM at all) is the ordinary way of running module FABADA4. This causes module FABADA4 to use the sorted DAR4ERRS data set from the SORT4 step as input.

PARM=DAR4ERRS
    This causes module FABADA4 to use the sorted DAR4ERRS data set from the SORT4 step as input.

PARM=DAR4USER
This causes module FABADA4 to use the DAR4USER data set that you provide as input.

PARM=BOTH
This causes module FABADA4 to use both DAR4ERRS data set from the SORT4 step and DAR4USER data set that you provide as input.

DD statements

The following table summarizes the DD statements.

For the required output data sets as determined by the FABADA1 TYPRUN= value, see .

Table 14. FABADA4 DD statements

DDNAME	Use	Format	Required or optional
ACBLIB	Input	PDS	Required when IMSCATHLQ=*NO
DFSUDUMP	Input	Image copy, or Image copy 2	Optional
DARVSAM	Input	VSAM ESDS cluster	Optional
DAR4ERRS	Input		Optional
DAR4USER	Input	LRECL=80	Optional
DAR13SI	Input		Required
SORTIN	Input		Required
SORT3CS	Input		Required
SYSOUT	Output	SYSOUT	Required
SYSPRINT	Output	LRECL=133	Required
REPORTS	Output	LRECL=133	Required
RMODLIB	Input	PDS	Optional
SNAPPIT	Output	LRECL=133	Required
SORTOUT	Work data set		Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

ACBLIB DD
This statement defines the library that contains the DBD member that describes the database analyzed by FABADA1. For example:

```
//ACBLIB DD DSN=IMSVS.ACBLIB,DISP=SHR
```

If the IMSCATHLQ=*bsdshlq* keyword is specified as a site default keyword (specifies to retrieve ACB definitions from the IMS directory instead of the ACB library), ACBLIB DD statement is ignored.

DFSUDUMP DD
This statement defines the image copy data sets of the areas to be analyzed. The data sets must have been previously generated using the IMS Image Copy utility (DFSUDMP0), the image copy function of IMS Image Copy Extensions, or the Image Copy 2 utility (DFSUDMT0). Concatenating the data sets

from different utilities is prohibited. The data sets can reside on a tape or a direct-access device. You do not need to specify this DD statement if input is a VSAM data set.

Notes:

1. When you use compressed data sets generated by IMS Image Copy Extensions, you must specify RMODLIB DD, which defines the library in which the IMS Image Copy Extensions compression routine resides.

2. When you use data sets generated by the Image Copy 2 utility, DCB is required if the input data set is not labeled.

3. When you use compressed data sets generated by the Image Copy 2 utility, you must apply DFSMSdss APAR 0W50226.

4. LBI (Large Block Interface) image copy can be specified on this DD statement, but under the following conditions, LBI image copy cannot be processed:

   - DEDB Pointer Checker runs on an operating system that does not support LBI.

   - An LBI compressed image copy that is produced by IMS Image Copy Extensions.

DARVSAM DD
This statement defines the VSAM cluster (area) which was analyzed by FABADA1. This DD statement is omitted if input is an image copy data set.

DAR4ERRS DD
This statement defines the input data set that contains the segment information records for all segments with invalid pointer. It is the SORTOUT data set from SORT4.

DAR4USER DD
This statement defines the input control statement data set. The data set can reside on a direct-access device, or be routed through the input stream.

DAR13SI DD
This statement defines the input data set that contains the segment information records. It is the DAR13SO data set from FABADA1.

SORTIN DD
This statement defines the input data set that contains the sorted segment and pointer information records. It is the SORTOUT data set from SORT3CS.

SORT3CS DD
This input data set contains information for DFSORT control statements. It is the output data set of FABADA1 defined by SORT3CS DD statement.

SYSOUT DD
This output data set contains the message produced by DFSORT.

SYSPRINT DD
This statement defines the output message data set. The data set can reside on a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

REPORTS DD
This statement defines the output data set for the Pointer Chain Reconstruction report. The data set can reside on a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

RMODLIB DD
This statement defines the library in which the IMS Image Copy Extensions compression routine resides. This DD statement is needed only if the input data set is compressed by IMS Image Copy Extensions.

SNAPPIT DD

This statement defines the output data set for the CI Map/CI Dump report. This data set can reside on a direct-access device, or be routed through the input stream. You should code your DD statement as follows:

```
//SNAPPIT DD SYSOUT=A
```

SORTOUT DD

This work data set contains the sorted records. Required space is as same size as the SORTIN data set. Do not code DCB information in your JCL.

SORTWK*nn* DD

These are intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about coding the SORTWK*nn* DD statements.

Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

DFSORT JCL (STEP SORT12)

The SORT12 program sorts the DAR12O data set that is created by FABADA1. Sorted data set is used by FABADA5.

Space requirements for the various work data sets created by SORT12 are summarized in the following table. Most output data sets do not require DCB information to be specified in your JCL.

Table 15. DEDB Pointer Checker SORT12 data set sizes

DDNAME	Space required
SORTOUT	# CI/area X 1.5 X 20 bytes

EXEC statement

This statement must be in the following form:

```
//      EXEC PGM=SORT
```

DD statements

To run DFSORT, you have to supply the appropriate DD statements. The following table summarizes the DD statements needed to run DFSORT. All statements in this table are required.

Table 16. DFSORT DD statements (Step SORT12)

DDNAME	Use	Format	Required or optional
SORTIN	Input		Required
SYSIN	Input		Required
SORTOUT	Output		Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

SORTIN DD

This input data set is the DAR12O file from FABADA1.

SYSIN DD

This input data set contains DFSORT control statements. It is the SORT12 file from FABADA1.

SORTOUT DD

This output data set contains the sorted records. It is used by FABADA5. Required space is as same size as the SORTIN data set. Do not specify DISP=MOD for this DD statement.

SYSOUT DD

This output data set contains the message produced by DFSORT.

SORTWK*nn* DD

These are intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about coding SORTWK*nn* DD statements.

Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

DFSORT JCL (STEP SORTR13)

The SORTR13 program sorts the DAR13O data set that is created by FABADA3. Sorted data set is used by FABADA5. It is the SRDA13 input data set to FABBME1 of the DEDB Tuning Aid.

The following table summarizes the space requirements for the various work data sets created by SORT13. Most output data sets do not require DCB information to be specified in your JCL.

Table 17. DEDB Pointer Checker SORT13 data set sizes

DDNAME	Space required
SORTOUT	(DAR13SO + DAR3CSO)/2

EXEC statement

The EXEC statement must be in the following form:

```
//      EXEC PGM=SORT
```

DD statements

To run DFSORT, you have to supply the appropriate DD statements. The following table summarizes the DD statements needed to run DFSORT. All statements in this table are required.

Table 18. DFSORT DD statements (Step SORTR13)

DDNAME	Use	Format	Required or optional
SORTIN	Input		Required
SYSIN	Input		Required
SORTOUT	Output		Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

SORTIN DD

This input data set is the DAR13O file from FABADA3.

SYSIN DD

This input data set contains DFSORT control statements. It is the SORTR13 file from FABADA3.

SORTOUT DD

This output data set contains the sorted records. It is used by FABADA5. It is the MEDARI input data set to FABBME1 of the DEDB Tuning Aid. Required space is the same size as the SORTIN data set. Do not specify DISP=MOD for this DD statement.

SYSOUT DD

This output data set contains the message produced by DFSORT.

SORTWK*nn* DD

These are intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about coding SORTWK*nn* DD statements.

Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

FABADA5 JCL

The FABADA5 program uses the area and free space information generated by FABADA1 and the segment and pointer information resolved by FABADA3 (sorted) to generate and print the analysis reports.

Optionally, it can perform threshold analysis, generate control statements for the High-Speed DEDB Direct Reorganization utility, or generate History and History2 file entries. The History file entry contains the key space utilization and performance values for each area. The History2 file entry contains the history data that is totaled by UOW range. These files allow you to do statistical or trend analysis using user-written programs. For more information about History and History2 file records, see the topic "Historical records layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

FABADA5 is run as a standard z/OS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

The EXEC statement must be in the form:

```
//       EXEC PGM=FABADA5,REGION=768K
```

DD statements

The following table summarizes the DD statements.

For the required output data sets as determined by the FABADA1 TYPRUN= value, see .

Table 19. FABADA5 DD statements

DDNAME	Use	Format	Required or optional
DAR12I or DAR12	Input		Required
DAR13I or DAR13R	Input		Required (See note)
PARMCTL	Input	LRECL=80	Optional
THRCTL	Input	LRECL=80	Optional
UOWCTL	Input	LRECL=80	Optional
SYSPRINT	Output	LRECL=133	Required
REPORTS	Output	LRECL=133	Required
UOWRPT	Output	LRECL=133	Optional
UOWEXCP	Output	LRECL=133	Optional
REORGCTL	Output	LRECL=80	Optional

Table 19. FABADA5 DD statements (continued)

DDNAME	Use	Format	Required or optional
HISTORY	Output	DISP=MOD	Optional
HISTORY2	Output	DISP=MOD	Optional
LARGEREC	Output	LRECL=296	Optional

Note: This data set is required when TYPRUN=RPT or TYPRUN=MODEL is specified in FABADA1.

DAR12I DD

This statement defines the input data set that contains free space and area information for report generation. It is the SORTOUT data set from SORT12.

If this DD statement is not specified in the JCL statement, the DAR12 DD statement is used as an alternative DD.

DAR12 DD

If DAR12I DD is not specified in the JCL statement, this statement is used instead of the DAR12I DD statement.

DAR13I DD

This statement defines the input data set that contains segment information for report generation. It is the SORTOUT data set from SORTR13.

If this DD statement is not specified in the JCL statement, the DAR13R DD statement is used as an alternative DD.

Note: This data set is required when TYPRUN=RPT or TYPRUN=MODEL is specified in FABADA1.

DAR13R DD

If DAR13I DD is not specified in the JCL statement, this statement is used instead of the DAR13I DD statement.

Note: This data set is required when TYPRUN=RPT or TYPRUN=MODEL is specified in FABADA1.

PARMCTL DD

This statement defines the optional input parameter control statement data set. This data set can reside on a direct-access device, or be routed through the input stream. This DD statement is optional. It may be omitted, or specified as:

```
//PARMCTL DD DUMMY
```

THRCTL DD

This statement defines the optional input threshold control statement data set. This data set can reside on a direct-access device, or be routed through the input stream. This DD statement is optional. It may be omitted, or specified as:

```
//THRCTL DD DUMMY,DCB=BLKSIZE=80
```

UOWCTL DD

This statement specifies the range of UOW of the History2 file record. This data set can reside on a direct-access device, or it can be routed through the input stream. This DD statement is optional, but if it is specified, the HISTORY2 DD statement is required. The HISTORY2 DD statement can be omitted when this DD statement is not specified, or is specified as:

```
//UOWCTL DD DUMMY
```

SYSPRINT DD

This statement defines the output message data set. The data set can reside on tape, a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

REPORTS DD
> This statement defines the output analysis reports data set. The data set can reside on tape, a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

UOWRPT DD
> This statement defines the optional output data set for the UOW report. This data set can reside on tape, direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//UOWRPT DD SYSOUT=A
```

UOWEXCP DD
> This statement defines the optional output data set for the UOW Exception report. This data set can reside on tape, a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//UOWEXCP DD SYSOUT=A
```

> Entries are generated for this report in response to user-specified reorganization criteria.

REORGCTL DD
> This statement defines the optional output data set for the reorganization control statements that may be generated for input to the High-Speed Direct Reorganization utility.

```
(RECFM=FB, LRECL=80, BLKSIZE= must be specified)
```

> Do not specify DISP=MOD for this DD statement.

> **Note:** Generating reorganization control statements for the High-Speed Direct Reorganization utility requires the inclusion of the REORGCTL DD statement in the JCL stream.

> For additional requirements, see *Note* under "FABADA5 PARMCTL DD data set" on page 48.

HISTORY DD
> This statement defines the output History file data set. This data set can reside on either a direct-access device or a tape. This DD statement is optional. If it is not present in the JOB stream, a History file record is not created or journaled. If the DD statement is present in the JOB stream, a History file record is generated. If this feature is used, the History file must be pre-allocated with the following attributes:
>
> DSORG=PS
> RECFM=FB
> LRECL=100
> BLKSIZE=*user-specified*

> The DD statement in the JOB stream is specified as:

```
//HISTORY DD DSN=HPFP.DA.HISTORY,
//          DISP=(MOD,KEEP,KEEP)
```

> The following is also valid:

```
//HISTORY DD DUMMY,DCB=BLKSIZE=100
```

HISTORY2 DD
> This statement defines the output History2 file data set. This data set can reside on either a direct-access device or a tape. This DD statement is optional.

If UOWCTL DD is specified and it is not dummy, the HISTORY2 DD statement is required. If only the UOWCTL DD is specified, and the HISTORY2 DD statement is not specified, FABADA5 ends with FABA3565E message and an abend code of 3565. If the DD statement is not specified in the JOB stream, a History2 file record will not be created or journaled. If the DD statement is specified in the JOB stream, a History2 file record will be generated. When you use this feature, the History2 file must be pre-allocated with the following attributes:

```
DSORG=PS
RECFM=FB
LRECL=200
BLKSIZE=user-specified
```

The DD statement in the JOB stream is specified as:

```
//HISTORY2 DD DSN=HPFP.DA.HISTORY2,
//          DISP=(MOD,KEEP,KEEP)
```

The following syntax is also valid:

```
//HISTORY2 DD DUMMY
```

LARGEREC DD

This statement defines the optional data set for the largest database records. This statement is effective only when TYPRUN=MODEL is selected in the FABADA1 step.

This DD statement is optional. It may be omitted, or specified as:

```
//LARGEREC  DD  DUMMY
```

Do not code DCB information in your JCL.

For more information about largest database records, see the topic "Largest database records layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Related reference

FABADA5 PARMCTL DD data set
The PARMCTL DD data set contains the FPB user's specification of execution control parameters used by module FABADA5.

Input for DEDB Pointer Checker

To run the DEDB Pointer Checker function, you must specify input definitions, including the various control statements, for each program.

FABADA1 SYSIN DD data set

The SYSIN DD data set contains the FPB user's description of the processing to be done by module FABADA1. It describes the database and area to be analyzed, and it contains optional user's requests that further define the operating mode.

This topic contains product sensitive programming interface information.

Syntax of the control statement

FABADA1 requires a control statement.

Keywords and the associated values can be coded in free format (columns 1 - 72), provided certain syntactical coding rules are followed:

1. If specification of a keyword value is required, the keyword must be separated from its associated value by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or

more blanks. The value must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.

2. For keywords that do not have associated values, the keyword must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.

3. In case of duplicate keywords, the last one coded is used.

Format

This control-statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte fixed-length records. Block size, if coded, must be a multiple of 80.

Control records can be coded as shown in the following figure.

```
//SYSIN DD *
   DBDNAME=VRSDSRF,AREA=VRSTSS1,VSAM,STATS,TYPRUN=RPT
   SDEP,TBLENTRY=70000,SPTFALL
/*
```

Figure 4. FABADA1 SYSIN DD data set

Record format

There is only one statement type in the SYSIN file. It contains the following keywords:

```
DBDNAME=dbdname
[AREA=areaname]
[TYPRUN={RPT|PTRSCAN|PTRALL|FS|MODEL|DATA|DATACMP}]
[VSAM]
[SDEP]
[OVRD]

[STATS]
[STATS=YES|NO]
[TBLENTRY={10000|nnnnnnnn}]
[SPTFALL]

VSAM={YES|NO}
SDEP={YES|NO}
OVRD={YES|NO}
SPTFALL={YES|NO}
IMSCATHLQ={*NO|bsdshlq}
IMSCATACB_INPUT={CURRENT|PENDING}
```

DBDNAME=
This statement specifies the DBD name of the database to be analyzed. This operand must be the name of a DBD with a DEDB organization. DBDNAME is a required keyword.

AREA=
This statement specifies the area name that is to be analyzed.

The DEDB Pointer Checker determines an input data set by analyzing the combination of the VSAM keyword and AREA statement. This is done as follows:

• If the VSAM keyword is not specified and the AREA= statement is specified, then image copy input is processed and the specified area name is used to verify that the correct image copy data set has been provided. When concatenated image copy data sets for multiple areas are used as input, this operand must be omitted.

• If both the VSAM keyword and the AREA= statement are specified, then VSAM data set input is processed and the specified area name is used to verify that the correct area data set has been provided. When DBRC=Y is specified, then the area status in DBRC is verified.

• If the VSAM keyword is specified but the AREA= statement is not provided, then an area name is obtained from the second CI of the VSAM data set.

- If neither the VSAM keyword nor the AREA= statement is specified, then VSAM data set input (DARVSAM DD) is processed. If DARVSAM DD is not provided or DD DUMMY is specified, then image copy input (DFSDUMP DD) is processed.

To make DBRC=Y or area data set dynamic allocation effective, both the VSAM keyword and the AREA= statement must be specified.

TYPRUN=

This statement specifies the operating mode. The following describes the various modes supported. When the operating mode is not specified, DEDB Pointer Checker assumes the default operating mode of RPT.

The following table summarizes the functions performed for each TYPRUN= value.

Table 20. TYPRUN= values and associated functions

TYPRUN=	Integrity verification	Analysis and report
PTRSCAN	• Fast scan of database integrity. • Optional CI Map/CI Dump report.	
PTRALL	• Fast scan of database integrity. • In-depth pointer analysis and resolution. • Optional CI Map/CI Dump report. • Optional Pointer Chain Reconstruction report.	
FS	• Fast scan of database integrity. • Optional CI Map/CI Dump report.	• Freespace Analysis reports only. • Optional Reorg control statement generation. • Optional History file update. • Optional History2 file update. • Optional threshold analysis (free-space variables only).
RPT	• Fast scan of database integrity. • In-depth pointer analysis and resolution. • Optional CI Map/CI Dump report. • Optional Pointer Chain Reconstruction report.	• Complete analysis report generated. • Optional Reorg control statement generation. • Optional History file update. • Optional History2 file update. • Optional threshold analysis (free-space variables only).

Table 20. TYPRUN= values and associated functions (continued)

TYPRUN=	Integrity verification	Analysis and report
MODEL	• Fast scan of database integrity. • In-depth pointer analysis and resolution. • Optional CI Map/CI Dump report. • Optional Pointer Chain Reconstruction report.	• Complete analysis report generated. • Root key value appended to data records for tuning. • Optional Largest Database Records report. • Optional Reorg control statement generation. • Optional History file update. • Optional History2 file update. • Optional threshold analysis (free-space variables only).
DATA	• Fast scan of database integrity. • In-depth pointer analysis and resolution. • Segment data appended to segment information records. – Expanded segment data is appended if the segment edit/compression routine defined for the segment. • Optional CI Map/CI Dump report. • Optional Pointer Chain Reconstruction report.	
DATACMP	• Fast scan of database integrity. • In-depth pointer analysis and resolution. • Segment data appended to segment information records. – Segment data as is in CI is appended regardless of the definition of the segment edit/compression routine for the segment. • Optional CI Map/CI Dump report. • Optional Pointer Chain Reconstruction report.	

PTRSCAN

When this mode is specified, DEDB Pointer Checker scans the database for pointer integrity errors without creating any output records for further analysis and reporting. Messages are generated if any pointer integrity errors are detected. The exact RBA of some errors cannot be reported in this mode.

PTRSCAN provides a very fast method of integrity verification, using minimal computer resources. Although unlikely, it is possible that some pointer errors are undetected in this mode. Use PTRALL or RPT when complete integrity verification is needed.

PTRALL

When this mode is specified, DEDB Pointer Checker scans the database and generates the segment records and the pointer information records for detecting and reporting pointer integrity errors in the database. The exact RBA of the error is reported in this mode. Any output records associated with further analysis of the database are not generated in this mode.

FS

When this mode is specified, DEDB Pointer Checker scans the database and generates only the output records required to analyze and report free space information for the database.

RPT

When this mode is specified, FABADA1 generates output records required for a complete analysis and reporting of the database. FABADA1 also performs all functions of the PTRALL mode.

MODEL

When this mode is specified, FABADA1 appends the root key value to the segment information records for tuning the database using DEDB Tuning Aid. FABADA1 also performs all functions of the RPT mode.

DATA

When this mode is specified, DEDB Pointer Checker appends the segment data to the segment information records. If the segment edit/compression routine is defined for the segment, the routine is called to expand the segment and the expanded segment data is appended. This option allows you to access the segment data from an image copy data set in physical hierarchical data sequence. Also, DEDB Pointer Checker performs all functions of PTRALL mode.

PSPI

The formats of the records produced when TYPRUN=DATA are given by the mapping macro FABAMDAR. The file contains one type 1,1 record, one or more type 1,0 records, and many type 3 records.

PSPI

DATACMP

When this mode is specified, DEDB Pointer Checker appends the segment data to the segment information records. Regardless of the definition of the segment edit/compression routine for the segment, the segment data as is in CI is appended. This option allows you to access the segment data from a physical image copy data set in physical hierarchical data sequence. Also, DEDB Pointer Checker performs all functions of PTRALL mode.

PSPI

The formats of the records produced when TYPRUN=DATACMP are given by the mapping macro FABAMDAR. The file contains one type 1,1 record, one or more type 1,0 records, and many type 3 records.

PSPI

VSAM

The VSAM keyword instructs the DEDB Pointer Checker to process the VSAM data set. This parameter must be omitted when image copy data sets are used as input.

DEDB Pointer Checker determines an input data set by analyzing the combination of the VSAM keyword and AREA statement. This is done as follows:

- If the VSAM keyword is not specified and the AREA= statement is specified, then image copy input is processed and the specified area name is used to verify that the correct image copy data set has been provided. When concatenated image copy data sets for multiple areas are used as input, this operand must be omitted.

- If both the VSAM keyword and the AREA= statement are specified, then VSAM data set input is processed and the specified area name is used to verify that the correct area data set has been provided. When DBRC=Y is specified, then the area status in DBRC is verified.
- If the VSAM keyword is specified but the AREA= statement is not provided, then an area name is obtained from the second CI of the VSAM data set.
- If neither the VSAM keyword nor the AREA= statement is specified, then VSAM data set input (DARVSAM DD) is processed. If DARVSAM DD is not provided or DD DUMMY is specified, then image copy input (DFSDUMP DD) is processed.

To make DBRC=Y or area data set dynamic allocation effective, both the VSAM keyword and the AREA= statement must be specified.

SDEP
This keyword requests DEDB Pointer Checker to verify and report any SDEP pointer integrity errors in the database. This keyword is ignored for a database that does not specify SDEP segments.

For how to process SDEP segments, see the topic "Determining how to process SDEP segments" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

OVRD
This keyword requests DEDB Pointer Checker to override a DBD member date (that is, the date of an ACBGEN for the DBD) and image copy date discrepancy.

FABADA1 normally terminates with an abend code of 3527 if the DBD member date is after the date of the image copy. Processing can be forced to continue by specifying OVRD keyword.

⚠️ **CAUTION:** If structural or physical changes have been made to the DBD, specifying OVRD keyword causes unpredictable results.

STATS
This keyword specifies that statistics and run totals are to be generated. The default is none. It is always preferable to generate run totals and statistics.

STATS=
This optional keyword parameter specifies whether DEDB Pointer Checker is to report the run totals and statistics.

'YES' specifies that DEDB Pointer Checker is to report the run totals and statistics. STATS=YES is the same as the STATS keyword.

'NO' specifies that DEDB Pointer Checker is not to report the run totals and statistics. 'NO' is the default value.

TBLENTRY=
This keyword parameter specifies the entry number of the SDEP table (SDEPTBL). The table contains the data of the parent root segment and the RBA data of the deleted SDEP segment. When the TBLENTRY= keyword is specified with the SPTFALL keyword, information about all SDEP RBAs between Logical Begin and Logical End of the SDEP part are also kept in the table. This keyword is ignored when the SDEP keyword is not specified.

The maximum value that can be specified for TBLENTRY is 99999999; the minimum is 0. When the specified value is greater than 63161283, FPB internally replaces the specified value to 63161283. When the specified value is 0, FPB bypasses the SDEP validation process by using the SDEP table. FABADA1 generates, in the DAR3CSO data set, all pointer information records and the segment information records for all logically deleted SDEP segments as it does for the valid SDEP segments so that the information of the deleted segment can be resolved in program FABADA3.

The default value is 10000. The length of an entry in the SDEP table is 34 bytes. The table resides above 16M by using GETMAIN. So if you need to define many entries for a deleted SDEP check, increase your region size as well.

How to estimate the entry number

Estimate the following numbers first:

- Number of root segments that have an SDEP pointer
- Number of valid SDEP segments
- Number of deleted SDEP segments between the logical begin and the logical end

Then estimate the number of entries as follows:

```
number of entries = number of root segments with SDEP pointer
   + number of valid SDEP segments
                     + (number of deleted SDEP segments/8)
```

One entry saves one root segment's information. The RBA of the deleted SDEP segment (which existed between the Logical Begin and the Logical End) is kept in the table. A maximum of 8 RBAs is kept in one entry. When SPTFALL is specified, all non-deleted SDEP segment information is kept in the table. One entry saves the information for one non-deleted SDEP segment.

SPTFALL
When this keyword is specified, DEDB Pointer Checker validates whether the SPTF in a valid SDEP segment points to a deleted RBA that is located at a physically higher RBA number. If you specify this keyword, pointer information for all valid SDEP segment RBAs is kept in the SDEP table. You should specify large enough entry numbers in TBLENTRY=. When this keyword is omitted, the SPTF RBA will be checked only to determine whether it points to a deleted segment whose RBA is physically smaller than the SPTF RBA.

VSAM=
This optional keyword parameter specifies whether DEDB Pointer Checker is to process a VSAM data set or image copy data sets.

'YES' specifies that DEDB Pointer Checker is to process a VSAM data set.

VSAM=YES has the same meaning as the VSAM keyword.

'NO' specifies that DEDB Pointer Checker is to process image copy data sets. 'NO' is the default value.

SDEP=
This optional keyword parameter specifies whether DEDB Pointer Checker is to verify and report any SDEP pointer integrity errors in the database.

'YES' specifies that DEDB Pointer Checker is to verify and report any SDEP pointer integrity errors. SDEP=YES has the same meaning as the SDEP keyword.

'NO' specifies that DEDB Pointer Checker is not to verify and report any SDEP pointer integrity errors. 'NO' is the default value.

OVRD=
This optional keyword parameter specifies whether DEDB Pointer Checker is to override a DBD member date (that is, the date of an ACBGEN for the DBD) and image copy date discrepancy.

'YES' specifies that DEDB Pointer Checker is to override the date discrepancy. OVRD=YES has the same meaning as the OVRD keyword.

'NO' specifies that DEDB Pointer Checker is not to override the date discrepancy. 'NO' is the default value.

SPTFALL=
This optional keyword parameter specifies whether DEDB Pointer Checker is to validate the SPTF in a valid SDEP segment points to a deleted RBA that is located at a physically higher RBA number.

'YES' specifies that pointer information for all valid SDEP segment RBAs is kept in the SDEP table. SPTFALL=YES has the same meaning as the SPTFALL keyword.

'NO' specifies that SPTF RBA will be checked only to determine whether it points to a deleted segment whose RBA is physically smaller than the SPTF RBA. 'NO' is the default value.

IMSCATHLQ=
This optional keyword specifies to retrieve ACBs from the IMS directory (an extension of the IMS catalog) instead of from the ACB library.

'*bsdshlq*' specifies to read ACBs from the IMS directory by using the IMS Tools Catalog Interface. *bsdshlq* specifies the high-level qualifier of the bootstrap data set of the IMS directory. IMSCATHLQ=*bsdshlq* is effective when the IMS catalog and the IMS management of ACBs are enabled.

If a TYPE=CATDSHLQ statement was used to create a DFSMDA member in your MDA library, you must specify the SYSDSHLQ parameter used in that DFSMDA member for the IMSCATHLQ keyword parameter.

'*NO' specifies to read ACBs from the ACB library. *NO is the default value.

IMSCATACB_INPUT=
This optional keyword specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. IMSCATACB_INPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* option is specified.

'CURRENT' specifies that currently active ACB members are retrieved from the IMS directory data sets. CURRENT is the default value.

'PENDING' specifies that pending ACB members are retrieved from the staging data set.

Related reference
Site default support for FPB
This topic describes the site default support for FPB.

FABADA3 SYSIN DD data set

The SYSIN DD data set contains the FPB user's specification of some parameters used by module FABADA3. It describes the size of the internal segment table and the size of the address stack.

Syntax of the control statement

FABADA3 may require one control statement.

Keywords and the associated values can be coded in free format (columns 1 - 72), provided certain syntactical coding rules are followed:

1. The keyword and its value must be on one control statement.

2. If specification of a keyword value is required, the keyword must be separated from its associated value by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or more blanks. The value must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.

3. For keywords that do not have associated values, the keyword must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.

4. In case of duplicate keywords, the last one coded is used.

Format

This control-statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte fixed-length records. Block size, if coded, must be a multiple of 80.

This data set must contain only one (optional) control statement. It can be coded as shown in the following figure.

```
//FABADA3.SYSIN DD *
  TBLENTRY=15000 STATS
/*
```

Figure 5. FABADA3 SYSIN DD data set

Record format

There is only one statement type in the SYSIN file. It contains the following keywords:

```
[TBLENTRY={5000|nnnnnnn}]

[STATS]
[STATS=YES|NO]
```

TBLENTRY=
> This keyword specifies the number of entries in the internal UOW table. The size of this table must reflect the number of segments in a UOW. This includes segments in IOVF CIs that are logically owned by a UOW. For most databases with a UOW specification under (50,n), the default value should suffice. The maximum value is 9999999, and the minimum value is 1. A value of zero is not allowed.

STATS
> This keyword specifies that statistics and run totals are to be generated. The default is none. It is always preferable to generate run totals and statistics.

STATS=
> This optional keyword parameter specifies whether DEDB Pointer Checker is to report the run totals and statistics.

> 'Yes' specifies that DEDB Pointer Checker is to report the run totals and statistics. STATS=YES is the same as the STATS keyword.

> 'No' specifies that DEDB Pointer Checker is not to report the run totals and statistics. 'No' is the default value.

FABADA4 DAR4USER DD data set

The DAR4USER DD data set contains the user's description of the processing to be done by FABADA4. It contains the target relative byte addresses (RBAs). FABADA4 prints a list of all pointers that point to each target RBA. This data set is processed when DAR4USER or BOTH is specified on the FABADA4 EXEC PARM parameter.

Format

This data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain an 80-byte fixed-length record for each target RBA to be processed. Block size, if coded, must be a multiple of 80. The DAR4USER data set can be coded as shown in the following figure.

```
//DAR4USER DD *
001 0000E538
002 00A812C0
010 00019F06
/*
```

Figure 6. FABADA4 DAR4USER DD data set

Record format

There is only one record type in the DAR4USER data set.

```
  areanum    rba
```

areanum
> The area number of the DEDB database that contains the target RBA. This must start in the first column. It must be a 1-to-5 digit decimal number, followed by at least one blank.

rba
> The target RBA, an eight-digit hexadecimal number with leading zeros if necessary.

FABADA5 PARMCTL DD data set

The PARMCTL DD data set contains the FPB user's specification of execution control parameters used by module FABADA5.

Syntax of the control statement

FABADA5 may require one control statement.

Keywords and the associated values can be coded in free format (columns 1 - 72), provided certain syntactical coding rules are followed:

1. The keyword and its value must be on one control statement.
2. If specification of a keyword value is required, the keyword must be separated from its associated value by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or more blanks. The value must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.
3. For keywords that do not have associated values, the keyword must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.
4. In case of duplicate keywords, the last one coded is used.

Format

This control-statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte fixed-length records. Block size, if coded, must be a multiple of 80.

This data set must contain only one (optional) control statement. It can be coded as shown in the following figure:

```
//PARMCTL DD *
  RDOVFFS=40 SYNTBL=20 UDOVFFS=35 RBASEFS=40
/*
```

Figure 7. FABADA5 PARMCTL DD data set

Record format

There is only one statement type in the PARMCTL file. It contains the following keywords:

```
[SYNTBL={41|nn}]
[RBASEFS={100|nn}]
[RDOVFFS={100|nn}]
[UDOVFFS={0|nn}]
[MAXLREC={20|nnnn}]
[ARDIGIT={3|5}]
[RGROUP=xxxx]
```

SYNTBL=
> This keyword controls the number of reporting entries in the Synonym Chain Analysis table. The default value is 41 (that is, all synonym chains longer than 40 are grouped and reported together). The maximum value is 999, and the minimum value is 12.

RBASEFS=
> If a RAP CI contains free space greater than the percentage specified, and the RAP CI also uses overflow CIs, a reorganization control statement is generated for the UOW, and the UOW is included in the UOW Exception report. Default is 100% (that is, never generate reorganization control statements). The maximum value is 99, and the minimum value is 0.

RDOVFFS=
> If the DOVF part of a UOW contains free space greater than the percentage specified, and the UOW also uses IOVF CIs, a reorganization control statement is generated for the UOW and the UOW is

included in the UOW Exception report. Default is 100% (that is, never generate reorganization control statements). The maximum value is 99, and the minimum value is 0.

UDOVFFS=
If the DOVF UOW contains free space less than the percentage specified, the UOW is included in the UOW Exception report. Default is 0% (that is, never include in UOW Exception report). The maximum value is 99, and the minimum value is 0.

MAXLREC=
This keyword controls the number of reporting entries in the Largest Database Records report. This statement is effective only when TYPRUN=MODEL is selected in the FABADA1 step. The default value is 20. The maximum value is 9999, and the minimum value is 0. When the minimum value 0 is specified, neither the Largest Database Records report nor the LARGEREC DD data set records are generated.

ARDIGIT=
This keyword specifies the digit of the area number field for the UOW group definition records that are defined in the UOWCTL file. The value is 3 or 5. The default is 3. If the UOWCTL file is not specified on JCL statement, this keyword will be ignored.

RGROUP=
This keyword is used to fill the value in the Repository Group field in the History2 file. There is no default value. If this keyword is not specified, FABADA5 will fill the Repository Group field with 4 bytes of blank characters. If the UOWCTL file is not specified on JCL statement, this keyword will be ignored.

Note: Generating reorganization control statements for the High-Speed DEDB Direct Reorganization utility requires specification of the free space control percentages.

For additional requirements, see *Note* under REORGCTL DD statement for the "FABADA5 JCL" on page 36.

FABADA5 THRCTL DD data set

The THRCTL DD data set contains an optional control statement used by module FABADA5. It is used to specify threshold parameter values, which cause warning messages to be generated when the actual values do not meet the specified criteria.

Format

This control-statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte fixed-length records. Block size, if coded, must be a multiple of 80.

This data set must contain only one control statement. It can be coded as shown in the following figure:

```
//THRCTL     DD *
T5002 02004 01202 02004 101010 80 02004 20 050050050 101010 1010
/*
```

Figure 8. FABADA5 THRCTL DD data set

Record format

There is only one record type in the THRCTL file:

- The control statement is optional.
- Only 1 control statement is used. If multiple statements are provided, only the first statement is used. The additional statements are ignored.
- If a control statement is present, it must contain a "T" in position 1.
- All threshold parameter values are optional (that is, they can be blank).
- Only non-blank threshold parameter values are tested.
- Threshold parameters can be used to monitor the following:

1. Synonym Chain Length:

   xx% of the synonym chains are > *yy* in length.
   average length > *xx.x*
   maximum length > *yy*

2. Root I/O:

   average I/O > *xx.x*
   maximum I/O > *yy*

3. Record I/O:

   average I/O > *xx.x*
   maximum I/O > *yy*

4. Overflow usage:

   – available IOVF CIs < *xx*%
   – RAP CIs using overflow > *xx*%
   – UOWs using DOVF > *xx*%
   – UOWs using IOVF > *yy*%
   – number of IOVF CIs used by a UOW

     average > *xx.x*
     maximum > *yy*
   – number of records using IOVF CIs > *xx*%

5. Percent free space:

   total RAA BASE free space < *xx*%
   total DOVF free space < *yy*%
   total IOVF free space < *zz*%

6. Free space relationships:

   RAA BASE > *xx*% and DOVF< *yy*% or IOVF < *zz*%
   DOVF > *xx*% and IOVF < *yy*%

The format of the threshold parameter control statement is as follows:

Note: Leading and trailing zeros can be omitted.

POSITION
   DEFINITION

1

   T is a mandatory value.

2 - 5

   xxyy - Synonym Chain Length

   xx

   The percentage of the synonym chains that exceed the specified length. The maximum value is 99, and the minimum value is 0.

   yy

   The default length of the maximum chain value is 40. The maximum value is 99, and the minimum value is 0. The value must be less than the value that is specified by SYNTBL in the PARMCTL DD data set.

   Both values must be present. A warning message is issued if xx% of the synonym chains exceed length *yy*.

6

   Not used.

7 - 9

xxx - Average Synonym Chain Length. There is an implied decimal point before the third digit. A warning message is issued if the average synonym chain length is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

10 - 11

xx - Maximum Synonym Chain Length. A warning message is issued if the maximum synonym chain length is greater than *xx*. The maximum value is 99, and the minimum value is 0.

12

Not used.

13 - 15

xxx - Average Root I/O. There is an implied decimal point before the third digit. A warning message is issued if the average number of I/O to access a root segment is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

16 - 17

xx - Maximum Root I/O. A warning message is issued if the maximum number of I/O to access a root segment is greater than *xx*. The maximum value is 99, and the minimum value is 0.

18

Not used.

19 - 21

xxx - Average Record I/O. There is an implied decimal point before the third digit. A warning message is issued if the average number of I/O to access a database record is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

22 - 23

xx - Maximum Record I/O. A warning message is issued if the maximum number of I/O to access a database record is greater than *xx*. The maximum value is 99, and the minimum value is 0.

24

Not used.

25 - 26

xx - Percentage of RAA BASE CIs that use either DOVF or IOVF. A warning message is issued if the percentage of RAA BASE CIs that use overflow is greater than *xx*%. The maximum value is 99, and the minimum value is 0.

27 - 28

xx - Percentage of UOWs that use DOVF. A warning message is issued if the percentage of UOWs that use DOVF is greater than *xx*%. The maximum value is 99, and the minimum value is 0.

29 - 30

xx - Percentage of UOWs using IOVF. A warning message is issued if the number of UOWs using IOVF is greater than *xx*% of the total UOWs. The maximum value is 99, and the minimum value is 0.

31

Not used.

32 - 33

xx - Percentage of Available IOVF CIs. A warning message is issued if the available IOVF CIs are less than *xx*% of the total IOVF CIs. The maximum value is 99, and the minimum value is 0.

34

Not used.

35 - 37

xxx - Average Number of IOVF CIs used by UOW. There is an implied decimal point before the third digit. A warning message is issued if the average number of CIs used by a UOW is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

38 - 39

xx - Maximum Number of IOVF CIs used by UOW. A warning message is issued if the maximum number of IOVF CIs used by a UOW is greater than *xx*. The maximum value is 99, and the minimum value is 0.

40

Not used.

41 - 42

xx - Percentage of Records that use IOVF CIs. A warning message is issued if the number of records using IOVF CIs is greater than *xx*%. The maximum value is 99, and the minimum value is 0.

43

Not used.

44 - 46

xxx - Percentage of the free space in RAA BASE. There is an implied decimal point before the third digit. A warning message is issued if the free space in RAA BASE is less than *xxx*% of total space. The maximum value is 999, and the minimum value is 0.

47 - 49

xxx - Percentage of the free space in DOVF. There is an implied decimal point before the third digit. A warning message is issued if the free space in DOVF is less than *xxx*% of total space. The maximum value is 999, and the minimum value is 0.

50 - 52

xxx - Percentage of free space in IOVF. There is an implied decimal point before the third digit. A warning message is issued if the free space in IOVF is less than *xxx*% of total space. The maximum value is 999, and the minimum value is 0.

53

Not used.

54 - 59

xxyyzz - Free space relationships between RAA BASE, DOVF, and IOVF.

xx

The percentage of free space in RAA BASE. The maximum value is 99, and the minimum value is 0.

yy

The percentage of free space in DOVF. The maximum value is 99, and the minimum value is 0.

zz

The percentage of free space in IOVF. The maximum value is 99, and the minimum value is 0.

A value for xx must be present. Either *yy* or *zz* must be present. Both *yy* and *zz* may be present.

A warning message is issued under the following conditions:

- When RAA BASE free space is greater than xx% and DOVF free space is less than *yy*%.
- When RAA BASE free space is greater than xx% and IOVF free space is less than *zz*%.

60

Not used.

61 - 64

xxyy - Free space relationship between DOVF and IOVF.

xx

The percentage of free space in DOVF. The maximum value is 99, and the minimum value is 0.

yy

The percentage of free space in IOVF. The maximum value is 99, and the minimum value is 0.

Both values must be present. A warning message is issued if the DOVF free space is greater than xx% and if the IOVF free space is less than *yy*%.

65 - 72

Not used.

FABADA5 UOWCTL DD data set

The UOWCTL DD data set contains optional control records that are used by module FABADA5. Each record of the data set specifies a range of UOW numbers of an area, which defines a UOW group of the

area. The information about the UOW range is recorded in HISTORY2 DD in the order that is specified in the UOWCTL data set, even if the UOW ranges overlap or are in reverse order.

The data set can reside in the JCL input stream. The data set can also be defined as a sequential data set or as a member of a partitioned data set.

If the UOWCTL data set is empty, all UOWs of the area are regarded as the target UOW range.

For the record layout, see the topic "UOW group definition record layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Output for DEDB Pointer Checker

DEDB Pointer Checker provides analysis information for database administrators and technical support personnel. It provides integrity verification reports, and summary and detailed analysis reports for those in database management, maintenance, and performance tuning.

If invalid pointer chains are encountered, messages are generated to provide the relative byte address (RBA) of the "orphan" segment, the RBA of the segment that contains the pointer that refers to a nonexistent segment, or the RBA of broken pointer chains. Messages are also generated if PCF/PCL/SSP pointer interdependency errors are detected. Messages are generated if invalid FSE chains or other IMS or VSAM control field errors are encountered. Any space utilization discrepancies are reported (that is, if the total of all segment lengths plus FSE and scrap lengths plus VSAM and IMS overhead does not equal CI size).

DEDB Pointer Checker reports can be used for monitoring and tuning DEDBs. For more information, see the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

FABADA1 SYSPRINT DD data set

The SYSPRINT DD data set contains the messages issued by the FABADA1 program. These include all messages that report integrity problems with your DEDB area.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. You can code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

Scan DEDB Area-Messages

The following figure shows an example of the Scan DEDB Area-Messages.

```
IMS HPFP UTILITIES - DEDBPC                      "SCAN DEDB AREA - MESSAGES"                          PAGE:    1
5698-FPP                                    DATE: 11/22/2020  TIME: 20.13.18                     FABADA1  -  V2R1


   FABA0105I - SDEP PROCESSING STARTED - LB: 00000001 000EC000
                                         LE: 00000001 000EC004


   FABA0104I - SDEP PART PROCESSED FOR AREA: VRSTSS3


   FABA0101I - PROCESSING COMPLETE FOR DBDNAME : VRSDSRF
                                      AREA     : VRSTSS3


   FABA0102I - DEDB ANALYSIS RECORDS WRITTEN :

             DDNAME  DAR120            389
                 (1,0,       :           1)
                 (1,1,       :           4)
                 (2, ,T      :           1)
                 (2, ,F      :         271)
                 (2, ,R      :         112)

             DDNAME  DAR13SO          1,446
                 (1,0,       :           1)
                 (1,1,       :           4)
                 (3,S,       :       1,441)

             DDNAME  DAR3CSO          2,469
                 (3,C,   "0" :          55)
                 (3,C,   "1" :       1,255)
                 (3,C,   "2" :         514)
                 (3,C,   "3" :         254)
                 (3,C,   "4" :           8)
                 (3,S,       :         383)
                 (4,P,       :           0)
                 (4,S,       :           0)

   FABA0103I - SORT CONTROL CARDS WRITTEN TO DDNAME: SORT12
               SORT FIELDS=(5,13,CH,A),SIZE=00000389
               RECORD TYPE=V,LENGTH=(,,,020,020)
```

Figure 9. Scan DEDB Area-Messages (FABADA1 message)

FABADA1 SNAPPIT DD data set

The SNAPPIT DD data set contains the CI Map/CI Dump report produced by the FABADA1 program.

Format

The data set contains 133-byte fixed-length records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//SNAPPIT DD SYSOUT=A
```

CI Map/CI Dump report

This report provides the CI map/CI dump as a part of the processing. With this report, you can get all the CIs necessary to analyze any broken database without having to perform another run to obtain such information.

Purpose

The CI Map/CI Dump report is used to analyze DEDB area CIs in order to determine the best way to repair them. The report provides the following information:

- CI map that is a list of CI header, all the segments, free space elements, scrap bytes, unknown area, and CI trailer that the DEDB Pointer Checker found in the CI.
- CI dump that is logically formatted based on the CI map.

Report content: (CI Map section)

The following figure shows the CI map section of the CI Map/CI Dump report.

```
IMS HPFP UTILITIES - DEDBPC                              "CI MAP / CI DUMP REPORT"                              PAGE:    1
5698-FPP                                               DATE: 11/22/2020  TIME: 20.13.22                        FABASNAP -  V2R1


DBNAME: VRSDSRF    AREA#: 02          DDNAME: VRSTSS2   DSNAME: HPFP.DBTIVP1.DEDB.VRSTSS2.TST2                 DUMP NO. 0001
CI-NO:    11  CI-RBA:     5800
      STORAGE    TYPE(HEX)       RBA      ADDRESS   FLAGS   SC   COUNTERS          LENGTH ROOT#      TYPE(CHAR)

        49000   C8C5C1C4C5D94040  00005800  0003B000  000000  00  0000000000000000  0008   0000      (HEADER  )
        49024   E3E2E2D9D6D6E340  00005808  0003B008  000000  01  0000000000000000  013E   0000      (TSSROOT )
        49048   E3E2E2C4C9D9F340  00005946  0003B146  000000  08  0000000000000000  00A6   0000      (TSSDIR3 )
        4906C   E3E2E2C4C9D9F140  000059EC  0003B1EC  000000  03  0000000000000000  00D2   0000      (TSSDIR1 )
```

Figure 10. CI Map/CI Dump report (CI Map section)

The following are the descriptions for this figure.

STORAGE
   The memory address of the first byte in that line of the CI map.

TYPE
   The one of the following entries in database CI:

   segment-name
      The segment that includes prefix and data.

   FSE
      The free space element.

   SHORT-FS
      Scraps (that is, free space less than 4 bytes).

   UNKNOWN
      Area that the DEDB Pointer Checker could not identify as segment, FSE, or scrap bytes area. If
      the CI is in SDEP part, segments and unused area are treated as unknown area.

   HEADER
      The first 4 or 8 bytes of the CI that contains the following items:

      FSEAP 2 bytes
         Offset to the first Free Space Element (FSE). These two bytes are unused if the CI is in the
         SDEP part as it does not contain FSEs.

      CI 2 bytes
         Describe the use of this CI and the meaning of the four next bytes. DSECT DBFDBLK in the
         distributed IMS libraries provides a detailed mapping of these fields.

      RAP 4 bytes
         Root Anchor Point if this CI belongs to the base section of the root addressable part. All
         root segments randomizing to this CI are chained off this RAP in ascending key sequence.
         There is only one RAP per CI.

         Note: In the dependent and independent overflow parts, these 4 bytes are used by Fast
         Path for control information. There is no RAP in SDEP CIs.

   TRAILER
      The last 13 bytes of the CI that contains following items:

      CUSN 2 bytes
         CI update sequence number (CUSN) maintained in each CI except in the SDEP part.

      RBA 4 bytes
         Relative byte address (RBA) of this CI.

      RDF 3 bytes
         Record definition field (VSAM control information).

      CIDF 4 bytes
         CI definition field (VSAM control information).

   RBA
      The relative byte address of the area defined by this CI map entry.

ADDRESS

The address in memory.

FLAGS

Always contains zero.

SC

The segment code. The value is zero if TYPE contains the segment name.

COUNTERS

Always contains zero.

LENGTH

The length of the area defined by this CI map entry.

ROOT#

Always contains zero.

Report content: (CI Dump section)

The following figure shows the CI dump section of the CI Map/CI Dump report.

```
IMS HPFP UTILITIES - DEDBPC                               "CI MAP / CI DUMP REPORT"                                    PAGE:    2
5698-FPP                                                  DATE: 11/22/2020  TIME: 20.13.22                    FABASNAP -  V2R1


DBNAME: VRSDSRF    AREA#: 02           DDNAME: VRSTSS2   DSNAME: HPFP.DBTIVP1.DEDB.VRSTSS2.TST2                 DUMP NO. 0001
CI-NO:    11  CI-RBA:     5800

   RBA  OFFSET

   5800 0000  069C0102 00005808                                                    *........                           *
   5808 0008  01900005 5DA20000 00000005 5DA20000 59EC0005 2C200005 30080000 5DCE0000   *....).......)................)...*
        0028  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        0048  59460005 28080000 00000000 00000000 00000005 F45A0000 5B900005 36BC0000   *....................4...$.......*
        0068  000000DC D9F1F0F6 D2C5E840 4040D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6   *....R106KEY   ROOTROOTROOTROOTRO*
        0088  D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6   *OTROOTROOTROOTROOTROOTROOTROOTRO*
   LINES 000058A8-00005908   SAME AS ABOVE
        0128  D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3      *OTROOTROOTROOTROOTROOTROOTROOT  *
   5946 0146  08000000 000000A0 E2D9F1F0 F660E2F0 F140C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9   *........SR106-S01 DIR3DIR3DIR3DI*
        0166  D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9   *R3DIR3DIR3DIR3DIR3DIR3DIR3DIR3DI*
   LINES 00005986-000059C6   SAME AS ABOVE
        01E6  D9F3C4C9 D9F3                                                       *R3DIR3                         *
   59EC 01EC  03000000 5ABE0000 00000000 00000000 00000000 00000000 00000000 00000000   *....*...........................*
        020C  00000000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0 F2404040   *.....................SR106-S02  *
        022C  C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1   *DIR1DIR1DIR1DIR1DIR1DIR1DIR1DIR1*
   LINES 00005A4C-00005A8C   SAME AS ABOVE
        02AC  C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9                              *DIR1DIR1DIR1DIR1DI             *
   5ABE 02BE  03000000 5CFC0000 00000000 00000000 00000000 00000000 00000000 00000000   *....*...........................*
        02DE  00000000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0 F3404040   *.....................SR106-S03  *
        02FE  C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1   *DIR1DIR1DIR1DIR1DIR1DIR1DIR1DIR1*
   LINES 00005B1E-00005B5E   SAME AS ABOVE
        037E  C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9                              *DIR1DIR1DIR1DIR1DI             *
   5B90 0390  0C000000 5C460000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0   *....*....................SR106-S0*
        03B0  F4C4C9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *4DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
        03D0  F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *6DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
   LINES 00005BF0-00005C10   SAME AS ABOVE
        0430  F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4                    *6DIR6DIR6DIR6DIR6DIR6D         *
   5C46 0446  0C000005 2B6A0000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0   *...................SR106-S0*
        0466  F5C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *5DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
        0486  F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *6DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
   LINES 00005CA6-00005CC6   SAME AS ABOVE
        04E6  F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4                    *6DIR6DIR6DIR6DIR6DIR6D         *
   5CFC 04FC  03000005 2C200000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        051C  00000000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0 F6404040   *.....................SR106-S06  *
        053C  C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1   *DIR1DIR1DIR1DIR1DIR1DIR1DIR1DIR1*
   LINES 00005D5C-00005D9C   SAME AS ABOVE
        05BC  C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9                              *DIR1DIR1DIR1DIR1DI             *
   5DCE 05CE  07000000 000000C8 E2D9F1F0 F660E2F0 F7404040 404040C4 C9D9F2C4 C9D9F2C4   *.......HSR106-S07      DIR2DIR2D*
        05EE  C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4   *IR2DIR2DIR2DIR2DIR2DIR2DIR2DIR2D*
   LINES 00005E0E-00005E6E   SAME AS ABOVE
        068E  C9D9F2C4 C9D9F2C4 C9D9                                              *IR2DIR2DIR2DIR                 *
   5E9C 069C  80000157 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        06BC  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
   LINES 00005EDC-00005FBC   SAME AS ABOVE
        07DC  00000000 00000000 00000000 00000000 00000000 000000                *.....................          *
   5FF3 07F3  00030000 58000007 F907F900 00                                       *........9.9..                  *
```

Figure 11. CI Map/CI Dump report (CI Dump section)

The following are the descriptions for this figure:

RBA

The relative byte address of the first byte in that line of the printed CI dump.

OFFSET

The offset from the first bytes of the CI to the first byte in that line of the printed CI dump.

FABADA1 REPORTS DD data set

The REPORTS DD data set contains the area information report produced by the FABADA1 program.

Format

The data set contains 133-byte fixed-length records, and the block size (if coded in your JCL statement) must be a multiple of 133. It is recommended that you code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

AREA Information report

Purpose

The report for AREA Information report provides the following:

- Definition of the AREA
- Summary of free space in each section of the database (that is, RAA BASE, DOVF, IOVF, and SDEP)
- Statistics on the number and length (defined and actual) of SDEP segments

Report content

The following figure shows the area information report produced by the FABADA1 program.

```
IMS HPFP UTILITIES - DEDBPC                       "DEDB AREA INFORMATION REPORT"                       PAGE:    1
5698-FPP                                        DATE: 11/22/2020  TIME: 17.59.35                    FABADA1  -  V2R1

                DBDNAME: DEDBJN22    AREA NAME:  DB22AR0
                                    AREA NUMBER:   1


                                    AREA INFORMATION

                                        RMNAME : RMOD5
                                        CI-SIZE:  1024

                                        UOW=(5,1)  ROOT=(5,1)

                                        UOW'S IN RAA:        4
                                        NO. OF RAPS :       16
                                        CI'S IN DOVF:        4
                                        CI'S IN IOVF:        5
                                        NO. IOVF DIR:        1
                                        CI'S IN SDEP:      463


                                    FREESPACE INFORMATION

                                        %FREESPACE - RAA BASE:  93%
                                                     DOVF    :   75%
                                                     IOVF    :   67%
                                                     SDEP    : 100%


                                    SDEP SEGMENT INFORMATION

                                        NO. OF ROOT SEGS THAT HAVE SDEP SEGS:          0
                                        NO. OF SDEP SEGMENTS :         0

                                        SDEP SEGMENT DEFINITION LENGTH (MAX):        900
                                        SDEP SEGMENT DEFINITION LENGTH (MIN):         20

                                        MAXIMUM SDEP SEGMENT LENGTH :          0
                                        MINIMUM SDEP SEGMENT LENGTH :          0
                                        AVERAGE SDEP SEGMENT LENGTH :          0
```

Figure 12. AREA Information report

Note: FPB adds "(XCI)" after the randomizer name in this report when an XCI randomizer is used.

FABADA3 SYSPRINT DD data set

The SYSPRINT DD data set contains the messages issued by the FABADA3 program. These include all messages that report integrity problems with your DEDB area.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. You can code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

Resolve Pointers-Messages

The following figure shows an example of the Resolve Pointers Messages.

```
IMS HPFP UTILITIES - DEDBPC                              "RESOLVE POINTERS - MESSAGES"                              PAGE:    1
5698-FPP                                                DATE: 11/22/2020  TIME: 20.13.20                          FABADA3  -  V2R1


    FABA0301I - PROCESSING STARTED FOR AREA:   3


    FABA0305I - INPUT/OUTPUT TOTALS AND PROCESSING STATISTICS

              INPUT  FILE TOTALS: DAR13SI :      1,446
                                  DAR3CSI :      2,469

              OUTPUT FILE TOTALS: DAR13O  :      1,829

              INPUT RECORD TYPES: TYPE 1  :          5
                                  TYPE 3S :      1,824
                                  TYPE 3C0:         55
                                  TYPE 3C1:      1,255
                                  TYPE 3C2:        514
                                  TYPE 3C3:        254
                                  TYPE 3C4:          8
                                  TYPE 4P :          0
                                  TYPE 4S :          0

              NO. TIMES FAST BUFFER USED  :        909
              NO. TIMES LOOK-AHEAD USED   :        574
              NO. TIMES ADDRESS STACK USED:        276
              NO. TIMES ADDRESS STACK "1" :        168
              MAXIMUM TABLE ENTRIES USED  :        158
              MAXIMUM STACK ENTRIES USED  :         18

    FABA0302I - FABADA3 ENDED NORMALLY - NO POINTER ERRORS DETECTED


    FABA0315I - SORT CONTROL CARDS WRITTEN TO DDNAME: SORTR13

                  SORT FIELDS=(5,29,CH,A),SIZE=00001829
                  RECORD TYPE=V,LENGTH=(,,,0056,0056)


    FABA0315I - SORT CONTROL CARDS WRITTEN TO DDNAME: SORT4CTL

                  SORT FIELDS=(1,06,BI,A),SIZE=00000000
                  RECORD TYPE=F
```

Figure 13. Resolve Pointers-Messages

FABADA4 SYSPRINT DD data set

The SYSPRINT DD data set contains the messages issued by the FABADA4 program. These include all messages that report integrity problems with your DEDB area.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. You can code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

Pointer Chain Reconstruction-Messages

The following figure shows an example of the Pointer Chain Reconstruction Messages.

```
IMS HPFP UTILITIES - DEDBPC                   "POINTER CHAIN RECONSTRUCTION - MESSAGES"                   PAGE:    1
5698-FPP                                         DATE: 11/22/2020  TIME: 20.13.21                   FABADA4  -  V2R1




   FABA0400I - FABADA4 ENDED NORMALLY
```

Figure 14. Pointer Chain Reconstruction-Messages

FABADA4 REPORTS DD data set

The REPORTS DD data set contains the Pointer Chain Reconstruction report produced by the FABADA4 program.

Format

This data set contains 133-byte fixed-length records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

Pointer Chain Reconstruction report

This report shows segments analyzed as wrong by FABADA3 and segments that point to this segment.

Purpose

The Pointer Chain Reconstruction report provides the following information:

- A list of all pointers to user-specified targets (obtained from FABADA3 and/or input control statements of the FABADA4 program) that were detected during the DEDB Pointer Checker runs.
- CI dump number of the CI dump that contains the user-specified target. The CI dump is on the CI Map/CI Dump report produced by the FABADA4 program.

Report content

The following figure shows a sample Pointer Chain Reconstruction report.

```
IMS HPFP UTILITIES - DEDBPC                          "POINTER CHAIN RECONSTRUCTION - REPORT"                              PAGE:    1
5698-FPP                                              DATE: 11/22/2020  TIME: 20.13.21                                    FABADA4  -  V2R1


                    <--- TARGET ----->  <---- SOURCE ---->  <----------------------------- MESSAGES ------------------------------->
AREA      AREA  DUMP REC  RBA          SC  RBA        SC PTR  NUMBER      DESCRIPTION
NAME      NO    NO   TYPE                                TYPE
PC30AR1    1   0001      0000E008      1                      FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA0322E MESSAGE (XXXXXXXX)
               3,S       0000E008      1                      FABA0412I ADDRESS FOUND IN DAR13SI DATA SET
               3,C       0000E008      1  0000E000     RAP    FABA0413I SEGMENT POINTS TO ABOVE TARGET RBA

                                                             FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

           0001          0000FFFF      3                      FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA0322E MESSAGE (ZZZZZZZZ)
               3,C       0000FFFF      3  0000E008    1 PCL   FABA0413I SEGMENT POINTS TO ABOVE TARGET RBA

                                                             FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

           0002          0002A008      1                      FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA0329E MESSAGE (XXXXXXXX)
               3,S       0002A008      1                      FABA0412I ADDRESS FOUND IN DAR13SI DATA SET
               3,C       0002A008      1  0002A000     RAP    FABA0413I SEGMENT POINTS TO ABOVE TARGET RBA

                                                             FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

           0003          000A8000                             FABA0411I RBA FROM DAR4USER DATA SET

                                                             FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

           0003          000A81EA      2                      FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA0330E MESSAGE (XXXXXXXX)
               4,S       000A81EA      2                      FABA0412I ADDRESS FOUND IN DAR3CSI DATA SET

                                                             FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

           0003          000A8514      2                      FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA0330E MESSAGE (XXXXXXXX)
               4,S       000A8514      2                      FABA0412I ADDRESS FOUND IN DAR3CSI DATA SET

                                                             FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

PC30AR2    2   0001      0001C008      1                      FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA0323E MESSAGE (XXXXXXXX)
               3,S       0001C008      1                      FABA0412I ADDRESS FOUND IN DAR13SI DATA SET
               3,C       0001C008      1  0001C000     RAP    FABA0413I SEGMENT POINTS TO ABOVE TARGET RBA
    ...                                          ....
    ...    For brevity, many lines have been removed  ....
    ...                                          ....

                                                             FABA0415I END OF FILE ON DAR4ERRS AND DAR4USER DATA SET
```

Figure 15. Pointer Chain Reconstruction report

The following are the descriptions of the fields of this report:

DUMP NO
> The number specified on CI Map/CI Dump report of FABADA4 SNAPPIT data set.

The following three fields all pertain to the target of a pointer:

REC TYPE
> The type of record that is written on the DAR3CSI data set. The DEDB Pointer Checker classifies its work records into types (that is, 3s, 4p, or 4s).

RBA (TARGET)
> The relative byte address (in hexadecimal) of the target of a pointer. This is the actual of the pointer itself.

SC (TARGET)
> The segment code (in hexadecimal) of the target of the pointer.

The following three fields all pertain to the segment that contains the pointer (also called the source of the pointer):

RBA (SOURCE)
> The relative byte address (in hexadecimal) of the segment that contains the pointer.

SC (SOURCE)
> The segment code (in hexadecimal) of the segment that contains the pointer.

PTR TYPE
> The type of pointer as follows:

The following two fields pertain to error message.

NUMBER
> The message number. You can find information about each message in this report.

DESCRIPTION
The message text.

FABADA4 SNAPPIT DD data set

The SNAPPIT data set contains the CI Map/CI Dump report produced by the FABADA4 program.

Format

The data set contains 133-byte fixed-length records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//SNAPPIT DD SYSOUT=A
```

CI Map/CI Dump report

This report provides the CI map/CI dump as a part of the processing. With this report, you can get all the CIs necessary to analyze any broken database without having to perform another run to obtain such information.

Purpose

The CI Map/CI Dump report is used to analyze DEDB area CIs in order to determine the best way to repair them. The report provides the following information:

- CI map that is a list of CI header, all the segments, free space elements, scrap bytes, unknown area, and CI trailer that the DEDB Pointer Checker found in the CI.
- CI dump that is logically formatted based on the CI map.

Report content: (CI Map section)

The following figure shows the CI map section of the CI Map/CI Dump report.

```
IMS HPFP UTILITIES - DEDBPC                             "CI MAP / CI DUMP REPORT"                          PAGE:    1
5698-FPP                                          DATE: 11/22/2020  TIME: 20.13.22                  FABASNAP -  V2R1


DBNAME: VRSDSRF    AREA#: 02           DDNAME: VRSTSS2    DSNAME: HPFP.DBTIVP1.DEDB.VRSTSS2.TST2              DUMP NO. 0001
CI-NO:    11  CI-RBA:    5800

       STORAGE   TYPE(HEX)        RBA       ADDRESS   FLAGS   SC  COUNTERS          LENGTH ROOT#    TYPE(CHAR)

        49000   C8C5C1C4C5D94040  00005800  0003B000  000000  00  0000000000000000  0008   0000     (HEADER  )
        49024   E3E2E2D9D6D6E340  00005808  0003B008  000000  01  0000000000000000  013E   0000     (TSSROOT )
        49048   E3E2E2C4C9D9F340  00005946  0003B146  000000  08  0000000000000000  00A6   0000     (TSSDIR3 )
        4906C   E3E2E2C4C9D9F140  000059EC  0003B1EC  000000  03  0000000000000000  00D2   0000     (TSSDIR1 )
```

Figure 16. CI Map/CI Dump report (CI Map section)

The following are the descriptions for this figure.

STORAGE
The memory address of the first byte in that line of the CI map.

TYPE
The one of the following entries in database CI:

segment-name
The segment that includes prefix and data.

FSE
The free space element.

SHORT-FS
Scraps (that is, free space less than 4 bytes).

UNKNOWN
Area that the DEDB Pointer Checker could not identify as segment, FSE, or scrap bytes area. If the CI is in SDEP part, segments and unused area are treated as unknown area.

HEADER

The first 4 or 8 bytes of the CI that contains the following items:

FSEAP 2 bytes

Offset to the first Free Space Element (FSE). These two bytes are unused if the CI is in the SDEP part as it does not contain FSEs.

CI 2 bytes

Describe the use of this CI and the meaning of the four next bytes. DSECT DBFDBLK in the distributed IMS libraries provides a detailed mapping of these fields.

RAP 4 bytes

Root Anchor Point if this CI belongs to the base section of the root addressable part. All root segments randomizing to this CI are chained off this RAP in ascending key sequence. There is only one RAP per CI.

Note: In the dependent and independent overflow parts, these 4 bytes are used by Fast Path for control information. There is no RAP in SDEP CIs.

TRAILER

The last 13 bytes of the CI that contains following items:

CUSN 2 bytes

CI update sequence number (CUSN) maintained in each CI except in the SDEP part.

RBA 4 bytes

Relative byte address (RBA) of this CI.

RDF 3 bytes

Record definition field (VSAM control information).

CIDF 4 bytes

CI definition field (VSAM control information).

RBA

The relative byte address of the area defined by this CI map entry.

ADDRESS

The address in memory.

FLAGS

Always contains zero.

SC

The segment code. The value is zero if TYPE contains the segment name.

COUNTERS

Always contains zero.

LENGTH

The length of the area defined by this CI map entry.

ROOT#

Always contains zero.

Report content: (CI Dump section)

The following figure shows the CI dump section of the CI Map/CI Dump report.

```
IMS HPFP UTILITIES - DEDBPC                         "CI MAP / CI DUMP REPORT"                              PAGE:    2
5698-FPP                                       DATE: 11/22/2020  TIME: 20.13.22                       FABASNAP -  V2R1


DBNAME: VRSDSRF    AREA#: 02         DDNAME: VRSTSS2   DSNAME: HPFP.DBTIVP1.DEDB.VRSTSS2.TST2                 DUMP NO. 0001
CI-NO:    11  CI-RBA:    5800

    RBA  OFFSET

   5800 0000   069C0102 00005808                                                        *........                        *
   5808 0008   01900005 5DA20000 00000005 5DA20000 59EC0005 2C200005 30080000 5DCE0000   *....).......)................)...*
        0028   00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        0048   59460005 28080000 00000000 00000000 00000005 F45A0000 5B900005 36BC0000   *.....................4...$......*
        0068   000000DC D9F1F0F6 D2C5E840 4040D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6   *....R106KEY   ROOTROOTROOTROOTRO*
        0088   D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6   *OTROOTROOTROOTROOTROOTROOTROOTRO*
   LINES  000058A8-00005908    SAME AS ABOVE
        0128   D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3D9D6 D6E3        *OTROOTROOTROOTROOTROOTROOTROOT  *
   5946 0146   08000000 000000A0 E2D9F1F0 F660E2F0 F140C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9   *........SR106-S01 DIR3DIR3DIR3DI*
        0166   D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9 D9F3C4C9   *R3DIR3DIR3DIR3DIR3DIR3DIR3DIR3DI*
   LINES  00005986-000059C6    SAME AS ABOVE
        01E6   D9F3C4C9 D9F3                                                            *R3DIR3                          *
   59EC 01EC   03000000 5ABE0000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        020C   00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0 F2404040            *.................SR106-S02   *
        022C   C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1   *DIR1DIR1DIR1DIR1DIR1DIR1DIR1DIR1*
   LINES  00005A4C-00005A8C    SAME AS ABOVE
        02AC   C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9                                  *DIR1DIR1DIR1DIR1DI              *
   5ABE 02BE   03000000 5CFC0000 00000000 00000000 00000000 00000000 00000000 00000000   *....*...........................*
        02DE   00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0 F3404040            *.................SR106-S03   *
        02FE   C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1   *DIR1DIR1DIR1DIR1DIR1DIR1DIR1DIR1*
   LINES  00005B1E-00005B5E    SAME AS ABOVE
        037E   C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9                                  *DIR1DIR1DIR1DIR1DI              *
   5B90 0390   0C000000 5C460000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0   *....*...................SR106-S0*
        03B0   F4C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *4DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
        03D0   F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *6DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
   LINES  00005BF0-00005C10    SAME AS ABOVE
        0430   F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4                                  *6DIR6DIR6DIR6DIR6D              *
   5C46 0446   0C000005 2B6A0000 00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0   *...................SR106-S0*
        0466   F5C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *5DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
        0486   F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9   *6DIR6DIR6DIR6DIR6DIR6DIR6DIR6DIR*
   LINES  00005CA6-00005CC6    SAME AS ABOVE
        04E6   F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4C9D9 F6C4                                  *6DIR6DIR6DIR6DIR6D              *
   5CFC 04FC   03000005 2C200000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        051C   00000000 00000000 00000000 000000A0 E2D9F1F0 F660E2F0 F6404040            *.................SR106-S06   *
        053C   C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1   *DIR1DIR1DIR1DIR1DIR1DIR1DIR1DIR1*
   LINES  00005D5C-00005D9C    SAME AS ABOVE
        05BC   C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9D9F1 C4C9                                  *DIR1DIR1DIR1DIR1DI              *
   5DCE 05CE   07000000 000000C8 E2D9F1F0 F660E2F0 F7404040 404040C4 C9D9F2C4 C9D9F2C4   *.......HSR106-S07    DIR2DIR2D*
        05EE   C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9F2C4   *IR2DIR2DIR2DIR2DIR2DIR2DIR2DIR2D*
   LINES  00005E0E-00005E6E    SAME AS ABOVE
        068E   C9D9F2C4 C9D9F2C4 C9D9F2C4 C9D9                                          *IR2DIR2DIR2DIR                  *
   5E9C 069C   80000157 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
        06BC   00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000   *................................*
   LINES  00005EDC-00005FBC    SAME AS ABOVE
        07DC   00000000 00000000 00000000 00000000 00000000 000000                      *.........................       *
   5FF3 07F3   00030000 58000007 F907F900 00                                            *........9.9..                   *
```

Figure 17. CI Map/CI Dump report (CI Dump section)

The following are the descriptions for this figure:

RBA
The relative byte address of the first byte in that line of the printed CI dump.

OFFSET
The offset from the first bytes of the CI to the first byte in that line of the printed CI dump.

FABADA5 SYSPRINT DD data set

The SYSPRINT data set contains the messages issued by the FABADA5 program.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

Generate Reports-Messages

The following figure shows an example of the Generate Reports-Messages.

```
IMS HPFP UTILITIES - DEDBPC                   "GENERATE REPORTS - MESSAGES"                        PAGE:    1
5698-FPP                                   DATE: 11/22/2020  TIME: 20.13.24                   FABADA5  -  V2R1


   FABA0501I - PROCESSING STARTED FOR  AREA:   3  (DDNAME: VRSTSS3 )


   FABA0500I - FABADA5 ENDED NORMALLY
```

Figure 18. Generate Reports-Messages

FABADA5 REPORTS DD data set

The REPORTS data set contains most of the reports produced by the FABADA5 program.

This data set contains several reports. The header of each report is "DEDB Area Analysis Reports". DATE and TIME in the report header are the date and time of the image copy data set that is used as the input to the analysis process. If the input data set is a VSAM ESDS, the date and time show the time the FABADA1 program was run. They are printed as *mm/dd/yyyy* and *hh.mm.ss*.

Optionally, the REPORTS data set contains the Largest Database Records report.

Format

This data set contains 133-byte fixed-length records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

Subsections:

- "Freespace Analysis report" on page 64
- "DB Record Profile Analysis report" on page 70
- "DB Record Placement Analysis report" on page 71
- "Segment Placement Analysis report" on page 72
- "Segment I/O Analysis report" on page 73
- "Synonym Chain Analysis report" on page 75

Freespace Analysis report

The summary section shows the percentage of free space in each of the area components (that is, RAA BASE, DOVF, IOVF, and SDEP). The detailed analysis section shows how the free space is distributed, statistics on the number and size of the free space elements, and statistics on references to other component sections of the area.

Purpose

The report for freespace analysis provides the following information:

- Summary and detailed information about the amount of free space in each section of the database (that is, RAA BASE, DOVF, and IOVF)
- Statistics on the number and size of the free space elements in each section of the database
- Statistics on the number of pointers to other CIs in the database
- Overflow usage analysis (DOVF and IOVF).

Free space calculation notes

- The total usable space in a CI is CI size minus 21 bytes.
- The length of the free space element (FSE) is included in usable free space.
- "Scraps" (that is, free space less than 4 bytes in length) are also included in the free space calculation.

- If the calculated value for any report field exceeds the field size, "***" is printed in the report field.

Report contents

This part contains DEDB area analysis reports and their descriptions. Percentage calculations are truncated. This permits more accurate monitoring of overflow usage (that is, DOVF or IOVF < 100%).

The following figure shows the summary section of the DEDB Area Analysis report for Freespace Analysis.

```
IMS HPFP UTILITIES - DEDBPC                      "DEDB AREA ANALYSIS REPORTS"                          PAGE:    1
5698-FPP                                    DATE: 11/22/2020  TIME: 11.52.00                    FABADA5  -  V2R1




              DBDNAME: VRSDSRF      AREA NAME :VRSTSS3
                                    AREA NUMBER:   3


                                    AREA DESCRIPTION

                                      RMNAME : DBFHDC40                                      CI-SIZE:  4096


                                      UOW=(6,2)  ROOT=(38,13)

                                      UOW'S IN RAA:       25
                                      NO. OF RAPS :      100
                                      CI'S IN DOVF:       50
                                      CI'S IN IOVF:       78
                                      NO. IOVF DIR:        1
                                      CI'S IN SDEP:        4
```

Figure 19. DEDB Area Analysis reports (Freespace Analysis: Summary section)

Note: FPB adds "(XCI)" after the randomizer name in this report when an XCI randomizer is used.

Overview section

The following figure shows the RAA BASE section of the DEDB Area Analysis report for Freespace Analysis.

```
IMS HPFP UTILITIES - DEDBPC                      "DEDB AREA ANALYSIS REPORTS"                          PAGE:    2
5698-FPP                                    DATE: 11/22/2020  TIME: 11.52.00                    FABADA5  -  V2R1


DBDNAME: VRSDSRF      AREA NAME :VRSTSS3
                      AREA NUMBER:   3



FREESPACE ANALYSIS - *** OVERVIEW ***       %FREESPACE - RAA BASE:  52%  %USABLE FREESPACE:   51%
                                                         DOVF   :   36%                       35%
                                                         IOVF   :   71%                       70%
                                                         SDEP   :   75%



FREESPACE ANALYSIS - RAA BASE

  %FS      NO. CI'S  %CI   ---NO. OF FSE'S---  -----SIZE OF FSE'S------    # USE    ---PTR. TO DOVF---  ---PTR. TO IOVF---
                           AVG  S/D MAX  MIN    AVG  S/D   MAX    MIN     OVFLOW    AVG  S/D MAX  MIN   AVG  S/D MAX  MIN

  100         45     45
 90-99         0
 80-89         0
 70-79         0
 60-69         0
 50-59         0
 40-49         0
 30-39         5      5   2.6   .5    3    2    566  421  1231    166        2      .6  1.2    3    0   .4   .5    1    0
 20-29         8      8   3.2   .7    4    2    291  186   740     25        8     2.6  3.3   11    0   .7   .4    1    0
 10-19        11     11   3.0   .6    4    2    185  118   528      4        9     2.2  2.9    8    0  2.0  3.4   12    0
  0-9         31     31   2.0   .7    3    1    126   68   210      1       30     2.6  2.2    7    0   .7  1.5    6    0


         OVFL USAGE SUMMARY

              CI'S USING OVERFLOW:  NO.:       49      %CI'S:   89%
```

Figure 20. DEDB Area Analysis reports (Freespace Analysis: RAA BASE section)

The following are the descriptions for the upper half of this report.

%FREESPACE-RAA BASE
The total amount of unused space in the RAA BASE section, expressed as a percentage of total usable space.

%FREESPACE DOVF
The total amount of unused space in the DOVF section, expressed as a percentage of total usable space.

%FREESPACE IOVF
the total amount of unused space in the IOVF section, expressed as a percentage of total usable space.

> **Note:** IOVF bit maps are excluded from the free space calculations.

%FREESPACE SDEP
The total amount of usable, unused space in the SDEP part at the time the image copy was taken, or at the time of execution if input is VSAM. Data for this calculation is obtained from the DMAC.

RAA BASE %USABLE FREESPACE
The total amount of usable free space in the RAA BASE section, expressed as a percentage of total usable space. %USABLE FREESPACE provides the percentage that IMS can use to insert new segments. It does not include FSEs and the free space that is smaller than the segment length of the DBD definition.

DOVF %FREESPACE
The total amount of usable free space in the DOVF section, expressed as a percentage of total usable space.

IOVF %FREESPACE
The total amount of usable free space in the IOVF section, expressed as a percentage of total usable space.

> **Note:** IOVF bit maps are excluded from the free space calculations.

RAA BASE section

The following are the descriptions for the lower half of this report.

%FS
A free space percentage range used for classifying CIs for reporting purposes.

NO. CI'S
The total number of CIs in a section of the database that contain free space within the bounds of the reporting range.

%CI
The percentage of CIs in a database section that contain free space within the bounds of the reporting range.

NO. OF FSE'S
Provides statistics on the number of free space elements (FSEs) found in CIs within a free space reporting range.

AVG
The average number of FSEs in a CI.

S/D
The standard deviation for the average number of FSEs and provides a measure of variation.

MAX
The maximum number of FSEs in any CI within a free space reporting range.

MIN
The minimum number of FSEs in any CI within a free space reporting range.

SIZE OF FSE'S
Provides statistics on the size of free space elements (FSEs) found in CIs within a free space reporting range.

AVG
The average size of the FSEs.

S/D
The standard deviation for the average FSE size.

MAX
The length of the longest FSE in any CI within a free space reporting range.

MIN
The length of the shortest FSE in any CI within a free space reporting range.

 # USE OVFLOW
The number of RAA BASE CIs (that is, RAPs) within the free space reporting range that use (reference) either dependent overflow (DOVF) or independent overflow (IOVF) CIs.

PTR. TO DOVF
(PTR. TO IOVF)
Provides statistics on the number of references from a CI to CIs in the dependent overflow section (and independent overflow section). A reference is a PTF, PCF, PCL, or SSP pointer that points to a segment that is not in the same CI. For RAA BASE, references also include a RAP that points to a root that is in a different CI (that is, a DOVF or IOVF CI).

Note: Each specific reference is tallied. Two references from a CI to the same external CI are tallied as 2.

AVG
The average number of external references for all CIs within a free space reporting range.

S/D
The standard deviation for the average number of external references.

MAX
The maximum number of external references from any CI within a free space reporting range.

MIN
The minimum number of external references from any CI within a free space reporting range.

OVFL USAGE SUMMARY
Provides the total number and percentage of used RAA BASE CIs (that is, RAPs) that use DOVF or IOVF (that is, "Dead RAPs" are excluded from the calculation).

DOVF section

The following figure shows the DOVF and IOVF sections of the DEDB Area Analysis report for Freespace Analysis.

```
DBDNAME: VRSDSRF     AREA NAME :VRSTSS3
                     AREA NUMBER:   3
```

FREESPACE ANALYSIS - DOVF

%DOVF FREE	#UOWS	%UOW	DOVF CI'S USED AVG	S/D	MAX	MIN	NO. OF FSE'S AVG	S/D	MAX	MIN	SIZE OF FSE'S AVG	S/D	MAX	MIN	PTR TO BASE AVG	S/D	MAX	MIN	PTR TO DOVF AVG	S/D	MAX	MIN	PTR TO IOVF AVG	S/D	MAX	MIN
100	2	8																								
90-99	1	4	1.0	.0	1	1	1.0	.0	1	1	3793	0	3793	3793	1.0	.0	1	1	0				0			
80-89	1	4	1.0	.0	1	1	1.0	.0	1	1	2839	0	2839	2839	1.0	.0	1	1	0				0			
70-79	1	4	1.0	.0	1	1	1.0	.0	1	1	2367	0	2367	2367	1.0	.0	1	1	0				0			
60-69	1	4	1.0	.0	1	1	1.0	.0	1	1	1257	0	1257	1257	2.0	.0	2	2	0				0			
50-59	3	12	1.0	.0	1	1	1.0	.0	1	1	359	306	791	121	5.3	4.7	12	2	0				0			
40-49	2	8	2.0	.0	2	2	1.5	.5	2	1	1269	***	3587	63	.2	.4	1	0	2.0	.0	2	2	0			
30-39	2	8	2.0	.0	2	2	1.0	.0	1	1	1263	912	2412	77	3.5	2.3	7	1	.7	.8	2	0	0			
20-29	0																									
10-19	1	4	2.0	.0	2	2	2.5	.5	3	2	212	197	528	7	3.0	2.0	5	1	.5	.5	1	0	.5	.5	1	0
0-9	11	44	2.0	.0	2	2	1.4	.5	2	1	137	96	429	8	1.0	.9	3	0	2.0	2.9	8	0	2.3	3.3	9	0

```
        DOVF USAGE SUMMARY

            UOW'S USING DOVF:    NO.:       23     S:   92%

            NO. DOVF CI'S USED:  AVG:      1.69      SDEV :   .46     MAX:   2     MIN:   1
```

FREESPACE ANALYSIS - IOVF

%DOVF FREE	#UOWS	%UOW	#USE IOVF	IOVF CI'S USED AVG	S/D	MAX	MIN	NO. OF FSE'S AVG	S/D	MAX	MIN	SIZE OF FSE'S AVG	S/D	MAX	MIN	PTR TO BASE AVG	S/D	MAX	MIN	PTR TO DOVF AVG	S/D	MAX	MIN	PTR TO IOVF AVG	S/D	MAX	MIN
100	2	8																									
90-99	1	4	0																								
80-89	1	4	0																								
70-79	1	4	0																								
60-69	1	4	0																								
50-59	3	12	0																								
40-49	2	8	0																								
30-39	2	8	0																								
20-29	0																										
10-19	1	4	1	2.0	.0	2	2	1.5	.5	2	1	811	962	2169	55	3.0	3.0	6	0	7.0	4.0	11	3	1.5	.5	2	1
0-9	11	44	11	2.4	1.2	5	1	1.2	.5	3	1	776	***	3457	1	3.9	4.9	16	0	1.2	1.5	6	0	2.0	3.1	9	0

```
        IOVF USAGE SUMMARY

            UOW'S USING IOVF:    NO.:       12     S:   48%

            NO. IOVF CI'S USED:  AVG:      2.41      SDEV : 1.19     MAX:   5     MIN:   1

            AVAILABLE IOVF CI'S: NO.:       48      %CI'S:   62%
```

Figure 21. DEDB Area Analysis reports (Freespace Analysis: DOVF and IOVF sections)

The following are the descriptions for the upper half of this report.

% DOVF FREE
A free space percentage range used for classifying UOWs for reporting purposes. DOVF part free space for a UOW includes all FSEs in used DOVF CIs plus all empty DOVF CIs.

#UOWS
The total number of UOWs in the RAA BASE section of the database that contain free space within the bounds of the reporting range.

%UOW
The percentage of UOWs in the RAA BASE section that contain free space within the bounds of the reporting range.

DOVF CI'S USED
Provides statistics on the number of DOVF CIs used by each UOW within the free space reporting range.

Note: All other DOVF part free space calculations do not include empty DOVF CIs. Example: Number and size of FSEs do not include the FSE in a totally empty CI.

DOVF USAGE SUMMARY
Provides the total number and percentage of UOWs that use DOVF, and statistics on the number of DOVF CIs used.

AVG
> The average number of DOVF CIs used by a UOW. This average excludes UOWs that do not use DOVF.

S/D
> The standard deviation for the average number of DOVF CIs.

MAX
> The maximum number of DOVF CIs used by any UOW within the free space reporting range.

MIN
> The minimum number of DOVF CIs used by any UOW within the reporting range. (Since UOWs that do not use DOVF are excluded from the calculations, the minimum cannot be less than 1.)

IOVF section

The following are the descriptions for the lower half of this report.

% DOVF FREE
> A free space percentage range used for classifying UOWs for reporting purposes. DOVF part free space for a UOW includes all FSEs in used DOVF CIs plus all empty DOVF CIs.

#UOWS
> The total number of UOWs in the RAA BASE section of the database that contain free space within the bounds of the reporting range.

%UOW
> The percentage of UOWs in the RAA BASE section that contain free space within the bounds of the reporting range.

#USE IOVF
> The number of UOWs within the DOVF part free space reporting range that also use one or more IOVF CIs.
>
> **Note:** Reorganization may be beneficial for those UOWs that have a high percentage of DOVF free space but also use IOVF.

IOVF CI'S USED
> Provides statistics on the number of IOVF CIs used by each UOW within the free space reporting range.
>
> **Note:** All other IOVF part free space calculations are based on the actual number of IOVF CIs used.

IOVF USAGE SUMMARY
> provides the total number and percentage of UOWs that use IOVF, and statistics on the number of IOVF CIs that are used and available. (Bit maps are excluded from all calculations.)

AVG
> The average number of IOVF CIs used by a UOW. This average excludes UOWs that do not use IOVF.

S/D
> The standard deviation for the average number of IOVF CIs.

MAX
> The maximum number of IOVF CIs used by any UOW within the free space reporting range.

MIN
> The minimum number of IOVF CIs used by any UOW within the reporting range. (Since UOWs that do not use IOVF are excluded from the calculations, the minimum cannot be less than 1.)

AVAILABLE IOVF CI'S
The number and percentage of unallocated and available IOVF CIs (that is, the number of totally empty IOVF CIs). A warning message is generated if the calculated value differs from the DMAC (DMACOCNT).

Usage

The report for freespace analysis provides information that can be used to determine:

- When to reorganize or expand an area
- The optimal values for the UOW and ROOT parameters.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

DB Record Profile Analysis report

This report provides a complete profile of the database records; the number of database records; average, maximum, and minimum record lengths; segment frequencies; and segment lengths.

Purpose

The report for DB record profile analysis provides the following:

- Statistics on the number and length of the database records
- Dependent segment frequencies and length statistics.

Report content

The following figure shows an example of the DEDB Area Analysis report for the DB record profile analysis and the DB record placement analysis.

```
IMS HPFP UTILITIES - DEDBPC                    "DEDB AREA ANALYSIS REPORTS"                        PAGE:    4
5698-FPP                                    DATE: 11/22/2020  TIME: 11.52.00                 FABADA5  -  V2R1


DBDNAME: VRSDSRF     AREA NAME :VRSTSS3
                     AREA NUMBER:   3


DB RECORD PROFILE ANALYSIS

   NO. DB RECORDS:         83
   AVG REC LENGTH:      4,994
   STD. DEV.    :       1,575
   MAX REC LENGTH:     10,882
   MIN REC LENGTH:      2,266


   SEGNAME    SEG  SEG    TOT #OCCS   ------SEGMENT FREQUENCY-------   DEFINED LENGTH   ------ACTUAL LENGTH------
              CD   LVL               AVG.    SDEV    MAX     MIN       MAX      MIN     AVG.   SDEV    MAX    MIN

   TSSROOT     1   1          83      N/A     N/A     N/A     N/A      500       12     327     78    500    210
   TSSDIR1     3   2         317     3.82    0.73       5       2      800       14     182     83    800    160
   TSSD11      4   3           6     0.07    0.65       6       0      600       12     160      0    160    160
   TSSD111     5   4          21     0.25    2.29      21       0      600       14     160      0    160    160
   TSSD12      6   3           0                                      600       14
   TSSDIR2     7   2         676     8.14    3.90      18       2      200        4     200      0    200    200
   TSSDIR3     8   2         173     2.08    2.08       9       1      300       12     179     44    300    110
   TSSDIR4     9   2         164     1.98    1.24       6       1      300       12     300      0    300    300
   TSSD41     10   3           3     0.04    0.24       2       0      300       11     300      0    300    300
   TSSDIR5    11   2          83     1.00    0.00       1       1      900       13     163     33    460    140
   TSSDIR6    12   2         267     3.22    1.18       8       2      500       11     186     71    500    100
   TSSD61     13   3          11     0.13    1.10      10       0      500       13     160      0    160    160
   TSSD611    14   4           4     0.05    0.44       4       0      500        4     130     10    140    120
   TSSD612    15   4           6     0.07    0.65       6       0      500        4     140     20    160    120
   TSSD613    16   4          10     0.12    0.78       6       0      500        9     139     30    155     80
```

Figure 22. DEDB Area Analysis reports (DB record profile analysis)

The following are the descriptions for this report:

NO. DB RECORDS
The number of database records in the area.

AVG REC LENGTH
> The total length of the average database record (that is, the root and all dependent segments). This length includes the lengths of the data portions and the segment prefixes.

STD. DEV.
> The standard deviation for the average length and provides a measure of variability for the length.

MAX REC LENGTH
> The total length of the longest database record.

MIN REC LENGTH
> The length of the shortest database record in the area.

SEGNAME
> The name of the segment as defined in the DBD.

SEG CD
SEG LVL
> The segment code and the hierarchical level of the segment.

TOT #OCCS
> The total number of occurrences of the segment in an area.

SEG FREQUENCY
> Provides statistics on the number of occurrences of each segment type.

> **AVG**
> > The average number of occurrences of a dependent segment within a database record.

> **SDEV**
> > The standard deviation for the average number of occurrences.

> **MAX**
> > The maximum number of occurrences of a dependent segment for a database record.

> **MIN**
> > The minimum number of occurrences of a dependent segment for a database record.

DEFINED LENGTH
> The maximum and minimum segment lengths as defined in the DBD.

> **Note:** ACBGEN ignores the minimum length specified in the DBD and substitutes key length plus key offset with a minimum value of four (4).

ACTUAL LENGTH
> The actual average, maximum, and minimum segment lengths encountered in the database area.

> **Note:** This length is the data portion only.

Usage

The report for DB record profile analysis provides information on the characteristics of the database records. Also, the information is used in determining:

- When to reorganize an area
- The optimal values for the ROOT and UOW parameters in the DBD.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

DB Record Placement Analysis report

This report provides a profile of record placement within an area, including statistics on the number of DOVF and IOVF CIs used.

Purpose

The report for DB record placement analysis provides the following:

- The number of database records contained in each of the seven placement categories
- Statistics on the number of DOVF and IOVF CIs used by a record.

Report content

```
DB RECORD PLACEMENT ANALYSIS                  ---DOVF CI'S USED---   ---IOVF CI'S USED---
                                 NO. RECS  P/C  AVG.  SDEV MAX MIN   AVG.  SDEV MAX MIN

    RECORDS IN: BASE ONLY             9  10.8
                BASE + DOVF          27  32.5  1.15  .36   2   1
                BASE + DOVF + IOVF   19  22.9  1.21  .41   2   1   1.11  .31   2   1
                BASE + IOVF           6   7.2                     1.67  .47   2   1
                DOVF ONLY             6   7.2  2.00  .00   2   2
                DOVF + IOVF           9  10.8  1.22  .42   2   1   1.44  .50   2   1
                IOVF ONLY             7   8.4                     2.29  .70   4   2
```

Figure 23. DEDB Area Analysis reports (DB record placement analysis)

The following are the descriptions for this report:

NO. RECS
    The total number of database records in the placement category.

P/C
    The percentage of the total number of database records in the placement category.

DOVF CI'S USED
IOVF CI'S USED
    Provide statistics about DOVF and IOVF CI usage.

    AVG.
        The average number of DOVF or IOVF CIs used by records in the placement category.

    SDEV
        The standard deviation for the average number of records in the placement category.

    MAX
        The maximum number of DOVF or IOVF CIs used by any record in the placement category.

    MIN
        The minimum number of DOVF or IOVF CIs used by any record in the placement category. (Minimum can never be less than 1.)

Usage

The report for DB record placement analysis provides information that can be used to determine:

- When to reorganize a DEDB area
- When to expand CI size (if it is less than 4 K)
- The optimal value for the UOW parameter in the DBD.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Rules of thumb

1. DOVF usage is preferable to IOVF usage.
2. Less than 5 to 10% of the database records should use IOVF.
3. Expand the DOVF part of the UOW to minimize IOVF usage.

Segment Placement Analysis report

This report provides a detailed analysis of segment placement in each of the component parts of an area.

Purpose

The report for segment placement analysis provides an overview of where the segments reside in the database and is introductory information for I/O analysis.

Report content

The following figure shows an example of the DEDB Area Analysis report for the segment placement analysis.

```
IMS HPFP UTILITIES - DEDBPC                          "DEDB AREA ANALYSIS REPORTS"                          PAGE:    5
5698-FPP                                        DATE: 11/22/2020  TIME: 11.52.00                      FABADA5  -  V2R1


DBDNAME: VRSDSRF     AREA NAME :VRSTSS3
                     AREA NUMBER:   3


SEGMENT PLACEMENT ANALYSIS

                                      ----IN RAA BASE---    ------IN DOVF-----    ------IN IOVF-----
        SEGNAME   SCD  LVL  TOT #OCCS   NO. OCCS    P/C       NO. OCCS    P/C       NO. OCCS    P/C
        TSSROOT    1    1       83          44    53.0           17    20.5           22    26.5
        TSSDIR1    3    2      317         189    59.6           76    24.0           52    16.4
        TSSD11     4    3        6           0     0.0            6   100.0            0     0.0
        TSSD111    5    4       21           0     0.0           21   100.0            0     0.0
        TSSD12     6    3        0
        TSSDIR2    7    2      676         225    33.3          261    38.6          190    28.1
        TSSDIR3    8    2      173          93    53.8           39    22.5           41    23.7
        TSSDIR4    9    2      164          98    59.8           42    25.6           24    14.6
        TSSD41    10    3        3           1    33.3            2    66.7            0     0.0
        TSSDIR5   11    2       83          54    65.1           18    21.7           11    13.3
        TSSDIR6   12    2      267         144    53.9           80    30.0           43    16.1
        TSSD61    13    3       11           8    72.7            3    27.3            0     0.0
        TSSD611   14    4        4           4   100.0            0     0.0            0     0.0
        TSSD612   15    4        6           6   100.0            0     0.0            0     0.0
        TSSD613   16    4       10          10   100.0            0     0.0            0     0.0
```

Figure 24. DEDB Area Analysis reports (Segment Placement Analysis)

The following are the descriptions for this report:

SEGNAME
　　The name of the segment as defined in the DBD.

SCD
LVL
　　The segment code and the hierarchical level of the segment, respectively.

TOT #OCCS
　　The total number of occurrences of the segment in an area.

NO. OCCS
　　The number of occurrences of a segment in a specific section of an area.

P/C
　　The percentage of the total number of occurrences of a segment in a specific section of an area.

Usage

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Segment I/O Analysis report

This report provides statistics on the placement and physical I/O required to access the dependent segments for the "typical" database record. Statistics on the average number of physical I/O required to access a complete database record are also shown.

Purpose

The report for segment I/O analysis provides the following information for the "average" database record:

• Statistics on the number of dependent segments in each section of the area
• Statistics on the number of physical I/O required to retrieve the dependent segments

- Statistics on the number of physical I/O required to retrieve the average database record
- Statistics on the number of physical I/O required to retrieve the average root segment.

Report content

The following figure shows an example of the DEDB Area Analysis report for the segment I/O analysis.

```
IMS HPFP UTILITIES - DEDBPC                              "DEDB AREA ANALYSIS REPORTS"                              PAGE:    6
5698-FPP                                                 DATE: 11/22/2020  TIME: 11.52.00                          FABADA5  -  V2R1


DBDNAME: VRSDSRF     AREA NAME :VRSTSS3
                     AREA NUMBER:   3

SEGMENT I/O ANALYSIS - AVERAGE DB RECORD

                          ----------SEGMENT PLACEMENT STATISTICS----------     ---PHYSICAL I/O STATS----
                          --IN RAABASE--   ---IN DOVF----   ---IN IOVF----
    SEGNAME  SEG SEG  AVG.    AVG./   MAX/   AVG./   MAX/    AVG./   MAX/      AVG.    SDEV   MAX MIN
             CD  LVL  FREQ    SDEV    MIN    SDEV    MIN     SDEV    MIN

    TSSROOT   1   1   1.00    0.53    N/A    0.20    N/A     0.27    N/A       1.61    0.76    4   1
                              0.50           0.40            0.44
    TSSDIR1   3   2   3.82    2.28     4     0.92     4      0.63     4        0.57    0.61    2   0
                              1.60     0     1.28     0      1.20     0
    TSSD11    4   3   0.07    0.00           0.07     6      0.00              0.01    0.11    1   0
                                             0.65     0
    TSSD111   5   4   0.25    0.00           0.25    21      0.00              0.00
                                             2.29     0
    TSSD12    6   3   0.00

    TSSDIR2   7   2   8.14    2.71     7     3.14    16      2.29    18        0.65    0.86    3   0
                              2.61     0     3.95     0      4.32     0
    TSSDIR3   8   2   2.08    1.12     8     0.47     8      0.49     8        0.27    0.47    2   0
                              1.67     0     1.11     0      1.30     0
    TSSDIR4   9   2   1.98    1.18     5     0.51     6      0.29     2        0.36    0.48    1   0
                              1.18     0     1.10     0      0.67     0
    TSSD41   10   3   0.04    0.01     1     0.02     1      0.00              0.01    0.11    1   0
                              0.11     0     0.15     0
    TSSDIR5  11   2   1.00    0.65     1     0.22     1      0.13     1        0.01    0.11    1   0
                              0.48     0     0.41     0      0.34     0
    TSSDIR6  12   2   3.22    1.73     5     0.96     6      0.52     5        0.52    0.59    2   0
                              1.30     0     1.37     0      1.08     0
    TSSD61   13   3   0.13    0.10     7     0.04     3      0.00              0.01    0.11    1   0
                              0.77     0     0.33     0
    TSSD611  14   4   0.05    0.05     4     0.00            0.00              0.00
                              0.44     0
    TSSD612  15   4   0.07    0.07     6     0.00            0.00              0.00
                              0.65     0
    TSSD613  16   4   0.12    0.12     6     0.00            0.00              0.00
                              0.78     0


    ** RECORD I/O **   AVG:  2.73    SDEV:  1.03    MAX:    6    MIN:     1

    *** ROOT I/O ***   AVG:  1.61    SDEV:  0.76    MAX:    4    MIN:     1
```

Figure 25. DEDB Area Analysis reports (Segment I/O analysis)

The following are the descriptions for this report:

SEGNAME
The name of the segment as defined in the DBD.

AVG. FREQ
The average number of occurrences of a dependent segment within a database record.

SEGMENT PLACEMENT STATISTICS
Provide statistics on the average, maximum, and minimum number of occurrences of a dependent segment within a section of an area.

PHYSICAL I/O STATS
Provides statistics on the average, maximum, and minimum number of "incremental" physical I/O required to retrieve the dependent segments in a hierarchical path.

For example, to retrieve all occurrences of segment code 4 (TSSD11) which is at the third hierarchical level, the average number of physical I/O required is calculated as follows:

```
average physical I/O =  1.61 + .57 + .01
```

Maximum is seven (which is maximum record I/O), and the minimum is one.

RECORD I/O

Provides statistics on the average, maximum, and minimum number of physical I/O required (that is, the number of CIs accessed) to retrieve an entire database record.

ROOT I/O

Provides statistics on the average, maximum, and minimum number of physical I/O required to retrieve a root segment.

All numeric fields have multiple formats, and the presentation format depends on the value of the numeric item.

Usage

The report for segment I/O analysis provides information for database performance tuning, and for evaluating the I/O requirements for specific user transactions. It is also used in determining the optimal reorganization point, and for validating the effects of UOW and ROOT value parameter changes.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Synonym Chain Analysis report

This report provides statistics on the frequency and length of synonym chains, and the physical I/O required to access root segments.

Purpose

The report for synonym chain analysis provides detailed information on the frequency and length of synonym chains and statistics on root placement and physical I/O requirements.

Report content

The following figure shows an example of the DEDB Area Analysis report for the synonym chain analysis.

```
DBDNAME: VRSDSRF     AREA NAME :VRSTSS3
                     AREA NUMBER:   3


ROOTS PER RAP DISTRIBUTION GRAPH

                                                     1
              1   2   3   4   5   6   7   8   9   0
              0   0   0   0   0   0   0   0   0   0   % RAPS
              --------------------------------------------
   #    0-|**********************  (45)
        1-|*****************   (33)
   R    2-|*********   (17)
   O    3-|**   (4)
   O    4-|*   (1)
   T    5-|
   S    6-|
        7-|
   /    8-|
        9-|
   R   10-|
   A   11-|
   P   12-|


           SYNONYM CHAIN STATISTICS

              SYN CHAIN LONGEST:      4    RAP RBA: 0006F000
              SYN CHAIN MAX I/O:      4    RAP RBA: 0003E000

              SYN CHAIN LENGTH :  AVG:    2.27
              NO. ROOTS PER RAP:  AVG:    1.51


SYNONYM CHAIN ANALYSIS
```
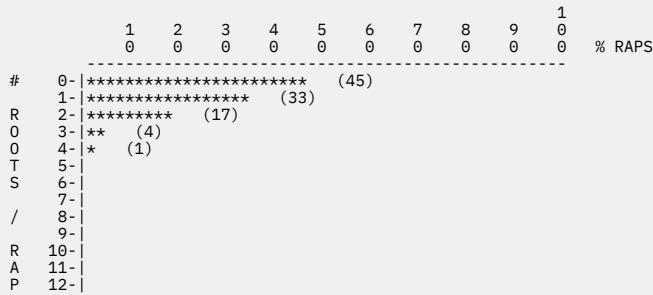
#ROOTS/ RAP	NO. RAPS	P/C	----------ROOT PLACEMENT STATISTICS-----------						---PHYSICAL I/O STATS----			
			--IN RAABASE--		---IN DOVF----		---IN IOVF----					
			AVG./ SDEV	MAX/ MIN	AVG./ SDEV	MAX/ MIN	AVG./ SDEV	MAX/ MIN	AVG.	SDEV	MAX	MIN
0	45	45.0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
1	33	33.0	0.79	1	0.06	1	0.15	1	1.21	0.41	2	1
				0		0		0				
2	17	17.0	0.88	2	0.53	1	0.59	2	2.12	0.58	3	1
			0.58	0	0.50	0	0.69	0				
3	4	4.0	0.50	1	1.50	2	1.00	2	3.50	0.50	4	3
			0.50	0	0.50	1	0.71	0				
4	1	1.0	1.00	1	0.00		3.00	3	4.00	0.00	4	4
			0.00	1			0.00	3				
5	0											

Figure 26. DEDB Area Analysis reports (Synonym chain analysis)

The following are the descriptions for this report:

#ROOTS/RAP (vertical axis)
> The number of roots chained from a RAP (that is, the length of a synonym chain).

> **Note:** The vertical axis increase in length as required, up to the user-specified number of synonym table entries. The default maximum is 40. All RAPs with synonym chains greater than the user-specified maximum (or default) are classified together.

Each graph line represents the percentage of RAPs that have the number of roots specified.

Each number is the number of RAPs that have the specified number of roots in a synonym chain.

SYN CHAIN LONGEST
SYN CHAIN MAX I/O
> The length of the longest synonym chain and the maximum number of physical I/O required for a synonym chain.

RAP RBA
> The RBA of the first CI encountered that met the condition.

SYN CHAIN LENGTH: AVG
> The average length of all synonym chains with a length greater than 1 (that is, RAPS with only one root are excluded from the calculation).

NO. ROOTS PER RAP: AVG
> The average number of roots per RAP excluding dead RAPs.

#ROOTS/RAP

The number of roots chained from a RAP (that is, it is the length of a synonym chain). This table increases in length as required, up to the user-specified (or default) maximum entries. All RAP with synonym chains of greater than the user-specified (or default) maximum are classified together.

NO. RAPS

The number of RAPS (that is, CIs) that have synonym chains of the corresponding #Roots/RAP length.

P/C

The percentage of RAP that have synonym chains of a specified length.

ROOT PLACEMENT STATISTICS

Provide statistics on the average, maximum, and minimum number of root segments that reside in the specified area section for a given synonym chain length.

PHYSICAL I/O STATS

Provide statistics on the average, maximum, and minimum number of physical I/O required to retrieve the root segments from the specified area section for a given synonym chain length.

All numeric fields have multiple formats, and the presentation format depends on the value of the numeric item.

Usage

The report for synonym chain analysis provides information for:

- Determining the effectiveness of a randomizing module. In theory, roots should be evenly distributed across all RAPs.
- Determining the optimal values for the ROOT and UOW parameters.
- Performance tuning the area.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Largest Database Records report

Purpose

The report for the largest database records provides a description of each large database record, beginning with the largest.

Report content

The following figure shows an example of the Largest Database Records report.

```
IMS HPFP UTILITIES - DEDBPC                           "LARGEST DATABASE RECORDS"                              PAGE:    1
5698-FPP                                              DATE: 11/22/2020  TIME: 10.11.54                   FABADA5   -   V2R1

DBDNAME: DEDBJN22   KEY LENGTH:   5

  RECORD SIZE    # OF SEG'S   AREANAME   AREA#   ROOT RBA    RAP#    RAP RBA    ROOT SEGMENT SEQUENCE FIELD (HEX)   (CHARACTER)
  -----------    ----------   --------   -----   --------    -----   --------   --------------------------------   *---------------*
       2,470             5    DB22AR0       1    0000798C        5   00007000   F0F1F0F0 F5                         *01005         *
       2,470             5    DB22AR0       1    0000E008       10   0000E000   F0F1F0F0 F4                         *01004         *
       2,470             5    DB22AR0       1    0001E204       17   00017000   F0F1F0F0 F6                         *01006         *
       2,470             5    DB22AR0       1    00030008       38   00030000   F0F1F0F0 F2                         *01002         *
       2,470             5    DB22AR0       1    00038032       44   00038000   F0F1F0F0 F3                         *01003         *
       2,470             5    DB22AR0       1    00043008       53   00043000   F0F1F0F0 F1                         *01001         *
       2,436             3    DB22AR0       1    00007008        5   00007000   F0F0F9F0 F3                         *00903         *
       2,436             3    DB22AR0       1    00010602       12   00010000   F0F0F9F0 F6                         *00906         *
       2,436             3    DB22AR0       1    000177FE       17   00017000   F0F0F9F0 F5                         *00905         *
       2,436             3    DB22AR0       1    00035008       41   00035000   F0F0F9F0 F2                         *00902         *
       2,436             3    DB22AR0       1    00048008       56   00048000   F0F0F9F0 F1                         *00901         *
       2,436             3    DB22AR0       1    00050008       56   00048000   F0F0F9F0 F4                         *00904         *
       2,038             3    DB22AR0       1    00017008       17   00017000   F0F0F8F0 F3                         *00803         *
       2,038             3    DB22AR0       1    0002E066       36   0002E000   F0F0F8F0 F2                         *00802         *
       2,038             3    DB22AR0       1    00032204       38   00030000   F0F0F8F0 F6                         *00806         *
       2,038             3    DB22AR0       1    0003F008       49   0003F000   F0F0F8F0 F4                         *00804         *
       2,038             3    DB22AR0       1    0003F7FE       49   0003F000   F0F0F8F0 F5                         *00805         *
       2,038             3    DB22AR0       1    0004E008       62   0004E000   F0F0F8F0 F1                         *00801         *
       1,928             2    DB22AR0       1    0000C008        8   0000C000   F0F0F7F0 F1                         *00701         *
       1,928             2    DB22AR0       1    00012406       14   00012000   F0F0F7F0 F6                         *00706         *
```

Figure 27. Largest Database Records report

The following values are tabulated in this report:

RECORD SIZE
The total length of the database record that includes prefix length of all segments in the record.

 # OF SEG'S
The total number of segments in the record.

AREANAME
The name of the area.

AREA#
The area number.

ROOT RBA
The value of root segment RBA. If the report is the result of DEDB Tuning Aid, the value of root segment RBA is null.

ROOT SEGMENT SEQUENCE FIELD (HEX)
(CHARACTER)
the value of the key in root segment.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

FABADA5 UOWRPT DD data set

The UOWRPT data set contains the report for UOW analysis produced by the FABADA5 program.

Note: DATE and TIME in the report header are the date and time of the image copy data set that is used as the input to the analysis process. If the input data set is a VSAM ESDS, the date and time show the time the FABADA1 program was run. They are printed as *mm/dd/yyyy* and *hh.mm.ss*.

Format

This data set contains 133-byte fixed-length records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//UOWRPT DD SYSOUT=A
```

UOW report

This optional report provides automated threshold reporting, which causes warning messages to be generated when conditions within the database do not meet user-specified criteria. This feature can be used to show when reorganization or expansion is required, or if database performance falls below acceptable standards.

Purpose

The optional report for UOW analysis provides detailed information on each UOW.

Report content

The following figure shows an example of the UOW report.

```
IMS HPFP UTILITIES - DEDBPC                                        "UOW REPORT"                                       PAGE:    1
5698-FPP                                              DATE: 11/22/2020  TIME: 20.13.18                           FABADA5  -  V2R1


DBDNAME: VRSDSRF     AREA NAME :VRSTSS3
                     AREA NUMBER:   3


             ---NUMBER CI'S USED---  #RAPS USE  ---%RAP FS---   % DOVF    #DB      ------RECORD LENGTH STATS------
    UOW NO.    #RAPS  #DOVF  #IOVF    OVFL CI'S  AVG  MAX  MIN   PART FS  RECORDS    AVERAGE   MAXIMUM   MINIMUM
        0        2      1      0         2        12   20    4     52       2         5,534     5,930     5,138
        1        1      1      0         1         8    8    8     79       1         5,440     5,440     5,440
        2        1      1      0         1         4    4    4     84       1         5,138     5,138     5,138
        3        2      2      0         2        19   22   15     30       3         4,080     4,540     3,704
        4        1      2      2         1         4    4    4      1       3         6,367     6,992     5,716
        5        4      2      3         4         9   14    3      1       7         4,503     6,892     2,266
        6        2      2      0         2         8    9    7     45       3         3,964     4,108     3,788
        7        3      1      0         1        17   34    5     59       3         4,442     6,810     2,678
        8        3      2      1         2        16   25   11      9       5         4,165     5,400     2,466
        9        3      1      0         2        12   34    0     65       3         4,492     6,710     2,678
       10        1      2      1         1        16   16   16      1       3         5,152     5,952     4,540
       11        1      1      0         1         8    8    8     96       1         3,998     3,998     3,998
       12        4      2      2         4         9   26    2      4       5         5,707     8,880     3,904
       13        3      2      1         2        21   39    8      6       4         4,536     5,590     2,466
       14        1      2      0         1         9    9    9     47       2         3,968     3,998     3,938
       15        4      2      0         4         7    9    3     31       5         4,140     5,771     2,566
       16        3      2      2         3        18   25    8      2       5         4,522     6,111     3,820
       17        0      0      0         0                         100      0
       18        2      2      5         2         6   10    1      9       5         6,528    10,882     4,540
       19        4      2      2         4        13   33    4     13       6         4,489     6,226     2,566
       20        1      0      0         0        11   11   11     100      1         3,604     3,604     3,604
       21        1      1      0         1         8    8    8     51       1         7,690     7,690     7,690
       22        2      2      4         2        22   39    6      8       5         5,501     7,690     3,984
       23        3      2      3         3        11   20    5      7       5         5,386     7,806     3,938
       24        3      2      3         3        15   22    8      2       4         6,943     9,533     5,138
```

Figure 28. UOW report

The following are the descriptions for this report:

UOW NO.
The number of the unit of work.

#RAPS
The number of RAPs in the UOW that contain data (that is, the remaining RAPs are "dead RAPs").

#DOVF
The number of DOVF CIs that contain data in this UOW.

#IOVF
The number of IOVF CIs that are allocated to the UOW.

#RAPS USE OVFL CI'S
The number of RAP CIs that contain a pointer that references either DOVF or IOVF.

%RAP FS
Provides statistics on the average, maximum, and minimum percentage free space in used RAP CIs. Empty RAP CIs are excluded form the calculations.

% DOVF PART FS
The percentage free space in the DOVF part of the UOW. DOVF part free space for a UOW includes all FSEs in used DOVF CIs plus all empty DOVF CIs.

#DB RECORDS
The number of database records in the UOW.

RECORD LENGTH STATS

Provides statistics on the average, maximum, and minimum record lengths. The record length includes both prefix and data lengths

Usage

The UOW report can be used to:

- List detailed information about each UOW
- Identify which UOWs require reorganization
- Manage space utilization when the randomizer has been modified to group related data into contiguous UOWs
- List only those UOWs that violate the user-specified space utilization criteria.

Since no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

FABADA5 UOWEXCP DD data set

The UOWEXCP data set contains the UOW Exception report produced by the FABADA5 program.

Note: DATE and TIME in the report header are the date and time of the image copy data set that is used as the input to the analysis process. If the input data set is a VSAM ESDS, the date and time show the time the FABADA1 program was run. They are printed as *mm/dd/yyyy* and *hh.mm.ss*.

Format

This data set contains 133-byte fixed-length records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//UOWEXCP DD SYSOUT=A
```

UOW Exception report

This optional report identifies the units of work (UOWs) that exceed a user-specified reorganization criteria. The control statements required to run the IMS High-Speed DEDB Direct Reorganization utility (DBFUMDR0) can be generated optionally.

Entries on the UOW Exception report are generated when one of the following user-specified reporting parameters is exceeded. These user-specified reporting parameters are controlled by the keywords that you specify on the PARMCTL DD statement.

- > *nn*% free space in a RAP CI and overflow CIs are used. Use the RBASEFS keyword to specify this parameter.
- > *nn*% free space in DOVF and the UOW uses IOVF CIs. Use the RDOVFFS keyword to specify this parameter.
- < *nn*% free space in the DOVF part of a UOW. Use the UDOVFFS keyword to specify this parameter.

Related reading: For information about the PARMCTL DD statement, see "FABADA5 PARMCTL DD data set" on page 48.

Report content

The following figure shows an example of the UOW Exception report.

```
IMS HPFP UTILITIES - DEDBPC                              "UOW EXCEPTION REPORT"                              PAGE:    1
5698-FPP                                        DATE: 11/22/2020  TIME: 20.13.18                        FABADA5  - V2R1


DBDNAME: VRSDSRF    AREA NAME :VRSTSS3
                    AREA NUMBER:   3


            ---NUMBER CI'S USED---  #RAPS USE ---%RAP FS---   % DOVF    #DB    ------RECORD LENGTH STATS------
     UOW NO.   #RAPS  #DOVF  #IOVF  OVFL CI'S  AVG MAX  MIN  PART FS  RECORDS    AVERAGE    MAXIMUM    MINIMUM
         22      2      2      4        2       22  39    6     8         5        5,501      7,690      3,984
```

Figure 29. UOW Exception report

The following are the descriptions for this report:

UOW NO.
> The number of the unit of work.

#RAPS
> The number of RAPs in the UOW that contain data (that is, the remaining RAPs are "dead RAPs").

#DOVF
> The number of DOVF CIs that contain data in this UOW.

#IOVF
> The number of IOVF CIs that are allocated to the UOW.

#RAPS USE OVFL CI'S
> The number of RAP CIs that contain a pointer that references either DOVF or IOVF.

%RAP FS
> Provides statistics on the average, maximum, and minimum percentage free space in used RAP CIs. Empty RAP CIs are excluded form the calculations.

% DOVF PART FS
> The percentage free space in the DOVF part of the UOW. DOVF part free space for a UOW includes all FSEs in used DOVF CIs plus all empty DOVF CIs.

#DB RECORDS
> The number of database records in the UOW.

RECORD LENGTH STATS
> Provides statistics on the average, maximum, and minimum record lengths. The record length includes both prefix and data lengths.

Usage

The UOW Exception report can be used to:

- List detailed information about each UOW
- Identify which UOWs require reorganization
- Manage space utilization when the randomizer has been modified to group related data into contiguous UOWs
- List only those UOWs that violate the user-specified space utilization criteria

Because no single report provides all the required information, usage of the reports is discussed in the topic "DEDB monitoring and tuning guide" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Examples for DEDB Pointer Checker

There are many ways to run DEDB Pointer Checker. The examples provided in the following topics show some of the typical ways that you can use.

By studying and understanding these examples, you can learn the techniques to use to effectively check the pointers and manage groups of UOWs.

You can find additional JCL examples that help you to monitor and tune DEDB areas in the topic "DEDB Pointer Checker JCL examples for monitoring and tuning DEDBs" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Example 1: Pointer checking and analyzing the DEDB area data set

The following figure shows example JCL statement for pointer checking and analyzing the DEDB area data set.

The complete procedure or subsets of the procedure, are used to generate all DEDB analysis reports except the SDEP Utilization report.

```
//FABADA1 EXEC PGM=FABADA1,REGION=1024K
//*****************************************************
//** EXTRACT DATA                                   **
//** FOR IMAGE COPY INPUT USE: //DFSUDUMP DD DSN=   **
//** FOR VSAM INPUT USE:       //DARVSAM  DD DSN=   **
//*****************************************************
//ACBLIB    DD DSN=IMSVS.ACBLIB,DISP=SHR
//DFSUDUMP  DD DSN=HPFP.IMGCPY.TSSAR01,DISP=OLD
//SYSPRINT  DD SYSOUT=A
//SNAPPIT   DD SYSOUT=A
//SYSIN     DD *
  DBDNAME=TSSDBD
/*
//DAR12O    DD DSN=HPFP.TSSDA.DA12,
//            DISP=(NEW,CATLG),
//            UNIT=SYSDA,
//            SPACE=(CYL,(5,2),RLSE)
//DAR13SO   DD DSN=HPFP.TSSDA.DA13S,
//            DISP=(NEW,CATLG),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,5),RLSE)
//DAR3CSO   DD DSN=HPFP.TSSDA.DA3CS,
//            DISP=(NEW,CATLG),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,5),RLSE)
//SORT12    DD DSN=&SORT12,
//            DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//SORT3CS   DD DSN=&SORT3CS,
//            DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//*
```

Figure 30. Sample JCL for pointer checking process for DEDB area data set (Part 1 of 4)

```
//SORT3CS EXEC PGM=SORT,REGION=2048K,PARM='CORE=MAX'
//*************************
//** SORT DA3CS RECORDS  **
//*************************
//SYSOUT    DD SYSOUT=A
//SYSIN     DD DSN=&SORT3CS,DISP=(OLD,PASS)
//SORTIN    DD DSN=HPFP.TSSDA.DA3CS,
//             DISP=(OLD,DELETE,KEEP)
//SORTOUT   DD DSN=HPFP.TSSDA.SDA3CS,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(50,2),RLSE)
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(25,5))
//*
//FABADA3 EXEC PGM=FABADA3,REGION=768K
//*******************************
//** RESOLVE 3,C AND 3,S RECORDS **
//*******************************
//SYSPRINT  DD SYSOUT=A
//DAR13SI   DD DSN=HPFP.TSSDA.DA13S,
//             DISP=(OLD,PASS)
//DAR3CSI   DD DSN=HPFP.TSSDA.SDA3CS,
//             DISP=(OLD,PASS)
//DAR4ERRS  DD DSN=HPFP.TSSDA.DA4ERRS,
//             DISP=(NEW,CATLG),
//             UNIT=SYSDA,
//             SPACE=(CYL,(2,2),RLSE)
//DAR13O    DD DSN=HPFP.TSSDA.RDA13,
//             DISP=(NEW,CATLG),
//             UNIT=SYSDA,
//             SPACE=(CYL,(50,2),RLSE)
//SORT4CTL  DD DSN=&SORT4,
//             DISP=(NEW,PASS),
//             UNIT=SYSDA,
//             SPACE=(TRK,(2,1),RLSE)
//SORTR13   DD DSN=&SORTR13,
//             DISP=(NEW,PASS),
//             UNIT=SYSDA,
//             SPACE=(TRK,(2,1),RLSE)
//*
```

Figure 31. Sample JCL for pointer checking process for DEDB area data set (Part 2 of 4)

```
//SORT4    EXEC PGM=SORT,REGION=2048K,PARM='CORE=MAX'
//**************************
//** SORT DA4ERRS RECORDS **
//**************************
//SYSOUT    DD SYSOUT=A
//SYSIN     DD DSN=&SORT4,DISP=(OLD,DELETE)
//SORTIN    DD DSN=HPFP.TSSDA.DA4ERRS,
//             DISP=(OLD,DELETE,KEEP)
//SORTOUT   DD DSN=HPFP.TSSDA.SDA4ERRS,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(TRK,(2,1),RLSE)
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//*
//FABADA4 EXEC PGM=FABADA4,REGION=1200K
//***************************************************
//** GENERATE CI MAP/DUMP & REPORT               **
//** FOR IMAGE COPY INPUT USE: //DFSUDUMP DD DSN= **
//** FOR VSAM INPUT USE:       //DARVSAM  DD DSN= **
//**                                             **
//**     ERROR DATA                              **
//** FOR SYSTEM INPUT USE:     //DAR4ERRS DD DSN= **
//**                           PARM='DAR4ERRS'   **
//**                           OR PARM=''        **
//** FOR USER INPUT USE:       //DAR4USER DD DSN= **
//**                           PARM='DAR4USER'   **
//** FOR BOTH INPUT USE:       //DAR4ERRS DD DSN= **
//**                           //DAR4USER DD DSN= **
//**                           PARM='BOTH'       **
//***************************************************
//ACBLIB    DD DSN=IMSVS.ACBLIB,DISP=SHR
//DFSUDUMP  DD DSN=HPFP.IMGCPY.TSSAR01,DISP=OLD
//SYSOUT    DD SYSOUT=A
//SYSPRINT  DD SYSOUT=A
//REPORTS   DD SYSOUT=A
//SNAPPIT   DD SYSOUT=A
//DAR4ERRS  DD DSN=HPFP.TSSDA.SDA4ERRS,
//             DISP=(OLD,DELETE,KEEP)
//DAR13SI   DD DSN=HPFP.TSSDA.DA13S,
//             DISP=(OLD,DELETE,KEEP)
//SORTIN    DD DSN=HPFP.TSSDA.SDA3CS,
//             DISP=(OLD,DELETE,KEEP)
//SORT3CS   DD DSN=&SORT3CS,
//             DISP=(OLD,DELETE)
//SORTOUT   DD DSN=&SDA3CS,
//             DISP=(NEW,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(50,2),RLSE)
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(2,2))
//*
```

Figure 32. Sample JCL for pointer checking process for DEDB area data set (Part 3 of 4)

```
//SORT12   EXEC PGM=SORT,REGION=2048K,PARM='CORE=MAX'
//*************************
//** SORT DA12 RECORDS    **
//*************************
//SYSOUT    DD SYSOUT=A
//SYSIN     DD DSN=&SORT12,DISP=(OLD,DELETE)
//SORTIN    DD DSN=HPFP.TSSDA.DA12,
//             DISP=(OLD,DELETE,KEEP)
//SORTOUT   DD DSN=HPFP.TSSDA.SDA12,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(10,5),RLSE)
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(5,2))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(5,2))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(5,2))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(5,2))
//*
//*
//SORTR13 EXEC PGM=SORT,REGION=2048K,PARM='CORE=MAX'
//*************************
//** SORT DA13 RECORDS    **
//*************************
//SYSOUT    DD SYSOUT=A
//SYSIN     DD DSN=&SORTR13,DISP=(OLD,DELETE)
//SORTIN    DD DSN=HPFP.TSSDA.RDA13,
//             DISP=(OLD,DELETE,KEEP)
//SORTOUT   DD DSN=HPFP.TSSDA.SRDA13,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(50,5),RLSE)
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(25,5))
//*
//*************************
//** GENERATE REPORTS     **
//*************************
//FABADA5 EXEC PGM=FABADA5,REGION=768K
//SYSPRINT  DD SYSOUT=A
//REPORTS   DD SYSOUT=A
//UOWRPT    DD SYSOUT=A
//UOWEXCP   DD SYSOUT=A
//DAR12I    DD DSN=HPFP.TSSDA.SDA12,
//             DISP=(OLD,DELETE,KEEP)
//DAR13I    DD DSN=HPFP.TSSDA.SRDA13,
//             DISP=(OLD,DELETE,KEEP)
//HISTORY   DD DSN=HPFP.TSSDA.HIST,
//             DISP=(MOD,KEEP,KEEP)
//REORGCTL  DD DSN=HPFP.TSSDA.REORGCTL,
//             DISP=(NEW,CATLG),
//             SPACE=(CYL,(2,1),RLSE),
//             UNIT=SYSDA,
//             DCB=BLKSIZE=6000
//PARMCTL   DD *
  SYNTBL=20 RBASEFS=40 RDOVFFS=40 UDOVFFS=35
/*
//THRCTL    DD *
T5002 02004 01202 02004 101010 80 02004 20 050050050 101010 1010
/*
```

Figure 33. Sample JCL for pointer checking process for DEDB area data set (Part 4 of 4)

Normally, only one database is analyzed per execution of this procedure. The image copy data sets for all areas of a database could be concatenated (in any order) for input to this procedure. However, it is usually advisable to analyze each area individually. This is primarily because of the size of the files created, and the sort work requirements.

If the VSAM cluster is used as input, only one area can be analyzed per execution of this procedure.

Example 2: Monitoring groups of UOWs

This topic provides a sample scenario for monitoring groups of UOWs.

For monitoring groups of UOWs, application requirements might dictate that associated data be grouped in contiguous CIs for ease of processing.

As an example:

> It might be desirable to have all savings accounts for each branch adjacent to avoid a sequential scan of the entire database when processing data for a single branch.

You can do this easily by modifying the randomizer to direct accounts for each branch to specific UOWs within specific areas. There is now a requirement to be able to monitor the amount of associated data in the UOWs (that is, when a range of UOWs is too small to hold the associated data, the data must be transferred to a larger interval). DEDB Pointer Checker addresses this requirement by optionally generating a UOW Detailed Analysis report and a UOW Exception report. These reports reflect free space, CI usage, and record statistics for each UOW. Monitoring is possible by user-specified reporting criteria (that is, report a UOW when DOVF free space is less than *nn*%). See the topic "DEDB Pointer Checker JCL examples for monitoring and tuning DEDBs" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Related reference

FABADA5 UOWRPT DD data set
The UOWRPT data set contains the report for UOW analysis produced by the FABADA5 program.

FABADA5 PARMCTL DD data set
The PARMCTL DD data set contains the FPB user's specification of execution control parameters used by module FABADA5.

Chapter 3. Tuning a DEDB offline with FPB

DEDB Tuning Aid is an extension of the DEDB Pointer Checker. This feature allows evaluation of the potential benefits, or impact, of changes to an existing DEDB without the time-consuming requirement to unload and reload the database.

Topics:

Functions of DEDB Tuning Aid

DEDB Tuning Aid facilitates multiple iterations of the database tuning process, thereby allowing you to select physical database attributes that meet performance and space utilization requirements.

Any or all of the following database specifications can be changed and evaluated with this program:

- Randomizing module
- UOW parameter values
- ROOT parameter values
- Number of database areas
- CI sizes.

DEDB Tuning Aid uses data previously extracted by the DEDB Pointer Checker using TYPRUN=MODEL. Both programs use the FABADA5 report generation module. This significantly simplifies database tuning evaluation and comparison.

DEDB Tuning Aid provides a comprehensive tuning facility for managing databases and tuning performance. It can help you accomplish the following tasks:

- Select database parameters and attributes with greater precision
- Identify and select physical database attributes that meet performance and space utilization requirements (that is, optimal values for the ROOT and UOW parameters)
- Simplify evaluation and selection of suitable randomizing routines
- Reduce maintenance frequency and costs
- Eliminate database maintenance required when a database, after an unload/reload, does not meet performance and/or space utilization expectations
- Evaluate whether tuning helps or hinders online performance
- Select optimal normal buffer allocation (NBA) for database reloading.

Data and system flow of DEDB Tuning Aid

The DEDB Tuning Aid utility uses the sorted and resolved data analysis records generated by the DEDB Pointer Checker, and a DMB that defines the database structure to be tuned.

If only the randomizing routine is being changed, the old DMB can be used and the new randomizer name can be supplied by means of a control statement parameter. Otherwise, a new DMB must be provided.

The first step reprocesses the data analysis records and, if required, substitutes the new AREA and RAP numbers. New analysis records are only generated for requested areas.

The second step uses the sorted segment information records to pseudo insert the segments into an area model, and the appropriate freespace, reference, and segment information records are generated.

The final step creates the result of the tuning process: a set of output reports formatted by the DEDB Pointer Checker report generation module.

DEDB Tuning Aid programs run in a virtual storage region as standard z/OS batch job steps. Region size varies with the size of a UOW and the number of IOVF CIs defined. Typical usage should start at 768K which includes requirements for the DEDB Tuning Aid programs, access methods and buffers, and report queues.

DEDB Tuning Aid contains three load modules. The following table lists these modules.

Table 21. Load modules of DEDB Tuning Aid

Load module name	Function
FABBME1	Extract/format segment data
FABBME3	Pseudo load area, generate records for reporting
FABADA5	Generate, format, and print reports

Processes of DEDB Tuning Aid

A typical DEDB Tuning Aid job contains five or six of the following steps:

Step 1. FABBME1
This FPB program sets up the environment required for invoking a randomizer. It reprocesses the data analysis records (that were generated by the DEDB Pointer Checker) substituting the new AREA and RAP numbers. It then generates analysis records for the requested areas. One output file containing all record types is created.

Step 2. DFSORT
The IBM Data Facility Sort Program Product 5740-SM1 (or its equivalent) sorts the analysis records created by FABBME1 (file MEDARO).

Step 3. FABBME3
This FPB program sets up an area model and pseudo inserts all segments into the model. It writes the area description, segment information, and free space records required by the reporting program to two output files:

- The first data set contains area description, segment definition, and free space information records.
- The second data set contains the segment information records.

Step 4. DFSORT
The IBM Data Facility Sort Program Product 5740-SM1 (or its equivalent) sorts the area description, segment definition, and free space information records created by FABBME3 (file MEDAR120).

Step 5. DFSORT
If more than one area is being processed, the IBM Data Facility Sort Program Product 5740-SM1 (or its equivalent) sorts the segment information records created by FABBME3 (file MEDAR30). Omit this step if only one area is requested. This is the only optional step.

Step 6. FABADA5
This FPB program uses the sorted files from the previous two steps to generate and print the analysis reports. Optionally, threshold analysis is performed, reorganization control statements are generated, and a history file is written.

The following figure shows the flow of the DEDB Tuning Aid procedure.
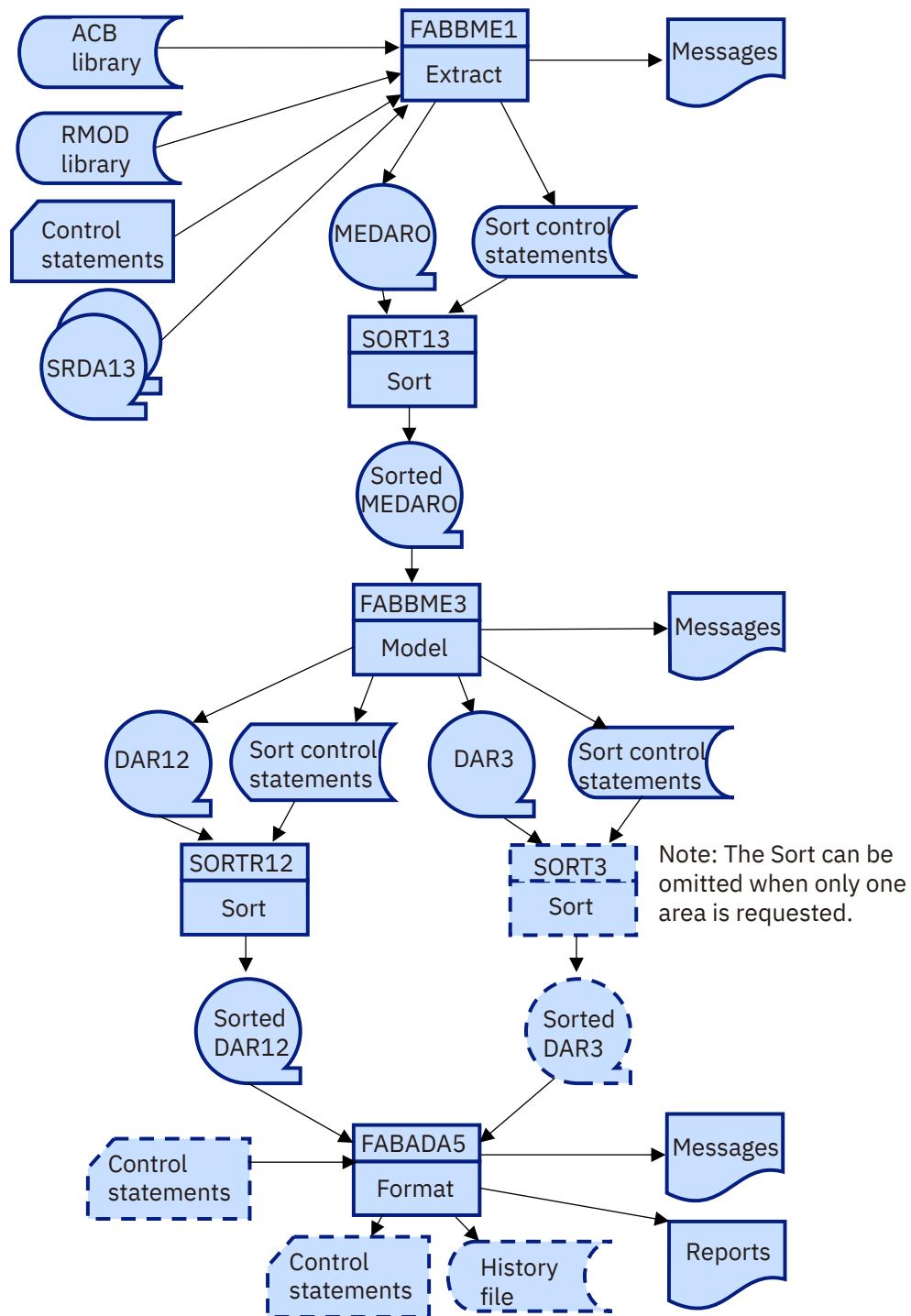
Figure 34. Flow of DEDB Tuning Aid

Running DEDB Tuning Aid

The DEDB Tuning Aid is used in relationship to the DEDB Pointer Checker, and therefore you must follow some steps to specify the control statement and JCL requirements.

Procedure

To run the DEDB Tuning Aid, you must complete the following tasks:

1. Select the DEDB areas to be tuned.

2. Run the DEDB Pointer Checker or the Online Pointer Checker for those areas to create input data that is required by the DEDB Tuning Aid.
3. Code the JCL for the DEDB Tuning Aid (FABBME1, FABBME3, and FABBME5) and DFSORT job steps to be run.
4. Select the database specifications to be changed and evaluated.
5. If you are changing parameters other than the randomizing routine, create a new DMB (that is, do a DBDGEN and ACBGEN).
6. Code the control statements needed for the FABBME1 and FABADA5 programs. The DEDB Tuning Aid creates all required DFSORT control statements.
7. Run the job.
8. Evaluate the output reports.
9. Repeat steps 4 through 8, as necessary, until you have enough information to decide how to tune your database.

DD statements for DEDB Tuning Aid

DD statements for DEDB Tuning Aid determine the input and output data sets and how DEDB Tuning Aid is run.

You must specify DD statements for the job control language (JCL) to run the DEDB Tuning Aid programs. There are space requirements for the various work data sets created by the DEDB Tuning Aid. Output data sets do not require DCB information to be specified in your JCL.

FABBME1 JCL

FABBME1 runs as a standard z/OS batch job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

The EXEC statement must be in the form:

```
//  EXEC  PGM=FABBME1,REGION=768K
```

DD statements

The following table summarizes the DD statements.

Table 22. FABBME1 DD statements

DDNAME	Use	Format	Required or optional
ACBLIB	Input	PDS	Required
RMODLIB	Input	PDS	Required
SYSIN	Input	LRECL=80	Required
MEDARI	Input		Required
SYSPRINT	Output	LRECL=133	Required
MESORTCD	Output	LRECL=80	Required
MEDARO	Output		Required

The following table shows space requirements for this JCL stream.

DDNAME	Space requirements
MEDARO	# segments X (28 + root key length + 4 X # levels) bytes
MESORTCD	< 1 track

ACBLIB DD

This statement defines the library that contains the DMB that describes the database configuration to be tuned. This would normally be a testing library, not IMSVS.ACBLIB. For example:

```
//ACBLIB DD DSN=IMSVS.TEST.ACBLIB,DISP=SHR
```

RMODLIB DD

This statement defines the library that contains the randomizing routine that is to be used.

SYSIN DD

This statement defines the input control statement data set. This data set can reside on a direct-access device or be routed through the input stream.

MEDARI DD

This statement defines the input data set(s) that contain the sorted and resolved data analysis records. These are the SORTOUT data sets from step SORTR13 of the DEDB Pointer Checker procedure.

Note: Data sets from different DEDB areas can be concatenated in any order. You can process as many areas as desired in a single DEDB Tuning Aid run.

SYSPRINT DD

This statement defines the output message and statistics data set. The data set can reside on a direct-access device or printer or be routed through the output stream. It is recommended that you code the following on your DD statement:

although the following code is also valid: RECFM=FBA,LRECL=133

```
//SYSPRINT DD SYSOUT=A
```

MESORTCD DD

This statement defines the output data set that contains the sort control statements required for step SORT13. This data set must reside on a direct-access device.

Space requirements can be satisfied with one track. DCB information is hard-coded in FABBME1. Do not specify DISP=MOD for this DD statement.

MEDARO DD

This statement defines the output data set for the area definition records, the segment definition records, and the segment records. The segment records contain details about the segment such as its RBA. The data set can reside on either a direct-access device or tape. Standard labels must be used for this data set.

The DCB attributes of this file are calculated by FABBME1. The RECFM is VB. For 3380, the default block size is 23476 bytes. For 3390, the default block size is 27998 bytes. For 9345, the default block size is 22928 bytes. For other output devices, the default block size is the maximum block size of the output device. Do not code any other DCB parameters in your JCL.

Space requirements vary depending on the number of segments in the tuned area(s), and the length of the root key. Approximation: size in bytes = number of segments X (28 + root key length + 4 X no. hierarchical levels in DMB). Do not specify DISP=MOD for this DD statement.

Related information

Input for DEDB Tuning Aid

To run the DEDB Tuning Aid, you must specify input definitions for each program.

DFSORT JCL (Step SORT13)

This reference topic summarizes the EXEC statement and DD statements for the DFSORT JCL (Step SORT13).

EXEC statement

The EXEC statement must be in the following form:

```
//       EXEC PGM=SORT
```

DD statements

The following table summarizes the DD statements needed to run DFSORT. All statements in this table are required.

Table 23. DFSORT DD statements (Step SORT13)

DDNAME	Use	Format	Required or optional
SORTIN	Input	LRECL=20	Required
SYSIN	Input		Required
SORTOUT	Output	LRECL=20	Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

The following table shows space requirements for this JCL stream.

DDNAME	Space requirements
SORTOUT	# segments X (28 + root key length + 4 X # levels) bytes

SORTIN DD
> This input data set is the MEDARO file from FABBME1.

SYSIN DD
> This input data set contains DFSORT control statements. It is the MESORTCD file from FABBME1.

SORTOUT DD
> This output data set contains the sorted records. Do not specify DISP=MOD for this DD statement.

SYSOUT DD
> This output data set contains the messages produced by DFSORT. It is used by FABBME3. Required space is the same size as the SORTIN data set.

SORTWK*nn* DD
> These are the intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about creating SORTWK*nn* DD statements.
>
> Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

FABBME3 JCL

FABBME3 runs as a standard z/OS batch job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

The EXEC statement must be in the form:

```
// EXEC  FABBME3,REGION=768K,PARM='x'
```

Region-size requirements vary depending on the number of DOVF CIs in a UOW, and the number of IOVF CIs in the area. REGION=768K should suffice for most databases.

FABBME3 requires a PARM= specification on the EXEC statement to specify the emulation mode desired: either "load" or "reorganization." Valid values for PARM= are L and R. If PARM= is not coded, or is specified as any other value, a default specification of L is assumed.

- A value of L requests generation of a model of the database that would be produced by loading its area, or areas, with the data in physical hierarchical sequence.
- A value of R simulates the effect of running the High- Speed Direct Reorganization Utility (DBFUMDR0) against the database.

DD statements

The following table summarizes the DD statements.

Table 24. FABBME3 DD statements

DDNAME	Use	Format	Required or optional
MEDARI	Input		Required
SYSPRINT	Output	LRECL=133	Required
MEDAR12O	Output		Required
MEDAR3O	Output		Required
SORT12	Output	LRECL=80	Required
SORT3	Output	LRECL=80	Required

The following table shows space requirements for this JCL stream.

DDNAME	Space requirements
MEDAR12O	# CI/area X 1.5 X 20 bytes
MEDAR3O	# segments X (30 + 4 X # levels)
SORT12	< 1 track
SORT3	< 1 track

MEDARI DD
This statement defines the input data set that contains the sorted area and segment information records from SORT13 step.

SYSPRINT DD
This statement defines the output message and statistics data set. The data set can reside on a direct-access device or printer, or be routed through the output stream. You can code RECFM=FBA,LRECL=133 on your DD statement, but it is better to use:

```
//SYSPRINT DD SYSOUT=A
```

MEDAR12O DD

This statement defines the output data set for the area and free space information records. The data set can reside on either a direct-access device or tape. Standard labels must be used for this data set.

DCB attributes are calculated by FABBME3. RECFM is VB. For 3380, the default block size is 23476 bytes. For 3390, the default block size is 27998 bytes. For 9345, the default block size is 22928 bytes. If LRECL is larger than the above block size, then this LRECL is used. For other output devices, the default block size is the maximum block size of the output device. A block size can be specified on the DD statement. Do not code any other DCB parameters in your JCL.

Space requirements vary depending on the number of CIs in the tuned area(s), and the number of intersection references. Approximation: number of CIs X 1.5 X 20 bytes. Do not specify DISP=MOD for this DD statement.

Note: The number of CIs includes RAA BASE, DOVF, and IOVF only.

MEDAR3O DD

This statement defines the output data set for the segment information records. The data set can reside on either a direct-access device or tape. Standard labels must be used for this data set.

DCB attributes are calculated by FABBME3. RECFM is VB. For 3380, the default block size is 23476 bytes. For 3390, the default block size is 27998 bytes. For 9345, the default block size is 22928 bytes. If LRECL is larger than the above block size, then this LRECL is used. For other output devices, the default block size is the maximum block size of the output device. A block size can be specified on the DD statement. Do not code any other DCB parameters in your JCL.

Space requirements vary depending on the hierarchical structure of the database, and the number of segments in the tuned area(s). Approximation: no. segments X (30 + 4 X no. hierarchical levels in DMB). Do not specify DISP=MOD for this DD statement.

SORT12

This statement defines the output data set that contains the SORT control statements required for step SORT12. This data set must reside on a direct-access device.

Space requirements are very small (1 track suffices). Do not specify DISP=MOD for this DD statement.

DCB information is hard-coded in FABBME3. Do not code DCB information in your JCL.

SORT3

This statement defines the output data set that contains the SORT control statements required for step SORT3. This data set must reside on a direct-access device.

Space requirements are very small (1 track suffices). Do not specify DISP=MOD for this DD statement.

DCB information is hard-coded in FABBME3. Do not code DCB information in your JCL.

DFSORT JCL (Step SORT12)

This program sorts the MEDAR12O data set that is created by FABBME3. The sorted data set is used by FABADA5.

EXEC statement

The EXEC statement must be in the following form:

```
//      EXEC PGM=SORT
```

DD statements

To run DFSORT, you must supply the appropriate DD statements. The following table summarizes the DD statements. Actual JCL requirements are as follows:

Table 25. DFSORT DD statements (Step SORT12)

DDNAME	Use	Format	Required or optional
SORTIN	Input	LRECL=20	Required
SYSIN	Input		Required
SORTOUT	Output	LRECL=20	Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

The following table shows space requirements for this JCL stream.

DDNAME	Space requirements
SORTOUT	# CI/area X 1.5 X 20 bytes

SORTIN DD
> This input data set is the MEDAR12O file from FABBME3.

SYSIN DD
> This input data set contains DFSORT control statements. It is the SORT12 file from FABBME3.

SORTOUT DD
> This output data set contains the sorted records. It is used by FABADA5. Required space is the same size as the SORTIN data set. Do not specify DISP=MOD for this DD statement.

SYSOUT DD
> This output data set contains the messages produced by DFSORT.

SORTWK*nn* DD
> These are the intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about creating SORTWK*nn* DD statements.
>
> Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

DFSORT JCL (Step SORT3)

This program sorts the MEDAR3O data set that is created by FABBME3. Sorted data set is used by FABADA5.

EXEC statement

This statement must be in the following form:

```
//      EXEC PGM=SORT
```

DD statements

To run DFSORT, you must supply the appropriate DD statements. The following table summarizes the DD statements. Actual JCL requirements are as follows:

Table 26. DFSORT DD statements (Step SORT3)

DDNAME	Use	Format	Required or optional
SORTIN	Input	LRECL=20	Required

Table 26. DFSORT DD statements (Step SORT3) (continued)

DDNAME	Use	Format	Required or optional
SYSIN	Input		Required
SORTOUT	Output	LRECL=20	Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required

The following table shows space requirements for this JCL stream.

DDNAME	Space requirements
SORTOUT	# segments X (30 + 4 X # levels)

SORTIN DD
> This input data set is the MEDAR3O file from FABBME3.

SYSIN DD
> This input data set contains DFSORT control statements. It is the SORT3 file from FABBME3.

SORTOUT DD
> This output data set contains the sorted records. It is used by FABADA5. Required space is the same size as the SORTIN data set. Do not specify DISP=MOD for this DD statement.

SYSOUT DD
> This output data set contains the messages produced by DFSORT.

SORTWK*nn* DD
> These are the intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about creating SORTWK*nn* DD statements.
>
> Allocating twice the space used by the SORTIN data set is usually adequate for each work data set.

FABADA5 JCL

The FABADA5 program uses the area and free space information generated by FABADA1 and the segment and pointer information resolved by FABADA3 (sorted) to generate and print the analysis reports.

Optionally, it can perform threshold analysis, generate control statements for the High-Speed DEDB Direct Reorganization utility, or generate History and History2 file entries. The History file entry contains the key space utilization and performance values for each area. The History2 file entry contains the history data that is totaled by UOW range. These files allow you to do statistical or trend analysis using user-written programs. For more information about History and History2 file records, see the topic "Historical records layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide.*

FABADA5 is run as a standard z/OS job step. An EXEC statement and DD statements that define inputs and outputs are required.

EXEC statement

The EXEC statement must be in the form:

```
//      EXEC PGM=FABADA5,REGION=768K
```

DD statements

The following table summarizes the DD statements.

For the required output data sets as determined by the FABADA1 TYPRUN= value, see Table 4 on page 17.

Table 27. FABADA5 DD statements

DDNAME	Use	Format	Required or optional
DAR12I or DAR12	Input		Required
DAR13I or DAR13R	Input		Required (See note)
PARMCTL	Input	LRECL=80	Optional
THRCTL	Input	LRECL=80	Optional
UOWCTL	Input	LRECL=80	Optional
SYSPRINT	Output	LRECL=133	Required
REPORTS	Output	LRECL=133	Required
UOWRPT	Output	LRECL=133	Optional
UOWEXCP	Output	LRECL=133	Optional
REORGCTL	Output	LRECL=80	Optional
HISTORY	Output	DISP=MOD	Optional
HISTORY2	Output	DISP=MOD	Optional
LARGEREC	Output	LRECL=296	Optional

Note: This data set is required when TYPRUN=RPT or TYPRUN=MODEL is specified in FABADA1.

DAR12I DD

This statement defines the input data set that contains free space and area information for report generation. It is the SORTOUT data set from SORT12.

If this DD statement is not specified in the JCL statement, the DAR12 DD statement is used as an alternative DD.

DAR12 DD

If DAR12I DD is not specified in the JCL statement, this statement is used instead of the DAR12I DD statement.

DAR13I DD

This statement defines the input data set that contains segment information for report generation. It is the SORTOUT data set from SORTR13.

If this DD statement is not specified in the JCL statement, the DAR13R DD statement is used as an alternative DD.

Note: This data set is required when TYPRUN=RPT or TYPRUN=MODEL is specified in FABADA1.

DAR13R DD

If DAR13I DD is not specified in the JCL statement, this statement is used instead of the DAR13I DD statement.

Note: This data set is required when TYPRUN=RPT or TYPRUN=MODEL is specified in FABADA1.

PARMCTL DD

This statement defines the optional input parameter control statement data set. This data set can reside on a direct-access device, or be routed through the input stream. This DD statement is optional. It may be omitted, or specified as:

```
//PARMCTL DD DUMMY
```

THRCTL DD
This statement defines the optional input threshold control statement data set. This data set can reside on a direct-access device, or be routed through the input stream. This DD statement is optional. It may be omitted, or specified as:

```
//THRCTL DD DUMMY,DCB=BLKSIZE=80
```

UOWCTL DD
This statement specifies the range of UOW of the History2 file record. This data set can reside on a direct-access device, or it can be routed through the input stream. This DD statement is optional, but if it is specified, the HISTORY2 DD statement is required. The HISTORY2 DD statement can be omitted when this DD statement is not specified, or is specified as:

```
//UOWCTL DD DUMMY
```

SYSPRINT DD
This statement defines the output message data set. The data set can reside on tape, a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

REPORTS DD
This statement defines the output analysis reports data set. The data set can reside on tape, a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//REPORTS DD SYSOUT=A
```

UOWRPT DD
This statement defines the optional output data set for the UOW report. This data set can reside on tape, direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//UOWRPT DD SYSOUT=A
```

UOWEXCP DD
This statement defines the optional output data set for the UOW Exception report. This data set can reside on tape, a direct-access device or printer, or be routed through the output stream. You should code your DD statement as follows:

```
//UOWEXCP DD SYSOUT=A
```

Entries are generated for this report in response to user-specified reorganization criteria.

REORGCTL DD
This statement defines the optional output data set for the reorganization control statements that may be generated for input to the High-Speed Direct Reorganization utility.

```
(RECFM=FB, LRECL=80, BLKSIZE= must be specified)
```

Do not specify DISP=MOD for this DD statement.

Note: Generating reorganization control statements for the High-Speed Direct Reorganization utility requires the inclusion of the REORGCTL DD statement in the JCL stream.

For additional requirements, see *Note* under .

HISTORY DD
This statement defines the output History file data set. This data set can reside on either a direct-access device or a tape. This DD statement is optional. If it is not present in the JOB stream, a History file record is not created or journaled. If the DD statement is present in the JOB stream, a History file

record is generated. If this feature is used, the History file must be pre-allocated with the following attributes:

```
DSORG=PS
RECFM=FB
LRECL=100
BLKSIZE=user-specified
```

The DD statement in the JOB stream is specified as:

```
//HISTORY DD DSN=HPFP.DA.HISTORY,
//           DISP=(MOD,KEEP,KEEP)
```

The following is also valid:

```
//HISTORY DD DUMMY,DCB=BLKSIZE=100
```

HISTORY2 DD
This statement defines the output History2 file data set. This data set can reside on either a direct-access device or a tape. This DD statement is optional.

If UOWCTL DD is specified and it is not dummy, the HISTORY2 DD statement is required. If only the UOWCTL DD is specified, and the HISTORY2 DD statement is not specified, FABADA5 ends with FABA3565E message and an abend code of 3565. If the DD statement is not specified in the JOB stream, a History2 file record will not be created or journaled. If the DD statement is specified in the JOB stream, a History2 file record will be generated. When you use this feature, the History2 file must be pre-allocated with the following attributes:

```
DSORG=PS
RECFM=FB
LRECL=200
BLKSIZE=user-specified
```

The DD statement in the JOB stream is specified as:

```
//HISTORY2 DD DSN=HPFP.DA.HISTORY2,
//            DISP=(MOD,KEEP,KEEP)
```

The following syntax is also valid:

```
//HISTORY2 DD DUMMY
```

LARGEREC DD
This statement defines the optional data set for the largest database records. This statement is effective only when TYPRUN=MODEL is selected in the FABADA1 step.

This DD statement is optional. It may be omitted, or specified as:

```
//LARGEREC  DD  DUMMY
```

Do not code DCB information in your JCL.

For more information about largest database records, see the topic "Largest database records layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Related reference
FABADA5 PARMCTL DD data set

The PARMCTL DD data set contains the FPB user's specification of execution control parameters used by module FABADA5.

Input for DEDB Tuning Aid

To run the DEDB Tuning Aid, you must specify input definitions for each program.

All input that the user must specify to run the DEDB Tuning Aid utility is described in detail in the following topics.

FABBME1 MEDARI DD data set

The FABBME1 MEDARI DD data set is the primary input to the DEDB Tuning Aid. It is the sorted and resolved segment-information-records data set from Step SORTR13 of the DEDB Pointer Checker procedure.

The detail of the procedure is shown in . For convenience, these records are also referred to as *SRDA13 data*.

DEDB Pointer Checker considerations

When the DEDB Tuning Aid is to be used, you must retain the SRDA13 data from the DEDB Pointer Checker run(s) for the area(s) involved. You must also use TYPRUN=MODEL on your DEDB Pointer Checker FABADA1 control statement.

Randomizing routine considerations

The number of areas for which SRDA13 data is required depends on the database specification being changed and the characteristics of the randomizing routine.

For purposes of this document, randomizers are classified as follows:

1. Area-specific: Defines the user-written randomizer that controls an area or area group; its selection is based on root segment key ranges or a similar technique.

   EXAMPLE: Keys starting with "A" through "L" are randomized across areas 1 through 5. Keys starting with "M" through "Z" are randomized across areas 6 through 10.

2. Nonspecific: Defines a randomizer where the area selection is not controlled by key values. DBFHDC40, as delivered with IMS Fast Path, is nonspecific.

Notes:

- The randomizer module interface environment created by FABBME1 conforms to the *published* interface as defined in *IMS Exit Routines* with one exception. When a randomizer is invoked by IMS, registers 10 and 11 contain the addresses of the EPST and ESCD, respectively. Some user-written or user-customized randomizers have been designed to use these addresses to gain access to IMS control blocks other than those passed as part of the published interface. When FABBME1 issues a call to a randomizing module, registers 10 and 11 are set to -1 and 0, respectively, to specify that the call is not being issued in a *live* IMS environment.

- The randomizer module must be capable of AMODE=31.

The areas for which you must provide SRDA13 data are a function of the kind of changes that you want to model. The following table shows how to select the right areas.

Table 28. DEDB Tuning Aid input requirements

Specification changed	Nonspecific randomizer	Area-specific randomizer
Randomizing routine	SRDA13 data for all areas	SRDA13 data for the specific area group if the area selection criteria is not changed

Table 28. DEDB Tuning Aid input requirements (continued)

Specification changed	Nonspecific randomizer	Area-specific randomizer
ROOT or UOW values	SRDA13 data for all areas	SRDA13 data for the specific area group
CI size only	SRDA13 data for the specific area being tuned	SRDA13 data for the specific area being tuned
Number of areas	SRDA13 data for all areas	SRDA13 data for all areas
None (that is, tune the Direct Reorganization utility)	SRDA13 data for the specific area being tuned	SRDA13 data for the specific area being tuned

FABBME1 SYSIN DD data set

The SYSIN data set contains the FPB user's description of the processing to be done by FABBME1. It describes the database and areas to be tuned.

Format

This control statement data set usually resides in the input stream. However, it can also be described as a sequential data set or as a member of a partitioned data set. It must contain 80-byte, fixed-length records. BLKSIZE, if coded, must be a multiple of 80.

This data set must contain only one control statement. It can be coded as shown in the following figure.

```
//SYSIN DD *
   DBDNAME=VRSDSRF,AREA=ALL,RMOD=R3DFN,RPT=YES
/*
```

Figure 35. FABBME1 SYSIN DD data set

Syntax of the control statement

FABBME1 requires a control statement. Keywords, and their associated parameter values, may be coded in free format (columns 1-72) provided certain coding syntax rules are followed:

1. All keyword/parameter-value sets must be contained on one control statement.
2. All keywords must be separated from their associated parameter values by an equal sign. The equal sign must not be preceded by blanks, but may be followed by one or more blanks.
3. Parameter values must be separated from the next keyword by one or more blanks, a comma, or a comma followed by one or more blanks.
4. Duplicate keywords are resolved on the basis that the last one coded is the value used.

Record format

There is only one statement type in the SYSIN file. It contains the following keywords:

```
DBDNAME=dbdname
AREA={n|ALL|(i,j,...)|(a-b,c-d,...)}
[RMOD=randomizer]
[RPT=YES]
```

where:

DBDNAME=
This required keyword specifies the name of the ACBLIB member that is the DMB for the DEDB database being tuned.

AREA=
This required keyword specifies the number of the areas to be tuned. Data analysis records and reports are generated only for the areas specified.

Only one format of the area keyword can be specified. The area numbers must exist within the DMB.

AREA= *n*
> Specifies the specific area number of the requested area.

```
EXAMPLE:   AREA=3
```

AREA=ALL
> Specifies that data analysis records and reports are to be generated for all areas of the database.

AREA=(*i,j, ...* **)**
> Specifies the specific area numbers of the requested areas. Numbers do not need to be contiguous. Parentheses are required.

```
EXAMPLE:   AREA=(1,3,5)
```

AREA=(*a-b,c-d,...* **)**
> Specifies specific ranges of requested area numbers. Parentheses are required.

```
EXAMPLE:   AREA=(1-3,7-9)
```

Area numbers *n, i, j, a, b, c,* and *d* are decimal numbers in the range of 1 - 2048.

RMOD=
> This optional keyword specifies the name of the randomizing routine to be used when it differs from that specified in the DMB. This name always overrides the name specified in the DMB. It is typically used when various randomizers are being tested, to avoid the need for DBD and ACB generation for each of the randomizers.

RPT=YES
> This optional keyword specifies that the detailed record count report is to be printed.

FABADA5 PARMCTL DD data set

The PARMCTL DD data set contains the FPB user's specification of execution control parameters used by module FABADA5.

Syntax of the control statement

FABADA5 may require one control statement.

Keywords and the associated values can be coded in free format (columns 1 - 72), provided certain syntactical coding rules are followed:

1. The keyword and its value must be on one control statement.

2. If specification of a keyword value is required, the keyword must be separated from its associated value by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or more blanks. The value must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.

3. For keywords that do not have associated values, the keyword must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.

4. In case of duplicate keywords, the last one coded is used.

Format

This control-statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte fixed-length records. Block size, if coded, must be a multiple of 80.

This data set must contain only one (optional) control statement. It can be coded as shown in the following figure:

```
//PARMCTL DD *
  RDOVFFS=40 SYNTBL=20 UDOVFFS=35 RBASEFS=40
/*
```

Figure 36. FABADA5 PARMCTL DD data set

Record format

There is only one statement type in the PARMCTL file. It contains the following keywords:

```
[SYNTBL={41|nn}]
[RBASEFS={100|nn}]
[RDOVFFS={100|nn}]
[UDOVFFS={0|nn}]
[MAXLREC={20|nnnn}]
[ARDIGIT={3|5}]
[RGROUP=xxxx]
```

SYNTBL=
   This keyword controls the number of reporting entries in the Synonym Chain Analysis table. The default value is 41 (that is, all synonym chains longer than 40 are grouped and reported together). The maximum value is 999, and the minimum value is 12.

RBASEFS=
   If a RAP CI contains free space greater than the percentage specified, and the RAP CI also uses overflow CIs, a reorganization control statement is generated for the UOW, and the UOW is included in the UOW Exception report. Default is 100% (that is, never generate reorganization control statements). The maximum value is 99, and the minimum value is 0.

RDOVFFS=
   If the DOVF part of a UOW contains free space greater than the percentage specified, and the UOW also uses IOVF CIs, a reorganization control statement is generated for the UOW and the UOW is included in the UOW Exception report. Default is 100% (that is, never generate reorganization control statements). The maximum value is 99, and the minimum value is 0.

UDOVFFS=
   If the DOVF UOW contains free space less than the percentage specified, the UOW is included in the UOW Exception report. Default is 0% (that is, never include in UOW Exception report). The maximum value is 99, and the minimum value is 0.

MAXLREC=
   This keyword controls the number of reporting entries in the Largest Database Records report. This statement is effective only when TYPRUN=MODEL is selected in the FABADA1 step. The default value is 20. The maximum value is 9999, and the minimum value is 0. When the minimum value 0 is specified, neither the Largest Database Records report nor the LARGEREC DD data set records are generated.

ARDIGIT=
   This keyword specifies the digit of the area number field for the UOW group definition records that are defined in the UOWCTL file. The value is 3 or 5. The default is 3. If the UOWCTL file is not specified on JCL statement, this keyword will be ignored.

RGROUP=
   This keyword is used to fill the value in the Repository Group field in the History2 file. There is no default value. If this keyword is not specified, FABADA5 will fill the Repository Group field with 4 bytes of blank characters. If the UOWCTL file is not specified on JCL statement, this keyword will be ignored.

Note: Generating reorganization control statements for the High-Speed DEDB Direct Reorganization utility requires specification of the free space control percentages.

For additional requirements, see *Note* under REORGCTL DD statement for the .

FABADA5 THRCTL DD data set

The THRCTL DD data set contains an optional control statement used by module FABADA5. It is used to specify threshold parameter values, which cause warning messages to be generated when the actual values do not meet the specified criteria.

Syntax of the control statement

FABADA5 may require one control statement.

Keywords and the associated values can be coded in free format (columns 1 - 72), provided certain syntactical coding rules are followed:

1. The keyword and its value must be on one control statement.
2. If specification of a keyword value is required, the keyword must be separated from its associated value by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or more blanks. The value must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.
3. For keywords that do not have associated values, the keyword must be separated from the next keyword by a blank, a comma, or a comma followed by one or more blanks.
4. In case of duplicate keywords, the last one coded is used.

Format

This control-statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte fixed-length records. Block size, if coded, must be a multiple of 80.

This data set must contain only one control statement. It can be coded as shown in the following figure:

```
//THRCTL    DD *
T5002 02004 01202 02004 101010 80 02004 20 050050050 101010 1010
/*
```

Figure 37. FABADA5 THRCTL DD data set

Record format

There is only one record type in the THRCTL file:

- The control statement is optional.
- Only 1 control statement is used. If multiple statements are provided, only the first statement is used. The additional statements are ignored.
- If a control statement is present, it must contain a "T" in position 1.
- All threshold parameter values are optional (that is, they can be blank).
- Only non-blank threshold parameter values are tested.
- Threshold parameters can be used to monitor the following:

    1. Synonym Chain Length:

        xx% of the synonym chains are > *yy* in length.
        average length > *xx.x*
        maximum length > *yy*

    2. Root I/O:

        average I/O > *xx.x*
        maximum I/O > *yy*

    3. Record I/O:

average I/O > *xx.x*
maximum I/O > *yy*

4. Overflow usage:
    - available IOVF CIs < *xx*%
    - RAP CIs using overflow > *xx*%
    - UOWs using DOVF > *xx*%
    - UOWs using IOVF > *yy*%
    - number of IOVF CIs used by a UOW

        average > *xx.x*
        maximum > *yy*
    - number of records using IOVF CIs > *xx*%

5. Percent free space:

    total RAA BASE free space < *xx*%
    total DOVF free space < *yy*%
    total IOVF free space < *zz*%

6. Free space relationships:

    RAA BASE > *xx*% and DOVF< *yy*% or IOVF < *zz*%
    DOVF > *xx*% and IOVF < *yy*%

The format of the threshold parameter control statement is as follows:

Note: Leading and trailing zeros can be omitted.

POSITION
  DEFINITION

1
  T is a mandatory value.

2 - 5
  xxyy - Synonym Chain Length

  xx
    The percentage of the synonym chains that exceed the specified length. The maximum value is 99, and the minimum value is 0.

  yy
    The default length of the maximum chain value is 40. The maximum value is 99, and the minimum value is 0. The value must be less than the value that is specified by SYNTBL in the PARMCTL DD data set.

  Both values must be present. A warning message is issued if xx% of the synonym chains exceed length *yy*.

6
  Not used.

7 - 9
  xxx - Average Synonym Chain Length. There is an implied decimal point before the third digit. A warning message is issued if the average synonym chain length is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

10 - 11
  xx - Maximum Synonym Chain Length. A warning message is issued if the maximum synonym chain length is greater than *xx*. The maximum value is 99, and the minimum value is 0.

12
  Not used.

13 - 15

xxx - Average Root I/O. There is an implied decimal point before the third digit. A warning message is issued if the average number of I/O to access a root segment is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

16 - 17

xx - Maximum Root I/O. A warning message is issued if the maximum number of I/O to access a root segment is greater than *xx*. The maximum value is 99, and the minimum value is 0.

18

Not used.

19 - 21

xxx - Average Record I/O. There is an implied decimal point before the third digit. A warning message is issued if the average number of I/O to access a database record is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

22 - 23

xx - Maximum Record I/O. A warning message is issued if the maximum number of I/O to access a database record is greater than *xx*. The maximum value is 99, and the minimum value is 0.

24

Not used.

25 - 26

xx - Percentage of RAA BASE CIs that use either DOVF or IOVF. A warning message is issued if the percentage of RAA BASE CIs that use overflow is greater than *xx*%. The maximum value is 99, and the minimum value is 0.

27 - 28

xx - Percentage of UOWs that use DOVF. A warning message is issued if the percentage of UOWs that use DOVF is greater than *xx*%. The maximum value is 99, and the minimum value is 0.

29 - 30

xx - Percentage of UOWs using IOVF. A warning message is issued if the number of UOWs using IOVF is greater than *xx*% of the total UOWs. The maximum value is 99, and the minimum value is 0.

31

Not used.

32 - 33

xx - Percentage of Available IOVF CIs. A warning message is issued if the available IOVF CIs are less than *xx*% of the total IOVF CIs. The maximum value is 99, and the minimum value is 0.

34

Not used.

35 - 37

xxx - Average Number of IOVF CIs used by UOW. There is an implied decimal point before the third digit. A warning message is issued if the average number of CIs used by a UOW is greater than *xx.x*. The maximum value is 999, and the minimum value is 0.

38 - 39

xx - Maximum Number of IOVF CIs used by UOW. A warning message is issued if the maximum number of IOVF CIs used by a UOW is greater than *xx*. The maximum value is 99, and the minimum value is 0.

40

Not used.

41 - 42

xx - Percentage of Records that use IOVF CIs. A warning message is issued if the number of records using IOVF CIs is greater than *xx*%. The maximum value is 99, and the minimum value is 0.

43

Not used.

44 - 46

xxx - Percentage of the free space in RAA BASE. There is an implied decimal point before the third digit. A warning message is issued if the free space in RAA BASE is less than *xxx*% of total space. The maximum value is 999, and the minimum value is 0.

47 - 49

xxx - Percentage of the free space in DOVF. There is an implied decimal point before the third digit. A warning message is issued if the free space in DOVF is less than *xxx*% of total space. The maximum value is 999, and the minimum value is 0.

50 - 52

xxx - Percentage of free space in IOVF. There is an implied decimal point before the third digit. A warning message is issued if the free space in IOVF is less than *xxx*% of total space. The maximum value is 999, and the minimum value is 0.

53

Not used.

54 - 59

xxyyzz - Free space relationships between RAA BASE, DOVF, and IOVF.

xx

The percentage of free space in RAA BASE. The maximum value is 99, and the minimum value is 0.

yy

The percentage of free space in DOVF. The maximum value is 99, and the minimum value is 0.

zz

The percentage of free space in IOVF. The maximum value is 99, and the minimum value is 0.

A value for xx must be present. Either *yy* or *zz* must be present. Both *yy* and *zz* may be present.

A warning message is issued under the following conditions:

- When RAA BASE free space is greater than xx% and DOVF free space is less than *yy*%.
- When RAA BASE free space is greater than xx% and IOVF free space is less than *zz*%.

60

Not used.

61 - 64

xxyy - Free space relationship between DOVF and IOVF.

xx

The percentage of free space in DOVF. The maximum value is 99, and the minimum value is 0.

yy

The percentage of free space in IOVF. The maximum value is 99, and the minimum value is 0.

Both values must be present. A warning message is issued if the DOVF free space is greater than xx% and if the IOVF free space is less than *yy*%.

65 - 72

Not used.

Output for DEDB Tuning Aid

The primary output from the DEDB Tuning Aid is the standard set of analysis reports from module FABADA5.

For more information, read about DEDB Pointer Checker.

Related reference

FABADA5 REPORTS DD data set

The REPORTS data set contains most of the reports produced by the FABADA5 program.

FABBME1 SYSPRINT DD data set

The SYSPRINT data set contains the messages issued by the FABBME1 program.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

Extract/Format-Messages

The following figure shows an example of the "Messages" report.

```
IMS HPFP UTILITIES - DEDBTA                            "EXTRACT/FORMAT - MESSAGES"                          PAGE:    1
5698-FPP                                            DATE: 11/22/2020  TIME:  20.29.38               FABBME1  -  V2R1


    FABB0101I - PROCESSING COMPLETE FOR DBDNAME : VRSDSRF

    FABB0102I -   - RECORDS READ FROM  MEDARI  :      5,671
                       (- TYPE "1,0, "         :          3)
                       (- TYPE "1,1, "         :         12)
                       (- TYPE "3 RESOLVED"    :      5,656)
                       (- OTHER                :          0)

    FABB0102I -   - RECORDS WRITTEN TO MEDARO  :      5,671
                       (- TYPE "1,0, "         :          3)
                       (- TYPE "1,1, "         :         12)
                       (- TYPE "3, , "         :      5,656)

    FABB0103I -   - DEDB ANALYZER RECORDS READ FOR AREAS:
                          1   2   3

    FABB0104I -   - DEDB MODELER RECORDS WRITTEN FOR AREAS:
                          1   2   3

    FABB0105I - SORT CONTROL CARDS:   SORT FIELDS=(5,0044,CH,A),SIZE=00005671
                                      RECORD TYPE=V,LENGTH=(,,,00054,00054)

    FABB0100I - FABBME1  ENDED NORMALLY
```

Figure 38. Extract/Format-Messages

FABBME3 SYSPRINT DD data set

The SYSPRINT data set contains the messages issued by the FABBME3 program.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. You should code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

DEDBTA Pseudo load area—Messages

The following figure shows an example of the Messages report.

```
IMS HPFP UTILITIES - DEDBTA                     "PSEUDO LOAD AREA - MESSAGES"                          PAGE:    2
5698-FPP                                    DATE: 11/22/2020  TIME: 20.29.39                       FABBME3  -  V2R1


     FABB0301I - PROCESSING STARTED FOR DBDNAME: VRSDSRF
                                AREA   : VRSTSS1


     FABB0302I - NUMBER OF RECORDS WRITTEN FOR THE AREA:
                 DDNAME  MEDAR120         262
                     (1,0       :         1)
                     (1,1       :         4)
                     (2,T       :         1)
                     (2,F       :       157)
                     (2,R       :        99)

                 DDNAME  MEDAR30        1,997
                     (3,S       :     1,997)


     FABB0301I - PROCESSING STARTED FOR DBDNAME: VRSDSRF
                                AREA   : VRSTSS2


     FABB0302I - NUMBER OF RECORDS WRITTEN FOR THE AREA:
                 DDNAME  MEDAR120         453
                     (1,0       :         1)
                     (1,1       :         4)
                     (2,T       :         1)
                     (2,F       :       261)
                     (2,R       :       186)

                 DDNAME  MEDAR30        1,835
                     (3,S       :     1,835)


     FABB0301I - PROCESSING STARTED FOR DBDNAME: VRSDSRF
                                AREA   : VRSTSS3


     FABB0302I - NUMBER OF RECORDS WRITTEN FOR THE AREA:
                 DDNAME  MEDAR120         248
                     (1,0       :         1)
                     (1,1       :         4)
                     (2,T       :         1)
                     (2,F       :       159)
                     (2,R       :        83)

                 DDNAME  MEDAR30        1,824
                     (3,S       :     1,824)

     FABB0303I - SORT CONTROL CARDS WRITTEN TO DDNAME: SORT12
                 SORT FIELDS=(5,13,CH,A),SIZE=00000963
                 RECORD TYPE=V,LENGTH=(,,,020,020)

     FABB0303I - SORT CONTROL CARDS WRITTEN TO DDNAME: SORT3
                 SORT FIELDS=(5,29,CH,A),SIZE=00005656
                 RECORD TYPE=V,LENGTH=(,,,0046,0046)

     FABB0300I - FABBME3 ENDED NORMALLY
```

Figure 39. DEDBTA Pseudo load area—Messages

Examples for DEDB Tuning Aid

There are many ways to run DEDB Tuning Aid. The examples provided in the following topics show some of the typical ways that you can use.

The examples presented in this topic represent some common actions that can be performed by DEDB Tuning Aid. These examples can assist you in learning techniques to help manage your IMS DEDBs effectively.

Example 1: Tuning the DEDB using DEDB Tuning Aid

The following figure shows example JCL statement for tuning the DEDB using DEDB Tuning Aid.

DEDB Tuning Aid is used to tune any database specification changes. For large databases, it is usually advisable to request the output for only one of the areas being tuned. This is primarily related to the potential size of the files created, and the sort work data set requirements. If the tuned area looks promising, the procedure can be rerun for the remaining areas.

Note: If DEDB Tuning Aid output is only requested for one area, the SORT step for segment information records (step SORT3) can be omitted.

The following figure shows a typical sample DEDB Tuning Aid JCL.

```
//FABBME1  EXEC PGM=FABBME1,REGION=768K
//**********************************
//** REPLACE AREA AND RAP NO'S.    **
//**********************************
//SYSPRINT DD SYSOUT=A
//ACBLIB   DD DSN=IMSVS.TEST.ACBLIB,DISP=SHR
//RMODLIB  DD DSN=IMSVS.PGMLIB,DISP=SHR
//MEDARI   DD DSN=HPFP.TSSDA.SRDA13,DISP=OLD
//MESORTCD DD DSN=&&SORT13,
//            DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
//MEDARO   DD DSN=HPFP.TSSME.ME13,
//            DISP=(NEW,CATLG),
//            UNIT=SYSDA,
//            SPACE=(CYL,(35,5),RLSE)
//SYSIN    DD *,DCB=BLKSIZE=80
     DBDNAME=TSSDBD,AREA=(1-3),RPT=YES
/*
//SORT13   EXEC PGM=SORT,PARM='CORE=MAX',COND=(4,LT),
//            REGION=2048K
//**********************************
//** SORT ME13 RECORDS             **
//**********************************
//SYSOUT   DD SYSOUT=A
//SYSIN    DD DSN=&&SORT13,DISP=(OLD,DELETE)
//SORTIN   DD DSN=HPFP.TSSME.ME13,
//            DISP=(OLD,DELETE,KEEP)
//SORTOUT  DD DSN=HPFP.TSSME.SME13,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(35,5),RLSE)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//*
//FABBME3  EXEC PGM=FABBME3,PARM='L',REGION=768K,COND=(4,LT)
//**********************************
//** BUILD AREA MODEL              **
//**********************************
//SYSPRINT DD SYSOUT=A
//MEDARI   DD DSN=HPFP.TSSME.SME13,
//            DISP=(OLD,DELETE,KEEP)
//MEDAR12O DD DSN=HPFP.TSSME.DA12,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(15,2),RLSE),
//            DCB=BLKSIZE=23000
//MEDAR3O  DD DSN=HPFP.TSSME.DA3,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,5),RLSE),
//            DCB=BLKSIZE=23000
//SORT12   DD DSN=&SORT12,
//            DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
//SORT3    DD DSN=&SORT3,
//            DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
//*
```

Figure 40. Sample JCL for DEDB Tuning Aid (Multi-area output) (Part 1 of 2)

```
//SORT12   EXEC PGM=SORT,PARM='CORE=MAX',COND=(4,LT),
//            REGION=2048K
//**********************************
//**   SORT MODELLER DA12 RECORDS   **
//**********************************
//SYSOUT   DD SYSOUT=A
//SYSIN    DD DSN=&SORT12,DISP=(OLD,DELETE)
//SORTIN   DD DSN=HPFP.TSSME.DA12,
//            DISP=(OLD,DELETE,KEEP)
//SORTOUT  DD DSN=HPFP.TSSME.SDA12,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(15,3),RLSE)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//*
//SORT3    EXEC PGM=SORT,PARM='CORE=MAX',COND=(4,LT),
//            REGION=2048K
//*****************************************
//**           SORT DA3 FILE           **
//**                                   **
//** NOTE: STEP CAN BE OMITTED IF      **
//**       MODELLER OUTPUT (STEP FABBME1) **
//**       IS ONLY FOR ONE AREA        **
//*****************************************
//SYSPRINT DD SYSOUT=A
//SYSOUT   DD SYSOUT=A
//SYSIN    DD DSN=&SORT3,DISP=(OLD,DELETE)
//SORTIN   DD DSN=HPFP.TSSME.DA3,
//            DISP=(OLD,DELETE,KEEP)
//SORTOUT  DD DSN=HPFP.TSSME.SDA3,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,5),RLSE)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//*
//FABADA5  EXEC PGM=FABADA5,REGION=512K,COND=(4,LT)
//**********************************************
//** GENERATE REPORTS                        **
//**                                         **
//** NOTE: ALTER DD STMT DAR13I AS REQUIRED  **
//**       IF STEP SORT3 WAS OMITTED         **
//**********************************************
//SYSPRINT DD SYSOUT=A
//REPORTS  DD SYSOUT=A
//UOWRPT   DD SYSOUT=A
//UOWEXCP  DD SYSOUT=A
//DAR12I   DD DSN=HPFP.TSSME.SDA12,
//            DISP=(OLD,DELETE,KEEP)
//DAR13I   DD DSN=HPFP.TSSME.SDA3,
//            DISP=(OLD,DELETE,KEEP)
//*
```

Figure 41. Sample JCL for DEDB Tuning Aid (Multi-area output) (Part 2 of 2)

Example 2: Requiring a new DMB

The following figure shows example JCL statement for requiring a new DMB.

If any database specifications, excluding the randomizing routine, are changed, a new DMB is required.

1. To avoid impacting existing components, allocate a small DBDLIB and ACBLIB.
2. Use DEDB Pointer Checker reports for the existing database to determine the new database specifications.
3. Code and generate the new DBD and ACB using the allocated libraries.
4. Determine the required input for the DEDB Tuning Aid process.

   Note: Unless an exact tune is required, it is not important that the SRDA13 data for all the required areas was created at the same time.
5. Perform the DEDB Tuning Aid procedure, requesting only the area of interest.

6. Evaluate and compare the results.

7. Repeat steps 2 through 6 until the desired space utilization and performance characteristics are obtained.

8. As a final check, perform the DEDB Tuning Aid procedure requesting a sampling of other areas or all areas.

Database TSSDBD currently has 10 areas, (TSSAR01 through TSSAR10.) The randomizer is area-specific. Keys starting with the characters "A" through "L" randomize across areas 1 through 5. Keys starting with "M" through "Z" use areas 6 through 10. Area 4 requires expansion. The ROOT and UOW values are to be increased.

```
Required input:  SRDA13 data for areas 1 through 5.

Control statement:  DBDNAME=TSSDBD,AREA=4
```

Note: FABBME1 generates a warning message stating that SRDA13 data for some areas was missing. This message can be ignored because the randomizer is area-specific. If the randomizer were nonspecific, the DEDB Tuning Aid results might not be valid.

Related information
Input for DEDB Tuning Aid
To run the DEDB Tuning Aid, you must specify input definitions for each program.

Example 3: Testing randomizing routines

The following figure shows example JCL statement for testing randomizing routines.

To effectively compare randomizing routines, none of the other database specifications should be changed.

1. Determine the required input for the DEDB Tuning Aid process.

   Note: Unless an exact tune is required, it is not important that the SRDA13 data for all the areas being tuned was created at the same time.

2. Perform the DEDB Tuning Aid procedure, requesting only the area of interest.

3. Evaluate and compare the results.

4. Repeat steps 2 and 3 until the desired space utilization and performance characteristics are obtained.

5. As a final check, perform the DEDB Tuning Aid procedure requesting a sampling of other areas, or all areas.

Database TSSDBD currently has four areas, TSSAR01 through TSSAR04. A new randomizer, TSSRAND4, is being evaluated. The current randomizer, DBFHDC40, is "nonspecific."

```
Required input:  SRDA13 data for areas 1 through 4.

Control statement: DBDNAME=TSSDBD,AREA=2,RMOD=TSSRAND4
```

Related information
Input for DEDB Tuning Aid
To run the DEDB Tuning Aid, you must specify input definitions for each program.

Example 4: Tuning the IBM high-speed DEDB direct reorganization utility

The following figure shows example JCL statement for tuning the IBM high-speed DEDB direct reorganization utility.

The potential effect of using the IBM high-speed DEDB direct reorganization utility can also be evaluated. To do so, run the DEDB Tuning Aid procedure, requesting only the area of interest.

Area TSSAR03 of database TSSDBD is to be tuned to determine the potential impact of reorganization.

```
Required input:  SRDA13 data for area 3.

Control statement:  DBDNAME=TSSDBD,AREA=3.
```

Chapter 4. Unloading a DEDB offline with FPB

You can unload a DEDB by using the DEDB Unload utility of FPB.

The unload function in FPB is run by program FABCUR1.

The DEDB Unload utility (FABCUR1) efficiently unloads a single DEDB area, or concurrently unload multiple DEDB areas, without impacting all areas of a database. They can be approximately 10 to 20 times faster than the equivalent function using "GN" and "ISRT" processing under the control of IMS.

Topics:

Functions of DEDB Unload

DEDB Unload can efficiently unload and reload a single DEDB area. DEDB Unload can concurrently unload and multiple DEDB areas, without impacting all areas of a database.

Subsections:

Features

This component provides besides the unload and the reload function, a set of support utilities that can be used for unload and reload. It offers the following features:

- Both the unload and reload programs function independently of the IMS Control Region. Hence, one or more areas of a multi-area database can be unloaded or reloaded while the application continues to use the areas that are not included in the maintenance process. This feature can significantly increase application availability.

  Note: This requires specific control of area selection and application code capable of handling 'FH' status codes.

- Both processors can concurrently process multiple database areas with no database contention. This concurrent processing capability significantly decreases the time required for database maintenance, and it further increases application availability.

- During the unload/reload process, you can change any or all of the following database specifications:

  - DBD name
  - Number of database areas

- – Randomizing module
- – Segment edit/compression routine
- – UOW parameter values
- – ROOT parameter values
- – CI size
- – Size of the VSAM data set
- – Pointer options
- – Addition of new segment
- – Change of existing segment hierarchical structure within the same parent

Notes:

1. These DBD definition changes will be applied only during unload process by specifying an ACB library that has new DBD definition member to the NEWACB DD statement. It implies that unloaded segment records produced by the unload process will be composed on the basis of the new DBD definition information. DBD definition change cannot be specified in the reload process (FABCUR3).

2. For how to add a new segment or to change an existing segment hierarchical structure, see "Hierarchical structure changes" on page 117. Existing segment names cannot be changed, nor can existing segments be deleted.

- During the unload process, a second copy option may be specified in order to prepare two sets of unloaded files. With this option specified, the unload processing will continue even if one of the copies encounters an I/O error. This function is very effective for the users with big databases.

- During the unload process, an empty area unloaded by this utility is clearly identified with a warning message. Then during the reload process, an output data set for this empty area is initialized with no segment in order to prevent IMS DB/DC accessing trouble. The Audit Control report also shows the empty DEDB area.

- During the reload process, abnormally long dependent segment twin chains can be controlled so that the impact on other database records in the same RAP CI is minimized. This feature is implemented via a user-specified limit on the number of segment occurrences that are to be placed "near" the root.

- The new area and RAP values for the database record are determined during unload processing. This allows all records for an area to be written to the same output data set, avoids an extra pass of the file, and simplifies the reload process.

- There are no source code modifications made to any user-written or IMS program or control block. Because IMS program integrity is maintained, new exposures are not introduced.

- DEDB Unload/Reload enables you to produce an expanded-format unloaded data set from compressed segments.

- The reload program can reload segment data in one area into multi-area data sets.

- The reload program can produce image copy data set(s) of an unloaded area.

- Under the image copy option, the reloaded program can do a fast scan (HASH check) of the integrity of the unloaded area.

General structure

The DEDB Unload/Reload utility consists of two functional components that operate independently of the IMS control region.

The unload processor is a z/OS batch program that can concurrently unload multiple areas of a DEDB. But there are basically two problems that hinder DEDB unload/reload processing if operated under the control of the IMS control region:

1. To prohibit access to the areas involved in the maintenance process, all transactions that access the database must be stopped or made logically unavailable to the application. If the transactions must be stopped, application availability is adversely impacted.

2. DEDB Unload/Reload processing (or any sequential process) that uses "GN" and "ISRT" calls is exceedingly slow. The basic problem with this approach is the number of EXCPs required. An unload using "GN" processing issues one EXCP for each CI that contains data. A reload using "GN" and "ISRT" processing requires one EXCP to retrieve a CI and another EXCP to rewrite the CI. Additional EXCPs are incurred if the NBA is not large enough to hold a logical UOW (that is, the RAA BASE section CIs plus all IOVF CIs logically owned by the UOW).

The DEDB Unload/Reload utility achieves significant performance improvements (that is, elapsed time reduction) by reducing the number of EXCPs issued. Both programs use the VSAM read ahead facility to minimize the number of EXCPs (that is, a conscious decision was made to reduce EXCPs at the expense of increased memory usage). Both the unload and reload programs, if buffered correctly, read and write a minimum of 23 2048-byte CIs per EXCP for the root addressable section of an area. This number can be further increased (that is, elapsed time reduced) by providing additional VSAM buffers (within limitations).

In addition, all IMS overheads (that is, logging or sync point processing) have been eliminated, and the instruction path length to retrieve or insert a segment is exceedingly short.

DEDB areas requiring maintenance are deallocated from the IMS control region. The unload processor, after first ensuring that the maintenance request is valid and that all required resources are present, attaches and manages a user-specified number of subtasks. Each subtask is responsible for unloading a specific area. When completed, the subtask returns to the main task for assignment of another area or termination. As each database segment is unloaded, the subtask invokes the randomizer (if required) to determine the new area/RAP values. All subtasks share a common output writer. This allows all database records for an area to be written to the same output file. This approach simplifies the reloading process, and eliminates an extra pass of the unloaded segment file.

The number of concurrent unload subtasks depends on the UOW size and the number of required IOVF buffers. This is explained in detail in "Hierarchical structure changes" on page 117.

After the VSAM clusters have been deleted and redefined, multiple reload jobs can be initiated to concurrently reload the areas. The only limitation on the number of concurrent reload jobs is the number of available initiators. You can also reload the areas in a single job step.

Hierarchical structure changes

During the unload/reload process, the segments can be moved in the hierarchical structure, and the new segments can be defined if the following rules are not violated. FABCUR1 detects invalid structure changes.

1. The existing segment names cannot be changed.
2. The new segments can be added, but the existing segments cannot be deleted.
3. When a segment is being moved in the hierarchical structure (that is, its segment code is being changed), it must remain at the same hierarchical level.
4. A segment must have the same parent after being moved in the hierarchical structure.

The following table shows examples of hierarchical structure change of the unload and reload processes.

Table 29. Examples of hierarchical structure change

Segname	Old segcode	New segcode
ROOTSEG	1	1
SEQDEP	2	2
NEWDIR	NEW	3
DIRDEP5	7	4
DIRDEP1	3	5
DIRDEP2	4	6

Table 29. Examples of hierarchical structure change (continued)

Segname	Old segcode	New segcode
DIRDEP3	5	7
DIRDEP4	6	8

Modes

There are two basic modes to use the DEDB Unload/Reload utility. These are called "reorg" mode and "change" mode. Setting your JCL and control statements depends heavily on the mode you are using. This topic describes the techniques about how to use for each type of unload/reload.

Reorg mode

This mode is used when you do not change the database structure, the DBD, or the randomizing routine. The only change you can make in "reorg" mode is to increase or decrease the size of the SDEP part of an area.

There are two ways to reduce the size of SDEP part of an area:

- Unload with the SDEP=LOGICAL option
- Unload with the SDEP=PHYSICAL option and reload with the SDEPRELOCATE=YES option

In both ways, the SDEP marker will be lost. When reducing the size of SDEP part, you must make sure that the SDEP size after being reduced has enough space to restore all SDEP CIs that are between SDEP logical begin and logical end.

A "reorg" mode FABCUR1 run should be set up in the following manner:

- Code REORG and STATS on your DBDNAME control statement.
- Do not code HIERCHNG= or RMODTYPE= on your DBDNAME control statement.
- Do not include NEWACB or RMODLIB DD statements in your JCL.

Change mode

This mode is used when you change the database structure, the DBD, or the randomizing routine.

A "change" mode FABCUR1 run should be set up in the following manner:

- Do not code REORG on your DBDNAME control statement.
- If you are changing the segment hierarchy (that is, one or more segment codes are being changed), then code HIERCHNG=YES/YESFORCE on your DBDNAME control statement.

  For additional information, see .

- If your randomizing routine is "area-specific," and if you are not unloading all areas, code RMODTYPE=S on your DBDNAME control statement.
- Include the NEWACB and RMODLIB DD statements in your JCL.

Change Database Definition
Specify an ACB library that includes the current DBD type ACB in the OLDACB DD, and an ACB library that includes the new database definition DBD in the NEWACB DD.

Change Randomizer module
Specify a library that includes the NEW randomizer module in RMODLIB. The old randomizer is not needed for unloading. Do not concatenate the library that includes the old randomizer ahead of the new randomizer library, if randomizer name is the same.

Notice: If you do not change the randomizer, you should specify the library that includes the current randomizer module in the RMODLIB DD of FABCUR1.

Change Compression exit
If the compression exit name is the same, check the following:

- Specify a library that includes the current compression exit in the RMODLIB DD when unloading, then provide a library that includes the new compression exit in the RMODLIB DD of FABCUR3. If the compression exit name is different, choose one of the following:
  - Specify COMP=Y in the SYSIN DD and the concatenated libraries that include the current compression exit and the new compression exit in the RMODLIB DD of FABCUR1.
  - Specify COMP=N in the SYSIN DD and specify the library that includes the current compression exit in the RMODLIB DD of FABCUR1, then specify the library that includes the old compression exit in the RMODLIB DD of FABCUR3.

You can specify Reorg mode and Change mode by using FABCOP1D (site default table).

Even if the TYPERUN/TYPRUN=REORG keyword is specified in FABCOP1D, it will be ignored if NEWDBDNM=, HIERCHNG=, or RMODTYPE= is specified in the SYSIN DBDNAME= control statement.

Data and system flow of DEDB Unload

The unload processor (FABCUR1) is a z/OS batch program that controls and manages the unloading of one or more DEDB areas to QSAM data sets. It achieves significant performance improvements (that is, reduced elapsed time) by implementing a parallel processing environment and by reducing the number of EXCPs issued.

If you enable the IMS catalog and the IMS management of ACBs, ACB definitions are retrieved from the IMS directory instead of from OLDACB and NEWACB libraries.

The following figure shows the flow of the unload procedure.
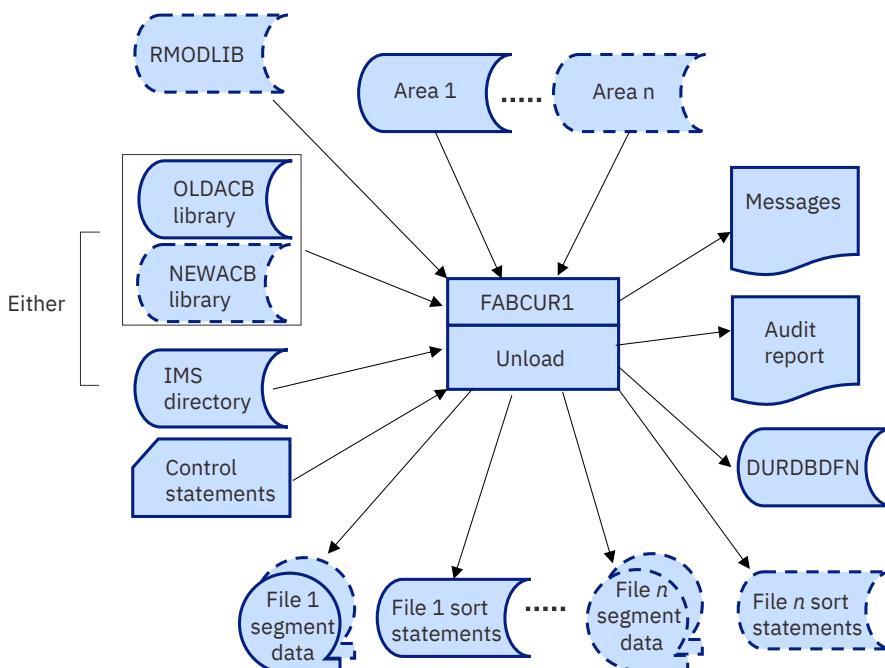
Figure 42. Flow of Unload

Load modules

Among the DEDB Unload/Reload modules, two are used for unload. The following table lists these modules.

Table 30. Load modules of DEDB Unload

Load module name	Function
FABCUR1	Controls and manages unload processing

Table 30. Load modules of DEDB Unload (continued)

Load module name	Function
FABCUR1U	Subtask to extract segment data

Restrictions of DEDB Unload

The DEDB Unload/Reload utilities have some restrictions.

When a randomizer is invoked by IMS, Registers 10 and 11 contain the addresses of the EPST and ESCD, respectively. Some user-written or user-customized randomizers are designed to use these addresses to gain access to IMS control blocks other than those passed as part of the published interface. When one of the DEDB Unload/Reload utilities issues a call to a randomizing module, Registers 10 and 11 are set to -1 to 0, respectively, to specify that the call is not being issued in a "live" IMS environment.

DEDB Unload (FABCUR1) does not support Automatic RECON Loss Notification.

Typical scenario and processes to run for unloading and reloading a DEDB

To use the DEDB Unload/Reload, you must do several things.

Typical scenario for DEDB Unload/Reload

The following is a checklist of the activities that are usually required to unload and reload one or more areas of a DEDB. Some steps are omitted when only database reorganization is being performed:

1. Perform the following pre-unload/reload planning.

   a. Determine the number of unload subtasks and output files, and calculate the storage requirements for FABCUR1.

      For the details, see the Unload Region Size Estimation Worksheet, .

   b. Set up and tailor the unload JCL:

      • Control statements
      • Input and output files
      • Number of sort steps.

   c. Set up and tailor the required number of reload jobs.

   d. Set up a temporary ACB library and generate the new DBD and ACB.

2. Stop (that is, deallocate from the IMS control region) all areas involved in the maintenance process. If the application does not handle "FH" status codes, all transactions that access the database must be stopped.

3. Image copy all areas that will be unloaded and reloaded.

4. Run the unload job.

5. Run DEDB Pointer Checker against the image copy data sets. This provides statistics on the number of segments in the areas being unloaded/reloaded. This could be run concurrently with the unload/reload jobs.

6. Check any messages generated by FABCUR1, and verify segment totals on the Unload Audit Control report.

7. Run the reload jobs by specifying image copy and HASH check options.

8. Check any messages generated by FABCUR3, and verify segment totals on the Reload Audit Control report.

9. If image copy option was not specified in step 7, image copy all areas that were unloaded and reloaded.

10. If image copy option was not specified in step 7, run the DEDB Pointer Checker utility against the image copy data sets. Verify that the unload/reload produced clean areas with the correct number of segments.
11. Migrate the new DMB to the production ACB library.
12. Start the reloaded areas.

Related reference

Region size requirements for DEDB Unload
The DEDB Unload utility operates in a virtual storage region as z/OS batch job steps. FABCUR1 and FABCUR1U run in 31-bit addressing mode.

Running DEDB Unload

The unload program (FABCUR1) is run as a standard z/OS batch job. You need to specify an EXEC statement and DD statements that define the input and output data sets in your JCL.

Procedure

1. Estimate the region size required to run the function.

   Note: The actual region size depends on the number of subtasks or output files.
2. Code the JCL for the unload (FABCUR1) job step.
3. Specify the DD statements to define input data sets, output data sets, and how the function is run.
4. Run the JCL.

   Tip: FPB allows you to specify site default parameters for DEDB Unload (FABCUR1). Macros and sample JCL streams are provided to generate the site default table.

Example

The following figure shows example JCL stream for FABCUR1.

```
//STEP1     EXEC PGM=FABCUR1,REGION=rrrrM
//RMODLIB  DD DSN=IMSVS.SDFSRESL,DISP=SHR
//NEWACB   DD DSN=IMSVS.ACBLIB,DISP=SHR
//OLDACB   DD DSN=IMSVS.BKUP.ACBLIB,DISP=SHR
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *,DCB=BLKSIZE=80
   ... control statements ...
//DURAUDIT DD SYSOUT=A
//areaxxx  DD DSN=VSAM.AREAxxx,DISP=OLD
//DURDBDFN DD DSN=HPFP.UR.DURDBDFN,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(TRK,(1,1))
//DURDzzzO DD DSN=HPFP.UR.FILEzzz.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),UNIT=TAPE
//DURDzzzE DD DSN=HPFP.UR.FILEzzz.SEGDATAE,
//            DISP=(NEW,CATLG,DELETE),UNIT=TAPE
//DURSzzzO DD DSN=HPFP.UR.FILEzzz.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(TRK,(1,1))
//*
```

Figure 43. Example JCL for DEDB unload

Related reference

Region size requirements for DEDB Unload

The DEDB Unload utility operates in a virtual storage region as z/OS batch job steps. FABCUR1 and FABCUR1U run in 31-bit addressing mode.

Region size requirements for DEDB Unload

The DEDB Unload utility operates in a virtual storage region as z/OS batch job steps. FABCUR1 and FABCUR1U run in 31-bit addressing mode.

Region size for the unload program depends on the number of unload subtasks, the size of a UOW, the number of IOVF buffers allocated, the number of output files, and the TASKCTL specifications.

Typically, database maintenance is a relatively important function. To increase the performance of the unload program, it would be advisable to run it in a region that has a relatively high dispatching priority (that is, at least equal to the dispatching priority of a message processing region).

Estimating the region size for unload

The region size required for the unload processor (FABCUR1) is a function of the number of areas being unloaded, the number of output files, and of the TASKCTL specifications used.

In most cases, the specification on the EXEC statement for FABCUR1 of a region that is too small, is detected during the main task's initialization processing. This causes processing to be terminated at that point. Under certain conditions, the region-size shortfall does not become evident until the actual unload has begun. If this occurs, one or more unload subtasks are quiesced in an attempt to free up sufficient storage to complete the requested processing. Should this fail, the main task ends with an abend code. There is no option to restart unload processing. The job must be completely rerun. Thus, the region-size calculations should always be performed prior to initiating unload processing.

The following worksheet is for estimating the region size that is required for a typical unload job. The actual region size depends on the keywords that are specified for the job. Therefore, when you specify the region size in your JCL, use a higher value than the value calculated from this worksheet.

```
Unload programs, internal tables, control blocks        260,000
Number of unloaded areas * nnn                          -------- (nnn=800 if ACCESS=VSAM, nnn=950 if ACCESS=FAST)
Number of areas defined in input DBD * 160              --------
Number of areas defined in output DBD * nnn             -------- (nnn=700 if reorganization mode, nnn=780 if restructure mode)
Number of output files * (1200 + BLKSIZE * BUFNO)       --------
    Internal tables, control blocks          53,000
    Iovfbufs * max CISZ in old DBD           -------
    max CISZ * (max CIs per track * 2 + 1)   -------
    aa * UOW1+ bb * iovfbufs                 -------            (aa=12,bb=24 if ACCESS=VSAM, aa=60,bb=72 if ACCESS=FAST)
    max CISZ * UOW1                          -------

                               Subtotal      -------
Number of unload subtasks X Subtotal                    --------

        Total bytes of unload storage               --------
```

Figure 44. Worksheet for estimating the region size for Unload

Reducing region size constraints

The unload program (FABCUR1) is implemented using various techniques aimed at extracting the segment data from DEDB areas in the shortest possible time. The trade-off is increased storage requirements. As seen in the examples, the region-size requirements for FABCUR1 can easily become very large. The following list shows a series of actions that may be taken to reduce the region-size requirements for a given unload request. They are listed in increasing order of impact on the performance of FABCUR1.

1. Reduce the number of the unload subtasks specified on the TASKCTL control statement.

   • This eliminates the storage required for the BASE/DOVF buffer area and a set of IOVF buffers for each of the unload subtasks removed.

   • It also reduces the amount of storage allocated at a given time by VSAM for the sequential ACBs used to access the BASE and DOVF CIs in the areas being unloaded.

2. If the maximum block size for the output file devices is large, specify the DCB=BLKSIZE=*nnnnn* parameter in the DD statement. Typically, reduce the block size to 15 - 20 K (depending on the number of output files being produced).

   • This reduces the storage required for the QSAM buffers for each of the files.

3. Reduce the number of the output files.

   • This eliminates the storage required for the QSAM buffers for each of the removed files.
   • It also might introduce the requirement to sort the output files prior to reload processing.

4. Reduce, within reason, the value of the TASKCTL *iovfbufs* parameter below the level of the maximum number of IOVF CIs used by a UOW being unloaded.

   • This reduces the storage required for the set of IOVF buffers allocated to each unload subtask.
   • It results in extra I/O for some UOWs, as the IOVF buffer sets do not contain all the CIs required to unload the UOWs with IOVF utilization greater than the specified value. FABCUR1 may be required to "steal" buffers and subsequently reread some IOVF CIs.

5. Specify a value for the TASKCTL *bdbufsz* parameter. If the UOW1 value for the database being processed is large, this is almost certainly required. The value is typically calculated as (UOW2 + ((UOW1 - UOW2) / 4)) X CISIZE, to a maximum of 250-300K. (Larger values may be specified, if the number of unload subtasks being used is very small.)

   • This reduces the storage required for the BASE/DOVF buffer area allocated to each unload subtask.
   • It results in increased elapsed processing time as the BASE portion of each UOW must be processed in several "pieces."

6. Specify an AMP=('BUFND=...') override on the DD statements for the areas being unloaded to reduce the number of buffers below the "2 X CIs-per-Track + 1" value. This should be considered only as a last resort. FABCUR1 automatically calculates an optimum value for this parameter when Opening the VSAM Sequential ACBs. The value calculated by FABCUR1 is designed to maximize the number of CIs read per EXCP. A JCL override takes precedence over the calculated value and can have a drastic effect on the number of EXCPs issued to extract the segment data in the areas being unloaded.

   • This reduces the storage required for the buffers acquired by VSAM.
   • It results in increased elapsed processing time as the EXCP count increases, and more channel contention and I/O wait time incurs.

DD statements for DEDB Unload

DD statements for DEDB Unload determine the input and output data sets and how DEDB Unload is run.

EXEC statement

Specify the EXEC statement in the following format:

```
//      EXEC PGM=FABCUR1,PARM='DBRC=x,IMSID=xxxxxxxxx,AREC=x,IMSPLEX=imsplex,
//      DBRCGRP=xxx',REGION=rrrrM
```

DBRC=
  Determines whether or not to use the DBRC interface.

  Y
    FABCUR1 uses the DBRC interface.

    DEDB Unload checks the following before and after completing the unloading of records in an IMS Data Entry Database (DEDB).

    • Status of an area and the area data set

      When an area is 'not recovery needed' and any of the specified area data sets are available in DBRC, DEDB Unload performs unloading. When an area is 'recovery needed' or none of the

specified area data sets are available in DBRC, the utility issues an error message and stops processing.

At the end of the unload processing, DEDB Unload rechecks the status of the area and the area data set. If the area and/or the area data set status is changed during unload processing, the utility issues a warning message.

- Authorization status of an area

DEDB Unload performs unloading when an area is not authorized or is authorized with read access intent by any IMS system. If an area is authorized with update/exclusive access intent, the utility issues an error message and stops processing.

At the end of unload processing, DEDB Unload rechecks the authorization status. If the area is authorized with update/exclusive access intent during unload processing, the utility issues a warning message.

- Extended Error Queue Element (EEQE)

DEDB Unload performs unloading when there is no Extended Error Queue Element (EEQE) for an area in DBRC. If any EEQEs exist in DBRC, the utility issues an error message and stops processing.

At the end of the unload processing, DEDB Unload rechecks the existence of EEQEs. If any EEQEs are registered in DBRC during the unload processing, the utility issues a warning message.

N
FABCUR1 does not use the DBRC interface.

Default: N

AREC=
Determines whether to include an area information record in an unloaded segment record data set.

Y
FABCUR1 puts an unloaded segment record data set that includes an area information record. AREC=YES can also be specified.

N
FABCUR1 puts an unloaded segment record data set that does not include an area information record. AREC=NO can also be specified.

Default: Y

Note: With AREC=N option, an empty area is not supported for reload purpose. Both FORMAT=TFMT and LRECL=SEGTFMT options must be specified with unload process AREC=N option to take care of the empty area for reload process. FABCUR1 generates a special record for an empty area to be recognized by FABCUR3 as an empty area. An application program that accesses the unloaded segment record file must be sensitive of the special record that has X'FFFFFFFF' value in the USRRAP field.

IMSPLEX=
IMSPLEX is a one- to five-character optional parameter that specifies which IMSplex DBRC should join in.

DBRCGRP=
DBRCGRP is a one- to three-character optional parameter, which is used by DBRC to identify which sharing group the notification belongs to.

IMSID=
Determines the IMS subsystem name.

The IMS subsystem name is an eight-character alphanumeric string that comprise a valid IMS subsystem identification name. If the IMSID parameter is specified, FABCUR1 generates the buffers for temporary ESCD and SCD. Register 11 contains the address of the temporary ESCD when FABCUR1 issues a call to a randomizing module.

FABCUR1 sets only the ESCDSCD field, which contains the address of SCD, in the temporary ESCD. FABCUR1 also sets only field SSCDIMID, which contains the IMS subsystem ID, in the temporary SCD. Other fields of temporary ESCD and temporary SCD cannot be referred to. If IMSID is specified, the IMS load module library must be concatenated with the JOBLIB or STEPLIB DD.

Default: None.

Note: The actual Region size depends on the number of subtasks or output files.

DD statements

You must specify DD statements for the job control language (JCL) to run the DEDB Unload utility. The following table summarizes the DD statements for FABCUR1 JCL.

Table 31. FABCUR1 DD statements

DDNAME	Use	Format	Required or optional
JOBLIB or STEPLIB	Input	PDS	Required
SYSIN	Input	LRECL=80	Required
area*xxx*	Input	VSAM ESDS cluster	Required
ads*xxx*	Input	VSAM ESDS cluster	Optional
OLDACB	Input	PDS	Required when IMSCATHLQ=*NO
NEWACB	Input	PDS	Optional
RMODLIB	Input	PDS	Optional
RECON1	Input	KSDS	Optional (see Note)
RECON2	Input	KSDS	Optional (see Note)
RECON3	Input	KSDS	Optional (see Note)
EXITLIB	Input	PDS	Optional
IMSDALIB	Input	PDS	Optional
SYSPRINT	Output	LRECL=133	Required
DURAUDIT	Output	LRECL=133	Required
DURDBDFN	Output	Do not code DCB	Required
DURD*zzz*O orXD*zzzzz*O	Output	Do not code any DCB other than BLKSIZE	Required
DURD*zzz*E orXD*zzzzz*E	Output	Do not code any DCB other than BLKSIZE	Optional
DURS*zzz*O orXS*zzzzz*O	Output	LRECL=80	Required
UR1TOTAL	Output	LRECL=60	Optional
EX*zzzzz*O or EX*zzzzz*E	Output	Do not code any DCB	Optional

Note: These data sets are required when DBRC=Y is specified as a parameter on the EXEC statement.

All output files are blocked to the maximum size of the output device (unless overridden in the execution JCL). Since the blocking factor is determined at execution time, standard labels must be used on all output data sets except SYSPRINT and DURAUDIT.

JOBLIB DD or STEPLIB DD

When DBRC=Y is specified on the EXEC statement, the IMS load module library (IMSVS.SDFSRESL) must be concatenated with the FPB DEDB Unload/Reload library on the JOBLIB or STEPLIB DD.

To allocate a DBRC RECON data set and/or a DBRC non-registered area data set dynamically, the DFSMDA library data set must be concatenated.

To allocate an area data set dynamically when DBRC=N is specified, the DFSMDA library data set must be concatenated.

If you specify the IMSCATHLQ=*bsdshlq* keyword, the IMS load module library (IMSVS.SDFSRESL) and the IMS Tools Base library (SGLXLOAD) must be concatenated.

SYSIN DD

This statement defines the control statement input data set. This data set can reside on a direct-access device, or be routed through the input stream.

area*xxx* DD

This statement defines an input data set that is the VSAM ESDS for an area to be unloaded. The ddname used must be the ddname of the area as defined in the old DMB. There must be one DD statement for each area included in the unload request.

When the DBRC=Y parameter is specified on the EXEC statement this area*xxx* DD is ignored for a DBRC registered area.

When DBRC=N is specified, or DBRC=Y is specified and the area is not registered in DBRC, this data set can be allocated dynamically by providing the associated DFSMDA member in the data set that is concatenated to the JOBLIB/STEPLIB DD statement.

ads*xxx* DD

This statement defines an input data set that is the VSAM ESDS for a DBRC registered area to be unloaded when DBRC=Y is specified as a parameter on the EXEC statement. The DD name and the data set name should be registered in DBRC. When the area data set is MADS (Multiple Area Data Set), more than one ads*xxx* DD can be specified. FABCUR1 unloads records from one of these data sets if DBRC shows that the status as available and that there are no EQEs (Error Queue Elements).

When DBRC=Y is specified and an area is registered in DBRC, all available area data sets can be allocated dynamically. If any ads*xxx* DD statement is specified explicitly for any area data sets that are available and registered in DBRC, dynamic allocation is not performed for all available area data sets that belong to the same area as the one with the ads*xxx* specification; Therefore, if you want to allocate available area data sets dynamically, do not specify any ads*xxx* DD statement.

When DBRC=N is specified on the EXEC statement, this ads*xxx* DD is ignored.

OLDACB DD

This statement defines the library that contains the old DMB for the database.

You must specify the library that contains ACB definitions either with the OLDACB DD statement or with the IMSCATHLQ=*bsdshlq* keyword. If IMSCATHLQ=*bsdshlq* is specified, ACB definitions are read from the IMS directory instead of the ACB library.

NEWACB DD

This statement defines the library that contains the new DMB for the database. This DD statement is ignored if an area is being reorganized (that is, the keyword REORG is specified). If this DD statement is present, REORG is not assumed even if there are no DBD specification changes (that is, the randomizer is called for each root segment).

If IMSCATHLQ=*bsdshlq* keyword is specified, NEWACB DD statement is ignored. If an ACB definition change is requested, the new DMB is retrieved from the IMS directory.

RMODLIB DD

This DD statement defines the library where the randomizer and/or segment edit/compression routines reside.

RMODLIB in the FABCUR1 step is required for segment edit/compression routines when the SYSIN COMPRESS=NO control statement (this literally means expand the segments as they are unloaded) is specified and the existing segment edit/compression routine must be used.

RMODLIB in the FABCUR1 step is required for randomizer routines when a new randomizer is to be used. The new randomizer MUST reside in the library pointed to by RMODLIB DD in the FABCUR1 step. If this new randomizer routine is required in conjunction with a segment restructure (change in DMB) and the FABCUR1 SYSIN DBDNAME control statement does NOT specify DBDNAME=REORG, then the NEWACB DD statement must also be specified.

If this DD statement is not provided, an attempt is made to load the randomizer and/or edit/compression routine from JOBLIB/STEPLIB.

RECON1 DD
This statement defines the first DBRC RECON data set. This RECON1 data set must be the RECON1 data set used in the control region.

RECON2 DD
This statement defines the second DBRC RECON data set. This RECON2 data set must be the RECON2 data set used in the control region.

RECON3 DD
This statement defines the third DBRC RECON data set. This RECON3 data set must be the RECON3 data set used in the control region.

Do not use these RECON data set ddnames if you specified dynamic allocation using the DFSMDA macro.

You need to specify the IMSPLEX and DBRCGRP parameters on the EXEC statement for parallel RECON access to make sure that all DBRC instances in a sharing group have access to the same RECON data sets.

EXITLIB DD
This statement defines the library in which the exit routine specified by the EXITRTN= control statement resides.

IMSDALIB DD
This statement defines the IMS library that contains DFSMDA members to allocate an area data set and/or RECON data sets dynamically. If this statement is provided, program FABCUR1 loads DFSMDA members from the IMSDALIB DD prior to the JOBLIB DD or the STEPLIB DD.

SYSPRINT DD
This statement defines the output data set that contains messages issued by FABCUR1. The data set can reside on a direct-access device or printer, or be routed through the output stream. You can code RECFM=FBA, LRECL=133 on your DD statement, but it is better to use:

```
//SYSPRINT DD SYSOUT=A
```

DURAUDIT DD
This statement defines the output data set that contains the Unload Audit Control report. This data set can reside on a direct-access device or printer, or be routed through the output stream. You can code RECFM=FBA,LRECL=133 on your DD statement, but it is better to use:

```
//DURAUDIT DD SYSOUT=A
```

DURDBDFN DD
This statement defines an output data set for the database definition record generated by FABCUR1. This contains data extracted from the "output" DMB that is used by the reload processor. The "output" DMB is that DMB which is used to access the database after reload processing. It is usually the DMB read from the NEWACB file; if REORG mode processing is being performed, it is the DMB from the OLDACB file. The data set must reside on a direct-access device. Space requirements depend on the size of the DMB, but a couple of tracks suffice.

The DCB attributes are calculated by FABCUR1. RECFM is VB.

- For 3380, the default block size is 23476 bytes.
- For 3390, the default block size is 27998 bytes.
- For 9345, the default block size is 22928 bytes.

For other output devices, the default block size is the maximum block size of the output device. Both LRECL and BLKSIZE are accepted when they are equal or smaller than the block size listed and also larger than the total DDT record size. The default BLKSIZE is the maximum block size of the output device, and the default LRECL is the total DDT record size. Do not code any other DCB parameters in your JCL.

Do not specify DISP=MOD for this DD statement.

DURD*zzz*O or XD*zzzzz*O DD

This statement defines an output data set for all of the database segment records produced for one or more of the areas defined in the "output" DMB. A DURD*zzz*O DD statement is for areas in the range of 1 - 999, and an XD*zzzzz*O DD statement is for areas in the range of 1 - 2048. If the area number of the unloaded area is greater than 999, you should provide the XD*zzzzz*O DD statement. The value of *zzz* or *zzzzz* is made up of right-aligned digits, with leading zeros if needed.

The data set can reside on either a direct-access device or tape. Standard labels must be used for this data set. The DCB attributes are calculated by FABCUR1. RECFM is VB.

- For 3380, the default block size is 23476 bytes.
- For 3390, the default block size is 27998 bytes.
- For 9345, the default block size is 22928 bytes.

For other output devices, the default block size is the maximum block size of the output device. Both LRECL and BLKSIZE are accepted when they are equal or smaller than the block size listed and also larger than the maximum possible unloaded segment size. The default BLKSIZE is the maximum block size of the output device, and the default LRECL is the maximum possible unloaded segment size. Do not code any other DCB parameters in your JCL. DUMMY or NULLFILE is allowed for the DD.

Do not specify DISP=MOD for these DD statements.

Notes: Whether the output area data set is empty, the DD statement of an output unloaded file is required to create an area information record on the following conditions:

- An associated DD statement of an output unloaded file is required for those areas specified in the FILECTL statement when RMODTYPE=S is specified on the DBDNAME control statement.
- An associated DD statement of an output unloaded file is required for all areas defined in the DMB of NEWACB when HIERCHNG=YES and/or RMODTYPE=G is specified on the DBDNAME control statement. AREACTL=ALL and 'ALL' or '(*)' on the FILECTL statement must be specified.

The rules for supplying the DURD*zzz*O or the XD*zzzzz*O data sets are discussed in "FILECTL control statement" on page 135.

DURD*zzz*E or XD*zzzzz*E DD

This statement defines the second copy data set for the DEDB unloaded file. For a DURD*zzz*E or XD*zzzzz*E DD statement, there must be a corresponding DURD*zzz*O or XD*zzzzz*O DD statement. A DURD*zzz*E DD statement is for areas in the range of 1 - 999, and an XD*zzzzz*E DD statement is for areas in the range of 1 - 2048. If the area number of the unloaded area is greater than 999, you should provide the XD*zzzzz*E DD statement. The value of *zzz* or *zzzzz* is made up of right-aligned digits, with leading zeros if needed.

The DCB attributes are calculated by FABCUR1. RECFM is VB.

- For 3380, the default block size is 23476 bytes.
- For 3390, the default block size is 27998 bytes.
- For 9345, the default block size is 22928 bytes.

For other output devices, the default block size is the maximum block size of the output device. Both LRECL and BLKSIZE are accepted when they are equal or smaller than the block size listed and also

larger than the maximum possible unloaded segment size. The default BLKSIZE is the maximum block size of the output device, and the default LRECL is the maximum possible unloaded segment size. Do not code any other DCB parameters in your JCL. DUMMY or NULLFILE is allowed for the DD.

Do not specify DISP=MOD for these DD statements.

DURS*zzz*O or XS*zzzzz*O DD

This statement defines an output data set that contains the SORT control statements for the segment data set associated with it. There must be a DURS*zzz*O or XS*zzzzz*O data set for each DURD*zzz*O or XD*zzzzz*O data set. A DURS*zzz*O DD statement is for areas in the range of 1 - 999, and an XS*zzzzz*O DD statement is for areas in the range of 1 - 2048. If the area number of the unloaded area is greater than 999, you should provide the XS*zzzzz*O DD statement. The value of *zzz* or *zzzzz* is made up of right-aligned digits, with leading zeros if needed. Space requirements are very small; one track suffices. The DCB information is hard-coded in FABCUR1. Do not code the DCB information in your JCL.

Do not specify DISP=MOD for these DD statements.

UR1TOTAL DD

This statement defines an output data set that contains the total number of unloaded segments in an area. The format of this record is described in the FABC1TAL macro, which is provided as a product-sensitive user interface.

EX*zzzzz*O or EX*zzzzz*E DD

This statement defines output data set for the SEGMCTL control statement. EX*zzzzz*O DD defines the primary output data set and EX*zzzzz*E DD defines the secondary one.

- Do not code the DCB information in your JCL.
- DCB attribute is the same as that of the Unloaded Segment File.
- DUMMY or NULLFILE is allowed for this DD.
- Do not specify DISP=MOD for these DD statements.

The EX*zzzzz*O/EX*zzzzz*E DD statement is for areas in the range of 1 - 2048. The value *zzzzz* is right-aligned digits, with leading zeros if needed. When the EX*zzzzz*/EX*zzzzz*E DD statements for the unloaded areas are not specified in the JCL statement, the segment records are written into EX00000O/EX00000E.

The extracted segment record file layout is as follows:

```
        LLZZ                    XL4
        Segment name            CL8


        ROOT segment key value  0X    (variable)
        Segment data            0X    (variable, include LL)
```

For more information about the layout of the unloaded segment records, see the topic "Unloaded segment records layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Related concepts

Compatibilities between IMS Fast Path Basic Tools and former releases of IMS Fast Path Basic Tools
There are considerations that apply when you migrate from IMS DBT 2.3 or FPB 1.x to this release of IMS Fast Path Basic Tools (FPB).

Related reference

Region size requirements for DEDB Unload

The DEDB Unload utility operates in a virtual storage region as z/OS batch job steps. FABCUR1 and FABCUR1U run in 31-bit addressing mode.

Input for DEDB Unload

You must specify the necessary input DD data sets to run the DEDB Unload utility.

DEDB Unload SYSIN DD data set

The SYSIN data set contains the user's description of the unload processing to be done by FABCUR1.

Format

This control statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte, fixed-length records. Block size, if coded, must be a multiple of 80.

This data set can contain several different types of control statements. It can be coded as shown in the following figure:

```
//SYSIN DD *
DBDNAME=VRSDSRF REORG STATS
AREACTL=ALL
TASKCTL=(,18,300)
/*
```

Figure 45. FABCUR1 SYSIN DD data set

Related reference
Site default support for FPB
This topic describes the site default support for FPB.

DEDB Unload SYSIN DD data set control statements

This topic describes the control statements used with DEDB Unload.

Subsections:

Syntax of the control statement

The FABCUR1 keywords and their associated parameter values may be coded in a free format (columns 1 - 71). The syntactical rules are as follows:

1. Control statements are coded on 80-byte records.
2. All control statement specifications must start in column 1. A control statement record can include only one control statement.
3. A "keyword=value" specification may not span the control statement.
4. There must be one DBDNAME control statement, and it must be the first in the control statement stream. There must be at least one AREACTL control statement.
5. Parameter values of the DBDNAME control statement must be separated from the next keyword by one or more blanks, or a comma followed by one or more blanks.
6. Any duplicate reference to an area in the AREACTL specifications is flagged with a warning message.
7. Any duplicate reference to an area or a file in the FILECTL specifications is flagged with an error message, and causes the program termination.
8. Any duplicate occurrence of the TASKCTL statement is flagged with a warning message, and the values from the last encountered statement are used.
9. Any duplicate reference to a segment in the LOADCTL specifications is flagged with a warning message, and the values from the last encountered statement are used.
10. Any duplicate occurrence of the PTRERROR statement is flagged with an error message, and causes the program to stop running.

DBDNAME control statement

The DBDNAME statement specifies the DBD name of the DEDB being unloaded. It also describes the type of unload operation you want to perform. There must be only one DBDNAME statement, and it must be the first statement in the SYSIN data set. It contains the following keywords:

```
DBDNAME=dbdname
[REORG]
[NEWDBDNM=new_dbdname]
[HIERCHNG={NO|YES|YESFORCE}]
[RMODTYPE={G|S}]
[STATS]
[COMPRESS={NO|YES}]
[SDEP={NO|LOGICAL|PHYSICAL}]
[ACCESS={VSAM|FAST}]
[PAD=X'nn']
```

Note: See "Modes" on page 118 for additional information.

This statement specifies the DBD name of the DEDB being unloaded. It is used for accessing both the OLDACB and the NEWACB libraries, unless the NEWDBDNM= keyword is not specified. DBDNAME is a required keyword.

REORG

This optional keyword specifies that one or more areas will be unloaded and reloaded as follows:

- No change is allowed to the database structure or DBD.
- No change is allowed to the RAP and area with which each segment is currently associated.
- The size of the SDEP part may be changed via your IDCAMS DEFINE CLUSTER parameters.

Thus, a simple reorganization of the database is being performed. REORG is mutually exclusive with the NEWDBDNM=, HIERCHNG=, and RMODTYPE= keywords. If the presented keywords and DD cards meets following cases, the REORG is assumed without the REORG keyword:

- Neither the NEWDBDNM= nor the RMODTYPE= keyword are specified, and the NEWACB DD statement is not present.
- Neither the HIERCHNG= nor the RMODTYPE= keyword are specified, and the NEWACB DD statement is not present.

If you want to run with REORG mode, you should not specify the NEWDBDNM= keyword, HIERCHNG= keyword, RMODTYPE= keyword, and NEWACB DD statement.

You can specify REORG in FABCOP1D (site default table) by using the TYPERUN/TYPRUN=REORG keyword to set the mode to Reorg Mode as the default mode. You can set the TYPERUN option without specifying this keyword in DBDNAME control statement. Even if TYPERUN/TYPRUN=REORG keyword is specified in FABCOP1D, NEWDBDNM=/HIERCHNG=/RMODTYPE= in DBDNAME control statement ignores it and sets the mode to Change Mode.

NEWDBDNM=
This optional keyword defines the new DBD name of DEDB being unloaded. It is used for accessing the NEWACB libraries. Even if TYPERUN/TYPRUN=REORG keyword is specified in FABCOP1D, NEWDBDNM= ignores it and sets the mode to Change Mode.

HIERCHNG=
This optional keyword determines FABCUR1 what to do if the DMB in the NEWACB data set has a different segment structure from that in the OLDACB data set. HIERCHNG= is mutually exclusive with the REORG keyword. Even if TYPERUN/TYPRUN=REORG keyword is specified in FABCOP1D, HIERCHNG= ignores it and sets the mode to Change Mode.

NO
Specifies that processing should stop. This is the default value.

YES
Allows processing to continue only when AREACTL=ALL is specified in the DBDNAME control statement.

YESFORCE
Allows processing to continue if RMODTYPE=S is specified. If RMODTYPE=G is specified, AREACTL=ALL must be specified in the DBDNAME control statement to continue processing.

RMODTYPE=
This optional keyword defines the type of randomizer specified in the DMB from the NEWACB data set. RMODTYPE= is mutually exclusive with the REORG keyword.

Even if TYPERUN/TYPRUN=REORG keyword is specified in FABCOP1D, RMODTYPE= ignores it and sets the mode to Change Mode.

G
Specifies a randomizer of *general* type that scatters database records across areas in a random fashion. DBFHDC40 (supplied with IMS) is an example of this kind of randomizer. This is the default value.

S
Specifies a randomizer of *area-specific* type that controls area selection in a non-random fashion. If the SDEP=PHYSICAL option and NEWACB data set are specified, RMODTYPE=S must also be specified.

STATS
This optional keyword requests printing of FABCUR1 scheduling parameter values, VSAM buffer statistics, and write-latch-contention statistics. You should always specify this keyword.

COMPRESS=
This optional keyword determines whether the segments of the unloaded record should be compressed or not.

NO
Specifies that unloaded record should contain segment that is expanded if segment edit/ compression routine is defined for the segment in DBDGEN. This is the default value.

YES

Specifies that unloaded record should contain compressed segment if segment edit/compression routine is defined for the segment in DBDGEN.

An abbreviation COMP is used for COMPRESS.

SDEP=

This optional keyword specifies whether or not to unload SDEP segments.

NO

Specifies that SDEP segments are not unloaded. This is the default value.

LOGICAL

Specifies that SDEP segments are being unloaded in logical order. The valid SDEP segments are extracted during the unload process. They will be inserted into the new database during the reload process and will retain their logical order (entry sequence) within each database record.

You must run the DFSORT job before you reload the segments with the DEDB Reload job.

PHYSICAL

Specifies that SDEP segments are being unloaded in physical order. All SDEP data between the SDEP logical beginning and the logical end are extracted during unload. The SDEP data will be reloaded into the SDEP part of the area created during reload. The reloaded SDEP data will be in the same physical order between the SDEP logical beginning and the logical end, as prior to the unload. When SDEP=PHYSICAL is specified, the performance of the unload processing will be faster than that of SDEP=LOGICAL. The following restrictions apply when the SDEP=PHYSICAL option is specified:

- If unload is accompanied by a DBD change (a NEWACB data set is specified), the characteristic of the randomizer defined in the new DBD must be area specific, that is, RMODTYPE=S is required.

- The CI size must be the same in the old and new databases.

- The definition of SDEP segment must be the same in the old and new databases.

- Compressed segments will not be expanded even if COMPRESS=YES is specified.

- Unload segment records of the SDEP CIs are unloaded as 512-byte blocks of CI data, not as discrete segments.

- The number of 512-byte blocks is shown on DURAUDIT of Unload/Reload. Text (PHYSICAL) is printed after the number so that the audit control report of Unload/Reload implies SDEP=PHYSICAL.

The SDEP marker concept at unload can be kept to reload. If SDEP=PHYSICAL is accompanied by DBD change or SDEPRELOCATE=YES option is selected at reload, absolute RBA value at unload to identify the SDEP marker might no longer be effective but root segment and the field value of the SDEP marker at unload will remain effective to identify the marker SDEP for SDEP Scan/Delete Utilities.

For how to process SDEP segments, see the topic "Determining how to process SDEP segments" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

ACCESS=

This optional keyword parameter specifies the type of access needed to read the area data set VSAM ESDS file.

VSAM

Specifies that the VSAM ICIP access method is used.

FAST

Specifies that DFSMS Media Manager is used. This is the default value.

If you want to use Media Manager, the IMS HP Fast Path Utilities load module library (HPFP.SHFPLMD0) must be authorized through the Authorized Program Facility (APF).

If the load module library is not authorized, FABCUR1 ignores the specification and uses the VSAM ICIP access method.

Performance considerations:
I/O buffers used for reading entire CIs of an UOW are page-fixed by the ACCESS=FAST option. If you cannot specify enough region size to process areas with large UOW, specify ACCESS=VSAM to avoid performance degrade due to frequent paging activity.

PAD=
This optional keyword defines the padding character to be used when the segment minimum length is increased due to DBD definition change. This keyword is effective for fixed-length segments and variable-length segments.

Code the value as two hexadecimal digits, as follows:

```
X'nn'
```

The default value is null (X'00')

Randomizing routine considerations

For purposes of this document, randomizers are classified as follows:

1. Area-specific: Defines the user-written randomizer that controls an area or area group; its selection is based on root segment key ranges or a similar technique.

   Example: Keys starting with "A" through "L" are randomized across areas 1 through 5. Keys starting with "M" through "Z" are randomized across areas 6 through 10.

2. Nonspecific: Defines a randomizer where the area selection is not controlled by key values. DBFHDC40, as delivered with IMS Fast Path, is nonspecific.

Notes:

1. The randomizer module interface environment that is created by FABCUR1 conforms to the *published* interface as defined in *IMS Exit Routines* with one exception.

2. When a randomizer is invoked by IMS, Registers 10 and 11 contain the addresses of the EPST and ESCD, respectively. Some user-written or user-customized randomizers have been designed to use these addresses to gain access to IMS control blocks other than those passed as part of the published interface. When FABCUR1 issues a call to a randomizing module, Registers 10 and 11 are set to -1 and 0, respectively, to specify that the call is not being issued in a "live" IMS environment. However, if "IMSID=*xxxxxxxx*" is specified in the FABCUR1 EXEC statement, register 11 contains the address of the temporary ESCD created by FABCUR1. Minimum required fields (ESCDSCD, SSCDIMID) in the temporary ESCD and temporary SCD are set by FABCUR1 so that the randomizer can get the IMSID like in the IMS online environment.

3. The randomizer module must be capable of AMODE=31.

AREACTL control statement

The AREACTL statement specifies the area numbers of the areas you are unloading. There must be at least one AREACTL statement in the SYSIN data set.

If HIERCHNG=YES is specified on the DBDNAME control statement, AREACTL=ALL must be specified. If RMODTYPE=G is specified in the DBDNAME control statement and the NEWACB DD statement is present, AREACTL=ALL must be specified. Duplicate references to an area in the AREACTL statements are flagged with a warning message. The AREACTL statement contains the following:

```
AREACTL={ALL|x|(x,y,...)|(x-y)}
```

There are four formats for specifying the areas that you are unloading:

ALL
Unload all areas.

x

    Unload area *x*.

(x,y,...)

    Unload area *x*, area *y*, ...

(x-y)

    Unload area *x*, area *x+1*, ... , area *y*.

Area numbers *x* and *y* are decimal numbers in the range of 1 - 2048.

Comment control statement

The comment statement allows the user to include comments in the control statement stream. It must contain an asterisk (*) in column 1.

FILECTL control statement

The optional FILECTL statement controls grouping of multiple areas' segment data into a single output file. The FILECTL statement contains the following:

```
[FILECTL={zzzzz},{ALL|x|(x,y,...)|(x-y)|(*)}]
```

The FILECTL control statement is composed of:

zzzzz

    Specifies the output file number described by this control statement. *zzzzz* is a decimal number 1 - 2048. The number is specified as a 1-to-5 digit decimal number. For an unloaded file, there must be related DD statements containing this number, DURD*zzz*O and an output data set for the SORT control statement, DURS*zzz*O in the JCL stream for each file specified on a FILECTL control statement. The value of *zzz* or *zzzzz* in the DD name is made up of right-aligned digits, with leading zeros if needed.

ALL

    Unloads the segment data records for all output areas into the DURD*zzz*O data set.

x

    Unloads the segment data records for output area *x* into the DURD*zzz*O data set.

(x,y,...)

    Unloads the segment data records for output area *x*, area *y*, ... into the DURD*zzz*O data set.

(x-y)

    Unloads the segment data records for output area *x*, area *x+1*, ..., area *y* into the DURD*zzz*O data set.

(*)

    Unloads the segment data records for all output areas that are not specified on other FILECTL= control statements into the DURD*zzz*O data set.

Area numbers *x* and *y* are decimal numbers in the range of 1 - 2048.

Instead of DURD*zzz*O, you can specify XD*zzzzz*O.

Instead of DURD*zzz*E, you can specify XD*zzzzz*E if it is specified.

Instead of DURS*zzz*O, you can specify XS*zzzzz*O.

Performance considerations

    The assignment of several areas to a single output file should *not* be done sequentially (for example, specifying file 1 for areas 1, 2, and 3; file 2 for areas 4, 5, and 6; and so on). The areas to be unloaded are processed in the order of their definition in the DMB from OLDACB. Sequentially assigning areas to output files degrades performance by causing output-writer-latch contention (several unload subtasks would be concurrently writing segment data records to the same output file).

    Assume that four subtasks are being used to unload eight areas to four output files. The following table shows the best and worst performance options.

Table 32. FILECTL performance considerations

Best performance	Worst performance
FILECTL=1,(1,5)	FILECTL=1,(1,2)
FILECTL=2,(2,6)	FILECTL=2,(3,4)
FILECTL=3,(3,7)	FILECTL=3,(5,6)
FILECTL=4,(4,8)	FILECTL=4,(7,8)

Empty area considerations

When there are no segment data records for output areas specified in the FILECTL statement, the areas are regarded as empty, and are initialized when reloading.

The area is regarded as empty when:

- The empty output area is specified explicitly in the FILECTL statement, that is, not by 'ALL' or '*' value but by the area number or by the area range.
- The area number of the empty output area is specified explicitly in AREACTL.
- AREACTL='ALL' is specified.

Important: Specifying all areas in AREACTL explicitly is not same as specifying 'ALL' in AREACTL from the viewpoint of the empty output area handling. You should specify 'ALL' in AREACTL if you want to unload all areas of the DEDB.

Default values

If FILECTL statements are not specified, each output area (number *zzzzz*) corresponds to its own DURD*zzz*O or XD*zzzzz*O data set. The DURD*zzz*O or XD*zzzzz*O data set contains all of the segments that FABCUR3 loads into area *zzzzz*, where *zzzzz* is the area number (field DMACRAID in the DMAC control block) assigned to the reference area during ACBGEN processing.

If unload processing is specified with HIERCHNG=YES and/or RMODTYPE=G in the DBDNAME control statement, an output file (and associated SORT control statement file) is required for every area defined in the "output" DMB.

Error conditions

Duplicate references to an area or file in the FILECTL control statements are flagged with an error message, and cause program termination.

TASKCTL control statement

The optional TASKCTL statement specifies the number of unload subtasks, the number of IOVF buffers, and the BASE/DOVF buffer area size parameters. The TASKCTL statement contains the following:

```
[TASKCTL=([subtasks],[iovfbufs],[bdbufsz])]
```

The TASKCTL control statement is composed of:

subtasks

This statement specifies the maximum number of subtasks that can be used to unload the database.

The default value is 8, and the maximum value is 99. If zero is specified, the default value is used. If the number you specify (or default to) is larger than the number of areas being unloaded, there is one subtask for each area being unloaded.

Recommendation: A large value could cause performance degradation because of excessive TCB dispatching activity.

iovfbufs
>This statement specifies the number of buffer areas that are to be allocated to each unload subtask for the processing of IOVF CIs. These buffers are all equal in size to the largest CI-size value defined in the OLDACB DMB for the area(s) being unloaded.

>The default value is calculated as follows:

```
28K x 5 / size of the largest CI of areas to be processed
```

>For example, the number 5 is used for 28 K CI; number 280 is used for a 512-byte CI. The maximum value is 9999. If zero is specified, the default value is used.

>Recommendation: The value of this parameter should reflect, within reason, the maximum number of IOVF CIs used by any unit-of-work in the area(s) being unloaded.

bdbufsz

>⚠️ **CAUTION:** The value of *bdbufsz* can seriously impair unload performance.

>This statement specifies the size of the buffer area to be allocated to each unload subtask for processing BASE and DOVF CIs. The buffer area is *bdbufsz* X 1024 bytes. "K" is not coded on the control statement.

>The default value is 200, and the maximum value is 9999.

>If zero is specified, the default value is used. If the size of the largest UOW is less than *bdbufsz* X 1024 bytes, then the buffer area is reduced to the size of the UOW. The buffer must always be large enough to contain at least one BASE CI plus all of the DOVF CIs in the largest UOW.

>If an explicitly specified number is not large enough to contain one base CI and DOVF CIs in the largest UOW of areas to be processed, then the value is replaced internally to save above the minimum required space.

Error conditions

>Duplicate occurrences of the TASKCTL statement are flagged with a warning message, and the values from the last encountered statement are used.

FORMAT control statement

FABCUR1 supports the enhanced format of an unloaded segment record in which the prefix part of the record is generated based on the maximum number of segment levels defined in DBD. Users can choose the type of the record format with the FORMAT control statement. Format information is stored in the area information record. The FORMAT statement contains the following:

```
[FORMAT={DBT|TFMT}]
```

The control statement specifies the format of the unloaded segment records. The FORMAT control statement is composed of:

DBT
>Specifies that the format of the unloaded segment records is same as DBT 2.x. This is the default value.

TFMT
>Specifies that the format of the unloaded segment records is enhanced, which means that the prefix part of the record is generated based on the maximum number of segment levels defined in DBD.

EXITRTN control statement

The optional EXITRTN statement specifies the name of the user exit routine that will be invoked with an EXPAND function.

For more information on the exit routine, see the topic "Exit routine option and its interface" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

The EXITRTN statement contains the following:

```
[EXITRTN=exit-routine]
```

This optional keyword specifies the name of the user exit routine to be invoked with an EXPAND function.

exit-routine
    Identifies the name of the user exit routine that will be called.

An EXITLIB DD statement must be provided when this control statement is specified.

LOADCTL control statement

The optional LOADCTL statement is a specification of "load control" information. It allows the user to specify an "Insert Limit Count" (ILC) for a dependent segment. ILC is the maximum number of occurrences of that segment which are inserted "close" to their root. The LOADCTL control statement contains the following:

```
[LOADCTL=(segname,nnnn)]
```

The LOADCTL control statement is composed of:

segname
    Specifies the segment name as coded on the SEGM statement in your DBD.

nnnn
    Specifies the maximum number of *segname* segments that can be stored "close" to their root segment. The maximum value is 9999 and the minimum value is 0. The minimum value 0 specifies that all specified segments are stored "far" from their root segment.

⚠ **CAUTION:** Specifying a low value for the Insert Limit Count parameter *nnnn* can seriously impair the performance of subsequent IMS access to the database. (The impact on Reload processing is relatively minimal.) A good "Rule of Thumb" is that no more than 10% of the database records being processed should exceed the ILC value.

Performance considerations

    Database performance can be adversely affected by long dependent segment twin chains. The normal segment insertion strategy, whether loading or reorganizing a database, results in the insertion into the database of all the dependent segments for a given root before processing the segments for the subsequent database records chained off of the same RAP.

    If long dependent segment twin chains are present, these subsequent database records are likely to be placed in the DOVF or IOVF part of the database. This tends to increase the number of I/Os required to access these subsequent database records.

Insert limit count

    Essentially, the ILC parameter *nnnn* allows the user to "split" long dependent segment twin chains. Up to *nnnn* occurrences of the specified dependent segment are inserted normally. The processing of the remainder of the segments on that twin chain is then deferred until all other database records (roots and those of their dependent segments that do not exceed any ILC rules) for the current RAP are processed.

Error conditions

    Duplicate references to a segment in the LOADCTL specifications are flagged with a warning message, and the values from the last encountered statement are used.

LOADPLACE control statement

The optional LOADPLACE statement is a specification of a place, RAP, DOVF, or IOVF. It allows the user to specify a place in which segments selected by the LOADCTL statement are loaded. The LOADPLACE control statement contains the following:

```
[LOADPLACE={RAP|DOVF|IOVF}]
```

The LOADPLACE control statement is composed of:

RAP
All segments that should be "far" from their root segment will be placed in the available RAP CI space. If the space is not enough in the RAP CI, they are placed in the DOVF or the IOVF part.

DOVF
All segments that should be "far" from their root segment will be placed in the available DOVF CI space. If the space is not enough in the DOVF CIs, they are placed in the IOVF part.

IOVF
All segments that should be "far" from their root will be placed in the available IOVF CI space.

If LOADCTL control statement is not specified, the function specified by this control statement is ignored.

PTRERROR control statement

This optional keyword specifies the action of FABCUR1 when a pointer error is detected. The PTRERROR statement contains the following:

```
PTRERROR={ABEND|BYPASS}
```

The PTRERROR control statement is composed of:

ABEND
Specifies that FABCUR1 ends abnormally and an error message is issued. This is the default value.

BYPASS
Specifies that the child or twin segments of any segment that has a pointer error are ignored in the unloading process:

Unloading resumes with the next valid segment in the hierarchical chain. If the error detected is a subset pointer error, the error pointer is ignored.

KEYSEQERROR control statement

The KEYSEQERROR control statement specifies the action for FABCUR1 when a sequence error in a key sequence field is detected.

```
KEYSEQERROR={NOCHECK|ABEND|BYPASS}
```

The KEYSEQERROR control statement is composed of:

NOCHECK
Specifies that FABCUR1 ignores the key sequence check function. This is the default.

ABEND
Specifies that FABCUR1 ends abnormally and an error message is issued.

BYPASS
Specifies that FABCUR1 issues a warning message and skips the error segment and its child segments, and continues the unload processing.

KEYCHGCHECK control statement

The KEYCHGCHECK control statement specifies whether FABCUR1 checks that the key sequence field is not changed by a user exit routine.

```
KEYCHGCHECK={YES|NO}
```

The KEYCHGCHECK control statement is composed of:

YES
Specifies that FABCUR1 checks altered key sequence field. This is the default.

NO
Specifies that FABCUR1 does not check altered key sequence field.

LRECL control statement

The LRECL control statement specifies how to determine the LRECL of an unloaded segment records file when it is not specified in the DD control statement explicitly.

```
LRECL={SEGMENT|SEGTFMT|BLOCK|nnnnn}
```

The LRECL control statement is composed of:

SEGMENT
LRECL is determined based on the maximum length of segments defined in DBD. This is the default.

SEGTFMT
LRECL is determined based on the basis of the maximum length of segments and the maximum number of segment levels defined in DBD by specifying LRECL=SEGTFMT together with the FORMAT=TFMT control statement.

> **Restriction:**
> Even though LRECL=SEGTFMT is specified, the number 3 is used as the maximum number of segment levels when the SDEP=PHYSICAL option is specified and the maximum number of segment level is less than 3.

BLOCK
LRECL is determined as BLKSIZE - 4.

nnnnn
Value of LRECL. The value can be up to 32760.

LRECL is determined based on the following rules.

1. When LRECL is specified in JCL DD, which is an appropriate one, the value is used.
2. The parameter of the LRECL= control statement.
3. The parameter of the LRECL= in site default.
4. When DISP=NEW and no LRECL is specified, LRECL=SEGMENT is used.

When the specified LRECL value is not appropriate, FABCUR1 overrides the value.

SEGMCTL control statement

This optional keyword tells FABCUR1 when the segment records for the specified segment names are written in the output EX*zzzzz*O or EX*zzzzz*E DD data set. The SEGMCTL statement contains the following.

```
SEGMCTL={segname1,....,segnameN|(segname1,....,segnameN)|ALL}
```

The SEGMCTL control statement is composed of:

***segname*1-*segname*N**
Shows the segment name that is defined in the OLDACB. The specified segment records are included in the output.

ALL
Shows the all segment names that are defined in the OLDACB. All segment records are included in the output.

Important: SEGMCTL=*segname* and SEGMCTL=ALL are exclusive. 127 SEGMCTL statements can be specified.

Default values

None.

SDEP segment considerations

The following table shows the relation between SEGMCTL, OLDACB, and SDEP segments:

Table 33. Relation between SEGMCTL, OLDACB, and SDEP segments

OLDACB	SDEP option	SEGMCTL control statement	FABCUR1 processing
SDEP seg defined	SDEP=NO	SEGMCTL=*segname*	U3728 ABEND
		SEGMCTL=ALL	FABC0176W message. No SDEP segment is extracted.
	SDEP=PHYSICAL	SEGMCTL=*segname*	U3728 ABEND
		SEGMCTL=ALL	FABC0176W message. No SDEP segment is extracted.
	SDEP=LOGICAL	SEGMCTL=*segname*	SDEP segments are extracted.
		SEGMCTL=ALL	SDEP segments are extracted.
SDEP seg not defined	SDEP=NO	SEGMCTL=*segname*	U3728 ABEND
		SEGMCTL=ALL	No SDEP segment is extracted.
	SDEP=PHYSICAL	SEGMCTL=*segname*	U3746 ABEND. SDEP option error.
		SEGMCTL=ALL	U3746 ABEND. SDEP option error.
	SDEP=LOGICAL	SEGMCTL=*segname*	U3746 ABEND. SDEP option error.
		SEGMCTL=ALL	U3746 ABEND. SDEP option error.

IMSCATHLQ control statement

The optional IMSCATHLQ statement specifies to retrieve ACBs from the IMS directory (an extension of the IMS catalog) instead of from the ACB library.

```
IMSCATHLQ={bsdshlq|*NO}
```

The IMSCATHLQ control statement is composed of:

bsdshlq
Reads ACBs from the IMS directory by using the IMS Tools Catalog Interface. *bsdshlq* specifies the high-level qualifier of the bootstrap data set of the IMS directory. IMSCATHLQ=*bsdshlq* is effective when the IMS catalog and the IMS management of ACBs are enabled.

If a TYPE=CATDSHLQ statement was used to create a DFSMDA member in your MDA library, you must specify the SYSDSHLQ parameter used in that DFSMDA member for the IMSCATHLQ keyword parameter.

*NO
Reads ACBs from the ACB library. This is the default value.

IMSCATACB_INPUT control statement

The optional IMSCATACB_INPUT statement specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. IMSCATACB_INPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* option is specified.

```
IMSCATACB_INPUT={CURRENT|PENDING}
```

The IMSCATACB_INPUT control statement is composed of:

CURRENT
Currently active ACB members are retrieved from the IMS directory data sets. This is the default value.

PENDING
Pending ACB members are retrieved from the staging data set.

IMSCATACB_OUTPUT control statement

The optional IMSCATACB_OUTPUT statement specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. This keyword specifies the new ACB definition, which is used when an ACB definition change is requested. IMSCATACB_OUTPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* is specified.

```
IMSCATACB_OUTPUT={CURRENT|PENDING}
```

The IMSCATACB_OUTPUT control statement is composed of:

CURRENT
Currently active ACB members are retrieved from the IMS directory data sets.

PENDING
Pending ACB members are retrieved from the staging data set.

Output for DEDB Unload

The DEDB Unload utility generates output reports (that is, Audit Control Reports and message reports).
Related reference
Site default support for FPB
This topic describes the site default support for FPB.

DEDB Unload DURAUDIT DD data set

The unload program (FABCUR1) generates a three-part Unload Audit Control report to provide verification totals.

Format

This data set contains 133-byte, fixed-length records. Block size, if coded in your JCL, must be a multiple of 133. It is better to code your DD statement as follows:

```
//DURAUDIT DD SYSOUT=A
```

Unload Audit Control report

1. Part 1: SEGMENTS UNLOADED FROM DATABASE *xxxxxxxx*

   This section of the report provides a count of the number of segments (by segment name and segment code) that were unloaded from each area of the database. A database total is also provided.

   The segment counts for each area should match the statistics generated by DEDB Pointer Checker when it is run against the same data set. Comparing the unload segment counts with those generated

by DEDB Pointer Checker provides a means of verifying that all segments were unloaded. If segment counts for an area is zero, it means that the unloaded area is empty.

The number of 512-byte blocks is shown on DURAUDIT of Unload/Reload in case of SDEP=PHYSICAL. Text '(PHYSICAL)' is printed after the number so that the audit control report of Unload/Reload implies SDEP=PHYSICAL.

```
IMS HPFP UTILITIES - DEDBUR                        "UNLOAD AUDIT CONTROL REPORT"                          PAGE:    1
5698-FPP                                           DATE: 11/22/2020  TIME: 20.41.39                       FABCUR1  -  V2R1


    SEGMENTS UNLOADED FROM DATABASE VRSDSRF  :

       AREA     AREA    SEG     SEG        NO SEGS
        NO     DDNAME   CODE    NAME       UNLOADED
       ----   --------  ----  --------   -----------

        1     VRSTSS1    1    TSSROOT            87
                         2    TSSSEQ1        (SDEP)
                         3    TSSDIR1           337
                         4    TSSDIR2           737
                         5    TSSDIR3           176
                         6    TSSDIR4           205
                         7    TSSDIR5            87
                         8    TSSDIR6           289
                                          -----------
                 **  AREA TOTALS  **           1,918  (VRSTSS1)

        2     VRSTSS2    1    TSSROOT            85
                         2    TSSSEQ1        (SDEP)
                         3    TSSDIR1           345
                         4    TSSDIR2           667
                         5    TSSDIR3           214
                         6    TSSDIR4           198
                         7    TSSDIR5            85
                         8    TSSDIR6           301
                                          -----------
                 **  AREA TOTALS  **           1,895  (VRSTSS2)

        3     VRSTSS3    1    TSSROOT            83
                         2    TSSSEQ1        (SDEP)
                         3    TSSDIR1           318
                         4    TSSDIR2           691
                         5    TSSDIR3           173
                         6    TSSDIR4           161
                         7    TSSDIR5            83
                         8    TSSDIR6           266
                                          -----------
                 **  AREA TOTALS  **           1,775  (VRSTSS3)
                                          -----------
                 *** DATABASE TOTAL ***        5,588
```

Figure 46. Unload Audit Control report (Segments unloaded)

2. Part 2: SEGMENTS TO BE RELOADED TO DATABASE *xxxxxxxx*

This section of the report provides a count of the number of segments that are to be reloaded to each area of the new database and to the database total.

If the unload/reload is a simple reorganization of one or more areas, the segment counts and area totals for Part 1 and Part 2 will be the same. The database totals should *always* be the same. If segment counts for an area are zero, it means that the reloaded area is empty.

The number of 512-byte blocks is shown on DURAUDIT of Unload/Reload in case of SDEP=PHYSICAL. Text '(PHYSICAL)' is printed after the number so that the audit control report of Unload/Reload implies SDEP=PHYSICAL.

```
        SEGMENTS TO BE RELOADED TO DATABASE VRSDSRF  :

          AREA    AREA    SEG     SEG        NO SEGS
           NO    DDNAME   CODE    NAME       UNLOADED
          ----   -------- ----   --------   ----------

           1     VRSTSS1    1     TSSROOT          87
                            2     TSSSEQ1       (SDEP)
                            3     TSSDIR1         337
                            4     TSSD11            0
                            5     TSSD111           0
                            6     TSSD12            0
                            7     TSSDIR2         737
                            8     TSSDIR3         176
                            9     TSSDIR4         205
                           10     TSSD41            0
                           11     TSSDIR5          87
                           12     TSSDIR6         289
                           13     TSSD61            0
                           14     TSSD611           0
                           15     TSSD612           0
                           16     TSSD613           0
                                              -----------
                   **  AREA TOTALS  **             1,918   (VRSTSS1)

           2     VRSTSS2    1     TSSROOT          85
                            2     TSSSEQ1       (SDEP)
                            3     TSSDIR1         345
                            4     TSSD11            0
                            5     TSSD111           0
                            6     TSSD12            0
                            7     TSSDIR2         667
                            8     TSSDIR3         214
                            9     TSSDIR4         198
                           10     TSSD41            0
                           11     TSSDIR5          85
                           12     TSSDIR6         301
                           13     TSSD61            0
                           14     TSSD611           0
                           15     TSSD612           0
                           16     TSSD613           0
                                              -----------
                   **  AREA TOTALS  **             1,895   (VRSTSS2)
```

```
          AREA    AREA    SEG     SEG        NO SEGS
           NO    DDNAME   CODE    NAME       UNLOADED
          ----   -------- ----   --------   ----------

           3     VRSTSS3    1     TSSROOT          83
                            2     TSSSEQ1       (SDEP)
                            3     TSSDIR1         318
                            4     TSSD11            0
                            5     TSSD111           0
                            6     TSSD12            0
                            7     TSSDIR2         691
                            8     TSSDIR3         173
                            9     TSSDIR4         161
                           10     TSSD41            0
                           11     TSSDIR5          83
                           12     TSSDIR6         266
                           13     TSSD61            0
                           14     TSSD611           0
                           15     TSSD612           0
                           16     TSSD613           0
                                              -----------
                   **  AREA TOTALS  **             1,775   (VRSTSS3)
                                              -----------
                   *** DATABASE TOTAL ***          5,588
```

Figure 47. Unload Audit Control report (Segments to be reloaded)

3. Part 3: SEGMENT TOTALS BY OUTPUT FILE

This section of the report provides segment counts and area totals by the output file ddname. File totals and a database total are also provided.

The area totals should match the area totals in Part 2. The file totals are ultimately verified against the reload file totals. The database total should match the two preceding database totals.

If FILECTL statements are not used (that is, ddnames default to area numbers), there is only one area per file. Conversely, if FILECTL statements are used, a file may contain data for more than one area.

The number of 512-byte blocks is shown on DURAUDIT of Unload/Reload in case of SDEP=PHYSICAL. Text '(PHYSICAL)' is printed after the number so that the audit control report of Unload/Reload implies SDEP=PHYSICAL.

```
        SEGMENT TOTALS BY OUTPUT FILE:

            FILE    AREA  SEG   SEG        NUMBER
            DDNAME   NO   CODE  NAME      SEGMENTS
           --------  ----  ----  --------  -----------

           DURD0010   1    1    TSSROOT          87
                           2    TSSSEQ1       (SDEP)
                           3    TSSDIR1         337
                           4    TSSD11            0
                           5    TSSD111           0
                           6    TSSD12            0
                           7    TSSDIR2         737
                           8    TSSDIR3         176
                           9    TSSDIR4         205
                          10    TSSD41            0
                          11    TSSDIR5          87
                          12    TSSDIR6         289
                          13    TSSD61            0
                          14    TSSD611           0
                          15    TSSD612           0
                          16    TSSD613           0
                                            -----------
                   **  AREA TOTALS  **            1,918
                                            -----------
                   *** FILE TOTALS  ***                        1,918


           DURD0020   2    1    TSSROOT          85
                           2    TSSSEQ1       (SDEP)
                           3    TSSDIR1         345
                           4    TSSD11            0
                           5    TSSD111           0
                           6    TSSD12            0
                           7    TSSDIR2         667
                           8    TSSDIR3         214
                           9    TSSDIR4         198
                          10    TSSD41            0
                          11    TSSDIR5          85
                          12    TSSDIR6         301
                          13    TSSD61            0
                          14    TSSD611           0
                          15    TSSD612           0
                          16    TSSD613           0
                                            -----------
                   **  AREA TOTALS  **            1,895
                                            -----------
                   *** FILE TOTALS  ***                        1,895
```

```
           DURD0030   3    1    TSSROOT          83
                           2    TSSSEQ1       (SDEP)
                           3    TSSDIR1         318
                           4    TSSD11            0
                           5    TSSD111           0
                           6    TSSD12            0
                           7    TSSDIR2         691
                           8    TSSDIR3         173
                           9    TSSDIR4         161
                          10    TSSD41            0
                          11    TSSDIR5          83
                          12    TSSDIR6         266
                          13    TSSD61            0
                          14    TSSD611           0
                          15    TSSD612           0
                          16    TSSD613           0
                                            -----------
                   **  AREA TOTALS  **            1,775
                                            -----------
                   *** FILE TOTALS  ***                        1,775
                                                            -----------
                   **** DATABASE TOTAL ****                     5,588
```

Figure 48. Unload Audit Control report (Segment totals by output file)

DEDB Unload SYSPRINT DD data set

The SYSPRINT data set contains the messages issued by the FABCUR1 program.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133. It is better to code your DD statement as follows:

```
//SYSPRINT DD SYSOUT=A
```

Unload processor-Messages

The following figure shows an example of the Messages report.

```
IMS HPFP UTILITIES - DEDBUR                    "UNLOAD PROCESSOR - MESSAGES"                          PAGE:    1
5698-FPP                                       DATE: 11/22/2020  TIME: 20.41.39                       FABCUR1 -  V2R1

  FABC0120I - CARD  1: DBDNAME=VRSDSRF    STATS    HIERCHNG=YES
  FABC0120I - CARD  2: AREACTL=ALL
  FABC0120I - CARD  3: TASKCTL=(5,25,67)

  FABC0165I - STRUCTURE CHANGE DETECTED FOR SEGMENT: TSSDIR2
              - "OLD" SEG-CD:   4
              - "NEW" SEG-CD:   7

  FABC0165I - STRUCTURE CHANGE DETECTED FOR SEGMENT: TSSDIR3
              - "OLD" SEG-CD:   5
              - "NEW" SEG-CD:   8

  FABC0165I - STRUCTURE CHANGE DETECTED FOR SEGMENT: TSSDIR4
              - "OLD" SEG-CD:   6
              - "NEW" SEG-CD:   9

  FABC0165I - STRUCTURE CHANGE DETECTED FOR SEGMENT: TSSDIR5
              - "OLD" SEG-CD:   7
              - "NEW" SEG-CD:  11

  FABC0165I - STRUCTURE CHANGE DETECTED FOR SEGMENT: TSSDIR6
              - "OLD" SEG-CD:   8
              - "NEW" SEG-CD:  12

  FABC0169I - AUTHORIZED STRUCTURE CHANGE(S) DETECTED

  FABC0115I - SCHEDULING PARAMETERS: NO. UTASK'S:  3  UOW BFR SIZE:    55,296 BYTES

  FABC0105I - PROCESSING COMMENCES FOR AREA   1  (AREANAME: VRSTSS1 )      (BUFND =  21)

  FABC0105I - PROCESSING COMMENCES FOR AREA   2  (AREANAME: VRSTSS2 )      (BUFND =  37)

  FABC0105I - PROCESSING COMMENCES FOR AREA   3  (AREANAME: VRSTSS3 )      (BUFND =  46 (OVERRIDDEN FROM  21) )

  FABC0106I - PROCESSING COMPLETED FOR AREA   1  (AREANAME: VRSTSS1 )      (   1 /          0)

  FABC0106I - PROCESSING COMPLETED FOR AREA   3  (AREANAME: VRSTSS3 )      (   3 /          0)

  FABC0106I - PROCESSING COMPLETED FOR AREA   2  (AREANAME: VRSTSS2 )      (   2 /          0)

  FABC0100W - FABCUR1 ENDED WITH WARNINGS
```

Figure 49. Unload processor-Messages

Examples for DEDB Unload

There are many ways to use the DEDB Unload utility. The examples presented in this topic represent some of the typical ways that you can use.

By studying and understanding these examples, you can learn the techniques to use and to effectively manage your IMS DEDBs.

Example 1: Reorganizing a single area

The following figure shows example JCL statement for reorganizing a single area. One output file is produced.

First, the following data must be assembled:

- The database name is TSSDBD1, and four areas are defined.

- The fourth area defined in the DMB is to be reorganized, has a data set name of TSSV.TSSDBD1.TSS1AR04, and is defined as follows:

```
        AREA      DD1=TSS1AR04,                                                     C
                  SIZE=4096,                                                        C
                  UOW=(20,5),                                                       C
                  ROOT=(100,12)
```

- The maximum number of IOVF CIs used by any UOW in this area is 7.

The JCL stream for this unload job can now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in .

2. The control statement stream is coded as follows:

```
DBDNAME=TSSDBD1 REORG STATS

AREACTL=4

TASKCTL=(1,7)
```

3. A DD statement for the area being processed must be included.

```
//TSS1AR04  DD DSN=TSSV.TSSDBD1.TSS1AR04,DISP=OLD
```

4. DD statements for the output file, and its associated SORT control statement data sets must be included.

```
//DURD0040 DD DSN=HPFP.UR.FILE004.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(18,3),RLSE)
//DURS0040 DD DSN=HPFP.UR.FILE004.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
```

5. Since a reorganization is being done, and the output file contains segment records from only one area, the SORT step may be omitted.

Now you can run the unload job.

Example 2: Reorganizing a database with three areas

The following figure shows example JCL statement for reorganizing three areas. Two output files are produced.

First, gather the region-size calculation data:

- The database name is TSSDBD2, and three areas are defined.
- The area data set names are TSSV.TSSDBD2.TSS2AR01, TSSV.TSSDBD2.TSS2AR02, and TSSV.TSSDBD2.TSS2AR03.

  The areas are defined as follows:

```
AREA      DD1=TSS2AR01,                                   C
          SIZE=4096,                                      C
          UOW=(20,5),                                     C
          ROOT=(100,12)
AREA      DD1=TSS2AR02,                                   C
          SIZE=2048,                                      C
          UOW=(30,8),                                     C
          ROOT=(160,12)
AREA      DD1=TSS2AR03,                                   C
          SIZE=4096,                                      C
          UOW=(25,5),                                     C
          ROOT=(120,12)
```

- Since the maximum block size for this device is large (32 K), and there are multiple output files, the output file DD statements are coded with DCB=BLKSIZE=22000.
- The maximum number of IOVF CIs used by any UOW in this area is nine.
- Three unload subtasks are used.

The JCL stream for this unload job can now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in .

2. The control statement stream is coded as follows:

```
DBDNAME=TSSDBD2 REORG STATS
```

```
            AREACTL=ALL
            TASKCTL=(3,9)
            FILECTL=1,(1,3)

            FILECTL=2,(2)
```

3. The DD statements for the areas being processed must be included:

```
//TSS2AR01  DD DSN=TSSV.TSSDBD2.TSS2AR01,DISP=OLD
//TSS2AR02  DD DSN=TSSV.TSSDBD2.TSS2AR02,DISP=OLD
//TSS2AR03  DD DSN=TSSV.TSSDBD2.TSS2AR03,DISP=OLD
```

4. The DD statements for the output files and associated SORT control statement data sets, must be included:

```
//DURD0010 DD DSN=HPFP.UR.FILE001.SEGDATA,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(65,7),RLSE),
//             DCB=BLKSIZE=22000
//DURS0010 DD DSN=HPFP.UR.FILE001.SORTCARD,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(TRK,(1,1))
//DURD0020 DD DSN=HPFP.UR.FILE002.SEGDATA,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(30,5),RLSE),
//             DCB=BLKSIZE=22000
//DURS0020 DD DSN=HPFP.UR.FILE002.SORTCARD,
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(TRK,(1,1))
```

5. Set up JCL to sort the output file containing data from multiple areas (HPFP.UR.FILE001.SEGDATA). The second output file does not need to be sorted because REORG processing is being performed and it contains data from only one area.

Now you can run the unload job.

Example 3: Expanding the data set of one area of a database

The following figure shows example JCL statement for expanding the data set of one area of a database.

The database in this unload example is under the control of an "area-specific" randomizer; this will expand the data set of one area.

First, gather the region-size calculation data:

- The database name is TSSDBD3, and three areas are defined.
- The area being expanded is the second area in the database, has a data set name of TSSV.TSSDBD3.TSS3AR02, and is currently defined as follows:

```
    AREA        DD1=TSS3AR02,                                    C
                SIZE=2048,                                       C
                UOW=(30,8),                                      C
                ROOT=(100,12)
```

- The area definition is being changed to:

```
    AREA        DD1=TSS3AR02,                                    C
                SIZE=4096,                                       C
                UOW=(20,4),                                      C
                ROOT=(160,10)
```

- The output file resides on a tape (Max. BLKSIZE = 32k).
- The maximum number of IOVF CIs used by any UOW in this area is 10.

The JCL stream for this unload job can now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in "Region size requirements for DEDB Unload" on page 122.

2. The control statements stream is coded as follows:

```
DBDNAME=TSSDBD3 RMODTYPE=S STATS

AREACTL=2
TASKCTL=(,10)

FILECTL=1,2
```

3. A DD statement for the area being processed must be included:

```
//TSS3AR02  DD DSN=TSSV.TSSDBD3.TSS3AR02,DISP=OLD
```

4. DD statements for the output file, and associated SORT control statement data set, must be included:

```
//DURD001O DD DSN=HPFP.UR.FILE001.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE
//DURS001O DD DSN=HPFP.UR.FILE001.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
```

5. Set up JCL to sort the output file.

   Note: The sort is required since the number of RAPs in the area has been changed. A given database record (root and dependents) is not necessarily chained off its original RAP. Therefore, the reloaded area is sequenced differently from the original area.

Now you can run the unload job.

Example 4: Expanding a three-area database to five areas (generating five outputs)

The following figure shows example JCL statement for expanding a three-area database to five areas (five outputs).

The database is under the control of a "general" randomizer. Five output files are produced.

First, gather the region-size calculation data:

- The database name is TSSDBD4.

- The three areas are defined with:

```
SIZE=4096,                                                      C
UOW=(20,8),                                                     C
ROOT=(120,14)
```

- The additional two areas in the new configuration are defined with the same values.

- The output files reside on tape. Since the maximum block size for this device is large (32 K), and there are multiple output files, the output file DD statements are coded with DCB=BLKSIZE=16000.

- The maximum number of IOVF CIs used by any UOW in this area is seven.

The JCL stream for this unload job can now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in "Region size requirements for DEDB Unload" on page 122.

2. The control statements stream is coded as follows:

```
DBDNAME=TSSDBD4 RMODTYPE=G STATS

AREACTL=ALL
```

```
        TASKCTL=(,7)
```

3. DD statements for the areas being processed must be included:

```
//TSS4AR01  DD DSN=TSSV.TSSDBD4.TSS4AR01,DISP=OLD
//TSS4AR02  DD DSN=TSSV.TSSDBD4.TSS4AR02,DISP=OLD
//TSS4AR03  DD DSN=TSSV.TSSDBD4.TSS4AR03,DISP=OLD
```

4. The DD statements for the output files, and associated SORT control statement data sets, must be included:

```
//DURD0010 DD DSN=HPFP.UR.FILE001.SEGDATA,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=TAPE,
//           DCB=BLKSIZE=16000
//DURS0010 DD DSN=HPFP.UR.FILE001.SORTCARD,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,
//           SPACE=(TRK,(1,1))
     .
     .
     .
//DURD0050 DD DSN=HPFP.UR.FILE005.SEGDATA,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=TAPE,
//           DCB=BLKSIZE=16000
//DURS0050 DD DSN=HPFP.UR.FILE005.SORTCARD,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,
//           SPACE=(TRK,(1,1))
```

5. Set up JCL to sort the output files. The sort is required since a "general" randomizer distributes segments across all the RAPs in the database, and the total number of RAPs in the database has been changed (that is, a given database record *root and dependents* is usually not chained off of the same RAP as in the old configuration of the database).

Now you can run the unload job.

Example 5: Expanding a three-area database to five areas (generating three outputs) when IMS management of ACBs is enabled

The following figure shows example JCL statement for expanding a three-area database to five areas (three outputs).

The database is under the control of a "general" randomizer. Three output files are produced.

This unload example is similar to . The only differences are:

• The number of output files
• ACB definitions are retrieved from the IMS directory

First, gather the region-size calculation data:

• The database name is TSSDBD5.
• The three areas are defined with:

```
SIZE=4096,                                                       C
UOW=(20,8),                                                      C
ROOT=(120,14)
```

• The additional two areas in the new configuration are defined with the same values.
• The output files reside on tape. Since the maximum block size for this device is large (32 K), and there are multiple output files, the output file DD statements are coded with DCB=BLKSIZE=16000.
• The maximum number of IOVF CIs used by any UOW in this area is seven.

The JCL stream for this unload job can now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in "Region size requirements for DEDB Unload" on page 122.

2. The control statements stream is coded as follows:

```
DBDNAME=TSSDBD5 RMODTYPE=G STATS

AREACTL=ALL
TASKCTL=(,7)
FILECTL=1,(1,4)
FILECTL=2,(2,5)
FILECTL=3,3
IMSCATHLQ=IMSVS.CATALOG
IMSCATACB_INPUT=CURRENT
IMSCATACB_OUTPUT=PENDING
```

3. DD statements for the areas being processed must be included:

```
//TSS5AR01  DD DSN=TSSV.TSSDBD5.TSS5AR01,DISP=OLD
//TSS5AR02  DD DSN=TSSV.TSSDBD5.TSS5AR02,DISP=OLD
//TSS5AR03  DD DSN=TSSV.TSSDBD5.TSS5AR03,DISP=OLD.
```

4. DD statements for the output files, and associated SORT control statement data sets, must be included:

```
//DURD0010 DD DSN=HPFP.UR.FILE001.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=BLKSIZE=16000
//DURS0010 DD DSN=HPFP.UR.FILE001.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
     .
     .
     .
//DURD0030 DD DSN=HPFP.UR.FILE003.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=BLKSIZE=16000
//DURS0030 DD DSN=HPFP.UR.FILE003.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
```

5. Set up JCL to sort the output files. The SORT is required since a "general" randomizer distributes segments across all the RAPs in the database, and the total number of RAPs in the database has been changed (that is, a given database record *root and dependents* is usually not chained off of the same RAP as in the old configuration of the database).

Now you can run the unload job.

Example 6: Unloading a twenty-area database

The following figure shows example JCL statement for unloading a twenty-area database.

In this unload example, the segment hierarchy is changed in the new DMB. Six unload subtasks are used. Five output files are produced.

First, gather the region-size calculation data:

- The database name is TSSDBD6.

- The 20 areas are defined with:

```
SIZE=4096,                                                      C
UOW=(20,8),                                                     C
ROOT=(120,14)
```

- The output files reside on tape. Since the maximum block size for this device is large (32 K), and there are multiple output files, the output file DD statements are coded with DCB=BLKSIZE=16000.

- The maximum number of IOVF CIs used by any UOW in this area is five.

The JCL stream for this unload job can now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in "Region size requirements for DEDB Unload" on page 122.

2. The control statement stream is coded as follows:

```
DBDNAME=TSSDBD6 RMODTYPE=G STATS

AREACTL=ALL
TASKCTL=(6,5)
FILECTL=1,(1,6,11,16)
FILECTL=2,(2,7,12,17)
FILECTL=3,(3,8,13,18)
FILECTL=4,(4,9,14,19)

FILECTL=5,(*)
```

3. The DD statements for the areas being processed must be included:

```
//TSS6AR01  DD DSN=TSSV.TSSDBD6.TSS6AR01,DISP=OLD
    .
    .
    .
//TSS6AR20  DD DSN=TSSV.TSSDBD6.TSS6AR20,DISP=OLD
```

4. The DD statements for the output files and the associated SORT control statement data sets must be included:

```
//DURD0010 DD DSN=HPFP.UR.FILE001.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=BLKSIZE=16000
//DURS0010 DD DSN=HPFP.UR.FILE001.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
      .
      .
      .
//DURD0050 DD DSN=HPFP.UR.FILE005.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=BLKSIZE=16000
//DURS0050 DD DSN=HPFP.UR.FILE005.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
```

5. Set up JCL to sort the output files. The SORT is required since the segment data records are not in the correct order (because of the segment hierarchy being changed).

Now you can run the unload job.

Example 7: Unloading a database

The following figure shows example JCL statement for unloading a database.

The database in this example is defined with a very large unit-of-work size.

First, gather the region-size calculation data:

- The database name is TSSDBD7.

- The five areas are defined with:

```
SIZE=4096,                                                          C
UOW=(300,35),                                                       C
ROOT=(800,85)
```

- The output files reside on tape. Since the maximum block size for this device is large (32 K), and there are multiple output files, the output file DD statements are coded with DCB=BLKSIZE=20000.
- The maximum number of IOVF CIs used by any UOW in this area is 18.
- Because of the large UOW size, the TASKCTL *bdbufsz* parameter is specified as 300K.

The JCL stream for this unload job may now be set up:

1. The EXEC statement for FABCUR1 is coded with REGION=*xxxx*. Where *xxxx* is the value that is estimated based on the worksheet provided in "Region size requirements for DEDB Unload" on page 122.

2. The control statement stream is coded as follows:

```
DBDNAME=TSSDBD7 REORG STATS

AREACTL=ALL

TASKCTL=(,18,300)
```

3. The DD statements for the areas being processed must be included:

```
//TSS7AR01  DD DSN=TSSV.TSSDBD7.TSS7AR01,DISP=OLD
       .
       .
//TSS7AR05  DD DSN=TSSV.TSSDBD7.TSS7AR05,DISP=OLD
```

4. The DD statements for the output files and the associated SORT control statement data sets must be included:

```
//DURD001O DD DSN=HPFP.UR.FILE001.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=BLKSIZE=20000
//DURS001O DD DSN=HPFP.UR.FILE001.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
      .
      .
      .
//DURD005O DD DSN=HPFP.UR.FILE005.SEGDATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=BLKSIZE=20000
//DURS005O DD DSN=HPFP.UR.FILE005.SORTCARD,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1))
```

5. Since reorganizing is being done, and the output files contain segment records from only one area, the SORT step may be omitted.

Now you can run the unload job.

Chapter 5. Reloading a DEDB offline with FPB

You can reload a DEDB by using the DEDB Reload utility of FPB.

The reloading function in DEDB Unload/Reload is run by program FABCUR3.

The DEDB Reload utility (FABCUR3) efficiently reloads a single DEDB area, or concurrently reload multiple DEDB areas, without impacting all areas of a database. They can be approximately 10 to 20 times faster than the equivalent function using "GN" and "ISRT" processing under the control of IMS.

Topics:

- "Functions of DEDB Reload" on page 155
- "Data and system flow of DEDB Reload" on page 159
- "Restrictions of DEDB Reload" on page 160
- "Typical scenario and processes to run for unloading and reloading a DEDB" on page 160
- "Running DEDB Reload" on page 161
- "Region size requirements for DEDB Reload" on page 162
- "DD statements for DEDB Reload" on page 163
- "Input for DEDB Reload" on page 172
- "Output for DEDB Reload" on page 178
- "Examples for DEDB Reload" on page 181

Functions of DEDB Reload

DEDB Reload can efficiently unload and reload a single DEDB area. DEDB Unload can concurrently unload and multiple DEDB areas, without impacting all areas of a database.

Subsections:

- "Features" on page 155
- "General structure" on page 156
- "Hierarchical structure changes" on page 157
- "Modes" on page 158

Features

This component provides besides the unload and the reload function, a set of support utilities that can be used for unload and reload. It offers the following features:

- Both the unload and reload programs function independently of the IMS Control Region. Hence, one or more areas of a multi-area database can be unloaded or reloaded while the application continues to use the areas that are not included in the maintenance process. This feature can significantly increase application availability.

  Note: This requires specific control of area selection and application code capable of handling "FH" status codes.

- Both processors can concurrently process multiple database areas with no database contention. This concurrent processing capability significantly decreases the time required for database maintenance, and it further increases application availability.

- During the unload/reload process, any or all of the following database specifications can be changed:

  - DBD name
  - Number of database areas

- Randomizing module
- Segment edit/compression routine
- UOW parameter values
- ROOT parameter values
- CI size
- Size of the VSAM data set
- Pointer options
- Addition of new segment
- Change of existing segment hierarchical structure

Notes:

1. These DBD definition changes will be applied only during unload process by specifying an ACB library that has new DBD definition member to the NEWACB DD statement. It implies that unloaded segment records produced by the unload process will be composed on the basis of the new DBD definition information. DBD definition change cannot be specified in the reload process (FABCUR3).

2. For how to add a new segment or to change an existing segment hierarchical structure, see "Hierarchical structure changes" on page 157. Existing segment names cannot be changed, nor can existing segments be deleted.

- During the unload process, a second copy option may be specified in order to prepare two sets of unloaded files. With this option specified, the unload processing will continue even if one of the copies encounters an I/O error. This function is very effective for the users with big databases.

- During the unload process, an empty area unloaded by this utility is clearly identified with a warning message. Then during the reload process, an output data set for this empty area is initialized with no segment in order to prevent IMS DB/DC accessing trouble. The Audit Control report also shows the empty DEDB area.

- During the reload process, abnormally long dependent segment twin chains can be controlled so that the impact on other database records in the same RAP CI is minimized. This feature is implemented via a user-specified limit on the number of segment occurrences that are to be placed "near" the root.

- The new area and RAP values for the database record are determined during unload processing. This allows all records for an area to be written to the same output data set, avoids an extra pass of the file, and simplifies the reload process.

- There are no source code modifications made to any user-written or IMS program or control block. Because IMS program integrity is maintained, new exposures are not introduced.

- DEDB Unload/Reload enables users to produce an expanded-format unloaded data set from compressed segments.

- The reload program can reload segment data in one area into multi-area data sets.

- The reload program can produce image copy data set(s) of an unloaded area.

- Under the image copy option, the reloaded program can do a fast scan (HASH check) of the integrity of the unloaded area.

General structure

The DEDB Unload/Reload utility consists of two functional components that operate independently of the IMS control region.

The unload processor is a z/OS batch program that can concurrently unload multiple areas of a DEDB. But there are basically two problems that hinder DEDB unload/reload processing if operated under the control of the IMS control region:

1. To prohibit access to the areas involved in the maintenance process, all transactions that access the database must be stopped or made logically unavailable to the application. If the transactions must be stopped, application availability is adversely impacted.

2. DEDB Unload/Reload processing (or any sequential process) that uses "GN" and "ISRT" calls is exceedingly slow. The basic problem with this approach is the number of EXCPs required. An unload using "GN" processing issues one EXCP for each CI that contains data. A reload using "GN" and "ISRT" processing requires one EXCP to retrieve a CI and another EXCP to rewrite the CI. Additional EXCPs are incurred if the NBA is not large enough to hold a logical UOW (that is, the RAA BASE section CIs plus all IOVF CIs logically owned by the UOW).

The DEDB Unload/Reload utility achieves significant performance improvements (that is, elapsed time reduction) by reducing the number of EXCPs issued. Both programs use the VSAM read ahead facility to minimize the number of EXCPs (that is, a conscious decision was made to reduce EXCPs at the expense of increased memory usage). Both the unload and reload programs, if buffered correctly, read and write a minimum of 23 2048-byte CIs per EXCP for the root addressable section of an area. This number can be further increased (that is, elapsed time reduced) by providing additional VSAM buffers (within limitations).

In addition, all IMS overheads (that is, logging or sync point processing) have been eliminated, and the instruction path length to retrieve or insert a segment is exceedingly short.

DEDB areas requiring maintenance are deallocated from the IMS control region. The unload processor, after first ensuring that the maintenance request is valid and that all required resources are present, attaches and manages a user-specified number of subtasks. Each subtask is responsible for unloading a specific area. When completed, the subtask returns to the main task for assignment of another area or termination. As each database segment is unloaded, the subtask invokes the randomizer (if required) to determine the new area/RAP values. All subtasks share a common output writer. This allows all database records for an area to be written to the same output file. This approach simplifies the reloading process, and eliminates an extra pass of the unloaded segment file.

The number of concurrent unload subtasks depends on the UOW size and the number of required IOVF buffers. This is explained in detail in "Hierarchical structure changes" on page 157.

After the VSAM clusters have been deleted and redefined, multiple reload jobs can be initiated to concurrently reload the areas. The only limitation on the number of concurrent reload jobs is the number of available initiators. You can also reload the areas in a single job step.

Hierarchical structure changes

During the unload/reload process, the segments can be moved in the hierarchical structure, and the new segments can be defined if the following rules are not violated. FABCUR1 detects invalid structure changes.

1. The existing segment names cannot be changed.

2. The new segments can be added, but the existing segments cannot be deleted.

3. When a segment is being moved in the hierarchical structure (that is, its segment code is being changed), it must remain at the same hierarchical level.

4. A segment must have the same parent after being moved in the hierarchical structure.

The following table shows examples of hierarchical structure change of the unload and reload processes.

Table 34. Examples of hierarchical structure change

Segname	Old segcode	New segcode
ROOTSEG	1	1
SEQDEP	2	2
NEWDIR	NEW	3
DIRDEP5	7	4
DIRDEP1	3	5
DIRDEP2	4	6

Table 34. Examples of hierarchical structure change (continued)

Segname	Old segcode	New segcode
DIRDEP3	5	7
DIRDEP4	6	8

Modes

There are two basic modes to use the DEDB Unload/Reload utility. These are called "reorg" mode and "change" mode. Setting your JCL and control statements depends heavily on the mode you are using. This topic describes the techniques about how to use for each type of unload/reload.

Reorg mode

This mode is used when you do not change the database structure, the DBD, or the randomizing routine. The only change you can make in "reorg" mode is to increase or decrease the size of the SDEP part of an area. When SDEP=LOGICAL|PHYSICAL is specified, you should not decrease the size of the SDEP part of an area.

A "reorg" mode FABCUR1 run should be set up in the following manner:

- Code *REORG* and *STATS* on your DBDNAME control statement.
- Do *not* code *HIERCHNG=* or *RMODTYPE=* on your DBDNAME control statement.
- Do *not* include *NEWACB* or *RMODLIB* DD statements in your JCL.

Change mode

This mode is used when you change the database structure, the DBD, or the randomizing routine.

A "change" mode FABCUR1 run should be set up in the following manner:

- Do *not* code *REORG* on your DBDNAME control statement.
- If you are changing the segment hierarchy (that is, one or more segment codes are being changed), then code *HIERCHNG=YES/YESFORCE* on your DBDNAME control statement.

- If your randomizing routine is "area-specific," and if you are not unloading all areas, code the *RMODTYPE=S* on your DBDNAME control statement.
- Include the *NEWACB* and *RMODLIB* DD statements in your JCL.

Change Database Definition

Specify an ACB library that includes the current DBD type ACB in the OLDACB DD, and an ACB library that includes the new database definition DBD in the NEWACB DD.

Change Randomizer module

Specify a library that includes the NEW randomizer module in RMODLIB. The old randomizer is not needed for unloading. Do not concatenate the library that includes the old randomizer ahead of the new randomizer library, if randomizer name is the same.

Notice: If you do not change the randomizer, you should specify the library that includes the current randomizer module in the RMODLIB DD of FABCUR1.

Change Compression exit

If the compression exit name is the same, check the following:

- Specify a library that includes the current compression exit in the RMODLIB DD when unloading, then provide a library that includes the new compression exit in the RMODLIB DD of FABCUR3. If the compression exit name is different, choose one of the following:

  - Specify COMP=Y in the SYSIN DD and the concatenated libraries that include the current compression exit and the new compression exit in the RMODLIB DD of FABCUR1.

– Specify COMP=N in the SYSIN DD and specify the library that includes the current compression exit in the RMODLIB DD of FABCUR1, then specify the library that includes the old compression exit in the RMODLIB DD of FABCUR3.

You can specify Reorg mode and Change mode by using FABCOP1D (site default table).

Even if the TYPERUN/TYPRUN=REORG keyword is specified in FABCOP1D, it will be ignored if NEWDBDNM=, HIERCHNG=, or RMODTYPE= is specified in the SYSIN DBDNAME= control statement.

Data and system flow of DEDB Reload

The reload processor (FABCUR3) is a z/OS batch program that reloads one or more DEDB areas from the formatted QSAM data sets created by the unload processor. It achieves significant performance improvements (that is, elapsed time reduction) by reducing the number of EXCPs issued and eliminating the need to run the DEDB Initialization utility (DBFUMIN0). The only limitation on the number of concurrent reload jobs is the number of available initiators.

ACB definitions are retrieved from the database definition record data set specified by the DURDBDFN DD statement or from the ACB library specified by the ACBLIB DD statement. However, if the IMS catalog and the IMS management of ACBs are enabled, ACB definitions can be retrieved from the IMS directory.

The following figure shows the flow of the reload procedure.
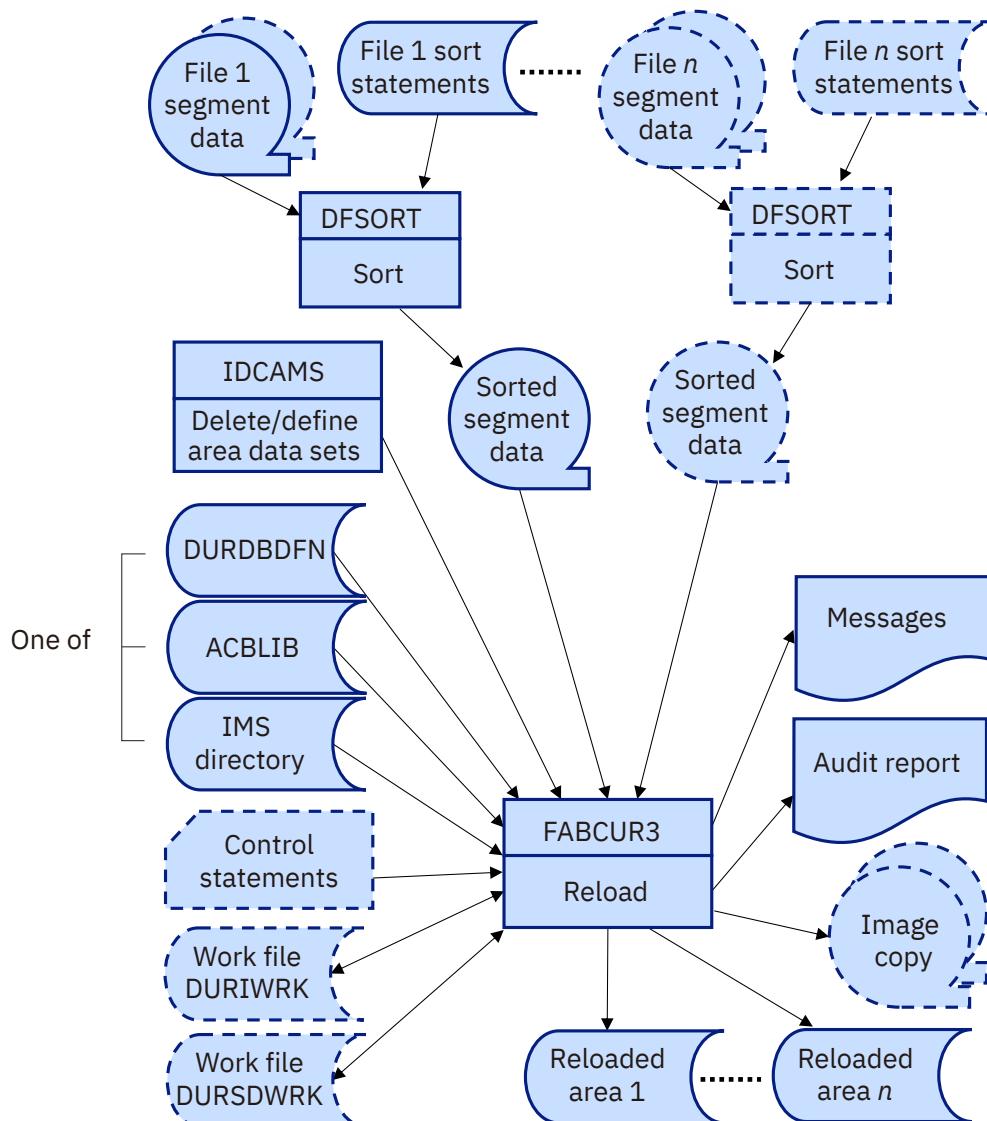
Figure 50. Flow of Reload

Load modules

Among the DEDB Unload/Reload modules, three are used for reload. The following table lists these modules.

Table 35. Load modules of DEDB Reload

Load module name	Function
FABCUR3	Reload processor
FABCUR3W	Subtask to write IOVF CIs for a UOW
FABC3ICT	Subtask to process image copies

Restrictions of DEDB Reload

The DEDB Unload/Reload utilities have some restrictions.

When a randomizer is invoked by IMS, Registers 10 and 11 contain the addresses of the EPST and ESCD, respectively. Some user-written or user-customized randomizers are designed to use these addresses to gain access to IMS control blocks other than those passed as part of the published interface. When one of the DEDB Unload/Reload utilities issues a call to a randomizing module, Registers 10 and 11 are set to -1 to 0, respectively, to specify that the call is not being issued in a "live" IMS environment.

Typical scenario and processes to run for unloading and reloading a DEDB

To use the DEDB Unload/Reload, you must do several things.

Typical scenario for DEDB Unload/Reload

The following is a checklist of the activities that are usually required to unload and reload one or more areas of a DEDB. Some steps are omitted when only database reorganization is being performed:

1. Perform the following pre-unload/reload planning.

   a. Determine the number of unload subtasks and output files, and calculate the storage requirements for FABCUR1.

      For the details, see the Unload Region Size Estimation Worksheet, .

   b. Set up and tailor the unload JCL:

      - Control statements
      - Input and output files
      - Number of sort steps.

   c. Set up and tailor the required number of reload jobs.

   d. Set up a temporary ACB library and generate the new DBD and ACB.

2. Stop (that is, deallocate from the IMS control region) all areas involved in the maintenance process. If the application does not handle "FH" status codes, all transactions that access the database must be stopped.

3. Image copy all areas that will be unloaded and reloaded.

4. Run the unload job.

5. Run DEDB Pointer Checker against the image copy data sets. This provides statistics on the number of segments in the areas being unloaded/reloaded. This could be run concurrently with the unload/reload jobs.

6. Check any messages generated by FABCUR1, and verify segment totals on the Unload Audit Control report.

7. Run the reload jobs by specifying image copy and HASH check options.
8. Check any messages generated by FABCUR3, and verify segment totals on the Reload Audit Control report.
9. If image copy option was not specified in step 7, image copy all areas that were unloaded and reloaded.
10. If image copy option was not specified in step 7, run the DEDB Pointer Checker utility against the image copy data sets. Verify that the unload/reload produced clean areas with the correct number of segments.
11. Migrate the new DMB to the production ACB library.
12. Start the reloaded areas.

Running DEDB Reload

The reload program (FABCUR3) is run as one or more standard z/OS batch jobs. You need to specify an EXEC statement and DD statements that define the input and output data sets in your JCL.

Procedure

1. Estimate the region size required to run the function.
2. Code the JCL for reload (DFSORT, IDCAMS, and FABCUR3) job steps.
3. Specify the DD statements to define input data sets, output data sets, and how the function is run.
4. Run the JCL.

   Note: FPB allows you to specify site default parameters for DEDB Reload (FABCUR3). Macros and sample JCL streams are provided to generate the site default table.

Example

The following figure shows example JCL for DEDB reload.

```
//STEP2A    EXEC PGM=SORT
//SYSOUT    DD SYSOUT=A
//SYSIN     DD DSN=HPFP.UR.FILEzzz.SORTCARD,
//             DISP=(OLD,DELETE,KEEP)
//SORTIN    DD DSN=HPFP.UR.FILEzzz.SEGDATA,
//             DISP=(OLD,DELETE,KEEP)
//SORTOUT   DD DSN=HPFP.UR.FILEzzz.SORTED.SEGDATA,
//             DISP=(NEW,CATLG,DELETE),UNIT=TAPE
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//*
//*
//STEP3A    EXEC PGM=IDCAMS,REGION=1024K
//*************************************************************
//**  DELETE AND DEFINE AREA AND WORK DATA SET DATASETS  **
//*************************************************************
//SYSPRINT DD SYSOUT=A
//SYSIN     DD *,DCB=BLKSIZE=80
   ... control statements ...
/*
//STEP4A    EXEC PGM=FABCUR3,REGION=1024K
//*****************************
//**  RELOAD THE AREA(S)       **
//*****************************
//STEPLIB  DD DSN=HPFP.SHFPLMD0,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DURAUDIT DD SYSOUT=A
//SYSIN     DD *
   ... control statements ...
/*
//DURDBDFN DD DSN=HPFP.UR.DURDBDFN,
//             DISP=(OLD,DELETE,KEEP)
//DURIWRK  DD DSN=VSAM.RUNn.WORK,
//             DISP=OLD
//DURDATA  DD DSN=HPFP.UR.FILEzzz.SORTED.SEGDATA,
//             DISP=(OLD,DELETE,KEEP)
//areaxxx  DD DSN=VSAM.AREAxxx,DISP=OLD
```

Figure 51. Example JCL for DEDB reload (FABCPRD)

Related reference

Region size requirements for DEDB Reload

The DEDB Reload utility operates in a virtual storage region as z/OS batch job steps. FABCUR3, FABCUR3W, and FABC3ICT run in 31-bit addressing mode.

Region size requirements for DEDB Reload

The DEDB Reload utility operates in a virtual storage region as z/OS batch job steps. FABCUR3, FABCUR3W, and FABC3ICT run in 31-bit addressing mode.

Region size for the reload program depends on the size of a UOW, the number of IOVF buffers allocated. Typically, database maintenance is a relatively important function. To increase the performance of the reload program, it would be advisable to run it in a region that has a relatively high dispatching priority (that is, at least equal to the dispatching priority of a message processing region).

Subsections:

Estimating the region size for reload

Region size for the reload program must satisfy the requirements for the program, and requires access method modules and buffers—that is, input sequential file BUFNO= and VSAM BUFND=. Elapsed time could be further reduced by increasing the BUFND value.

The following worksheet is for estimating the region size that is required for a typical reload job. The actual region size depends on the keywords that are specified for the job. Therefore, when you specify the region size in your JCL, use a higher value than the value calculated from this worksheet.

```
Reload programs, internal tables, control blocks      180,000

1024 + BLKSIZE * BUFNO                                -------

(IOVFBUF+UOW2) * max CISZ in DBD                      -------

Number of areas defined in DBD * 1300                 -------   (If ACBLIB DD and DBDNAME keyword are
specified.)

Number of areas defined in DBD * 72                   -------

(Root key length+388) * TBLENTRY                      -------

max(CISZ * UOW1)                                      -------

Total bytes of reload storage                         -------
```

Figure 52. Worksheet for estimating the region size for Reload

Reducing region size constraints

The reload program (FABCUR3) is implemented using various techniques aimed at reloading DEDB areas from the segment data in the shortest possible time. The trade-off is increased storage requirements. The following list contains a series of actions that may be taken to reduce the region size requirements for a given reload request. They are listed in increasing order of impact on the performance of FABCUR3.

1. If the maximum block size for the input file devices is large, specify the DCB=BLKSIZE=*nnnnn* parameter in the DD statement. Typically, reduce the block size to 15-20K.

   - This reduces the storage required for the QSAM buffers for each of the files.

2. Specify a value for the IOVFBUF parameter. This reduces the storage required for the set of IOVF buffers.

3. Specify an AMP=('BUFND=...' override on the DD statements for the areas being reloaded to reduce the number of buffers below the "2 X CIs-per-Track + 1" value. This should be considered only as a last resort. FABCUR3 automatically calculates an optimum value for this parameter when opening the VSAM Sequential ACBs. The value calculated by FABCUR3 is designed to maximize the number of CIs read per EXCP. A JCL override takes precedence over the calculated value and can have a drastic effect on the number of EXCPs issued to extract the segment data in the areas being unloaded.

   - This reduces the storage required for the buffers acquired by VSAM.

   - It results in increased elapsed processing time as the EXCP count increases, and more channel contention and I/O wait time incurs.

DD statements for DEDB Reload

DD statements for DEDB Reload determine the input and output data sets and how DEDB Reload is run.

You must specify DD statements for the job control language (JCL) to run the DEDB Reload utility.

DFSORT JCL

To run DFSORT, you have to supply an EXEC statement and the appropriate DD statements.

However, unless the DEDB Unload utility (FABCUR1) has been run in REORG mode, with each area unloaded into its own DURD*zzz*0 or DURD*zzz*E data set, it is not necessary to run DFSORT unless one of the following was also specified:

- LOADCTL statement
- SDEP=LOGICAL

EXEC statement

The EXEC statement must be in the following form:

```
//      EXEC PGM=SORT
```

DD statements

The following table summarizes the DD statements.

Table 36. DFSORT DD statements

DDNAME	Use	Format	Required or optional
SORTIN	Input		Required
SYSIN	Input		Required
SORTOUT	Output		Required
SYSOUT	Output	SYSOUT	Required
SORTWK01	Work data set		Required
SORTWK02	Work data set		Required
SORTWK03	Work data set		Required
SORTWK04	Work data set		Required
SORTWK05	Work data set		Required
SORTWK06	Work data set		Required

SORTIN DD
 This input data set is the DURD*zzz*O (or DURD*zzz*E) file from FABCUR1.

SYSIN DD
 This input data set contains DFSORT control statements. It is the DURS*zzz*O file from FABCUR1.

SORTOUT DD
 This output data set contains the sorted records. Do not specify DISP=MOD for this DD statement.

SYSOUT DD
 This output data set contains the messages produced by DFSORT.

SORTWK*nn* DD
 These are intermediate storage data sets used by DFSORT. See *DFSORT Application Programming Guide* for more information about coding the SORTWK*nn* DD statements.

Example JCL stream is shown in the following figure.

```
//STEP2A    EXEC PGM=SORT
//SYSOUT    DD SYSOUT=A
//SYSIN     DD DSN=HPFP.UR.FILEzzz.SORTCARD,
//             DISP=(OLD,DELETE,KEEP)
//SORTIN    DD DSN=HPFP.UR.FILEzzz.SEGDATA,
//             DISP=(OLD,DELETE,KEEP)
//SORTOUT   DD DSN=HPFP.UR.FILEzzz.SORTED.SEGDATA,
//             DISP=(NEW,CATLG,DELETE),UNIT=TAPE
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(???,??))
//*
//*
```

IDCAMS JCL

To run IDCAMS, you have to supply an EXEC statement and the appropriate DD statements.

EXEC statement

The EXEC statement must be in the following form:

```
//      EXEC PGM=IDCAMS
```

DD statements

The following table summarizes the DD statements.

Table 37. IDCAMS DD statements

DDNAME	Use	Format	Required or optional
SYSIN	Input	LRECL=80	Required
SYSPRINT	Output	SYSOUT	Required

SYSIN DD
> This input data set contains the Access Method Services control statements that delete the old area data sets and define VSAM clusters for the new ones. For additional information, see *DFSMS Access Method Services for the Integrated Catalog Facility*.

SYSPRINT DD
> This output data set contains the messages produced by IDCAMS.

Example JCL stream is shown in the following figure.

```
//STEP3A   EXEC PGM=IDCAMS,REGION=1024K
//*********************************************************
//**  DELETE AND DEFINE AREA AND WORK DATA SET DATASETS  **
//*********************************************************
//SYSPRINT DD SYSOUT=A
//SYSIN    DD *,DCB=BLKSIZE=80
   ... control statements ...
/*
```

FABCUR3 JCL

This topic describes the EXEC statement and DD statements of FABCUR3.

EXEC statement

Specify the EXEC statement in the following format:

```
//      EXEC PGM=FABCUR3,REGION=rrrrM,
//          PARM='DBRC=x,GSGNAME=gsgname,IMSPLEX=imsplex,DBRCGRP=xxx,AREC=x'
```

DBRC=
> Specifies whether or not to use the DBRC interface.
>
> **Y**
>> FABCUR3 uses the DBRC interface.
>>
>> DEDB Reload uses the DBRC interface in exactly the same manner as the IMS FP DEDB Initialization utility (DBFUMIN0), as follows:
>>
>> • Authorizes an area when the area is 'recovery needed' in DBRC.

- Notifies the completion of the reload processing, that is, the area is changed to 'not recovery needed,' reloaded area data set(s) are made available, and an image copy recommended flag is set on in DBRC.

N

FABCUR3 does not use the DBRC interface.

Default: N

GSGNAME=

GSGNAME is a 1-to-8 character optional parameter specifying the global service group.

IMSPLEX=

IMSPLEX is a one- to five-character optional parameter specifying which IMSplex DBRC should join in.

DBRCGRP=

DBRCGRP is a one- to three-character optional parameter, which is used by DBRC to identify which sharing group the notification belongs to.

AREC=

Determines whether to include an area information record in an unloaded segment record data set.

Y

The input unloaded segment record files are created by FABCUR1 with the AREC=Y option.

N

The input unloaded segment record files are created by FABCUR1 with the AREC=N option.

Default: Y

DD statements

The following table summarizes the DD statements of FABCUR3.

Table 38. FABCUR3 DD statements

DDNAME	Use	Format	Required or optional
JOBLIB or STEPLIB	Input	PDS	Required
SYSIN	Input	LRECL=80	Optional
ACBLIB	Input	PDS	Optional
RMODLIB	Input	PDS	Optional
RECON1	Input	KSDS	Optional[1]
RECON2	Input	KSDS	Optional[1]
RECON3	Input	KSDS	Optional[1]
EXITLIB	Input	PDS	Optional
IMSDALIB	Input	PDS	Optional
DURDBDFN	Input		Required when the DBDNAME= keyword is not specified
DURDATA	Input		Required
DURIWRK	Input/Output	VSAM ESDS cluster	Optional
DURSDWRK	Input/Output		Optional
SYSPRINT	Output	LRECL=133	Required
DURAUDIT	Output	LRECL=133	Required

Table 38. FABCUR3 DD statements (continued)

DDNAME	Use	Format	Required or optional
area*xxx*	Input/Output	LRECL=80/ VSAM ESDS cluster	Required
ads*xxx*	Output	VSAM ESDS cluster	Optional
HDIAUDIT	Output	Do not code DCB	Optional
HDI*xxx*P or XI*xxxxx*P	Output	Do not code parameters other than BLKSIZE for the DCB statement.	Optional
HDI*xxx*S or XI*xxxxx*S	Output	Do not code parameters other than BLKSIZE for the DCB statement.	Optional
ERRORSEG	Output	Do not code parameters other than BLKSIZE for the DCB statement.	Optional
IMS	Input	PDS	Optional[2]
MSGOUT	Output	LRECL=133	Optional[2]
REPORTS	Output	LRECL=133	Optional[2]
SNAPDPIT	Output	LRECL=133	Optional[2]
SORTIN	Work data set		Optional[2]
SORTOUT	Work data set		Optional[2]
SORTWK*nn*	Work data set		Optional[2]
SYSOUT	Output	SYSOUT	Optional[2]

Notes:

1. These data sets are required when DBRC=Y is specified as a parameter on the EXEC statement.

2. These data sets are required when IMGCPY=YES|DUAL and ICHASH=(YES) parameters are specified.

JOBLIB DD or STEPLIB DD
>   When DBRC=Y is specified on the EXEC statement or when IMGCPY=YES|DUAL and ICHASH=(YES) parameters are specified, the IMS load module library (IMSVS.SDFSRESL) must be concatenated with the FPB DEDB Unload/Reload library on the JOBLIB or STEPLIB DD.
>
>   To allocate a DBRC RECON data set and/or a DBRC non-registered area data set dynamically, the DFSMDA library data set must be concatenated.
>
>   To allocate an area data set dynamically when DBRC=N is specified, the DFSMDA library data set must be concatenated.
>
>   If you specify the IMSCATHLQ=*bsdshlq* keyword, the IMS load module library (IMSVS.SDFSRESL) and the IMS Tools Base library (SGLXLOAD) must be concatenated.

SYSIN DD
>   This statement defines the input control statement data set. This data set can reside on either a direct-access device or a printer; or it can be routed through the output stream.

ACBLIB DD
>   This statement defines the library that contains the DMB for the database.
>
>   The location of the DMB must be supplied by the ACBLIB DD statement, the DURDBDFN DD statement, or the IMSCATHLQ=*bsdshlq* keyword.

FABCUR3 retrieves the DMB according to the following rules:

- The DBDNAME= keyword is specified.
  - If the IMSCATHLQ=*bsdshlq* keyword is specified, DMB is retrieved from the IMS directory.
  - If the IMSCATHLQ=*bsdshlq* keyword is not specified, DMB is retrieved from the ACBLIB library (ACBLIB DD).
- The DBDNAME= keyword is not specified. DMB is retrieved from the DURDBDFN data set (DURDBDFN DD).

RMODLIB DD

This statement defines the library where the randomizer and/or segment edit/compression routines reside.

The DD statement is not needed when the COMPRESS=Y parameter is used even if the DBD contains a segment edit/compression routine except if the USEREXIT option is specified.

If the compression routine is changed, then the old compression routine should reside in the FABCUR1 step RMODLIB DD library (with the FABCUR1 COMPRESS=N parameter specified). Make sure that the new compression routine resides in the FABCUR3 step RMODLIB DD library before attempting the DEDB reload.

RMODLIB in the FABCUR3 step is required when a randomizer is to be used for the RAPERROR=ABEND option. If this DD statement is not provided, an attempt is made to load the randomizer and/or edit/compression routine from JOBLIB/STEPLIB.

RECON1 DD

This statement defines the first DBRC RECON data set. This RECON1 data set must be the RECON1 data set used in the control region.

RECON2 DD

This statement defines the second DBRC RECON data set. This RECON2 data set must be the RECON2 data set used in the control region.

RECON3 DD

This statement defines the third DBRC RECON data set. This RECON3 data set must be the RECON3 data set used in the control region.

Do not use these RECON data set ddnames if you specified dynamic allocation using the DFSMDA macro.

You need to specify the IMSPLEX and DBRCGRP parameters on the EXEC statement for parallel RECON access to make sure that all DBRC instances in a sharing group have access to the same RECON data sets.

EXITLIB DD

This statement defines the library in which the exit routine specified by the EXITRTN= control statement resides.

IMSDALIB DD

This statement defines the IMS library that contains DFSMDA members to allocate an area data set and/or RECON data sets dynamically. If this statement is provided, program FABCUR3 loads DFSMDA members from the IMSDALIB DD prior to the JOBLIB DD or the STEPLIB DD.

DURDBDFN DD

This statement defines the data set that contains a formatted copy of the DMB. It is the DURDBDFN data set from FABCUR1.

The location of the DMB must be supplied by the ACBLIB DD statement, the DURDBDFN DD statement, or the IMSCATHLQ=*bsdshlq* keyword.

FABCUR3 retrieves the DMB according to the following rules:

- The DBDNAME= keyword is specified.
  - If the IMSCATHLQ=*bsdshlq* keyword is specified, DMB is retrieved from the IMS directory.

- If the IMSCATHLQ=*bsdshlq* keyword is not specified, DMB is retrieved from the ACBLIB library (ACBLIB DD).
- The DBDNAME= keyword is not specified. DMB is retrieved from the DURDBDFN data set (DURDBDFN DD).

DURDATA DD

This statement defines the sorted input data set that contains the unloaded (or created) segment data records for one or more areas. It is the SORTOUT data set from DFSORT.

DURIWRK DD

This statement defines a work file (VSAM ESDS) for controlling IOVF creation and usage. If DURIWRK DD omitted or defined either DUMMY or DSN=NULLFILE, the Data Space is used for controlling IOVF creation and usage.

If you use a work file, the following characteristics are required:

- It must be defined with the REUSE option, since it is reused for each area in the input data stream.
- It must have at least as many CIs as the maximum number of IOVF CIs in any of the areas being loaded.
- Its CI size must be as large as the largest CI size of the areas being specified in the DMB.
- The VSAM SPEED option can be effected for performance.

When using the data space, the maximum size of the data space is equal to the IOVF size in reloaded area.

DURSDWRK DD

This statement defines a work file (OS sequential data set) for controlling SDEP CIs. This DD statement is required for the unloaded data set that is created by the DEDB Unload utility (FABCUR1) with SDEP=LOGICAL option, and for SDEP Relocation processing.

The following characteristics are required.

- DISP=MOD is not allowed, since the data set is reused for each area in the input data stream.
- The data set must allocate enough space to hold all of the 512-byte blocks of every SDEP CI between the logical begin and logical end CI of any area being reloaded.
- DO not code DCB information in your JCL.

SYSPRINT DD

This statement defines the output data set that contains messages issued by FABCUR3. The data set can reside on either a direct-access device or a printer; or it can be routed through the output stream. You can code RECFM=FBA,LRECL=133 on your DD statement, but it is better to use:

```
//SYSPRINT DD SYSOUT=A
```

DURAUDIT DD

This statement defines the output data set that contains the Reload Audit Control report. The data set can reside on either a direct-access device or a printer; or it can be routed through the output stream. You can code RECFM=FBA,LRECL=133 on your DD statement, but it is better to use:

```
//DURAUDIT DD SYSOUT=A
```

area*xxx* DD

This statement is used when DBRC=N is specified, or for a DBRC non-registered area when DBRC=Y is specified.

This statement defines an output area data set (that is, the VSAM ESDS for an area to be reloaded), or an input data set that contains DD names of output multiple area data sets for area*xxx*. The ddname used must be the ddname of the area as defined in the output DMB used by FABCUR1. There must be one DD statement for each area for which data exists in the input file (DURDATA).

All area VSAM data sets must be empty (that is, deleted and redefined).

To allocate an area VSAM data set dynamically, a DFSMDA member for the area must be provided.

When the area*xxx* DD is defined as an output area data set, consider the following.

1. Device and channel contention should be avoided. Allocating the work data set on the same device as the area data set is not recommended.

2. Multiple work data sets can be allocated on the same device without affecting performance. A reasonable maximum work data set is six to eight (that is, if six reload jobs are being processed concurrently, the six work data sets should reside on the same device with negligible performance degradation).

3. It could be possible that no segments are reloaded. In this case, the output area data set is initialized (that is, empty area).

When area*xxx* DD defines an input data set that contains DD names of output multiple area data sets, see the detailed FABCUR3 area*xxx* data set in the input.

ads*xxx* DD

When 'DBRC=Y' is specified, or when 'DBRC=N' is specified and the area*xxx* DD defines an input data set that contains DD names of output multiple area data sets, this statement defines the VSAM ESDS output data set for the area to be reloaded.

When the DBRC=Y parameter is specified on the EXEC statement, the DD name and the data set name should be registered in DBRC.

Consider the following:

1. Avoid device and channel contention. Do not allocate the work data set on the same device as the area data set.

2. Multiple work data sets can be allocated on the same device without affecting performance. A reasonable maximum work data set is six to eight (that is, if six reload jobs are being processed concurrently, the six work data sets should reside on the same device with negligible performance degradation).

If the empty output area data set is initialized, no segments are reloaded by FABCUR3.

When DBRC=Y is specified and an area is registered in DBRC, all area data sets can be allocated dynamically. If any of the area data sets registered in DBRC, are explicitly specified by ads*xxx* DD statements, no dynamic allocation is done for all area data sets that belong to the same area as the one with the ads*xxx* specification; Therefore, if you want to allocate area data sets dynamically, do not specify any ads*xxx* DD statement.

HDIAUDIT DD

This statement defines the output data set that contains messages issued by the IBM IMS Image Copy Extensions for z/OS (ICE). The data set can reside on either a direct-access device or a printer; or it can be routed through the output stream. Do not code the DCB= parameter; it is recommended that you use:

```
//HDIAUDIT DD SYSOUT=A
```

HDI*xxx*P or XI*xxxxx*P DD

This statement defines the first copy of the image copy output data set. One DD statement is required for each area data set to be dumped. The output device must be either a direct-access or a tape. The standard label must be used. The BLKSIZE used is the largest multiple of the logical record length that does not exceed the maximum BLKSIZE. If BLKSIZE is specified in the JCL stream, that BLKSIZE is considered as the maximum.

The infix *xxx*/*xxxxx* specifies the area number of the area to be dumped. Zero cannot be suppressed.

HDI*xxx*S or XI*xxxxx*S DD

This statement defines the second copy of the image copy output data set. If IMGCPY=DUAL is specified, this DD statement is required. The output device must be either a direct-access or a tape. The standard label must be used. The BLKSIZE used is the largest multiple of the logical record length that does not exceed the maximum BLKSIZE. If BLKSIZE is specified in the JCL stream, that BLKSIZE is considered as the maximum.

ERRORSEG DD
>This statement defines the segment file that contains the key sequence error segments and its child segments. This DD statement is effective only when the KEYSEQERROR=BYPASS control statement is specified in the SYSIN DD statement or the site default table. The DCB attributes are calculated by FABCUR3. RECFM is VB.
>
>- For 3380, the default block size is 23476 bytes.
>- For 3390, the default block size is 27998 bytes.
>- For 9345, the default block size is 22928 bytes.
>
>For other output devices, the default block size is the maximum block size of the output device. A block size can be specified on the DD statement. Do not code any other DCB parameters in your JCL.
>
>For its record layout, see the topic "Unloaded segment records layout" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Example JCL stream is shown in the following figure.

```
//STEP4A    EXEC PGM=FABCUR3,REGION=1024K
//****************************
//**  RELOAD THE AREA(S)     **
//****************************
//STEPLIB  DD DSN=HPFP.SHFPLMD0,DISP=SHR
//SYSPRINT DD SYSOUT=A
//DURAUDIT DD SYSOUT=A
//SYSIN    DD *
   ... control statements ...
/*
//DURDBDFN DD DSN=HPFP.UR.DURDBDFN,
//           DISP=(OLD,DELETE,KEEP)
//DURIWRK  DD DSN=VSAM.RUNn.WORK,
//           DISP=OLD
//DURDATA  DD DSN=HPFP.UR.FILEzzz.SORTED.SEGDATA,
//           DISP=(OLD,DELETE,KEEP)
//areaxxx  DD DSN=VSAM.AREAxxx,DISP=OLD
```

Function-unique DD statements for the HASH Check option

To use the HASH check option, you must specify your DD statements the following format.

IMS DD
>This statement defines the library that contains the DBD that describes the database to be processed. The data set must reside on a direct-access volume.

MSGOUT DD
>This statement defines the output data set that contains messages produced by the DEDB Pointer Checker. The data set can reside on either a direct-access device or a printer; or it can be routed through the output stream. Do not code the DCB= parameter; it is recommended that you use:
>
>```
>//MSGOUT DD SYSOUT=A
>```

REPORTS DD
>This statement defines the output data set for the Pointer Chain Reconstruction report of the DEDB Pointer Checker. The data set can reside on either a direct-access device or a printer; or it can be routed through the output stream. Do not code the DCB= parameter; it is recommended that you use:
>
>```
>//REPORTS DD SYSOUT=A
>```

SNAPDPIT DD
>This statement defines the output data set for the DEDB CI Map/CI Dump report. The data set can reside on a direct-access device or it can be routed through the input stream. Do not code the DCB= parameter; it is recommended that you use:
>
>```
>//SNAPDPIT DD SYSOUT=A
>```

SORTIN DD

This work data set is used for the subset pointer check function of the DEDB Pointer Checker. If you want to use the data space for the subset pointer check, or if you do not want to use the subset pointer check function at all, this DD statement can be omitted.

SORTOUT DD

This work data set is used for the subset pointer check function of the DEDB Pointer Checker. If you want to use the data space for the subset pointer check, or if you do not want to use the subset pointer check function at all, this DD statement can be omitted.

SORTWK*nn* DD

These are intermediate storage data sets used by DFSORT. For more information about coding the SORTWK*nn* DD statements, see the *DFSORT Application Programming Guide*. Allocating a space twice the size used by the SORTIN data set is usually adequate for each work data set.

SYSOUT DD

This statement defines the output data set that contains the messages produced by DFSORT.

Input for DEDB Reload

You must specify the necessary input DD data sets to run the DEBD Reload utility.

DEDB Reload SYSIN DD data set

The SYSIN data set contains the user's specification of the control statements that control the reload processing.

Format

This control statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte, fixed-length records. Block size, if coded in your JCL, must be a multiple of 80.

This data set can contain several optional keyword parameters. It can be coded as follows:

```
//SYSIN DD *
   IOVFBUF=9
/*
```

Figure 53. FABCUR3 SYSIN DD data set

Related reference
Site default support for FPB
This topic describes the site default support for FPB.

DEDB Reload SYSIN DD data set control statements

This topic describes the control statements used with DEDB Reload.

Subsection:

Syntax of the control statement

The FABCUR3 keywords and their associated parameter values may be coded in free format (columns 1 - 72) provided certain syntactical rules are followed:

1. All keyword parameter sets can be contained in one or more control statements.
2. All keywords must be separated from their associated parameter values by an equal sign. The equal sign must not be preceded by blanks, but may be followed by one or more blanks.
3. Parameter values must be separated from the next keyword by one or more blanks, a comma, or a comma followed by one or more blanks.
4. Parentheses are always required for keyword parameters that use them.

Control statement keywords

The FABCUR3 control statement contains the following keywords:

```
[DBDNAME=dbdname]
[IOVFBUF=nn]
[STARTAREA=zzz]
[TBLENTRY=mmmm]
[EXITRTN=(exit-routine,function)]
[IMGCPY=NO|YES|DUAL]
[ICHASH=(NO|YES,SSP)]
[ICCOMPRESS=(NO|YES,routine)]
[KEYSEQERROR={NOCHECK|ABEND|BYPASS}]
[KEYCHGCHECK={YES|NO}]
[SDEPRELOCATE={YES|NO}]
[IMSCATHLQ={*NO|bsdshlq}]
[IMSCATACB_INPUT={CURRENT|PENDING}]
[RAPERROR=NOCHECK|ABEND]
```

DBDNAME control statement

This optional statement specifies the DBD name of the DEDB for the unloaded segment records file. This statement is required if ACBLIB DD or IMSCATHLQ=*bsdshlq* is specified instead of DURDBDFN DD to obtain the DEDB definition.

If DURDBDFN DD is specified, this statement is not necessary.

When DBDNAME= is specified:

- If both DURDBDFN DD and ACBLIB DD statements are specified, or both DURDBDFN DD statement and the IMSCATHLQ=*bsdshlq* keyword are specified, ACBLIB DD or the IMSCATHLQ=*bsdshlq* keyword takes precedence over DURDBDFN DD.
- If the ACB library or the IMS directory is used, the ACB library or the IMS directory is validated to make sure that the correct ACB library or IMS directory is provided.

dbdname
    Specifies the name of the DBD that is to be used.

IOVFBUF control statement

Optional statement to specify the number of IOVF buffers to allocate for direct processing of the IOVF work data set. This should be equal to the maximum number of IOVF CIs used by any UOW being reloaded. The maximum value specified should not exceed approximately 99. Too small a value has the potential to cause excessive flushing and rereading of IOVF CIs.

The default value is calculated as follows:

```
28K x 5 / size of the largest CI of areas to be processed
```

If the calculated value is greater than 99, then 99 is used.

For example, the number 5 is used for a 28 K CI; the number 99 is used for a 512-byte CI.

STARTAREA control statement

Optional statement to specify which area to start processing (that is, data for areas less than this value are bypassed). This statement should be used only in restart situations.

Example: The input file contains segment data for areas 3 and 6. Area 3 was successfully loaded, but during the loading of area 6 a physical I/O error occurred. The reload job can be restarted after a new area 6 is defined on a different pack by specifying STARTAREA=6 (that is, data for area 3 is bypassed).

The area number is 1 - 2048.

TBLENTRY control statement

Optional statement to increase the number of:entries in the internal segment parentage table. This table is only used when the insert limit count (ILC) feature of FABCUR1 is used.

The default value is 200 entries per RAP CI. The maximum value is 9999 and the minimum value is 1.

The ILC value is specified on the FABCUR1 LOADCTL control statement. If a single RAP has more than 200 dependent-segment twin chains that exceed the ILC value, you must use this keyword. Just make *mmm* larger than the maximum number of long twin chains that can occur on one RAP.

EXITRTN control statement

This optional statement specifies the name of the user exit routine to be invoked, and the function to be performed by it.

For more information on the exit routine, see the topic "Exit routine option and its interface" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

The EXITLIB DD statement must be provided when this control statement is specified.

exit-routine
    Specifies the name of the user exit routine to be invoked.

function
    Specifies the function to be performed by the exit routine.

    EXP
        Expansion

    CMP
        Compression

    EXC
        Expansion and then compression; invoke the exit routine twice

IMGCPY control statement

Optional statement to specify whether image copies of the areas being reloaded are to be created.

To create an image copy, IMS High Performance Image Copy (included in IMS Fast Path Solution Pack) must be installed, and its load module library data set must be concatenated to the JOBLIB DD or the STEPLIB DD statement.

NO
No image copies are created. This is the default.

YES
A single image copy of each area being reloaded is created.

DUAL
Dual image copies of each area being reloaded are created.

ICHASH control statement

Optional statement to specify whether the HASH Check option of DEDB Pointer Checker is invoked while the image copy is being created.

NO
HASH check is not performed. This is the default.

YES
HASH check is performed. This subparameter is effective only when IMGCPY=YES or when IMGCPY=DUAL is specified.

Additional DD statements described in "Function-unique DD statements for the HASH Check option" on page 171 are required for this option.

SSP
Subset pointer check is performed during the HASH check. This subparameter is effective only when ICHASH=(YES,) is specified.

ICCOMPRESS control statement

Optional statement to specify whether the ICE Image Copy function compresses the output image copy data.

NO
The compression routine is not invoked. This is the default.

YES
The compression routine has been invoked. This subparameter is effective only when IMGCPY=YES or when IMGCPY=DUAL is specified.

routine
The compression routine. The ICE provides two compression exit routines: FABJCMP1 and FABJCMP2. Specify one of these compression routines to be called by the ICE Image Copy function. If no compression routine is specified, FABJCMP1 is used as the default. This subparameter is effective only when ICCOMPRESS=(YES,) is specified.

Recommendation: To check that the area reloaded has no error, specify the creation of image copies and run of HASH check when loading an area. If you do this, you do not need to run additional jobs: Image Copy Utility and the DEDB Pointer Checker.

KEYSEQERROR control statement

This optional statement specifies whether FABCUR3 checks the key sequence, and what to do when a sequence error in a key sequence field is detected.

NOCHECK
Specifies that FABCUR3 ignores the key sequence check function. This is the default.

ABEND
Specifies that FABCUR3 ends abnormally, and an error message is issued.

BYPASS

Specifies that any segments that have a sequence error in a key sequence field, as well as any children, are ignored in the reloading process.

KEYCHGCHECK control statement

This optional statement determines whether FABCUR3 should check alternation of key sequence field after returning from a user exit routine.

YES

Specifies that FABCUR3 checks altered key sequence field. This is the default.

NO

Specifies that FABCUR3 does not check altered key sequence field.

SDEPRELOCATE control statement

This optional statement determines whether FABCUR3 should relocate areas of the SDEP segments. If SDEP segments are unloaded with SDEP=PHYSICAL accompanied by a DBD change, this optional statement is not necessary because the reload process (FABCUR3) automatically sets SDEPRELOCATE=YES. SDEPRELOCATE=YES can be specified only when the SDEP segments are unloaded with the SDEP=PHYSICAL and AIR=YES options.

For how to process SDEP segments, see the topic "Determining how to process SDEP segments" in the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

YES

Specifies that FABCUR3 relocates SDEP pointers.

NO

Specifies that FABCUR3 does not relocate SDEP pointers. This is the default.

IMSCATHLQ control statement

This optional statement specifies to retrieve ACBs from the IMS directory (an extension of the IMS catalog) instead of from the ACB library.

bsdshlq

Specifies to read ACBs from the IMS directory by using the IMS Tools Catalog Interface. *bsdshlq* specifies the high-level qualifier of the bootstrap data set of the IMS directory. IMSCATHLQ=*bsdshlq* is effective when the IMS catalog and the IMS management of ACBs are enabled.

If a TYPE=CATDSHLQ statement was used to create a DFSMDA member in your MDA library, you must specify the SYSDSHLQ parameter used in that DFSMDA member for the IMSCATHLQ keyword parameter.

*NO

Specifies to read ACBs from the ACB library. This is the default.

IMSCATACB_INPUT control statement

This optional statement specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. IMSCATACB_INPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* option is specified.

CURRENT

Specifies that currently active ACB members are retrieved from the IMS directory data sets. This is the default.

PENDING

Specifies that pending ACB members are retrieved from the staging data set.

RAPERROR control statement

This optional statement specifies whether FABCUR3 checks the validity of RAP data in the prefix part of the unloaded segment record by using the result of the randomizer call.

NOCHECK

Specifies that FABCUR3 does not check the validity of the RAP data in the prefix part of the unloaded segment record by using a randomizer. This is the default.

ABEND

Specifies that FABCUR3 checks the validity of the RAP data in the prefix part of the unloaded segment record by using a randomizer. FABCUR3 ends abnormally with an error message if it finds an error.

The DBDNAME control statement must be specified to enable RAPERROR=ABEND. Also, either the ACBLIB DD statement or the IMSCATHLQ=*bsdshlq* control statement must be specified.

Area*xxx* data set for input

The area*xxx* data set contains the ddname for output area data set of area name area*xxx*.

Format

This control statement data set usually resides in the input stream. However, it can also be defined as a sequential data set or as a member of a partitioned data set. It must contain 80-byte, fix-length records. Block size, if coded in your JCL, must be a multiple of 80.

This data set can contain up to three optional keyword parameters. It can be coded as shown in the following figure.

```
//areaxxx  DD *
DDNAME adsxx1
DDNAME adsxx2
```

Figure 54. FABCUR3 areaxxx data set

Syntax of the control statement

The "area*xxx*" keywords and their associated parameter values can be coded in free format (columns 1 - 72) if certain syntax rules are met:

1. All keyword parameter sets must be contained on one control statement.
2. All keywords must be separated from their associated parameter values by an equal sign. The equal sign must not be preceded by blanks, but can be followed by one or more blanks.
3. Parameter values must be separated from the next keyword by one or more blanks, a comma, or a comma followed by one or more blanks.

Control statement keywords

The area*xxx* control statement contains the following keyword:

```
[DDNAME adsxxx]
```

DDNAME

This statement specifies the ddname of output area data set. The combination of DDNAME keyword and ads*xxx* can be specified up to 7 sets, and at least 1 in an area*xxx*. If the ddname is specified with more than 8 characters, the first 8 characters are taken as a ddname.

An abbreviation DDN is used for DDNAME.

Consideration for ads*xxx* naming

The ddname of ads*xxx* must be specified the same as the one in the area*xxx* control statement. You can use any ddname except a reserved ddname or a ddname that has already been used in the same JCL.

Output for DEDB Reload

DEDB Unload/Reload generates output reports (that is, Audit Control reports and message reports).

DEDB Reload DURAUDIT DD data set

The reload program (FABCUR3) generates a Reload Audit Control report to provide verification totals.

Format

This data set contains 133-byte, fixed-length records. Block size, if coded in your JCL, must be a multiple of 133. It is better to code your DD statement as follows:

```
//DURAUDIT DD SYSOUT=A
```

Reload Audit Control report

Each reload job generates a Reload Audit Control report as shown as follows:

```
IMS HPFP UTILITIES - DEDBUR                          "RELOAD AUDIT CONTROL REPORT"                        PAGE:    1
5698-FPP                                             DATE: 11/22/2020  TIME: 20.36.26                    FABCUR3  -  V2R1


    SEGMENTS RELOADED FOR AREA:   1  (AREANAME: VRSTSS1 )

       SEG NAME   S/C    TYPE   SEG TOTAL
       --------   ---    ----   -----------

       TSSROOT     1     ROOT         87
       TSSSEQ1     2     SDEP          0
       TSSDIR1     3     DDEP        337
       TSSD11      4     DDEP          0
       TSSD111     5     DDEP          0
       TSSD12      6     DDEP          0
       TSSDIR2     7     DDEP        737
       TSSDIR3     8     DDEP        176
       TSSDIR4     9     DDEP        205
       TSSD41     10     DDEP          0
       TSSDIR5    11     DDEP         87
       TSSDIR6    12     DDEP        289
       TSSD61     13     DDEP          0
       TSSD611    14     DDEP          0
       TSSD612    15     DDEP          0
       TSSD613    16     DDEP          0

                               ------------
           ** AREA TOTAL **         1,918  (VRSTSS1)


    SEGMENTS RELOADED FOR AREA:   2  (AREANAME: VRSTSS2 )

       SEG NAME   S/C    TYPE   SEG TOTAL
       --------   ---    ----   -----------

       TSSROOT     1     ROOT         85
       TSSSEQ1     2     SDEP          0
       TSSDIR1     3     DDEP        345
       TSSD11      4     DDEP          0
       TSSD111     5     DDEP          0
       TSSD12      6     DDEP          0
       TSSDIR2     7     DDEP        667
       TSSDIR3     8     DDEP        214
       TSSDIR4     9     DDEP        198
       TSSD41     10     DDEP          0
       TSSDIR5    11     DDEP         85
       TSSDIR6    12     DDEP        301
       TSSD61     13     DDEP          0
       TSSD611    14     DDEP          0
       TSSD612    15     DDEP          0
       TSSD613    16     DDEP          0

                               ------------
           ** AREA TOTAL **         1,895  (VRSTSS2)
```

Figure 55. Reload Audit Control report (Part 1 of 2)

```
     SEGMENTS RELOADED FOR AREA:   3  (AREANAME: VRSTSS3 )

        SEG NAME   S/C    TYPE   SEG TOTAL
        --------   ---    ----   -----------

        TSSROOT     1     ROOT           83
        TSSSEQ1     2     SDEP            0
        TSSDIR1     3     DDEP          318
        TSSD11      4     DDEP            0
        TSSD111     5     DDEP            0
        TSSD12      6     DDEP            0
        TSSDIR2     7     DDEP          691
        TSSDIR3     8     DDEP          173
        TSSDIR4     9     DDEP          161
        TSSD41     10     DDEP            0
        TSSDIR5    11     DDEP           83
        TSSDIR6    12     DDEP          266
        TSSD61     13     DDEP            0
        TSSD611    14     DDEP            0
        TSSD612    15     DDEP            0
        TSSD613    16     DDEP            0

                                  ------------
           ** AREA TOTAL **             1,775  (VRSTSS3)


                                  ------------
           *** FILE TOTAL ***           5,588
```

Figure 56. Reload Audit Control report (Part 2 of 2)

This report contains a count of the number of segments loaded into each area. An area total and file total are also provided. The reload area total should match the corresponding area total in Part 2 and Part 3 of the Unload Audit Control report. If the segment counts for an area are zero, it means that the reloaded area is empty.

The file total should match the corresponding file total in Part 3 of the Unload Audit Control report.

The integrity of the reloaded database can be verified by running the DEDB Pointer Checker integrity verification procedure.

Note: The number of 512-byte blocks is shown on DURAUDIT of Unload/Reload in case of SDEP=PHYSICAL. Text '(PHYSICAL)' is printed after the number so that the Audit Control report of Unload/Reload implies SDEP=PHYSICAL.

DEDB Reload SYSPRINT DD data set

The SYSPRINT data set contains the messages issued by the FABCUR3 program.

Format

This data set contains 133-byte records, and block size (if coded in your JCL statement) must be a multiple of 133.

```
//SYSPRINT DD SYSOUT=A
```

Reload processor-Messages

The following figure shows examples of the "Messages" report.

```
IMS HPFP UTILITIES - DEDBUR                              "RELOAD PROCESSOR - MESSAGES"                              PAGE:    1
5698-FPP                                                 DATE: 11/22/2020  TIME: 20.36.26                          FABCUR3  -  V2R1


   FABC0300I - PROCESSING STARTED  FOR AREA   1  (AREANAME VRSTSS1 )


   FABC0302I - AREA INITIALIZATION STATISTICS - DBDNAME: VRSDSRF   AREANAME: VRSTSS1

            - ROOT PORTION:
                   CI'S/UOW ADDRESSABLE BY RMOD:        9
                   CI'S/UOW USED AS OVERFLOW:           4
                   TOTAL CI'S/UNIT OF WORK:            13
                   TOTAL UOW'S:                        10

            - INDEPENDENT OVERFLOW PORTION:
                   TOTAL CI'S:                         91

            - SEQUENTIAL DEPENDENT PORTION:
                   TOTAL CI'S:                         64

   FABC0303I - VSAM "BUFND" VALUES: AREA (SEQ):  45  WORK (SEQ):  45  WORK (DIR):  26


   FABC0301I - PROCESSING COMPLETE FOR AREA   1  (AREANAME VRSTSS1 )
IMS HPFP UTILITIES - DEDBUR                              "RELOAD PROCESSOR - MESSAGES"                              PAGE:    2
5698-FPP                                                 DATE: 11/22/2020  TIME: 20.36.26                          FABCUR3  -  V2R1


   FABC0300I - PROCESSING STARTED  FOR AREA   2  (AREANAME VRSTSS2 )


   FABC0302I - AREA INITIALIZATION STATISTICS - DBDNAME: VRSDSRF   AREANAME: VRSTSS2

            - ROOT PORTION:
                   CI'S/UOW ADDRESSABLE BY RMOD:       18
                   CI'S/UOW USED AS OVERFLOW:           9
                   TOTAL CI'S/UNIT OF WORK:            27
                   TOTAL UOW'S:                         5

            - INDEPENDENT OVERFLOW PORTION:
                   TOTAL CI'S:                        135

            - SEQUENTIAL DEPENDENT PORTION:
                   TOTAL CI'S:                        241

   FABC0303I - VSAM "BUFND" VALUES: AREA (SEQ):  45  WORK (SEQ):  45  WORK (DIR):  26


   FABC0309W - IOVF INTERVAL AT RBA: 00080800 IS FULL


   FABC0309W - IOVF INTERVAL AT RBA: 00080800 IS FULL


   FABC0309W - IOVF INTERVAL AT RBA: 00080800 IS FULL


   FABC0309W - IOVF INTERVAL AT RBA: 00080800 IS FULL


   FABC0309W - IOVF INTERVAL AT RBA: 00080800 IS FULL


   FABC0309W - IOVF INTERVAL AT RBA: 00080800 IS FULL


   FABC0301I - PROCESSING COMPLETE FOR AREA   2  (AREANAME VRSTSS2 )
```

Figure 57. Reload processor-Messages (Part 1 of 2)

```
IMS HPFP UTILITIES -DEDBUR                        "RELOAD PROCESSOR - MESSAGES"                              PAGE:    3
5698-FPP                                          DATE: 11/22/2020  TIME: 20.36.26                          FABCUR3  -  V2R1


    FABC0300I - PROCESSING STARTED  FOR AREA   3  (AREANAME VRSTSS3 )


    FABC0302I - AREA INITIALIZATION STATISTICS - DBDNAME: VRSDSRF   AREANAME: VRSTSS3

             - ROOT PORTION:
                   CI'S/UOW ADDRESSABLE BY RMOD:        4
                   CI'S/UOW USED AS OVERFLOW:           2
                   TOTAL CI'S/UNIT OF WORK:             6
                   TOTAL UOW'S:                        25

             - INDEPENDENT OVERFLOW PORTION:
                   TOTAL CI'S:                         78

             - SEQUENTIAL DEPENDENT PORTION:
                   TOTAL CI'S:                         64

    FABC0303I - VSAM "BUFND" VALUES: AREA (SEQ):  45  WORK (SEQ):  45  WORK (DIR):  26


    FABC0301I - PROCESSING COMPLETE FOR AREA   3  (AREANAME VRSTSS3 )


    FABC0310W - FABCUR3 ENDED WITH WARNINGS
```

Figure 58. Reload processor-Messages (Part 2 of 2)

Examples for DEDB Reload

There are many ways to use the DEDB Reload utility. The examples presented in this topic represent some of the typical ways that you can use.

By studying and understanding these examples, you can learn the techniques to use and to effectively manage your IMS DEDBs.

Example 1: Reloading from an input data set with three areas when IMS management of ACBs is enabled

This example JCL statement reloads from an input data set that has three areas.

The following table shows a reload example where the input data set contains segment data for three areas.

AREA #	UOW=	ROOT=	CI SIZE	Max IOVF
1	(16,3)	(6883,347)	2048	11
4	(15,2)	(4881,286)	4096	9
7	(24,5)	(3886,192)	2048	8

The work data set is allocated as follows:

```
        DEFINE CLUSTER (NAME(TSSV.RUN5.WORK) -
                   VOLUME(TSS001) -
                   NONINDEXED -
                   CISZ(4096) -
                   RECSZ(4089 4089) -
                   RECORDS(5552) -
                   REUSE) -
            DATA     (NAME(TSSV.RUN5.WORK.DATA))
```

Notes:

1. The largest CI size is 4 K.

2. The maximum number of IOVF CIs in any area being reloaded is 5552 (area number 1: 347 X 16).

The execution JCL for the reload Job must contain DD statements for each of the three areas (that is, the ddnames specified in the new DMB).

To retrieve ACB definitions from the staging IMS directory data set instead of the ACBLIB library, specify the following control statement keywords:

```
IMSCATHLQ=IMSVS.CATALOG
IMSCATACB_INPUT=PENDING
```

For optimum performance, the following control statement keyword should be specified:

```
IOVFBUF=11
```

Example 2: Restarting the reload process

The following figure shows example JCL statement for restarting the reload process.

This example shows a restart of the reload process from an error.

In this reload example, the reload process (from) must be restarted because of an I/O error during the load of area 4. Area 1 was successfully loaded.

Area 4 must be deleted and redefined (on another pack). The work data set and area 7 are not impacted.

The control statement is modified as follows:

```
IOVFBUF=9 STARTAREA=4
```

Example 3: Reloading multi-area data sets using data space as the work data set

The following figure shows example JCL statement for reloading multi-area data sets using data space as the work data set.

In this reload segment data from one area is combined into 3 multi-area data sets. First, the following data must be assembled:

- The area name is TSSDBD1, with one area defined.
- A single-area data set consists of 3 multi-area data sets.
- The area defined in the DMB is defined as follows:

```
AREA      DD1=TSSAAR0,                                             C
          SIZE=1024,                                              C
          UOW=(20,5),                                             C
          ROOT=(100,12)
```

- Each of the three data sets in the multi-area data set has data name of HPFP.TSSADBD.TSSAAR01, HPFP.TSSADBD.TSSAAR02, or HPFP.TSSADBD.TSSAAR03.

The JCL stream for this reloaded job can now be set up:

1. The output multi-area data sets must be allocated before processing FABCUR3.
2. DD statement for the work file for controlling IOVF must be removed, or specified as a dummy DD.

```
//DURIWRK DD  DUMMY
```

3. DD statements for the area*xxx* control statement must be included.

```
//TSSAAR0  DD *
DDNAME TSSAAR01
DDNAME TSSAAR02
DDNAME TSSAAR03
/*
```

4. DD statements for the output area data sets must be included.

```
//TSSAAR01  DD  DISP=OLD,DSN=HPFP.TSSADBD.TSSAAR01
//TSSAAR02  DD  DISP=OLD,DSN=HPFP.TSSADBD.TSSAAR02
//TSSAAR03  DD  DISP=OLD,DSN=HPFP.TSSADBD.TSSAAR03
```

Chapter 6. Site default support for FPB

This topic describes the site default support for FPB.

FPB allows you to specify site default parameters for its utilities: DEDB Pointer Checker (FABADA1 and FABADA4), DEDB Unload (FABCUR1), and DEDB Reload (FABCUR3).

The macros and sample JCL in the following table are provided to generate site default tables. If you want to change default values for keywords, you can use the following macros to generate a default table.

Table 39. Macros and JCL members for site default table

Program	Macro	Sample JCL
FABADA1	FABAOP1M	FABAOP1J
FABADA4	FABAOP1M	FABAOP1J
FABCUR1	FABCOP1M	FABCOP1J
FABCUR3	FABCOP3M	FABCOP3J

You might specify only the keywords that you want to change. This site default table library must be concatenated to the IMS HP Fast Path Utilities load module library in the JOBLIB or STEPLIB DD statement.

Use the TABLESET= parameter to specify the type of the table to be generated. The TABLESET= parameter is common to all macros described in this topic.

The keywords for the TABLESET= parameter are as follows:

USER
Builds a site default table. This is the default value.

SYSTEM
Builds a system default table that is to be used internally by the FPB program. Users of FPB should not specify this value.

DSECT
Builds a DSECT to map default table entries. Users of FPB should not specify this value.

When coding the macros, note the following:

- Under TABLESET=USER, specifying system default value will cause FABD3675I message to be generated and a table entry for the keyword value will not be generated.

- Under TABLESET=USER, coding the same macro more than once will cause FABD3676E message to be generated and will end with return code of 8. All necessary site default values for a macro must be specified in the same macro.

Topics:

- "FABAOP1M macro" on page 186
- "FABCOP1M macro" on page 187
- "FABCOP3M macro" on page 190

Related tasks

Running the DEDB Pointer Checker process
To use the DEDB Pointer Checker, you must run several programs. To run the programs, you must code JCL statements for those programs. The steps in a DEDB Pointer Checker job stream vary, depending on your particular functional options. But there are series of steps you must follow to run DEDB Pointer Checker.

Running DEDB Unload

The unload program (FABCUR1) is run as a standard z/OS batch job. You need to specify an EXEC statement and DD statements that define the input and output data sets in your JCL.

Running DEDB Reload

The reload program (FABCUR3) is run as one or more standard z/OS batch jobs. You need to specify an EXEC statement and DD statements that define the input and output data sets in your JCL.

FABAOP1M macro

This topic describes the site default support for FPB, FABAOP1M macro.

The following keywords can be specified for the site default values:

DBRC=
> Specifies whether to connect to the DBRC interface. This keyword is applicable to FABADA1 only.
>
> **Y | YES**
>> Builds a communication interface with DBRC.
>
> **N | NO**
>> Does not build a communication interface with DBRC. This is the system default value.

FORCE=
> Specifies the continuation of FABADA1 when area status is 'RECOVERY NEEDED' or when the database is accessed by other IMS online programs. This keyword is applicable to FABADA1 only.
>
> **Y | YES**
>> Continues the processing.
>
> **N | NO**
>> Stops the processing. This is the system default value.

VSAM=
> Specifies the type of database to process. This keyword is applicable to FABADA1 only.
>
> **Y | YES**
>> Processes VSAM data sets.
>
> **N | NO**
>> Processes image copy data sets. This is the system default value.

SDEP=
> Specifies whether to verify and report any SDEP pointer integrity errors in the database. This keyword is applicable to FABADA1 only.
>
> **Y | YES**
>> Verifies and reports any SDEP pointer integrity errors.
>
> **N | NO**
>> Does not verify or report any SDEP pointer integrity errors. This is the system default value.

OVRD=
> Specifies whether to override a DBD member date (that is, the date of an ACBGEN for the DBD) and image copy date discrepancy. This keyword is applicable to FABADA1 only.
>
> **Y | YES**
>> Overrides date discrepancy.
>
> **N | NO**
>> Does not override date discrepancy. This is the system default value.

STATS=
> Specifies whether to generate the run totals and statistics. This keyword is applicable to FABADA1 only.
>
> **Y | YES**
>> Reports the run totals and statistics.

N | NO
Does not report the run totals and statistics. This is the system default value.

TBLENTRY=
Specifies the entry number of the SDEP table (SDEPTBL). This keyword is applicable to FABADA1 only.

1 - 99999999
The default value is 10000.

SPTFALL=
Specifies whether to validate the SPTF in a valid SDEP segment points to a deleted RBA that is located at a physically higher RBA number. This keyword is applicable to FABADA1 only.

Y | YES
Specifies that pointer information for all valid SDEP segment RBAs is kept in the SDEP table.

N | NO
Specifies that the SPTF RBA will be checked only to determine whether it points to a deleted segment whose RBA is physically smaller than the SPTF RBA. This is the system default value.

IMSCATHLQ=
Specifies to retrieve ACBs from the IMS directory (an extension of the IMS catalog) instead of from the ACB library. This keyword is applicable to both FABADA1 and FABADA4.

bsdshlq
Specifies to read ACBs from the IMS directory by using the IMS Tools Catalog Interface. *bsdshlq* specifies the high-level qualifier of the bootstrap data set of the IMS directory. IMSCATHLQ=*bsdshlq* is effective when the IMS catalog and the IMS management of ACBs are enabled.

*NO
Specifies to read ACBs from the ACB library. This is the system default value.

IMSCATACB_INPUT=
Specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. IMSCATACB_INPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* option is specified. This keyword is applicable to both FABADA1 and FABADA4.

CURRENT
Currently active ACB members are retrieved from the IMS directory data sets. This is the system default value.

PENDING
Pending ACB members are retrieved from the staging data set.

FABCOP1M macro

This topic describes the site default support for FPB, FABCOP1M macro.

The following keywords can be specified for the site default values:

TYPERUN= or TYPRUN=
Specifies whether to run the reorganization process.

REORG
Unloads or reloads one or more areas without changing the database structure. This is as same as REORG in the DBDNAME control statement. Even if this keyword is specified, it is overridden by NEWACBNM=, HIERCHNG=, and RMODTYPE= keywords of the DBDNAME= control statement.

NEWACB
Unloads or reloads with new ACBLIB. Reorganization is assumed if no NEWACB DD is provided. This is the system default value.

DBRC=
Specifies whether to connect to the DBRC interface.

Y | YES

Builds a communication interface with DBRC.

N | NO

Does not build a communication interface with DBRC. This is the system default value.

STATS=

Specifies whether to print the scheduling parameter values, VSAM buffer statistics, and write-latch-contention statistics.

Y | YES

Reports the statistics.

N | NO

Does not report the statistics. This is the system default value.

COMPRESS=

Specifies whether the segments of the unloaded record should be compressed if the segment edit/compression routine is defined for the segment in DBDGEN.

Y | YES

Specifies that the unloaded record should contain compressed segments.

N | NO

Specifies that the unloaded record should contain segments that are expanded. This is the system default value.

SDEP=

Specifies whether to unload SDEP segments.

N | NO

Does not unload SDEP segments. This is the system default value.

LOGICAL

Unloads SDEP segments in logical order.

PHYSICAL

Unloads SDEP segments in physical order.

ACCESS=

Specifies the type of access needed to read the area data set VSAM ESDS file.

VSAM

Uses the VSAM ICIP access method.

FAST

Uses DFSMS Media Manager. This is the system default value.

PAD=

Specifies the padding character to be used when the segment minimum length is increased due to DBD definition change. This keyword is effective for fixed-length segments and variable-length segments.

Code the value as two hexadecimal digits, as follows: X'NN'. The default value is null (X'00').

FORMAT=

Specifies the format of the unloaded segment records.

DBT

Specifies that the format of the unloaded segment records is same as IMS DBT 2.x. This is the system default value.

TFMT

Specifies that the format of the unloaded segment records is enhanced, which means that the prefix part of the record is generated based on the maximum number of segment levels defined in DBD.

PTRERROR=

Specifies the action when a pointer error is detected.

ABEND

Ends abnormally. This is the system default value.

BYPASS

Ignores the segments with pointer errors and continues the unloading process.

KEYSEQERROR=

Specifies whether to run a key sequence check process, and the action to take during the check process when a key sequence error in a sequence field is detected.

NOCHECK

Does not run a key sequence check process. This is the system default value.

ABEND

Ends abnormally.

BYPASS

Skips error segment and its child segments, and continues the unload processing for succeeding valid segments.

KEYCHGCHECK=

Specifies whether to check the key sequence field value for changes that are made by a user exit routine.

Y | YES

Checks the altered key sequence field. This is the system default value.

N | NO

Does not check the altered key sequence field.

LRECL=

Specifies the LRECL of the unloaded segment records file.

SEGMENT

LRECL is determined based on the maximum length of segments defined in DBD. This is the system default value.

SEGTFMT

LRECL is determined based on the TFMT format prefix and the maximum length of segments defined in DBD. This option is effective only when the FORMAT=TFMT option is specified. Otherwise, it is treated as LRECL=SEGMENT.

BLOCK

LRECL is determined as BLKSIZE -4.

nnnnn

Value of LRECL. The value can be up to 32760.

LOADPLACE=

Specifies the place of segments that are specified by the LOADCTL= control statement.

RAP

All segments that should be "far" from their root segment will be placed in the available RAP CI space. If the space is not enough in the RAP CI, they are placed in the DOVF or the IOVF part. This is the system default value.

DOVF

All segments that should be "far" from their root segment will be placed in the available DOVF CI space. If the space is not enough in the DOVF CIs, they are placed in the DOVF or the IOVF part.

IOVF

All segments that should be "far" from their root segment will be placed in the available IOVF CI space.

AREC=

Specifies whether to generate an area information record in an unloaded segment file.

Y | YES

Generates an area information record. This is the system default value.

N | NO

Does not generate an area information record.

IMSCATHLQ=

Specifies to retrieve ACBs from the IMS directory (an extension of the IMS catalog) instead of from the ACB library.

bsdshlq

Specifies to read ACBs from the IMS directory by using the IMS Tools Catalog Interface. *bsdshlq* specifies the high-level qualifier of the bootstrap data set of the IMS directory. IMSCATHLQ=*bsdshlq* is effective when the IMS catalog and the IMS management of ACBs are enabled.

*NO

Specifies to read ACBs from the ACB library. This is the system default value.

IMSCATACB_INPUT=

Specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. IMSCATACB_INPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* option is specified.

CURRENT

Currently active ACB members are retrieved from the IMS directory data sets. This is the system default value.

PENDING

Pending ACB members are retrieved from the staging data set.

FABCOP3M macro

This topic describes the site default support for FPB, FABCOP3M macro.

The following keywords can be specified for the site default values:

IOVFBUF=

Specifies the number of IOVF buffers to allocate for direct processing of the IOVF work data set. Specifying less than 4 will be replaced to 4 at run time. Specify a value in the range of 1 - 99.

IMGCPY=

Specifies whether image copies of the areas being reloaded are to be created.

Y | YES

A single image copy of each area being reloaded is created.

N | NO

No image copies are created. This is the system default value.

DUAL

Dual image copies of each area being reloaded are created.

ICHASH=

Specifies whether the HASH check option of DEDB Pointer Checker is invoked while the image copy is being created.

Y | YES

HASH check is performed.

N | NO

HASH check is not performed. This is the system default value.

SSP

Subset pointer check is performed during the HASH check. This subparameter is effective only when ICHASH=(YES,) is specified.

ICCOMPRESS=

Specifies whether the ICE Image Copy function compresses the output image copy data.

Y | YES

The compression routine has been invoked. This subparameter is effective only when IMGCPY=YES or IMGCPY=DUAL is specified.

N | NO

The compression routine is not invoked. This is the system default value.

ROUTINE

The compression routine. ICE provides two compression exit routines: FABJCMP1 and FABJCMP2. Specify one of these compression routines to be called by the ICE Image Copy function. If no compression routine is specified, FABJCMP1 is used as the default. This subparameter is effective only when ICCOMPRESS=(YES,) is specified.

DBRC=

Specifies whether to connect to the DBRC interface.

Y | YES

Builds a communication interface with DBRC.

N | NO

Does not build a communication interface with DBRC. This is the system default value.

KEYSEQERROR=

Specifies the action to take when a key sequence error is detected.

NOCHECK

Does not run a key sequence check process. This is the system default value.

ABEND

Ends abnormally.

BYPASS

Skips error segment and its child segments, and continues the reload processing for succeeding valid segments.

SDEPRELOCATE=

Specifies whether SDEP segments of the areas are to be relocated.

Y | YES

Relocates SDEP segments.

N | NO

Does not relocate SDEP segments. This is the system default value.

KEYCHGCHECK=

Specifies whether to check the key sequence field value for changes that are made by a user exit routine.

Y | YES

Checks the altered key sequence field. This is the system default value.

N | NO

Does not check the altered key sequence field.

RAPERROR=

Specifies whether FABCUR3 checks the validity of RAP data in the prefix part of the unloaded segment record by using the result of the randomizer call.

NOCHECK

Does not check the validity of the RAP data in the prefix part of the unloaded segment record by using a randomizer. This is the system default value.

ABEND

Checks the validity of the RAP data in the prefix part of the unloaded segment record by using a randomizer. FABCUR3 ends abnormally with an error message if it finds an error. The DBDNAME= keyword and the ACBLIB DD are required for the ABEND option.

IMSCATHLQ=
Specifies to retrieve ACBs from the IMS directory (an extension of the IMS catalog) instead of from the ACB library.

> **bsdshlq**
> Specifies to read ACBs from the IMS directory by using the IMS Tools Catalog Interface. *bsdshlq* specifies the high-level qualifier of the bootstrap data set of the IMS directory. IMSCATHLQ=*bsdshlq* is effective when the IMS catalog and the IMS management of ACBs are enabled.

> ***NO**
> Specifies to read ACBs from the ACB library. This is the system default value.

IMSCATACB_INPUT=
Specifies whether to retrieve currently active ACB definitions or pending ACB definitions from the IMS directory. IMSCATACB_INPUT keyword is effective only when the IMSCATHLQ=*bsdshlq* option is specified.

> **CURRENT**
> Currently active ACB members are retrieved from the IMS directory data sets. This is the system default value.

> **PENDING**
> Pending ACB members are retrieved from the staging data set.

Chapter 7. Reference: Initialization user exit routines (FABC1IE0 and FABC3IE0)

DEDB Unload (FABCUR1) and DEDB Reload (FABCUR3) support the interface for user exits FABC1IE0 and FABC3IE0.

Interface of the initialization user exit routines

FABC1IE0 and FABC3IE0 are invoked before the unload and reload processing by FABCUR1 and FABCUR3.

- Initialization user exit routine must be named FABC1IE0 for FABCUR1, and FABC3IE0 for FABCUR3, and reside in STEPLIB or JOBLIB. FABC1IE0, if provided, is invoked before the unload process by FABCUR1. FABC3IE0, if provided, is invoked before the reload process by FABCUR3.
- If either FABC1IE0, FABC3IE0, or both exist, they will be invoked before the unload or the reload processing.
- If the return code is not zero, FABCUR1 or FABCUR3 abends with a code of U3811 or U3812.

The following figure shows the input and output of the initialization user exit routine.
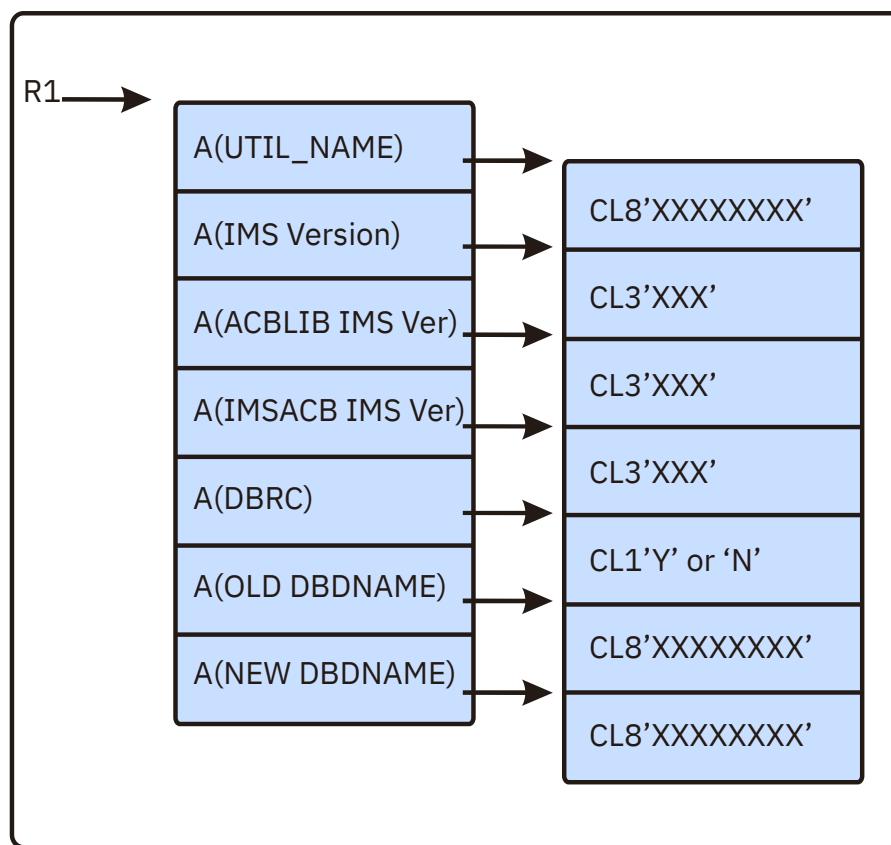
Figure 59. Parameter list

Input

> **R1**
> Parameter list address.
> **R13**
> Save area address (A(18F) This save area is cleared with null).

R14
    Return address.

R15
    Entry point of FABC1IE0 or FABC3IE0.

A(UTIL_NAME):
    FABCUR1 or FABCUR3.

A(IMS Version):
    The address of the SSCDIMSR field of the IMS SCD control block.

A(ACBLIB IMS Ver):
    OLDACB IMS version of ACBLIB for unload.

    DURDBDFN IMS version of ACBLIB for reload.

A(NEWACB IMS Ver):
    NEWACB IMS version of ACBLIB (only for unload).

A(DBRC):
    DBRC parameter on EXEC statement.

A(OLD DBDNAME):
    OLDACB DBD name.

A(NEW DBDNAME):
    NEWACB DBD name (only for unload).

Output
    R15 Return code.

Chapter 8. Reference: Return code user exit routine (FABARCEX)

Fast Path DEDB Pointer Checker supports a return code user exit routine that can be used to change the return codes of FABADA1 and FABADA3.

Overview of the return code user exit routine

This topic describes the overview of the return code user exit routine.

- This user exit routine must be named FABARCEX, and should be included in STEPLIB or JOBLIB.
- If FABARCEX is found by DEDB Pointer Checker, it will be called no matter what the return code is.
- You can change the return codes according to the message and return code table.
- FABARCEX must be specified with AMODE=31 and RMODE=ANY.
- DEDB Pointer Checker provides a sample JCL (FABARCET) in *.SHFPSAMP. FABARCET is composed of two steps: one to assemble FABARCEX, and another to link-edit FABARCEX.
- For this release, the interface is available only for FABADA1 and FABADA3.

Interface of the return code user exit routine

The following figure shows the input and output of the return code user exit routine.
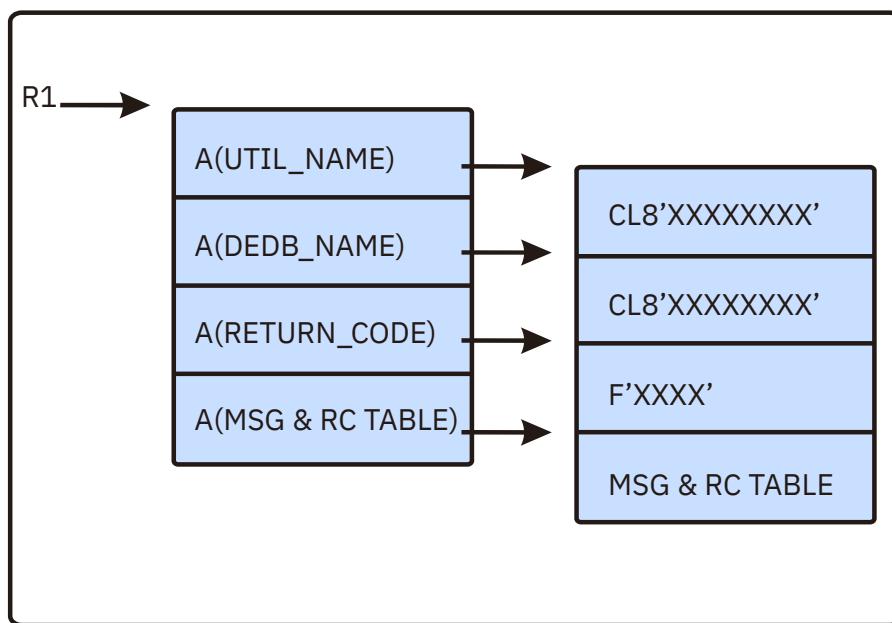
Figure 60. Parameter list

Input

R1
Parameter list address

R13
Save area address

R14
Return address

R15
Entry point of FABARCEX

A(UTIL_NAME):
Address of the 8-byte utility name; either FABADA1 or FABADA3.

A(DEDB_NAME):
Address of the 8-byte DEDB name. It is blank when the utility is FABADA3.

A(RETURN_CODE):
Address of the full-word return code. Input: The original return code. Output: The changed return code.

A(MSG &RC TABLE):
The address of the message and return code table. Every entry contains the following information: Message ID: 4-byte Message suffix: 1-byte Reserved area: 3-byte Return code: Full-word, currently this field is not used.

The following table shows an example of a message and return code table. If FABA0301I, FABA0330E, FABA0305I, FABA0302E, FABA0315I are issued, the message and return code table would look like:

Table 40. Example of a message and return code table

Message ID	Message suffix	Reserved area	Reserved area
X'00000000'	X'00'	X'000000'	X'00000005'
X'F0F3F0F1'	X'C9'	X'000000'	X'00000000'
X'F0F3F3F0'	X'C5'	X'000000'	X'00000000'
X'F0F3F0F5'	X'C9'	X'000000'	X'00000000'
X'F0F3F0F2'	X'C5'	X'000000'	X'00000000'
X'F0F3F1F5'	X'C9'	X'000000'	X'00000000'

Where X'00000005' is the number of entries in the message and return code table.

Output
The changed return code that is pointed to by address in A(RETURN_CODE).

Chapter 9. Troubleshooting

These topics provide technical references to help you troubleshoot and diagnose IMS Fast Path Basic Tools problems.

Topics:

Return codes of FPB

This topic provides detailed information about the return codes that are issued by DEDB Pointer Checker, DEDB Tuning Aid, DEDB Unload, and DEDB Reload utilities.

The following subsections describe the return codes of each component.

Subsections:

FABADA1

This program writes numbered messages to the SYSPRINT data set that more fully explain the results of program execution. FABADA1 supports a return code user exit routine that can change the return codes. The return codes are as follows:

Code
Meaning

0

The requested operation has been completed successfully.

4

Warning messages were issued, but the requested operation completed.

99

Pointer errors, FSE or space utilization discrepancies, or other IMS or VSAM control field errors were detected, but the requested operation completed.

FABADA3

This program writes numbered messages to the SYSPRINT data set that more fully explain the results of program execution. FABADA3 supports a return code user exit routine that can change the return codes. The return codes are as follows:

Code
  Meaning

0

  The requested operation has been completed successfully.

4

  Warning messages were issued, but the requested operation completed.

99

  Pointer errors, FSE or space utilization discrepancies, or other IMS or VSAM control field errors were detected, but the requested operation completed.

FABADA4

This program writes numbered messages to the SYSPRINT data set that more fully explain the results of program execution. The return codes are as follows:

Code
  Meaning

0

  The requested operation has been completed successfully.

4

  Warning messages were issued, but the requested operation completed.

FABADA5

This program writes numbered messages to the SYSPRINT data set that more fully explain the results of program execution. The return codes are as follows:

Code
  Meaning

0

  The requested operation has been completed successfully.

4

  There are no database records in the area. The free space report generation completed successfully.

97

  SDEP warning message was issued. SDEP information is not reported.

98

  The threshold parameter values were invalid. Threshold analysis is bypassed. Report generation completed successfully.

99

  Threshold warning messages were issued. Report generation completed successfully.

FABADA8

The return code is as follows:

Code
  Meaning

0

  The requested operation has been completed successfully.

FABADA9

This program writes numbered messages to the MSGOUT data set that more fully explain the results of program execution. The return codes are as follows:

Code
> **Meaning**

0
> The requested operation has been completed successfully.

4
> Warning messages were issued, but the requested operation completed.

99
> Threshold values were exceeded and warning messages were generated. The requested operation completed.

FABBME1

The FABBME1 program writes numbered messages to the SYSPRINT data set that more fully explain the result of program execution. The return codes are as follows:

Code
> **Meaning**

0
> The requested operation has been completed successfully.

4
> Warning messages were issued, but the requested operation was completed.

8
> Severe errors, causing job termination, occurred.

FABBME3

The FABBME3 program writes numbered messages to the SYSPRINT data set that more fully explain the result of program execution. The return codes are as follows:

Code
> **Meaning**

0
> The program has been completed successfully.

4
> Warning messages were issued, but the requested operation was completed.

8
> Errors causing job termination occurred.

FABCUR

The FABCUR programs write numbered messages to the SYSPRINT data set that more fully explain the result of program execution. The return codes are as follows:

Code
> **Meaning**

0
> The requested operation has been completed successfully.

4
> Warning messages were issued, but the requested operation was completed.

8
> Severe errors; the job was ended.

Abend codes of FPB

The abend code, which is a 4-digit number, is related to the numeric part of message ID.

FPB messages start with FAB*x*. If an abend code is issued, determine which utility issued the abend code by referring to the "Messages" on page 200. Then, add the four-digit number that was issued as the abend code, and look up the message explanation for that message.

For example, if abend code 3501 was issued when running a DEDB Pointer Checker job, add prefix FABA (message prefix for DEDB Pointer Checker) and suffix E to the abend code, and look up the explanation of message FABA3501E.

Report any undocumented 39*xx* or 40*xx* abend codes to IBM.

Messages

Use the information in these topics to help you diagnose and solve IMS Fast Path Basic Tools problems.

For each message, the following accompanying information is provided where applicable:

Explanation:
This explains what the message text means, what caused the message to be issued, and what its variable entry fields are (if any).

System action:
This explains what the system will do next

User response:
This describes whether a response is necessary, what the appropriate response should be, and what the resulting effect is on the system or program.

Message prefixes

The following table shows the prefixes of messages and the utility or the process that issues the messages.

Table 41. Message prefixes

Prefix	Utility or process
FABA	DEDB Pointer Checker of FPB
FABB	Tuning Aid of FPB
FABC	Unload or Reload of FPB
FABD	Common routines of FPB
FABU	Diagnostics Aid utilities. For Diagnostics Aid messages, see the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide.*

Message suffixes

Some messages provide additional information by including the following suffixes:

A

Indicates that operator intervention is required before processing can continue.

I

Indicates that the message is informational only.

W

Indicates that the message is a warning to alert you to a possible error condition.

E

Indicates that an error occurred, which might or might not require operator intervention.

Description for FABc messages

The messages describe conditions detected during program execution and any errors encountered. Very severe errors result in both an error message and a request for a "User Code" abend. For a User Abend condition, the error message is written to the SYSPRINT data set (MSGOUT data set in FABADA9) and issued as a WTO to the job log.

Each of the FPB messages has one of the following formats:

- FAB*c*0*nxxy text*
- FAB*c*35*xxy text*
- FAB*c*36*xxy text*
- FAB*c*37*xxy text*
- FAB*c*38*xxy text*
- FAB*c*39*xxy text*
- FAB*c*40*xxy text*

Note: FAB*c*35*xxy*, FAB*c*36*xxy*, FAB*c*37*xxy*, FAB*c*38*xxy*, FAB*c*39*xxy*, and FAB*c*40*xxy* is associated with an abend condition.

where:

c
Describes which component issues the message. See the message prefixes section in this topic.

n
Is the suffix of the program name (for example, FABADA1, *n* is 1).

xx
Is a two-digit number associated with the information or error message being issued.

y
Describes the severity of the message. See the message suffixes section in this topic.

FABA messages

The following information is about messages and codes that begin with FABA.

FABA0100I   **FABADA1 ENDED NORMALLY ("TYPRUN=*xxxxxxx*") NO POINTER ERRORS DETECTED**

Explanation

This message is generated when program FABADA1 completes successfully.

System action

FABADA1 ends normally with a completion code of zero.

User response

Proceed to perform the remainder of the job steps in the report generation procedure.

FABA0100W   **FABADA1 ENDED WITH WARNINGS ("TYPRUN=*xxxxxxx*") NO POINTER ERRORS DETECTED**

Explanation

One or more warning conditions were encountered during the execution of program FABADA1.

System action

FABADA1 ends normally with a completion code of four.

User response

Determine the cause of the warning using the other messages generated. Correct the problem and rerun FABADA1, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0100E   **FABADA1 ENDED WITH ERRORS ("TYPRUN=*xxxxxxx*")**

Explanation

One or more integrity error conditions were encountered during the execution of program FABADA1.

System action

FABADA1 ends normally with a completion code of ninety-nine.

User response

Determine the cause of the error using the other messages generated. Correct the problem and rerun FABADA1, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0101I	PROCESSING COMPLETE FOR DBDNAME: *dbdname* AREA: *areaname*

Explanation

This message is generated when all data for the area *areaname* of the database *dbdname* has been processed.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0102I	DEDB ANALYSIS RECORDS WRITTEN: DDNAME *ddname zzz,zzz,zz9* (*x,x,x :zzz,zzz,zz9*)

Explanation

This message is generated when the STATS keyword is specified on the control statement. The number and the type of records written are shown in the message.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0103I	SORT CONTROL CARDS WRITTEN TO DDNAME: *ddname* SORT FIELDS=, SIZE=, RECORD TYPE=, LENGTH=

Explanation

This message is generated when the sort control statements for the DEDB Pointer Checker record data sets have been generated and written to the ddname specified. The sort control statements generated are shown in the message.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0104I	SDEP PART PROCESSED FOR AREA: *areaname*

Explanation

This message is generated when the "SDEP" keyword has been specified on the control statement, and program FABADA1 has successfully processed the SDEP part associated with the area *areaname*.

System action

FABADA1 continues processing.

User response

None. This message is informational.

FABA0105I	SDEP PROCESSING STARTED - LB: CYCLE# REL-BYTE-ADDR LE: CYCLE# REL-BYTE-ADDR

Explanation

This message is generated when SDEP processing is about to start. LB specifies the logical beginning of the SDEP part of the area and LE specifies the logical end of the SDEP part of the area.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0106E	[POINTER ERRORS DETECTED | ERROR MESSAGES GENERATED]

Explanation

These two messages are generated as WTO messages at the end of FABADA1 job step if FABADA1 detected any pointer errors during processing.

System action

Program FABADA1 ends normally with a completion code of 99.

User response

Determine the cause of the error using the other messages generated. Correct the problem and rerun FABADA1, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0107E	[INVALID CI'S ENCOUNTERED | ERROR MESSAGES GENERATED]

Explanation

These two messages are generated as WTO messages at the end of FABADA1 job step if FABADA1 detected any invalid CIs during processing.

System action

Program FABADA1 ends normally with a completion code of 99.

User response

Determine the cause of the error using the other messages generated. Correct the problem and rerun FABADA1, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0108E	EQE DETECTED FOR CI AT RBA: *xxxxxxxx*

Explanation

Program FABADA1 found that the CI at RBA *xxxxxxxx* has an EQE.

System action

FABADA1 increments the internal error counter, bypasses the CI specified, and continues processing. FABADA1 ends with an abend code of 3520 if the value of the internal error counter exceeds 10.

User response

Run the Full Recovery Utility and rerun the job, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0109E	EEQE DETECTED FOR CI AT RBA: *xxxxxxxx*

Explanation

Program FABADA1 found that the CI at RBA *xxxxxxxx* has an EEQE.

System action

FABADA1 increments the internal error counter, bypasses the CI specified, and continues processing. FABADA1 ends with an abend code of 3520 if the RBA specifies a second CI.

User response

Run the Full Recovery utility and rerun the job, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0110E	INVALID CONTROL CARD ENCOUNTERED

Explanation

The user-supplied control statement was found to contain one or more errors.

System action

Program FABADA1 issues one or more of the following error messages describing the errors encountered:

UNKNOWN KEYWORD AT COLUMN *xx*

INVALID SPECIFICATION FOR "parm" PARM
- VALUE FOR DATABASE NAME NOT PRESENT
- DATABASE NAME IS LONGER THAN 8 CHARACTERS
- VALUE FOR AREA DDNAME NOT PRESENT
- AREA DDNAME IS LONGER THAN 8 CHARACTERS
- AREA DDNAME REQUIRED FOR INPUT=VSAM

FABADA1 ends with an abend code of 3503

User response

Make sure that there is a valid specification for the "DBDNAME=" parameter in columns 1 - 72. If the "VSAM" keyword is specified, make sure that there is a valid specification for the "AREA=" parameter on the control statement. Correct the control statement, and rerun the job.

FABA0111W	*xxxxxxxxxxxx* DATE/TIME STAMP LATER THAN DATE/TIME STAMP FROM IMAGE-COPY DATA SET

ACB MEMBER CREATED: *yyyy.ddd/ hh:mm:ss*
IMAGE COPY CREATED: *yyyy.ddd/ hh:mm:ss*; ===>> PROCESSING CONTINUES ("OVRD" REQUESTED)

Explanation

Program FABADA1 determined that the DMB read from the ACB library or the IMS directory was created after the image copy of the DEDB was created.

System action

If the control statement contained the "OVRD" keyword, FABADA1 sets an end-of-job return code of 4, and continues processing.

User response

Make sure that this situation is not an error condition, and proceed to perform the remainder of the job steps in the report generation procedure.

FABA0111E	*xxxxxxxxxxxx* DATE/TIME STAMP LATER THAN DATE/TIME STAMP FROM IMAGE-COPY DATA SET ACB MEMBER CREATED: *yyyy.ddd/ hh:mm:ss* IMAGE COPY CREATED: *yyyy.ddd/ hh:mm:ss*

Explanation

Program FABADA1 determined that the DMB read from the ACB library or the IMS directory was created after the image copy of the DEDB was created. The "OVRD" keyword is not specified on the control statement.

System action

FABADA1 ends with an abend code of 3527.

User response

Make sure that the DMB from the library associated with the DD statement ACBLIB is a correct specification for the DEDB areas contained in the files associated with the DFSUDUMP DD statement. If this is the case, rerun the job specifying the "OVRD" keyword on the control statement. Otherwise, re-create the image copy data sets and rerun the job.

FABA0113I	INFORMATION OF THE DB DEFINITION WAS OBTAINED FROM *resource*

Explanation

This message indicates the resource (ACB library or IMS directory) where FABADA1 obtained DMB definitions from.

System action

FABADA1 continues processing.

User response

None. This message is informational.

FABA0120E	I/O ERROR ATTEMPTING VSAM "READ" - REQUESTED RBA: *eeeeeeee* - VSAM ERROR DATA: RETURN CODE : *aaa* (*bb*) RPL FDBK: *ccc* (*dd*)

Explanation

Program FABADA1 received a nonzero return code from VSAM while attempting to read a record from the VSAM data set being analyzed. The RBA of the CI being read is shown in hexadecimal format. The return code and RPL FDBK code values are shown in both decimal (*aaa*, *ccc*) and hexadecimal (*bb*, *dd*) format.

System action

FABADA1 increments the internal error counter, and continues processing. FABADA1 ends with an abend code of 3520 if the value of the internal error counter exceeds 10.

User response

See *DFSMS Macro Instructions for Data Sets* for a further explanation of the error. Correct any errors. Rerun the job. If this situation persists, report it to systems operations personnel.

FABA0130E	INVALID [BLK TYPE ID | SEGM CODE] IN CI AT RBA: *xxxxxxxx* - CI BYPASSED (DATA VALUE: *yy* OFFSET: *zzz*)

Explanation

Program FABADA1 encountered an invalid block type (DBLKBTID) or an invalid segment code during the serial "deblocking" of the CI at RBA *xxxxxxxx*. An incorrect length of a segment or FSE contained in the specified CI could also cause this condition.

System action

FABADA1 increments the internal error counter, bypasses the CI specified, and continues processing. FABADA1 ends with an abend code of 3529 if the value of the internal error counter exceeds 10.

User response

Correct the errors, and rerun the job.

FABA0131E   **TOTAL FSE LENGTH DISCREPANCY - RBA:** *xxxxxxxx*

Explanation

Program FABADA1 determined that the total free space in the CI at RBA *xxxxxxxx* as calculated by "chasing" the FSE chain did not correspond to the value calculated during the serial "deblocking" of that CI. This condition may have been caused by an incorrect FSE chain, or by an incorrect FSE or segment length.

System action

FABADA1 increments the internal error counter, and continues processing. FABADA1 ends with an abend code of 3529 if the value of the internal error counter exceeds 10.

User response

See "DEDB integrity verification" on page 11 for a description of the corrective action required.

FABA0132E   **TOTAL NO. OF FSE DISCREPANCY - RBA:** *xxxxxxxx*

Explanation

Program FABADA1 determined that the number of Free Space elements in the CI at RBA *xxxxxxxx*, as calculated by "chasing" the FSE chain, did not match the value calculated during the serial "deblocking" of that CI. This condition may have been caused by an incorrect FSE chain, or by an incorrect FSE or segment length.

System action

FABADA1 increments the internal error counter, and continues processing. FABADA1 ends with an abend code of 3529 if the value of the internal error counter exceeds 10.

User response

See "DEDB integrity verification" on page 11 for a description of the corrective action required.

FABA0133E   **CI "SPACE USAGE" DISCREPANCY - RBA:** *xxxxxxxx*

Explanation

Program FABADA1 determined that the sum of the Free Space element, Scrap, and segment lengths in the CI at RBA *xxxxxxxx* encountered during the serial "deblocking" of that CI was not equal to the usable space of the CI. This condition may have been caused by an incorrect FSE chain, or by an incorrect FSE or segment length.

System action

FABADA1 increments the internal error counter, and continues processing. FABADA1 ends with an abend code of 3529, if the value of the internal error counter exceeds 10.

User response

See "DEDB integrity verification" on page 11 for a description of the corrective action required.

FABA0134E   **FSE CHAIN POINTS TO A NON FSE - RBA:** *xxxxxxxx*

Explanation

Program FABADA1 encountered an invalid Free Space Element while following the FSE chain for the CI at RBA *xxxxxxxx* (that is, first byte not X'80').

System action

FABADA1 increments the internal error counter, and continues processing. FABADA1 ends with an abend code of 3529, if the value of the internal error counter exceeds 10.

User response

See "DEDB integrity verification" on page 11 for a description of the corrective action required.

FABA0135E   **SEGMENT AT RBA:** *xxxxxxxx* **(SEGCD:** *yy***) HAS PCF/PCL/SSPTR DISCREPANCY**

Explanation

Program FABADA1 encountered an error in the segment at RBA *xxxxxxxx* while checking the PCF/PCL/SSPTR pointer inter dependencies.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

Make sure that the pointer values contained in the segment at RBA *xxxxxxxx* meet the following criteria:

1. If PCF pointer value is zero, the associated PCL pointer value and Subset pointer value must also be zero.
2. If PCF pointer value is nonzero, the associated PCL pointer value must also be nonzero.

For further information, see "DEDB integrity verification" on page 11. Correct any errors, and rerun the job if necessary.

FABA0136E	POINTER (PTF/PCF) ERRORS IN AREA: *areaname*

Explanation

Program FABADA1 performed a checksum validation test of the RBAs of all segments in the specified area versus the values of their PCF and PTF pointers. The test failed. This shows that the area *areaname* contained PTF/PCF pointer integrity problems.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

Rerun the job in "TYPRUN=PTRALL" mode if required, and run FABADA3 and FABADA4. See "Running the DEDB Pointer Checker process" on page 20. Also, see "DEDB integrity verification" on page 11 for further information.

FABA0137E	POINTER (PCL) ERRORS IN AREA: *areaname*

Explanation

Program FABADA1 performed a checksum validation test of the RBAs of all segments referenced by a PCL pointer versus the values of the PCL pointers. The test failed. This shows that the area *areaname* contained PCL pointer integrity problems.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

Rerun the job in "TYPRUN=PTRALL" mode if required, and run FABADA3 and FABADA4. See "Running the DEDB Pointer Checker process" on page 20. Also, see "DEDB integrity verification" on page 11 for further information.

FABA0138E	POINTER (SDEP) ERRORS IN AREA: *areaname*

Explanation

Program FABADA1 performed a checksum validation test of the RBAs of all ROOT and SDEP segments in the specified area versus the values of their SDEP pointers. The test failed. This shows that the area *areaname* contained SDEP pointer integrity problems.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

If the exact RBA of the errors is needed, rerun the job in "TYPRUN=PTRALL" mode, and run FABADA3 and FABADA4. See "Running the DEDB Pointer Checker process" on page 20. Also, see "DEDB integrity verification" on page 11 for further information.

FABA0139E	SEGMENT AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS INVALID SDEP POINTER VALUE

Explanation

The SDEP pointer value contained in the segment at RBA *xxxxxxxx* does not point to the SDEP part of the DEDB area.

System action

Program FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11. Correct the error and rerun the job, or continue with the execution of FABADA3, FABADA4, and FABADA5 as desired.

FABA0140E   INVALID VALUE IN "DMACXVAL"/"DMACNXTS" - SDEP PROCESSING BYPASSED FOR AREA: *areaname*

Explanation

Program FABADA1 encountered an error while validating the "logical beginning" and the "logical end" of the SDEP part associated with the specified area. The 8-byte field of "DMACXVAL" contained a value higher than the 8-byte field of "DMACNXTS".

System action

FABADA1 continues processing. The SDEP processing is bypassed.

User response

Correct the error and rerun the job, or continue with the execution of FABADA3, FABADA4 and FABADA5 as desired.

FABA0141W   NO SDEP SEGMENTS FOUND - SDEP PROCESSING BYPASSED FOR AREA: *areaname*

Explanation

"SDEP" keyword was specified on the control statement. However program FABADA1 determined that the SDEP part associated with the area *areaname* was empty. This condition is detected when the "logical beginning" and the "logical end" contained in DMAC have the same value, or if their values differ by 4.

System action

FABADA1 sets an end-of-job return code of 4, and continues processing. The SDEP processing is bypassed.

User response

Proceed with the remainder of the job steps in the report generation procedure.

FABA0142E   CI AT RBA: *xxxxxxxx* CONTAINS NEGATIVE SEGMENT/FSE LENGTH - CI BYPASSED

Explanation

Program FABADA1 encountered a segment or an FSE with a negative length during the serial "deblocking" of the CI at RBA *xxxxxxxx*.

System action

FABADA1 bypasses the CI in error, and continues processing.

User response

Correct any errors, and rerun the job. If this situation persists, report it to system operation personnel.

FABA0143E   CI AT RBA: *xxxxxxxx* CONTAINS INVALID SEGMENT/FSE LENGTH - CI BYPASSED

Explanation

Program FABADA1 encountered a segment or an FSE with an invalid length during the serial "deblocking" of the CI at RBA *xxxxxxxx*.

System action

FABADA1 bypasses the CI containing the error and continues processing.

User response

Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA0144I   CI MAP/DUMP FOR CI: *xxxxxxxx* GENERATED. DUMP No. =*nnn*

Explanation

This is an informational message. CI map/CI dump for CI *xxxxxxxx* is generated in a data set specified with the SNAPPIT DD statement. A dump no. *nnn* is assigned.

System action

See the user response section.

User response

For both system action and user response, see the message issued immediately before this message to determine the error that caused generation of the dump.

FABA0145W   CI MAP/DUMP FUNCTION IS IGNORED. REASON CODE *nn*

Explanation

A CI map/CI dump function is ignored for one of the following reasons:

- REASON CODE 01

  SNAPPIT data set was not provided, open failed, or I/O failed.

  The area to which CI is to be dumped is not specified by DARVSAM or DFSUDUMP DD statement.
- REASON CODE 05

  The CI to be dumped is not found in DARVSAM data set or DFSUDUMP data set.
- REASON CODE 06

  The maximum number of 100 CI maps/dumps are already generated.
- REASON CODE 07

  GETMAIN failed to obtain space for internal blocks.
- REASON CODE 08

  All entries in block map table (internal table) have been consumed.
- REASON CODE 09

  An error occurred during printing a CI map.
- REASON CODE 10

  An error occurred during printing a CI dump.

System action

Program FABADA1 bypasses the CI map/CI dump function and continues processing.

User response

Correct errors and rerun the utility if CI map/CI dump is needed.

FABA0146W	SPACE MAP AT RBA: *xxxxxxxx* OFFSET: *yyyy* HAS CONTROL WORD DISCREPANCY

Explanation

The CI corresponding to space map at RBA *xxxxxxxx* OFFSET *yyyy* should be the first allocatable CI; that is, 1 byte from OFFSET *yyy* should be x'80'. But, the space map specifies that the CI is already allocated; that is, 1 byte from OFFSET *yyyy* is x'40'.

System action

Program FABADA1 sets an end-of-job return code of 4 and continues processing.

User response

Correct the error, and rerun the job.

FABA0147I	IOVF CONTROL WORD DISCREPANCY FOUND - SPACE MAP CI RBA: *aaaaaaaa* CONTROL WORD: *bbbbbbbb* OFFSET: *cccc* - IOVF DATA CI RBA: *dddddddd* CONTROL WORD: *eeeeeeee*

Explanation

The first byte of a control word value *bbbbbbbb* in the space map CI at RBA *aaaaaaaa* OFFSET *cccc* is different from that of *eeeeeeee* in IOVF DATA CI at RBA: *dddddddd*.

System action

Program FABADA1 continues processing.

User response

Confirm the status of the first byte of control word value (DBLKBTUI).

- If DBLKBTUI in the space map CI at RBA *aaaaaaaa* OFFSET *cccc* is X'80', and DBLKBTUI in the IOVF DATA CI at RBA *dddddddd* is X'00', no problem exists.
- If DBLKBTUI in the space map CI at RBA *aaaaaaaa* OFFSET *cccc* is X'80', and DBLKBTUI in the IOVF DATA CI at RBA *dddddddd* is not X'40', IMS can insert segments into the IOVF DATA CI. In this case, check them for data integrity.
- It is recommended that you check them for data integrity.

FABA0148W	CI AT RBA: *xxxxxxxx* HAS UOW# DISCREPANCY - IOVF SPACE MAP: *yyyyyyyy* and IOVF IX: *zzzzzzzz*

Explanation

A UOW number *yyyyyyyy* in the space map that corresponds to the CI at RBA *xxxxxxxx* and the number *zzzzzzzz* in the IOVF CI prefix at RBA *xxx xxxxx* are different.

System action

Program FABADA1 sets an end-of-job return code of 4 and continues processing using the value of the space map.

User response

Correct the error and rerun the job.

FABA0149W   **INPUT DATA SET IS [ONLINE IMAGE COPY | CONCURRENT IMAGE COPY]**

Explanation

Program FABADA1 found that the input data set is an online image copy or a concurrent image copy data set. Pointer errors are expected with the input data set.

System action

FABADA1 sets an end-of-job return code of 4, and continues processing.

User response

If any pointer errors are reported, run the job using a batch image copy data set.

FABA0150I   **INPUT DATA SET IS IMAGECOPY 2. DUMPED AREA DATA SET NAME:** *dsname*

Explanation

This message is generated when the data set created by the Image Copy 2 Utility is defined on DFSUDUMP DD.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0151W   **SDEP NOT DEFINED IN DATABASE - SDEP PROCESSING BYPASSED FOR DBDNAME:** *dbdname*

Explanation

The "SDEP" keyword was specified, although the SDEP segment was not specified in the database.

System action

Program FABADA1 sets an end-of-job return code of 4, and continues processing.

User response

None.

FABA0152I   **AREA NOT REGISTERED IN DBRC FOR AREA** *areaname*

Explanation

Program FABADA1 found that the specified area was not registered in DBRC.

System action

FABADA1 continues processing.

User response

None. This message is informational.

FABA0153W   **AREA** *areaname* **STATUS CHANGED IN DBRC DURING POINTER CHECKER RUN**
- UPDATE/EXCLUSIVE ACCESS INTENT IMS SYSTEM STARTED
- CHANGED TO RECOVERY NEEDED
- CHANGED TO NON-REGISTERED AREA
- EEQE CREATED/DELETED
- ADS CHANGED TO UNAVAILABLE
- ADS DELETED FROM DBRC

Explanation

Program FABADA1 found that the status of the area specified was changed during the DEDB Pointer Checker run.

System action

FABADA1 sets an end-of-job return code of 4, and continues processing.

User response

If any pointer errors are detected, they may be caused by updates being made to the IMS system during DEDB Pointer Checker processing. Use the /STOP AREA or /DBR AREA command to stop updates, or change access to the area to 'read.' If any pointer error occurred, rerun the job.

FABA0154I   **DBRC=Y IS SPECIFIED**

Explanation

DBRC=Y is specified in the EXEC parameter of the FABADA1 JCL. Program FABADA1 will establish DBRC interface and obtain area information from DBRC.

System action

FABADA1 continues processing.

User response

None. This message is informational.

FABA0155I   **DBRC=N IS SPECIFIED - EEQE DETECTION NOT PERFORMED**

Explanation

DBRC=N is specified in the EXEC parameter of the FABADA1 JCL. Program FABADA1 does not establish a DBRC interface and does not check the existence of EEQEs.

System action

FABADA1 continues processing.

User response

If the area is registered in DBRC, it is recommended that you print the DBRC RECON list and to ensure that there are no EEQEs registered in DBRC for the area.

FABA0156I   **DBRC=Y IS IGNORED - KEYWORD VSAM AND/OR AREA NAME NOT SPECIFIED**

Explanation

DBRC=Y is specified in the EXEC parameter of the FABADA1 JCL. But as the keyword VSAM and/or AREA= is not specified on the control statement, program FABADA1 ignores DBRC=Y. FABADA1 does not establish DBRC interface and does not check the existence of EEQEs.

System action

FABADA1 continues processing.

User response

If the area is registered in DBRC, it is recommended that you print DBRC RECON list and to make sure that there are no EEQEs registered in DBRC for the area.

FABA0157E   **SEGMENT AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS A PTF OR AN SPTF PTR THAT POINTS TO ITS OWN SEGMENT**

Explanation

Program FABADA1 encountered an IMS pointer chain discrepancy. The segment at RBA *xxxxxxxx* is pointed to by its own PTF pointer.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11. Correct the error and rerun the job, or continue running FABADA3, FABADA4, and FABADA5.

FABA0158E   **SEGMENT AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS INVALID SDEP CYCLE COUNT VALUE**

Explanation

The SDEP cycle count value contained in the segment at RBA *xxxxxxxx* is incorrect.

System action

Program FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11. Correct the error and rerun the job, or continue with the run of FABADA3, FABADA4, and FABADA5 as desired.

FABA0159E   **INVALID VALUE IN "SPCF"/"SPTF" - SDEP CHECKSUM VALIDATION PROCESSING BYPASSED FOR AREA: *areaname***

Explanation

Program FABADA1 encountered an error while validating the "SPCF" and "SPTF" with the specified area.

This message could have been issued for an error free area because of one of the following conditions:

- A root segment, which has SDEP segments between the logical begin and the logical end of the SDEP part, has been deleted.
- IOVF part has been extended by the Online Area Extender (OAE) of IMS Fast Path Online Tools of IMS HP Fast Path Utilities.

System action

Program FABADA1 stops SDEP checksum validation processing, sets an end-of-job return code of 99, and continues processing.

User response

Correct the error and rerun the job, or continue with the run of FABADA3, FABADA4, and FABADA5 as desired.

If the area is error free, you can suppress the message by reorganizing the area by using FABCUR1 with the SDEP=LOGICAL option and FABCUR3.

FABA0160E	INVALID IMSID IN SDEP CI AT RBA: *xxxxxxxx* - CI BYPASSED

Explanation

Program FABADA1 encountered an incorrect IMSID in SDEP CI prefix.

System action

FABADA1 increments the internal error counter, bypasses the CI specified, and continues processing. If the value of the internal error counter exceeds 10, FABADA1 ends with an abend code of 3529.

User response

Correct the errors, and rerun the job.

FABA0161I	CI FORMATTED BY CREATE UTILITY ENCOUNTERED, CI WAS SKIPPED RBA OF CI: *nnnnnnnn*

Explanation

Program FABADA1 encountered a control interval containing dummy data formatted by the DEDB Area Data Create Utility. The dummy data CI specified by the subsystem ID "CREATFMT" in the subsystem name stored in the SDEP CI prefix was formatted during the copy phase of the DEDB Area Data Set Create Utility. This CI implies the pre-allocated one and will be overlaid by the SDEP CI image containing real SDEP segments or SDEP pre-allocated dummy CI image.

System action

This message is informational. Processing continues, but the "CREATFMT" dummy CI is skipped on this run.

User response

None. This message is informational.

FABA0162I	RETURN CODE USER EXIT FABARCEX IS CALLED

Explanation

Return code user exit routine FABARCEX is called.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0163I	RETURN CODE USER EXIT FABARCEX COMPLETED. ORIGINAL RC = *xxx*, RETURNED RC = *xxx*

Explanation

Return code user exit routine FABARCEX has completed. The original return code might be changed to the returned return code.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0164E	AREA MAY HAVE ORPHAN SDEP SEGMENTS

Explanation

The number of 4,P records is less than the number of 4,S records. This message is reported only when the return code exit FABARCEX is provided.

System action

Program FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

For more details, see "Restrictions of DEDB Pointer Checker" on page 13.

FABA0165W	THE NUMBER OF MESSAGES IS BEYOND THE LIMIT. EXCESS MESSAGES ARE NOT PUT IN THE MESSAGE TABLE

Explanation

The number of output messages is beyond the limit. Excess messages were not put in the message and

return code table that is used by return code user exit FABARCEX.

System action

Program FABADA1 sets an end-of-job return code of 04, and continues processing.

User response

None.

FABA0166I   **RETURN CODE USER EXIT FABARCEX IS NOT CALLED BECAUSE OF A MESSAGE TABLE GETMAIN FAILURE**

Explanation

Return code user exit routine FABARCEX is not called because of the GETMAIN failure of the message and return code table.

System action

Program FABADA1 continues processing.

User response

None. This message is informational.

FABA0167E   **SPACE MAP AT RBA:** *xxxxxxxx* **OFFSET:** *yyyy* **HAS INCORRECT CONTROL WORD VALUE:** *zzzzzzzz*

Explanation

The first byte of a control word value *zzzzzzzz* in the CI at RBA *xxxxxxxx* OFFSET *yyyy* is incorrect. Expected data value is X'40', which means 'Allocated', or X'80', which means 'Free'.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

Correct the errors, and rerun the job.

FABA0168E   **SEGMENT AT RBA:** *xxxxxxxx* **(SEGCD:** *yy***) HAS AN INVALID LENGTH OF SEGMENT**

Explanation

Program FABADA1 encountered a segment with an incorrect length during the serial "deblocking" of the

CI. The length value is out of the range between the minimum length and the maximum length defined in DBD. If it is a compressed segment, it is out of the range and is in between 3 and the maximum length added by 10.

System action

FABADA1 sets an end-of-job return code of 99, and continues processing.

User response

Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA0169E   **SDEP CI FREESPACE OFFSET DOES NOT POINT TO AN ADDRESS AFTER THE LAST SEGMENT IN CI. - CI BYPASSED - RBA:** *xxxxxxxx* **- SDEP CI FREESPACE OFFSET:** *yyyy* **- OFFSET OF THE BYTE AFTER THE LAST SEGMENT:** *zzzz*

Explanation

Program FABADA1 encountered an invalid free space offset value in the SDEP CI. The free space offset of the SDEP CI should point to an address after the last valid SDEP segment.

System action

FABADA1 bypasses the CI in error, and continues processing.

User response

Correct any errors, and rerun the job. If this situation persists, report it to system operation personnel.

FABA0170W   **TIMESTAMP IN IMAGE COPY HEADER IS INCONSISTENT WITH TIMESTAMP RETURNED FROM EXIT ROUTINE:***comprtn* **EXIT ASSEMBLE DATE TIMESTAMP : MM/DD/YYHH:MM TIMESTAMP IN IMAGE COPY HDR : MM/DD/YYHH:MM**

Explanation

The timestamp in the image copy header record is not equal to the assemble time of *comprtn*.

System action

The DEDB Pointer Checker (FABADA1) sets an end-of-job return code of 4, and continues the processing.

User response

Make sure that the correct IMS HP Image Copy library is concatenated to the JOBLIB or STEPLIB library and rerun the job.

FABA0177I   **- TBLENTRY=0 IS SPECIFIED - SDEP TABLE IS NOT USED FOR SDEP VALIDATION**

Explanation

TBLENTRY=0 is specified in the SYSIN DD control statement of the FABADA1 JCL. Program FABADA1 does not use the SDEP table for SDEP validation process. All pointer information records and the segment information records for all logically deleted SDEP segments are created in the DAR3CSO data set so that the information of the deleted segment can be resolved in program FABADA3.

System action

FABADA1 continues processing.

User response

None. This message is informational.

FABA0301I   **PROCESSING STARTED FOR AREA: _zzzzz_**

Explanation

This message is issued when program FABADA3 starts processing of the specified DEDB area number.

System action

FABADA3 continues processing.

User response

None. This message is informational.

FABA0302I   **FABADA3 ENDED NORMALLY - NO POINTER ERRORS DETECTED**

Explanation

This message is generated when the program completes successfully.

System action

Program FABADA3 ends normally with a completion code of zero.

User response

None. This message is informational.

FABA0302E   **FABADA3 ENDED WITH ERRORS [- POINTER ERRORS DETECTED]**

Explanation

Program FABADA3 processed all segment and pointer records, but errors or IMS pointer chain discrepancies were encountered.

System action

FABADA3 ends normally with a completion code of 99.

User response

See the other messages generated for further information. If the "POINTER ERRORS DETECTED" message is present, see "DEDB integrity verification" on page 11.

FABA0303E   **[POINTER ERRORS WERE DETECTED | IMMEDIATE ATTENTION REQUIRED]**

Explanation

These two messages are generated as WTO messages at the end of FABADA3 job step if program FABADA3 detected any pointer errors during processing.

System action

FABADA3 ends normally with a completion code of 99.

User response

Determine the cause of the error using the other messages generated. See "DEDB integrity verification" on page 11.

FABA0305I   **INPUT/OUTPUT TOTALS AND PROCESSING STATISTICS**

Explanation

This message is issued to display the number of input/output records processed by program FABADA3.

System action

FABADA3 continues processing.

User response

None. This message is informational.

FABA0310E   UNKNOWN KEYWORD (NEAR COLUMN *xx*)

Explanation

Program FABADA3 encountered an invalid keyword while parsing the user-supplied control statement.

System action

FABADA3 ends with an abend code of 3546.

User response

Correct the control statement, and rerun the job.

FABA0311E   "TBLENTRY=" VALUE INVALID

Explanation

Program FABADA3 encountered an invalid "TBLENTRY=" specification while parsing the user-supplied control statement.

System action

FABADA3 ends with an abend code of 3546.

User response

Correct the control statement, and rerun the job.

FABA0312E   "STKENTRY=" VALUE INVALID

Explanation

Program FABADA3 encountered an incorrect "STKENTRY=" specification while parsing the user-supplied control statement.

System action

FABADA3 ends with an abend code of 3546.

User response

Correct the control statement, and rerun the job.

FABA0313E   "STATS=" VALUE INVALID

Explanation

Program FABADA3 encountered an incorrect "STATS=" specification while parsing the user-supplied control statement.

System action

FABADA3 ends with and abend code of 3546.

User response

Correct the control statement, and rerun the job.

FABA0315I   SORT CONTROL CARDS WRITTEN TO DDNAME: *ddname*

Explanation

This message is generated when program FABADA3 has generated and written the required sort control statements to the file associated with the *ddname* (SORTR13 or SORT13R) DD statement.

System action

FABADA3 continues processing.

User response

None. This message is informational.

FABA0321E   SEGMENT DATA MISSING FOR UOW#: *aaaaaaaa*

Explanation

Pointer records were processed for segments in the specified UOW but no segment information records were found.

System action

Program FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

Make sure that the files associated with the DAR13SI and DAR3CSI DD statements are correctly specified, and that the DAR3CSI file has been sorted. Correct any errors, and rerun the job.

FABA0322E   SEG AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS INVALID PCL/SSPTR THAT REFERENCES SEG AT RBA: *zzzzzzzz*

Explanation

Program FABADA3 determined that the segment at RBA *xxxxxxxx* contains an invalid RBA value of *zzzzzzzz* for a PCL or SSPTR. The pointer value *zzzzzzzz* references a nonexistent segment.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

| FABA0323E | SEG AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS PCL/SSPTR THAT MAY POINT TO WRONG SEG TYPE AT RBA: *zzzzzzzz* |

Explanation

Program FABADA3 determined that the PCL or SSPTR contained in the segment at RBA *xxxxxxxx* may be pointing to a wrong segment type at RBA *zzzzzzzz*.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

| FABA0324E | SEGMENT AT RBA: *aaaaaaaa* (SEGCODE: *bb*) POINTED TO BY MULTIPLE SEGMENTS - POINTER IN SEGMENT AT RBA: *cccccccc* (SEGCODE: *dd*) |

Explanation

Program FABADA3 encountered an IMS pointer chain discrepancy. The segment at RBA *aaaaaaaa* is pointed to by multiple PCF/PTF pointers.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

| FABA0325E | SEGMENT AT RBA: *xxxxxxxx* POINTS TO NONEXISTENT SEGMENT (SEGCODE: *bb*) AT RBA: *yyyyyyyy* |

Explanation

Program FABADA3 encountered an invalid IMS pointer. No segment is found at RBA *yyyyyyyy*.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

| FABA0326E | SEGMENT AT RBA: *xxxxxxxx* IS AN ORPHAN |

Explanation

Program FABADA3 encountered a segment in the database that is not pointed to by another segment or a RAP.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

| FABA0327E | SEG AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS PTF/PCF PTR THAT POINTS TO WRONG SEG TYPE AT RBA: *zzzzzzzz* |

Explanation

Program FABADA3 determined that the PTF or the PCF contained in the segment at RBA *xxxxxxxx* is pointing to a wrong segment type at RBA *zzzzzzzz*.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

| FABA0328E | SEGMENT AT RBA: *xxxxxxxx* (SEGCODE: *yy*) NOT IN RAP CHAIN - POINTED TO BY SEGMENT AT RBA: *zzzzzzzz* |

Explanation

Program FABADA3 encountered an IMS pointer chain discrepancy.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

FABA0329E	[ROOT | SDEP] SEGMENT AT RBA: *xxxxxxxx* CONTAINS INVALID SDEP POINTER [(CYCLE# REL-BYTE-ADDR)]

Explanation

The SDEP pointer contained in the segment at RBA *xxxxxxxx* points to a nonexistent segment specified by CYCLE# and REL-BYTE-ADDR.

System action

Program FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

FABA0330E	SDEP SEGMENT AT RBA: *xxxxxxxx* IS AN ORPHAN

Explanation

The SDEP segment at RBA *xxxxxxxx* is not pointed to by another SDEP segment or a ROOT segment.

System action

Program FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

See "DEDB integrity verification" on page 11 for further information.

FABA0335I	HIGHEST(INVALID) SDEP POINTER VALUE ENCOUNTERED CYCLE# REL-BYTE-ADDR

Explanation

During processing of the area specified by preceding message FABA0301I, one or more SDEP pointer errors were detected, and the cycle count and the RBA of the highest invalid SDEP pointer are specified by CYCLE# and REL-BYTE-ADDR.

System action

Program FABADA3 continues processing

User response

See "DEDB integrity verification" on page 11 for further information.

FABA0336E	SEGMENT AT RBA: *xxxxxxxx* (SEGCD: *yy*) HAS A PTF OR AN SPTF PTR THAT POINTS TO ITS OWN SEGMENT

Explanation

Program FABADA3 encountered an IMS pointer chain discrepancy. The segment at RBA *xxxxxxxx* is pointed to by its own PTF pointer.

System action

FABADA3 sets an end-of-job return code of 99, and continues processing.

User response

For further information, see "DEDB integrity verification" on page 11.

FABA0341I	RETURN CODE USER EXIT FABARCEX IS NOT CALLED BECAUSE OF A MESSAGE TABLE GETMAIN FAILURE

Explanation

This message is generated when the return code user exit routine FABARCEX is not called due to GETMAIN failure of the message and return code table.

System action

Program FABADA3 continues processing.

User response

None. This message is informational.

FABA0342I	RETURN CODE USER EXIT FABARCEX IS CALLED

Explanation

Return code user exit routine FABARCEX is called.

System action

Program FABADA3 continues processing.

User response

None. This message is informational.

FABA0343I	RETURN CODE USER EXIT FABARCEX COMPLETED. ORIGINAL RC = *xxx*, RETURNED RC = *xxx*

Explanation

Return code user exit routine FABARCEX has completed. The original return code might be changed to the returned return code.

System action

Program FABADA3 continues processing.

User response

None. This message is informational.

FABA0344W	THE NUMBER OF MESSAGES IS BEYOND THE LIMIT. EXCESS MESSAGES ARE NOT PUT IN THE MESSAGE TABLE

Explanation

The number of output messages is beyond the limit. Excess messages were not put in the message and return code table that is used by return code user exit FABARCEX.

System action

Program FABADA3 sets an end-of-job return code of 04, and continues processing.

User response

None.

FABA0345I	- LOGICALLY DELETED ORPHAN SDEP SEGMENT DETECTED

Explanation

Program FABADA3 detected one or more logically deleted orphan SDEP segments. Even if program FABADA1 ended with message FABA0138E, program

FABADA3 will end with RC=0 if no other kinds of errors are detected.

System action

FABADA3 ignores such logically deleted orphan SDEP segments assuming that such logically deleted orphan SDEP segment exists due to the deletion of parent root segments, and continues processing.

User response

None. This message is informational.

FABA0400I	FABADA4 ENDED NORMALLY

Explanation

This message is generated when the program completes successfully.

System action

Program FABADA4 ends normally.

User response

None. This message is informational.

FABA0400W	FABADA4 ENDED WITH WARNINGS

Explanation

One or more warning conditions were encountered during the execution of FABADA4.

System action

Program FABADA4 ends normally with a completion code of four.

User response

Determine the cause of the warning using the other messages generated.

FABA0402I	INPUT DATA SET IS IMAGECOPY 2. DUMPED AREA DATA SET NAME: *dsname*

Explanation

This message is generated when the data set created by the Image Copy 2 Utility is defined on DFSUDUMP DD.

System action

Program FABADA4 continues processing.

User response

None. This message is informational.

FABA0410I RBA FROM DAR4ERRS DATA SET WITH FABA03xxE MESSAGE (cccccccc)

Explanation

The target segment on this report line was read by program FABADA4 from the DAR4ERRS data set. The RBA was shown in the message specified by *cccccccc* in FABADA3.

System action

Processing continues.

User response

None. This message is informational.

FABA0411I RBA FROM DAR4USER DATA SET

Explanation

The target segment on this report line was read by program FABADA4 from the DAR4USER data set.

System action

Processing continues.

User response

None. This message is informational.

FABA0412I ADDRESS FOUND IN xxxxxxx DATA SET

Explanation

The address of this report line was requested on your program FABADA4 input (see message FABA0410I or FABA0411I) and was found in the DAR13SI or DAR3CSI DATA SET.

System action

Processing continues.

User response

None. This message is informational.

FABA0413I SEGMENT POINTS TO ABOVE TARGET RBA

Explanation

The segment on this report line contains a pointer that points to the target RBA requested on your program FABADA4 input (see message FABA0410I or FABA0411I).

System action

Processing continues.

User response

None. This message is informational.

FABA0414I NO MORE RECORDS FOR SPECIFIED RBA

Explanation

All records on DAR13SI data set and DAR3CSI data set for this RBA (and data set) have been processed.

System action

Processing continues.

User response

None. This message is informational.

FABA0415I END OF FILE ON [DAR4ERRS | DAR4USER | DAR4ERRS AND DAR4USER] DATA SET

Explanation

All records on the DAR4ERRS data set, DAR4USER data set, or DAR4ERRS and DAR4USER data sets have been processed by module FABADA4.

System action

Processing continues.

User response

None. This message is informational.

FABA0416I END OF FILE ON DAR13SI AND DAR3CSI DATA SETS

Explanation

All records on the DAR13SI and DAR3CSI data set have been processed by module FABADA4.

System action

Processing continues.

User response

None. This message is informational.

FABA0445W **CI MAP/DUMP FUNCTION IS IGNORED. REASON CODE** *nn*

Explanation

A CI map/CI dump function is ignored for the following reasons:

- REASON CODE 01

  SNAPPIT data set was not provided, open failed, or I/O failed.

- REASON CODE 02

  ACBLIB data set was not provided; an unsupported level of ACBLIB data set was provided; the open failed; I/O failed; or necessary DMB was not found.

- REASON CODE 03

  DARVSAM data set or DFSUDUMP data set was not provided, open failed, or I/O failed.

- REASON CODE 04

  The area which CI need to be dumped is not specified by DARVSAM data set, or the area is not found in the DFSUDUMP data set.

- REASON CODE 05

  The CI need to be dumped is not found in DARVSAM data set or DFSUDUMP data set.

- REASON CODE 06

  100 CI maps/dumps are already generated.

- REASON CODE 07

  GETMAIN failed to obtain space for internal blocks.

- REASON CODE 08

  All entries in block map table (internal table) have been consumed.

- REASON CODE 09

  An error occurred during printing a CI map.

- REASON CODE 10

  An error occurred during printing a CI dump.

System action

System continues but the CI map/CI dump function no longer work.

User response

Correct errors, and rerun the utility if CI map/CI dump is needed.

FABA0470W **TIMESTAMP IN IMAGE COPY HEADER IS INCONSISTENT WITH TIMESTAMP RETURNED FROM EXIT ROUTINE:***comprtn* **EXIT ASSEMBLE DATE TIMESTAMP : MM/DD/YYHH:MM TIMESTAMP IN IMAGE COPY HDR : MM/DD/YYHH:MM**

Explanation

The timestamp in the image copy header record is not equal to the assemble time of *comprtn*.

System action

The DEDB Pointer Checker (FABADA4) sets an end-of-job return code of 4, and continues the processing.

User response

Make sure that the correct IMS HP Image Copy library is concatenated to the JOBLIB or STEPLIB library and rerun the job.

FABA0500I **FABADA5 ENDED NORMALLY**

Explanation

This message is generated when program FABADA5 completes successfully.

System action

FABADA5 ends normally with an end-of-job return code of 0.

User response

None. This message is informational.

FABA0500I **FABADA5 ENDED WITH WARNINGS - THRESHOLD CONTROL CARD ERRORS - THRESHOLD VIOLATIONS DETECTED - SDEP PROCESSING BYPASSED IN THE PREVIOUS STEP**

Explanation

Warning conditions were detected during the execution. This message is generated when program FABADA5 ends with a warning.

System action

FABADA5 ends with an end-of-job return code of a value other than zero.

User response

For additional descriptions of the warning conditions, see the other messages that were generated.

FABA0501I   **PROCESSING STARTED FOR AREA:** *zzzzz* **(AREANAME:** *ddname*)

Explanation

This message is generated when program FABADA5 starts processing of the area number specified. *zzzzz* is the area number.

System action

FABADA5 continues processing.

User response

None. This message is informational.

FABA0504I   **NUMBER OF UOW'S TO BE REORG-ED:** *xx,xxx*

Explanation

This message is issued when program FABADA5 has generated and written the REORG Utility control statements to the file associated with the REORGCTL DD statement. No control statements are generated by FABADA5, if *xx,xxx* is zero.

System action

FABADA5 continues processing.

User response

None. This message is informational.

FABA0505W   **DATA INDICATES "TYPRUN=RPT/ MODEL" - FILE:** *ddname* **MISSING**

Explanation

Program FABADA5 expected a file associated with the *ddname* (DAR13I or DAR13R) DD statement but the file was not present.

System action

FABADA5 sets an end-of-job return code of 4, and continues processing.

User response

Make sure that this situation is not an error condition. If it is an error condition, make sure that the *ddname* (DAR13I or DAR13R) DD statement is present and that it properly specifies the correct file. Rerun the job.

FABA0510E   **UNKNOWN KEYWORD (NEAR COLUMN** *xx*)

Explanation

Program FABADA5 encountered an invalid keyword while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the control statement, and rerun the job.

FABA0513E   **"RBASEFS=" VALUE INVALID**

Explanation

Program FABADA5 encountered an invalid "RBASEFS=" specification while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the control statement, and rerun the job.

FABA0514E   **"RDOVFFS=" VALUE INVALID**

Explanation

Program FABADA5 encountered an invalid "RDOVFFS=" specification while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the control statement, and rerun the job.

FABA0515E   **"UDOVFFS=" VALUE INVALID**

Explanation

Program FABADA5 encountered an invalid "UDOVFFS=" specification while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the control statement, and rerun the job.

FABA0516E   **INVALID THRESHOLD PARAMETER VALUE(S) - THRESHOLD ANALYSIS BYPASSED**

Explanation

Program FABADA5 encountered an invalid specification for the threshold parameter values in the file associated with the THRCTL DD statement.

System action

FABADA5 sets an end-of-job return code of 98, and continues processing.

User response

Make sure that the threshold parameter values are correctly specified in the file associated with the THRCTL DD statement. Make sure that the values specified do not contain any character other than numeric characters and blanks.

FABA0517E   **"SYNTBL=" VALUE INVALID**

Explanation

Program FABADA5 encountered an invalid "SYNTBL=" specification while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the control statement, and rerun the job.

FABA0520W   **COUNT OF UNUSED IOVF CI'S (DMACOCNT) INCORRECT IN DMAC**

Explanation

The number of the unused IOVF CIs calculated by program FABADA5 was not equal to the value contained in the "DMACOCNT" field of the DMAC.

System action

FABADA5 sets an end-of-job return code of 4, and continues processing.

User response

Correct the error in the DMAC, and rerun the job, if desired.

FABA0521E   **[THRESHOLD CONTROL CARD ERRORS | THRESHOLD ANALYSIS BY-PASSED]**

Explanation

This message is issued together with a FABA0516E message.

System action

Program FABADA5 sets an end-of-job return code of 98, and continues processing.

User response

See message FABA0516E.

FABA0522W   **THRESHOLD VIOLATIONS DETECTED**

Explanation

This message is issued when *FABA0599W* message was previously issued.

System action

Program FABADA5 sets an end-of-job return code of 99, and continues processing.

User response

None.

FABA0523E   **"MAXLREC=" VALUE INVALID**

Explanation

Program FABADA5 encountered an incorrect "MAXLREC=" specification while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the control statement, and rerun the job.

FABA0525W   **SDEP INFORMATION IS NOT REPORTED**

Explanation

Program FABADA5 has found that SDEP information was not obtained in the previous step. See FABA0140E message that was issued by program FABADA1.

System action

Program FABADA5 sets an end-of-job return code of 97, and continues processing. The SDEP information is not reported.

User response

None.

FABA0530I   **REPORTED TO HISTORY2 FILE BY UOW RANGE. START UOW=*nnnnnnn* STOP UOW=*nnnnnnn***

Explanation

This message is generated when program FABADA5 reported successfully the records that were totaled by the UOW range to the History2 file by a UOWCTL control card.

System action

FABADA5 continues processing.

User response

None. This message is informational.

FABA0532E   **UOWCTL FILE ERRORS. CARD: "*xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx x*"**

Explanation

Program FABADA5 encountered one or more errors in one of the UOWCTL control cards.

System action

FABADA5 ends with an abend code of 3563.

User response

Correct the UOWCTL control card, and rerun the job.

FABA0533W   **UOWCTL CONTROL CARDS FOR AREAS THAT ARE NOT PROCESSED WERE SPECIFIED. THESE CONTROL CARDS WERE IGNORED.**

Explanation

One or more UOWCTL control cards, of which area are not processed by FABADA5, were specified. These control cards were ignored.

System action

Program FABADA5 sets an end-of-job return code of 4, and continues processing.

User response

Make sure that this situation is not an error condition. If it is an error condition, specify the correct file, and rerun the job.

FABA0534W   **SDEP INFORMATION WAS NOT OBTAINED BECAUSE THE SDEP PROCESSING WAS BYPASSED IN THE PREVIOUS STEP**

Explanation

Program FABADA5 has found that SDEP information was not obtained in the previous step. See FABA0140E message that was issued by program FABADA1.

System action

Program FABADA5 sets an end-of-job return code of 97 and continues processing. The SDEP information is not reported.

User response

See FABA0140E message.

FABA0599W   **THRESHOLD EXCEEDED: *descriptive message* THRESHOLD: *value1* ACTUAL: *value2***

Explanation

Program FABADA5 encountered a situation where the user-supplied threshold value has been exceeded. The *descriptive message* in the text describes the condition

and the associated values. The *descriptive message* may be one of the following:

```
FS RELATIONSHIP ( RAA  > xx% AND DOVF < yy% )
FS RELATIONSHIP ( RAA  > xx% AND IOVF < yy% )
FS RELATIONSHIP ( DOVF > xx% AND IOVF < yy% )
% FS RAA   ( < xxx% )
% FS DOVF  ( < xxx% )
% FS IOVF  ( < xxx% )
RAP CI OVFL USAGE ( > xx% )
UOW'S USING DOVF  ( > xx% )
UOW'S USING IOVF  ( > xx% )
% AVAILABLE IOVF CI'S ( < xx% )
AVG. IOVF CI'S USED ( > xx.x )
MAX. IOVF CI'S USED ( > xx )
AVG. SYN LENGTH   ( > xx.x )
MAX. SYN LENGTH   ( > xx )
AVG. ROOT I/O ( > xx.x )
MAX. ROOT I/O ( > xx )
AVG. RECORD I/O ( > xx.x )
MAX. RECORD I/O ( > xx )
% RECS. USING IOVF ( > xx% )
SYN LENGTH ( xx% > LENGTH yy ).
```

System action

FABADA5 sets an end-of-job return code of 99 and continues processing.

User response

Check the threshold exceptions that were reported by this message and other FABA0599W messages. If you determine that any of those exceptions must be addressed, follow the guideline in the topic "Analyzing and tuning IMS data entry databases" in the *IMS Solution Packs IMS Database Space Tuning Guide*.

FABA0800I   FABADA8 ENDED NORMALLY NUMBER OF INPUT RECORDS READ: *xxx,xxx,xxx*

Explanation

This message is generated when the program completes successfully.

System action

Program FABADA8 ends normally.

User response

None. This message is informational.

FABA0900I   FABADA9 ENDED NORMALLY

Explanation

This message is generated when program FABADA9 completes successfully.

System action

FABADA9 ends normally with an end-of-job return code of 0.

User response

None. This message is informational.

FABA0900W   FABADA9 ENDED WITH WARNINGS

Explanation

One or more warning conditions were encountered during the execution of program FABADA9.

System action

FABADA9 ends normally with an end-of-job return code of 4.

User response

For a description of the warning conditions, see the other messages generated.

FABA0900E   FABADA9 ENDED WITH ERRORS

Explanation

One or more error conditions were encountered during the execution of program FABADA9.

System action

FABADA9 ends normally with an end-of-job return code of 99.

User response

See the other messages generated for the description of the errors. Correct the errors, and rerun the job.

FABA0910I   UPDATE PHASE BYPASSED (NO HISTORY FILE UPDATES PERFORMED) - SDEP DATA FILE NOT PROVIDED - SDEP DATA FILE IS EMPTY

Explanation

Program FABADA9 bypassed the SDEP History file update since the file associated with the DADARI DD statement was not present or was empty.

System action

FABADA9 bypasses the SDEP History file update, and continues processing.

User response

None. This message is informational.

FABA0911I   UPDATE PHASE ENDED NORMALLY

Explanation

Self-explanatory.

System action

Program FABADA9 continues processing.

User response

None. This message is informational.

FABA0911W   UPDATE PHASE ENDED WITH WARNINGS

Explanation

Self-explanatory.

System action

Program FABADA9 sets an end-of-job return code of 4, and continues processing.

User response

None.

FABA0920W   SDEP PART UTILIZATION EXCEEDS THRESHOLD FOR AREA *areaname* **IN DATABASE** *dbdname* **ACTUAL UTILIZATION AS AT** *yy.ddd: zzz%* **THRESHOLD:** *xxx%*

Explanation

This message is generated when the value specified in the "THR=" parameter has been exceeded.

System action

Program FABADA9 generates the SDEP space utilization graph of the area specified. The program sets an end-of-job return code of 99, and continues processing.

User response

None.

FABA0921W   SDEP DATA RECORD IGNORED FOR AREA *areaname* **IN DATABASE** *dbdname* **(REC#:** *x,xxx***)**

- DATA DATE PREVIOUS TO HISTORY FILE "LAST CHANGED" DATE
- DATE FOUND IN SDEP DATA RECORD: *aa.bbb*
- HISTORY FILE "LAST CHANGED" DATE: *cc.ddd*

Explanation

Program FABADA9 encountered a date discrepancy while attempting to update the History file.

System action

FABADA9 sets an end-of-job return code of 4, and continues processing. The History file updating is bypassed.

User response

Make sure that the files associated with the DADARI and DASDHIO DD statements are correctly specified.

FABA0930I   REPORT PHASE BYPASSED (NO UTILIZATION REPORTS WILL BE GENERATED)
- CONTROL CARD FILE NOT PROVIDED
- CONTROL CARD FILE IS EMPTY

Explanation

Program FABADA9 bypassed the SDEP Utilization report since the file associated with the SYSIN DD statement was not provided or empty.

System action

FABADA9 continues processing. The SDEP Utilization report is bypassed.

User response

None. This message is informational.

FABA0931I   REPORT PHASE ENDED NORMALLY

Explanation

Self-explanatory.

System action

Program FABADA9 continues processing.

User response

None. This message is informational.

**FABA0931W   REPORT PHASE ENDED WITH
                WARNINGS**

Explanation

Self-explanatory.

System action

Program FABADA9 continues processing.

User response

Correct the control statements, and rerun the job in
'report only' mode.

**FABA0932W   NO INFORMATION IN HISTORY
                FILE FOR [AREA *areaname* IN]
                DATABASE *dbdname***

Explanation

The file associated with the DASDHIO DD statement
did not contain any information about the specified
area, or the file was empty.

System action

Program FABADA9 sets an end-of-job return code of 4
and continues processing.

User response

None.

FABA0950I   CARD *xxx*: CONTROL STATEMENT

Explanation

CARD *xxx* shows the control statement specified by
user.

System action

Processing continues.

User response

None. This message is informational.

FABA0950E   CARD *xxx*: CONTROL STATEMENT

Explanation

Program FABADA9 encountered one or more errors in
the user-supplied control statement. *xxx* shows the

sequence of the control statement that contained the
errors.

System action

FABADA9 generates one or more messages to
describe the errors and continues processing.

User response

For a description of the errors, see the other messages
generated.

**FABA0951E   ERROR DETECTED - CONTROL
                CARD IGNORED**

Explanation

Program FABADA9 encountered one or more errors in
one of the user-supplied control statements.

System action

FABADA9 discards the control statement, and
continues processing.

User response

See the other messages generated for the description
of the errors. Correct the errors in the control
statement, and rerun the job in 'report only' mode.

**FABA0952E   CONTROL CARD RECORD IS
                BLANK**

Explanation

Self-explanatory.

System action

Program FABADA9 discards the blank control
statement, and continues processing.

User response

Self evident.

**FABA0953E   UNKNOWN KEYWORD NEAR
                COLUMN *xx***

Explanation

Program FABADA9 encountered an invalid keyword
while searching for a parameter on the user-supplied
control statement.

System action

FABADA9 discards the control statement and continues processing.

User response

Correct the control statement, and rerun the job in 'report only' mode.

FABA0954E DBDNAME VALUE MISSING/ INVALID

Explanation

Self-explanatory.

System action

Program FABADA9 discards the control statement and continues processing.

User response

Correct the control statement, and rerun the job in 'report only' mode.

FABA0955E AREA DDNAME VALUE MISSING/ INVALID

Explanation

Self-explanatory.

System action

Program FABADA9 discards the control statement and continues processing.

User response

Correct the control statement, and rerun the job in 'report only' mode.

FABA0956E NO VALID DBDNAME SPECIFICATION ON CONTROL CARD

Explanation

Self-explanatory.

System action

Program FABADA9 discards the control statement and continues processing.

User response

Correct the control statement, and rerun the job in 'report only' mode.

FABA0960E INVALID DATA IN "PARM" FIELD - IGNORED

Explanation

Program FABADA9 encountered an invalid "PARM" specification on the EXEC statement.

System action

FABADA9 assumes the default value for SDEP utilization threshold (85%) and continues processing.

User response

None.

FABA3501E "DEVTYPE" FAILED FOR DDNAME *ddname*

Explanation

Program FABADA1 issued a DEVTYPE macro to get information about the device associated with the specified ddname. The return code specified that the attempt to do so was unsuccessful.

System action

FABADA1 ends with an abend code of 3501.

User response

Make sure that a DD statement is present for *ddname* and that it properly identifies the correct data set. Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA3502E MISSING CONTROL CARD

Explanation

The data set associated with the SYSIN DD statement was empty or was a DUMMY data set.

System action

Program FABADA1 ends with an abend code of 3502.

User response

Make sure that the data set associated with the SYSIN DD statement contains a control statement, and rerun the job.

FABA3503E   CRITICAL CONTROL CARD ERROR ENCOUNTERED

Explanation

Program FABADA1 encountered one or more errors while parsing the user-supplied control statement.

System action

FABADA1 ends with an abend code of 3503.

User response

See the FABA0110E messages for the description of the error. Correct the error, and rerun the job.

FABA3504E   PROCESSING FAILED FOR *xxxxxx* MEMBER
- *xxxxxx* MEMBER IS NOT A DEDB DMB
- MEMBER NAME NOT EQUAL DEFINED DATABASE NAME
- *ddname* DD STATEMENT NOT FOUND
- MEMBER NOT FOUND IN ACBLIB
- INSUFFICIENT STORAGE
- INVALID PARAMETER LIST IN CALL TO *nnnnnnn*
- IMS LEVEL OF ACB MEMBER *acbname* NOT SUPPORTED
- ACB MEMBER *acbname* HAS NO DATA

Explanation

To obtain DMB information of the database that is being processed, program FABADA1 called either the FABAGDD program to obtain DMB information from the ACB library or the FABAGDD2 program to obtain DMB information from the IMS directory. The return code specified that the attempt to do so was unsuccessful.

System action

FABADA1 ends with an abend code of 3504.

User response

Ensure that the files associated with the DD statements DFSUDUMP (or DARVSAM) and ACBLIB or the IMS directory are correctly specified. Also, ensure that the ACBGEN and DBDGEN or IMS catalog population were correctly performed for the database being analyzed. Correct any errors, and rerun the job.

If "- INVALID PARAMETER LIST IN CALL TO FABAGDD" is shown, contact IBM Software Support.

FABA3505E   INSUFF. STORAGE FOR: *aaaa* - INCREASE REGION SIZE

Explanation

Program FABADA1 issued a GETMAIN macro to get storage for *aaaa*. The return code specified that the attempt was unsuccessful.

System action

FABADA1 ends with an abend code of 3505.

User response

Increase the region size parameter on the EXEC statement, and rerun the job.

FABA3506E   "TRKCALC" FAILED (RC = *xx*)

Explanation

Program FABADA1 issued a TRKCALC macro to get device information. The return code specified that the attempt was unsuccessful.

System action

FABADA1 ends with an abend code of 3506.

User response

Report this situation to systems operations personnel.

FABA3507E   VSAM ["GENCB" | "MODCB" | "SHOWCB"] ERROR
- REG 15: *xx* REG 0: *yy*

Explanation

Program FABADA1 issued a GENCB, MODCB, or SHOWCB macro to get or modify the data set specified in the DARVSAM DD statement. The return code specified that the attempt was unsuccessful.

System action

FABADA1 ends with an abend code of 3507.

User response

Make sure that the DARVSAM DD statement is correctly specified. Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA3508E   DARVSAM DEVICE INFORMATION UNAVAILABLE

Explanation

The "DARVSAM" DD STATEMENT CONTAINED A "DUMMY" KEYWORD PARAMETER OR AN UNRESOLVED "DDNAME" KEYWORD PARAMETER.

System action

Program FABADA1 ends with an abend code of 3508.

User response

Correct the "DARVSAM" DD statement and/or make sure that the DD statement associated with the "DDNAME" parameter is specified correctly. Rerun the job.

FABA3509E   *macro-name* FAILED FOR DMB MEMBER *member-name* FOR DDNAME *ddname* (RC=*rr*: REASON=*zz*)

Explanation

Program FABADA1 issued the specified macro to access the specified DMB member in the specified data set. The return code and the reason code from z/OS specify that the attempt was unsuccessful.

System action

FABADA1 ends with an abend code of 3509.

User response

Make sure that the DMB member specified is present in the data set and that the data set is correctly identified. Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA351*x*E   "OPEN" FAILED FOR DDNAME *ddname*
- DD STATEMENT NOT FOUND
- RECFM INCORRECT
- LRECL TOO SMALL
- BLKSIZE TOO SMALL
- FAILED BY OPERATING SYSTEM

Explanation

Program FABADA1 issued an OPEN macro for the file associated with the specified DD statement. The return code specified that the attempt was unsuccessful.

The probable causes of this problem are:

- The DD statement was missing.
- The DSN=NULLFILE or DUMMY operand was specified on DD card.
- A DCB parameter was not specified on the DD statement for a non-label ('LABEL=(,NL)') input tape.
- The specified data set record was invalid.

System action

FABADA1 ends with an abend code. The following describes the error message number and the abend code associated with each file:

Message No.	Abend code	DDNAME
FABA3511E	3511	SYSPRINT
FABA3512E	3512	SYSIN
FABA3512E	3512	SYSIN
FABA3513E	3513	DAR12O
FABA3514E	3514	SORT12
FABA3515E	3515	DAR13SO
FABA3516E	3516	DAR3CSO
FABA3517E	3517	SORT3CS
FABA3518E	3518	DFSUDUMP

User response

Make sure that a DD statement is present for the ddname specified, and that it properly identifies the data set. Correct any errors, and rerun the job.

FABA3519E   "OPEN" FAILED FOR VSAM DATA SET
- VSAM ERROR DATA: RETURN CODE: *aaa* (*bb*) ACB ERROR: *ccc* (*dd*)

Explanation

Because VSAM=YES was found either in the FABADA1 SYSIN data set or the site default table, program FABADA1 issued the OPEN macro for the VSAM data set that is specified by the DARVSAM DD statement. The return code indicates that the attempt was unsuccessful. The return code and ACB error code values are shown in both decimal (*aaa*, *ccc*) and hexadecimal (*bb*, *dd*) formats.

System action

FABADA1 ends with an abend code of 3519.

User response

Ensure that a DARVSAM DD statement is present and that it properly identifies the correct data set. If you want to process image copy data sets, remove the VSAM=YES entry from the FABADA1 SYSIN data set and the site default table, and then rerun the job. If this situation persists, report it to systems operations personnel.

FABA3520E   **MULTIPLE I/O ERRORS ATTEMPTING VSAM "READ"**

Explanation

Program FABADA1 encountered more than 10 CIs in error while attempting to read the data set specified by the DARVSAM DD statement.

System action

FABADA1 ends with an abend code of 3520.

User response

See the FABA0120E messages to find the CIs in error. Report this situation to systems operations personnel.

FABA3521E   **UNEXPECTED END-OF-FILE ENCOUNTERED ON IMAGE COPY D/S (SEARCHING FOR "HEADER RECORD") (PROCESSING DEDB AREA DATA) (I/C RECORD *xxx,xxx,xxx*)**

Explanation

Program FABADA1 encountered an end-of-file condition on the Image Copy data set while expecting more records to process. The number of records read is shown in the message.

System action

FABADA1 ends with an abend code of 3521.

User response

Verify that the data sets associated with the DFSUDUMP DD statement are correctly specified and contain the complete Image Copy dumps of the DEDB areas to be analyzed. Rerun the job.

FABA3522E   **UNEXPECTED DUMP HDR RCD READ FROM IMAGE COPY DATA SET (PROCESSING DEDB AREA DATA) (I/C RECORD *xxx,xxx,xxx*)**

Explanation

Program FABADA1 encountered an Image Copy header record while processing the area data. The number of records read is shown in the message.

System action

FABADA1 ends with an abend code of 3522.

User response

Verify that the data sets associated with the DFSUDUMP DD statement are correctly specified and contain the complete Image Copy dumps of the DEDB areas to be analyzed. Rerun the job.

FABA3523E   **AREA *areaname* CONTROL CI (SECOND CI) READ FAILED**

Explanation

Program FABADA1 cannot read the second CI.

System action

FABADA1 ends with an abend code of 3523.

User response

Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA3524E   **"DBDNAME" MISMATCH: CONTROL CARD: *dbdname1* IMAGE COPY D/S: *dbdname2* VSAM DATA SET: *dbdname2***

Explanation

The "DBDNAME=" parameter specification in the control statement does not match the dbdname in the Image Copy header record or VSAM data set.

System action

Program FABADA1 ends with an abend code of 3524.

User response

Make sure that the "DBDNAME=" parameter specification on the control statement is correct for the Image Copy data set associated with the DFSUDUMP statement or the VSAM data set associated with the DARVSAM DD statement. Correct and rerun the job.

FABA3525E   **"DEDB AREA NAME" MISMATCH: CONTROL CARD: *areaname* IMAGE COPY D/S: *areaname2***

VSAM DATA SET: *areaname2*

Explanation

The "AREA=" specification on the control statement does not match the database DDname in the Image Copy header record or VSAM data set.

System action

Program FABADA1 ends with an abend code of 3525.

User response

Make sure that the "AREA=" parameter specification on the control statement is correct for the Image Copy data set associated with the DFSUDUMP statement or the VSAM data set associated with the DARVSAM DD statement. Correct and rerun the job.

FABA3525E   "DEDB AREA NAMED *xxxxxxxx***"
 NOT FOUND IN DMB

Explanation

The "AREA=" specification on the control statement or the area name in the Image Copy header record does not match the area name in DMB. *xxxxxxxx* shows either of the following texts:

- ON CONTROL CARD
- IN I/C DATASET

DBD name, area name, or both might be printed with this message.

System action

Program FABADA1 ends with an abend code of 3525.

User response

Make sure that the files associated with the SYSIN AND DFSUDUMP (or DARVSAM) DD statements are correctly specified. Rerun the job.

**FABA3526E   DEDB DEFINITION MISMATCH:
 FIELD:** *fieldname*
 VALUE FROM *xxxxxx* **DMB:** *value-1*
 VALUE FROM I/C "DMAC" RCD:
 value-2

Explanation

Program FABADA1 determined that the specifications found in the file associated with the ACBLIB DD statement or the IMS directory did not match those found in the "DMAC" record of the Image Copy data set.

System action

FABADA1 ends with an abend code of 3526.

User response

Make sure that the file associated with the ACBLIB DD statement or the IMS directory contains the correct DMB for the database in the files associated with the DFSUDUMP DD statement. Rerun the job.

FABA3527E   OVERRIDE NOT REQUESTED

Explanation

This message is associated with the FABA0111E messages. Program FABADA1 determined that the DMB read from ACBLIB was created after the Image Copy of the DEDB was created, and that the "OVRD" keyword was not specified on the control statement.

System action

FABADA1 ends with an abend code of 3527.

User response

Make sure that the DMB from the file associated with the ACBLIB DD statement is a correct specification for the DEDB areas contained in the files associated with the DFSUDUMP DD statement. If this is the case, rerun the job specifying the "OVRD" keyword on the control statement.

**FABA3529E   MORE THAN 10 BAD CI'S
 ENCOUNTERED**

Explanation

Program FABADA1 encountered more than 10 CIs that contained one or more invalid data items.

System action

FABADA1 ends with an abend code of 3529.

User response

See the FABA0130E, FABA0131E, FABA0132E, FABA0133, and FABA0134E messages to find the CIs in error. Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA3530E   AREANAME: *areaname* **CANNOT
 BE PROCESSED DUE TO X'80' SET
 IN EQE LIST**

Explanation

Program FABADA1 detected that X'80' was set in the first byte of the Error Queue Element (EQE) list in the second CI, which means that there are more than 10 EQEs.

System action

FABADA1 ends with an abend code of 3530.

User response

Run the Full Recovery Utility, then rerun the job.

FABA3531E	CORRECT IMS RESLIB NOT CONCATENATED [- NO DFSBSCD0 MODULE FOUND] [- INVALID IMS LEVEL]

Explanation

The correct IMS load module library was not concatenated to the JOBLIB/STEPLIB because the JOBLIB/STEPLIB library has no DFSBSCD0 module, or the DFSBSCD0 module shows an unsupported IMS level.

System action

Program FABADA1 terminates with an abend code of 3531.

User response

Concatenate the correct IMS load module library to the JOBLIB/STEPLIB, and rerun the job.

FABA3531E	POINTER RECORD SAVE AREA IS FULL - INCREASE NUMBER OF ENTRIES

Explanation

The internal table used by program FABADA1 for keeping the data of root segment and deleted SDEP segment is full.

System action

FABADA1 terminates with an abend code of 3531.

User response

Either provide a 'TBLENTRY=' statement on the user-supplied control statement specifying a value greater than 10000, or increase the value currently specified. Rerun the job.

FABA3532E	RDJFCB FAILED FOR DDNAME *ddname* (RC=*xx*)

Explanation

Program FABADA1 issued an RDJFCB macro for the specified ddname. The macro failed. (Return Code is shown in decimal format.)

System action

FABADA1 terminates with an abend code of 3532.

User response

See *DFSMS DFSMSdfp Advanced Services* that explains the error return codes of the RDJFCB macro to determine the cause of the problem specified by the RDJFCB error return code *xx*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABA3533W	AREA RECOVERY NEEDED IN DBRC FOR AREA *areaname*

Explanation

Program FABADA1 found that the area specified was in recovery needed status in DBRC.

System action

If FORCE=Y is specified, then FABADA1 issues FABA3533W, sets an end-of-job return code of 4, and continues processing. If FORCE=N is specified, then FABADA1 issues FABA3533E, and terminates with an abend code of 3533.

User response

Get a LIST.RECON output report, and collect a valid area and an available area data set name. Then specify the name in DARVSAM DD, and rerun the job if necessary.

FABA3533E	AREA RECOVERY NEEDED IN DBRC FOR AREA *areaname*

Explanation

Program FABADA1 found that the area specified was in recovery needed status in DBRC.

System action

If FORCE=Y is specified, then FABADA1 issues FABA3533W, sets an end-of-job return code of 4, and continues processing. If FORCE=N is specified, then

FABADA1 issues FABA3533E, and terminates with an abend code of 3533.

User response

Get a LIST.RECON output report, and collect a valid area and an available area data set name. Then specify the name in DARVSAM DD, and rerun the job if necessary.

FABA3534W	AREA DATA SET NOT REGISTERED IN DBRC FOR AREA *areaname* DSNAME *dsname*

Explanation

Program FABADA1 found that the area data set specified in DARVSAM DD was not registered in DBRC.

System action

If FORCE=Y is specified, then FABADA1 issues FABA3534W, sets an end-of-job return code of 4, and continues processing. If FORCE=N is specified, then FABADA1 issues FABA3534E, and terminates with an abend code of 3534.

User response

Get a LIST.RECON output report, and collect an available area data set name. Then specify the name in DARVSAM DD, and rerun the job if necessary.

FABA3534E	AREA DATA SET NOT REGISTERED IN DBRC FOR AREA *areaname* DSNAME *dsname*

Explanation

Program FABADA1 found that the area data set specified in DARVSAM DD was not registered in DBRC.

System action

If FORCE=Y is specified, then FABADA1 issues FABA3534W, sets an end-of-job return code of 4, and continues processing. If FORCE=N is specified, then FABADA1 issues FABA3534E, and terminates with an abend code of 3534.

User response

Get a LIST.RECON output report, and collect an available area data set name. Then specify the name in DARVSAM DD, and rerun the job if necessary.

FABA3535W	AREA DATA SET UNAVAILABLE IN DBRC FOR AREA *areaname* DSNAME *dsname*

Explanation

Program FABADA1 found that the area data set specified in DARVSAM DD was unavailable in DBRC.

System action

Because FORCE=Y is specified, FABADA1 sets an end-of-job return code of 4 and continues processing.

User response

None. This message is informational.

FABA3535E	AREA DATA SET UNAVAILABLE IN DBRC FOR AREA *areaname* DSNAME *dsname*

Explanation

Program FABADA1 found that the area data set specified in DARVSAM DD was unavailable in DBRC.

System action

Because FORCE=N is specified, FABADA1 terminates with an abend code of 3535.

User response

Get a LIST.RECON output report, and collect a valid area and an available area data set name. Then specify the name in DARVSAM DD and rerun the job.

FABA3536W	IMS ONLINE SYSTEM IS ACCESSING AREA *areaname* WITH UPDATE/EXCLUSIVE INTENT

Explanation

Program FABADA1 found that the area specified was being used by an IMS system with update or exclusive intent.

System action

Because FORCE=Y is specified, FABADA1 sets an end-of-job return code of 4 and continues processing.

User response

None. This message is informational.

FABA3536E	IMS ONLINE SYSTEM IS ACCESSING AREA *areaname* WITH UPDATE/EXCLUSIVE INTENT

Explanation

Program FABADA1 found that the area specified was being used by an IMS system with update or exclusive intent.

System action

Because FORCE=N is specified, FABADA1 ends with an abend code of 3536.

User response

If FORCE=N is specified, then stop use of the area by entering the /STOP AREA or /DBR AREA command, or change access to the area to 'read'. Rerun the job.

FABA3537E	IMS TOOLS CATALOG INTERFACE CANNOT BE USED - UNSUPPORTED IMS RELEASE

Explanation

The IMSCATHLQ=*bsdshlq* keyword is specified, but FABADA1 or FABADA4 could not use the IMS Tools Catalog Interface to read ACBs from the IMS directory because the version of IMS is lower than 14.

System action

FABADA1 or FABADA4 ends with an abend code of 3537.

User response

Rerun the job using a supported version of IMS.

FABA3538E	IMS TOOLS CATALOG INTERFACE *function* FUNCTION (DEFINITION=CURRENT| PENDING) FAILED - RETURN CODE: *rc*, REASON CODE: *rsn*

Explanation

The IMS Tools Catalog Interface ended with an error. *function* shows the function code of the IMS Tools Catalog Interface. The return code and reason code from the IMS Tools Catalog Interface are shown in *rc* and *rsn*, respectively.

System action

FABADA1 or FABADA4 ends with an abend code of U3538.

User response

If the function is OPEN, check if the correct high-level qualifier of the bootstrap data set is specified for the IMSCATHLQ keyword. Otherwise, contact IBM Software Support.

FABA3540E	"OPEN" FAILED FOR DDNAME *ddname*

Explanation

OPEN processing failed for the file associated with the specified DD statement.

System action

Program FABADA3 issues one of the following messages after this message in order to describe the error, then ends with an abend code of 3540.

- DD STATEMENT NOT FOUND
- RECFM INCORRECT (S/B: VB)
- LRECL TOO SMALL (DO NOT SPECIFY)
- BLKSIZE TOO SMALL (GIVE US A BREAK...)
- FAILED BY OPERATING SYSTEM

User response

Make sure that a DD statement is present for the ddname specified, and that it is correctly specified. Correct any errors, and rerun the job.

FABA3545E	"DEVTYPE" FAILED FOR DDNAME *ddname* (RC = *rr*)

Explanation

Program FABADA3 issued a DEVTYPE macro to get information about the device associated with the specified ddname. The return code specified that the attempt was unsuccessful.

System action

FABADA3 ends with an abend code of 3545.

User response

Make sure that the DD statement for the specified ddname is correctly specified. Correct any errors, then rerun the job. If this situation persists, report it to systems operations personnel.

FABA3546E	CRITICAL CONTROL CARD ERROR ENCOUNTERED

Explanation

Self-explanatory.

System action

Program FABADA3 ends with an abend code of 3546.

User response

See the other messages generated for further information about the error. Correct the control statement. Rerun the job.

FABA3547E	INSUFFICIENT STORAGE FOR: *xxxx* - INCREASE REGION SIZE

Explanation

Self-explanatory.

System action

Program FABADA3 ends with an abend code of 3547.

User response

Increase the region size parameter on the EXEC statement. If a large size of the internal table is required, specify a region size greater than 16M. Rerun the job.

FABA3548E	UOW TABLE IS FULL - INCREASE NUMBER OF ENTRIES

Explanation

The internal table used by program FABADA3 to process segment and pointer information is full.

System action

FABADA3 ends with an abend code of 3548.

User response

Either provide a "TBLENTRY=" statement on the user-supplied control statement specifying a value greater than 5000, or increase the value currently specified. Rerun the job.

FABA3549E	UOW STACK IS FULL - INCREASE NUMBER OF ENTRIES

Explanation

The internal LIFO stack that is used by program FABADA3 to process segment and pointer information is full.

System action

FABADA3 ends with an abend code of 3549.

User response

Increase the value of the TBLENTRY= statement. If there is no current specification for the TBLENTRY= statement, specify the TBLENTRY= statement with a value greater than the default value 5000 and rerun the job.

FABA3550E	INPUT DATA ("*ddname1/ ddname2*") SEQUENCE ERROR - *ddname1* CTR: *zzz,zzz,zz9* *ddname2* CTR: *zzz,zzz,zz9*

Explanation

The segment and pointer information records in the input files are not in the sequence required by program FABADA3.

System action

FABADA3 ends with an abend code of 3550.

User response

Make sure that the files associated with the *ddname1* (DAR13SI or DAR13S) and *ddname2* (DAR3CSI or DAR3CS) DD statements are correctly specified, and that the *ddname2* (DAR3CSI or DAR3CS) file has been sorted. Correct any errors and rerun the job.

FABA3551E	INPUT FILE ERROR - FIRST RECORD NOT TYPE "1,0"

Explanation

Program FABADA3 determined that the first record read is not a type "1,0" record.

System action

FABADA3 ends with an abend code of 3551.

User response

Make sure that the files associated with the DAR13SI and DAR3CSI DD statements are correctly specified, and that the DAR3CSI file has been sorted. Correct any errors and rerun the job.

FABA3553E	"DEVTYPE" FAILED FOR DDNAME *ddname* (RC = *rr*)

Explanation

Program FABADA5 issued a DEVTYPE macro to get information about the device associated with the specified ddname. The return code specified that the attempt was unsuccessful.

System action

FABADA5 ends with an abend code of 3553.

User response

Make sure that the DD statement for the specified *ddname* is correctly specified. Correct any errors, and rerun the job.

FABA3554E   **"OPEN" FAILED FOR DDNAME SYSPRINT**

Explanation

OPEN processing failed for the file associated with the specified DD statement.

System action

Program FABADA5 issues one of the following messages to describe the error, then ends with an abend code equal to the message number:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

User response

Make sure that a DD statement is present for the specified ddname, and that it properly identifies the correct data set. Correct any errors, and rerun the job.

FABA3555E   **"OPEN" FAILED FOR DDNAME DAR12I**

Explanation

OPEN processing failed for the file associated with the specified DD statement.

System action

Program FABADA5 issues one of the following messages to describe the error, then ends with an abend code equal to the message number:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

User response

Make sure that a DD statement is present for the specified ddname, and that it properly identifies the correct data set. Correct any errors, and rerun the job.

FABA3556E   **"OPEN" FAILED FOR DDNAME REPORTS**

Explanation

OPEN processing failed for the file associated with the specified DD statement.

System action

Program FABADA5 issues one of the following messages to describe the error, then ends with an abend code equal to the message number:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

User response

Make sure that a DD statement is present for the specified ddname, and that it properly identifies the correct data set. Correct any errors, and rerun the job.

FABA3557E   **"OPEN" FAILED FOR DDNAME HISTORY**

Explanation

OPEN processing failed for the file associated with the specified DD statement.

System action

Program FABADA5 issues one of the following messages to describe the error, then ends with an abend code equal to the message number:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

User response

Make sure that a DD statement is present for the specified ddname, and that it properly identifies the correct data set. Correct any errors, and rerun the job.

FABA3558E   **"OPEN" FAILED FOR DDNAME REORGCTL**

Explanation

OPEN processing failed for the file associated with the specified DD statement.

System action

Program FABADA5 issues one of the following messages to describe the error, then ends with an abend code equal to the message number:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

User response

Make sure that a DD statement is present for the specified ddname, and that it properly identifies the correct data set. Correct any errors, and rerun the job.

FABA3560E	INPUT DATA SEQUENCE ERROR FILE: *ddname* CTR: *xxx,xxx,xxx*

Explanation

The records in the input file associated with the specified ddname are not in the sequence required by program FABADA5.

System action

FABADA5 ends with an abend code of 3560.

User response

Make sure that the file associated with the "ddname" has been sorted and is correctly specified. Correct any errors, and rerun the job.

FABA3561E	INPUT FILES ("*ddname1*"/"*ddname2*") ARE EMPTY

Explanation

No records were found in the input files associated with the *ddname1* (DAR12I or DAR12) and *ddname2* (DAR13I or DAR13R) DD statements.

System action

Program FABADA5 ends with an abend code of 3561.

User response

Make sure that the files associated with the *ddname1* (DAR12I or DAR12) and *ddname2* (DAR13I or DAR13R) DD statements are correctly specified. Correct any errors, and rerun the job.

FABA3562E	INPUT DATA ERROR - EXPECTING RECORD TYPE: "*a,b*"

Explanation

Program FABADA5 was expecting a record type *a,b*, but the record read was not of the correct type.

System action

FABADA5 ends with an abend code of 3562.

User response

Make sure that the files associated with the DAR12I and DAR13I DD statements are sorted and correctly specified. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABA3563E	CRITICAL CONTROL CARD ERROR ENCOUNTERED

Explanation

Program FABADA5 encountered one or more errors while parsing the user-supplied control statement.

System action

FABADA5 ends with an abend code of 3563.

User response

See the other messages generated for further information about the error. Correct the control statement, and rerun the job.

FABA3564E	INSUFF. STORAGE FOR: *aaaa* - INCREASE REGION SIZE

Explanation

Program FABADA5 issued a GETMAIN macro to get storage for *aaaa*. The return code specified that the attempt was unsuccessful.

System action

FABADA5 ends with an abend code of 3564.

User response

Increase the region size parameter value in the EXEC statement, and rerun the job.

FABA3565E	HISTORY2 DD STATEMENT MUST BE SPECIFIED WHEN UOWCTL DD STATEMENT IS SPECIFIED

Explanation

The UOWCTL DD statement for each UOW range has been specified, but the output HISTORY2 DD statement was not specified.

System action

Program FABADA5 ends with an abend code of 3565.

User response

Make sure that the HISTORY2 DD statement is correctly specified. Correct any errors, and rerun the job.

FABA3575E **"DEVTYPE" FAILED FOR DDNAME** *ddname* **(RC = *xx*)**

Explanation

Program FABADA8 issued a DEVTYPE macro to get information about the device associated with the specified ddname. The return code specified that the attempt was unsuccessful.

System action

FABADA8 ends with an abend code of 3575.

User response

Make sure that the file associated with the ddname is correctly specified. Correct any errors, and rerun the job.

FABA3576E **"OPEN" FAILED FOR DDNAME SYSPRINT**

Explanation

Program FABADA8 attempted to open the file associated with the specified ddname. The return code specified that the attempt was unsuccessful. One of the following messages describes the possible cause of the error:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

System action

FABADA8 ends with an abend code equal to the message number.

User response

Make sure that a DD statement is present, and that it correctly specifies the file.

FABA3577E **"OPEN" FAILED FOR DDNAME DADARI**

Explanation

Program FABADA8 attempted to open the file associated with the specified ddname. The return code specified that the attempt was unsuccessful. One of the following messages describes the possible cause of the error:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

System action

FABADA8 ends with an abend code equal to the message number.

User response

Make sure that a DD statement is present, and that it correctly specifies the file.

FABA3578E **"OPEN" FAILED FOR DDNAME DADARO**

Explanation

Program FABADA8 attempted to open the file associated with the specified ddname. The return code specified that the attempt was unsuccessful. One of the following messages describes the possible cause of the error:

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM.

System action

FABADA8 ends with an abend code equal to the message number.

User response

Make sure that a DD statement is present, and that it correctly specifies the file.

FABA3579E **"DATA EXTRACTION ERROR" RECORD ENCOUNTERED FOR DATABASE:** *dbdname*

Explanation

"DATA EXTRACTION ERROR" record was encountered while reading a record from the file generated by program FABADA7. FABADA8 determined that the DMAC is not from a recognizable release of IMS.

System action

FABADA8 ends with an abend code of 3579.

User response

Make sure that the file associated with the DADARI DD statement is correctly specified. If this situation persists, report it to system operations personnel.

FABA3580E **OPEN FAILED FOR DDNAME** *ddname*
VSAM ERROR DATA: RETURN CODE: *rr*
ACB "ERROR": *aaa* (*bb*)

Explanation

Program FABADA9 attempted to open the file associated with the specified ddname. The return code specified that the attempt was unsuccessful.

System action

FABADA9 ends with an abend code of 3580.

User response

Make sure that a DD statement is present and that it properly specifies the correct file. If the error is on a VSAM file, determine the cause of the error specified by the VSAM status code, correct it, and perform an AMS VERIFY on the KSDS. Rerun the job.

FABA3581E **DEVTYPE FAILED FOR DDNAME** *ddname* (**RC:** *rrr*)

Explanation

Program FABADA9 issued a DEVTYPE macro to get information about the device associated with the specified ddname. The return code indicates that the attempt was unsuccessful.

System action

FABADA9 ends with an abend code of 3581.

User response

Make sure that the ddname properly specifies the correct file. Correct any errors, and rerun the job.

FABA3581E **"GENCB"/"SHOWCB" FAILED FOR SDEP**
HISTORY FILE "ACB"/"RPL"
VSAM ERROR DATA: RETURN CODE: *xx*
REASON CODE: *aaa* (*bb*);

"GENCB"/"SHOWCB" FAILED FOR SDEP
HISTORY FILE "ACB"/"RPL"
- INSUFFICIENT STORAGE

Explanation

Program FABADA9 issued a GENCB or SHOWCB macro to generate or get information about the SDEP History file. The return code specified that the effort was unsuccessful.

System action

FABADA9 ends with an abend code of 3581.

User response

Determine the cause of the error specified by the VSAM status code, and correct it. Do an AMS VERIFY on the KSDS, and rerun the job. If this situation persists, report it to system operations personnel.

FABA3585E **I/O ERROR ACCESSING VSAM HISTORY**
FILE FOR
"POINT"/"READ"/"WRITE"
VSAM ERROR DATA: RETURN CODE: *xx*
RPL "FDBK": *aaa* (*bb*)

Explanation

Program FABADA9 encountered an error while attempting to access a record from the SDEP History file.

System action

FABADA9 ends with an abend code of 3585.

User response

Determine the cause of the error specified by the VSAM status code, and correct it. Perform an AMS VERIFY on the KSDS, and rerun the job. If this situation persists, report it to system operations personnel.

FABA3586E **DEDB ANALYZER RECORDS NOT IN**
DATE/TIME SEQUENCE (REC# *x,xxx*)

Explanation

Self-explanatory.

System action

Program FABADA9 ends with an abend code of 3586.

User response

Make sure that the output file from FABADA8 is sorted. Correct any errors, and rerun the job.

FABA3587E   **ERROR IN SDEP DATA FOR AREA** *areaname* **(DATE:** *yy.ddd***) (REC#:** *x,xxx***)**
L.B. DATA: *aaaaaaaa bbbbbbbb*
L.E. DATA: *cccccccc dddddddd*

Explanation

Program FABADA9 encountered an error while processing SDEP data associated with the specified area. The cycle count and RBA of the "logical beginning" and "logical end" are displayed in the message.

System action

FABADA9 ends with an abend code of 3587.

User response

Correct any errors, and rerun the job.

FABA3588E   **"OPEN" FAILED FOR DDNAME** *ddname*
DD STATEMENT NOT FOUND

Explanation

OPEN processing failed for the file associated with the DD statement specified.

System action

Program FABADA4 ends with an abend code of 3588.

User response

Make sure that a DD statement is present for the specified ddname, and that it is correctly specified. Rerun the job.

FABA3589E   **"DEVTYPE" FAILED FOR DDNAME** *ddname* **(RC :** *rr***)**

Explanation

Program FABADA4 issued a DEVTYPE macro to get information about the device associated with the specified ddname. The return code specified that the attempt was unsuccessful.

System action

FABADA4 ends with an abend code of 3589.

User response

Make sure that a DD statement for the ddname is correctly specified. Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABA3590E   **INPUT FILE ERROR - FIRST RECORD NOT TYPE "1,0"**

Explanation

Program FABADA4 determined that the first record read is not a type "1,0" record.

System action

FABADA4 ends with an abend code of 3590.

User response

Make sure that the files associated with the DAR13SI and DAR3CSI DD statements are correctly specified, and the DAR3CSI file has been sorted. Correct any errors, and rerun the job.

FABA3591E   **INPUT FILE ("DAR13SI"/"DAR3CSI") IS EMPTY**

Explanation

The input file that normally contains segment and pointer information is empty.

System action

Program FABADA4 ends with an abend code of 3591.

User response

Make sure that the files associated with the DAR13SI and DAR3CSI DD statements are correctly specified. Correct any errors, and rerun the job.

FABA3592E   **INPUT DATA ("DAR13SI"/"DAR3CSI") SEQUENCE ERROR**
- DAR13SI CTR: *zzz,zzz,zz9*
DAR3CSI CTR: *zzz,zzz,zz9*

Explanation

The segment and pointer information records in the input files are not in the sequence requested by program FABADA4.

System action

FABADA4 ends with an abend code of 3592.

User response

Make sure that the files associated with the DAR13SI and DAR3CSI DD statements are correctly specified, and that the DAR3CSI file has been sorted. Correct any errors, and rerun the job.

FABA3593E   **INVALID DATA IN "PARM" FIELD**

Explanation

Program FABADA4 encountered an invalid "PARM" specification on the EXEC statement.

System action

FABADA4 ends with an abend code of 3593.

User response

Correct any errors, and rerun the job.

FABA3594E   **INPUT DATA "DAR4ERRS" SEQUENCE ERROR - DAR4ERRS CTR: *z,zz9***

Explanation

The segment and pointer information records in the input file are not in the sequence requested by program FABADA4.

System action

FABADA4 ends with an abend code of 3594.

User response

Make sure that the file associated with the DAR4ERRS DD statement is correctly specified, and that the DAR4ERRS file has been sorted. Correct any errors, and rerun the job.

FABA3595E   **AREA: *zzzzz* SPECIFIED IN *ddname* DATA SET IS NOT FOUND IN DAR13SI DATA SET**

Explanation

Program FABADA4 detected that there was no TYPE "1,0" record found in the DAR13SI data set which corresponds to the specified area number in the record of the specified data set. *zzzzz* is the area number.

System action

FABADA4 ends with an abend code of 3595.

User response

Make sure that the files associated with the DAR13SI DD statements are correctly specified. Correct any errors, and rerun the job.

FABA3596E   **SORT CONTROL STATEMENT NOT FOUND IN SORT3CS DATA SET**

Explanation

Program FABADA4 detected that necessary sort control statements were not found in the data set associated with the SORT3CS DD statement. This message is issued if DUMMY is specified on SORT3CS DD statement.

System action

FABADA4 ends with an abend code of 3596.

User response

Make sure that the data set associated with the SORT3CS DD statement is correctly specified. Correct any errors. Rerun the job.

FABA3597E   **SORT REQUEST FAILED**

Explanation

Program FABADA4 linked the DFSORT program internally to sort a data set associated with the SORTIN DD statement and the DFSORT program returned an error code.

System action

FABADA4 ends with an abend code of 3597.

User response

Make sure that SYSOUT, SORTIN, SORTOUT, SORTWK01, SORTWK02, and SORTWK03 DD statements are correctly specified. If there are DFSORT messages on the data set associated with the SYSOUT DD statement, check those messages

and correct any errors. Make sure that the data set associated with the SORT3CS DD statement is correctly specified.

FABA3598E   LOAD FAILED FOR COMPRESSION ROUTINE *comprname* **(ABEND CODE S***xxxx* **/ REASON CODE** *yyyyyyyy***)**

Explanation

Program FABADA1 issued an SVC 8 (LOAD) to bring a copy of the segment edit/compression routine into the core. The return code from OS("S*xxxx*") specified that the attempt was unsuccessful.

System action

FABADA1 ends with an abend code of 3598.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABA3600E   SEGMENT IN AREA *zzzzz* **AT RBA** *xxxxxxx* **(SEGCODE** *yy***) RETURNED FROM COMPRESSION ROUTINE** *comprname* **EXCEEDS MAX DEFINED LENGTH**

Explanation

Program FABADA1 determined that after the specified segment expanded the edit/compression routine, the data in the segment was longer than the length defined in the DBD member of the ACB library specified by the ACBLIB DD statement. *zzzzz* is the area number.

System action

FABADA1 ends with an abend code of 3600.

User response

Make sure that the segment edit/compression routine is correct for the specified segment. Correct any errors, and rerun the job.

FABA3601E   SEGMENT IN AREA *zzzzz* **AT RBA** *xxxxxxx* **(SEGCODE** *yy***) RETURNED FROM COMPRESSION ROUITINE** *comprname* **TOO SHORT**

Explanation

Program FABADA1 determined that after the specified segment expanded the edit/compression routine, the data in the segment was shorter than the length defined in the DBD member of the ACB library specified by the ACBLIB DD statement. *zzzzz* is the area number.

System action

FABADA1 ends with an abend code of 3601.

User response

Make sure that the segment edit/compression routine is correct for the specified segment. Correct any errors, and rerun the job.

FABA3602E   SEGMENT IN AREA *zzzzz* **AT RBA** *xxxxxxx* **(SEGCODE** *yy***) RETURNED FROM COMPRESSION ROUTINE** *comprname* **KEY FIELD MODIFIED**

Explanation

Program FABADA1 determined that the key field in the specified segment was modified by the specified segment edit/compression routine. *zzzzz* is the area number.

System action

FABADA1 ends with an abend code of 3602.

User response

Make sure that the segment edit/compression routine is correct for the specified segment. Correct any errors, and rerun the job.

FABA3603E   "OPEN" FAILED FOR DDNAME RMODLIB

Explanation

Program FABADA1 issued an OPEN macro for the file associated with the RMODLIB DD statement. The return code specifies that the result was unsuccessful.

System action

FABADA1 issues one of the following messages after this message text, then ends with an abend code of 3603.

- DD STATEMENT NOT FOUND
- FAILED BY OPERATING SYSTEM

User response

Make sure that the RMODLIB DD statement is present, and that it specifies the correct data set. Correct any errors, and rerun the job.

FABA3604E	UNSUPPORTED LEVEL OF IMS IS BEING USED: *xxx* IMS LEVEL OF THIS RUN

Explanation

Program FABADA7 was run as an exit routine of the IMS Sequential Dependent Scan utility, but the level of the IMS system was not supported by the IMS DBT 2.x.

System action

IMS Sequential Dependent Scan utility ends with an abend code of 3604. IMS message DFS0782A follows this message.

User response

Rerun the IMS Sequential Dependent Scan utility, ensuring that FABADA7 is running under the correct IMS level.

FABA3605E	LARGE BLOCK SIZE IMAGE COPY DATA SET IS NOT SUPPORTED UNDER THIS OPERATING SYSTEM

Explanation

The DEDB Pointer Checker detected that one of the following image copy data set specified for input could not be processed on the operating system that does not support LBI.

- IMS standard LBI image copy data set
- IMS standard LBI image copy 2 data set
- IMS standard image copy 2 data set on TAPE that was produced without the BLKSIZE parameter specification
- IMS standard LBI image copy data set produced by the ICE

System action

The DEDB Pointer Checker issues a U3605 abend.

User response

You have run DEDB Pointer Checker on a system that does not support LBI. Rerun the job on a system that supports LBI.

FABA3606E	LARGE BLOCK SIZE COMPRESSED IMAGE COPY DATA SET

GENERATED BY ICE IS NOT SUPPORTED

Explanation

The DEDB Pointer Checker detected that the compressed LBI image copy data set produced by the ICE was specified for input. The data set is not supported.

System action

The DEDB Pointer Checker issues a U3606 abend.

User response

Specify the IMS standard image copy data set and rerun the job.

FABA3851E	RECORD LENGTH RETURNED FROM EXIT ROUTINE: *comprtn* IS NOT CONSISTENT WITH ORIGINAL LENGTH AT RBA: *rba*

Explanation

The record length increased by the image copy compression routine *comprtn* is not equal to the length of the original CI at *rba*.

System action

The DEDB Pointer Checker ends with an abend code of 3851.

User response

Make sure that the correct IMS HP Image Copy library is concatenated to the JOBLIB or STEPLIB library, and rerun the job.

FABA3852E	TERMINATION REQUIRED BY *comprtn* IN *function* CALL RC=*rc*

Explanation

The image copy compression routine *comprtn* returned *rc* in function call because errors were detected while compressing or decompressing the CI.

System action

The DEDB Pointer Checker ends with an abend code of 3852.

User response

Check the integrity of the area by using the Analyze function of FPA or the OPC utility, correct the integrity errors, and rerun the job.

FABA3853E	LOAD FAILED FOR IMAGE COPY COMPRESSION ROUTINE *comprtn* (ABEND CODE S*xxx* / REASON CODE *reason_code*)

Explanation

The DEDB Pointer Checker could not load the image copy compression routine *comprtn*. *xxx* shows the abend code and *reason_code* shows the reason code.

System action

The DEDB Pointer Checker ends with an abend code of 3853.

User response

Make sure that the IMS HP Image Copy library is concatenated to the JOBLIB or STEPLIB library. For further explanation of the error, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job.

FABA3854E	DELETE FAILED FOR IMAGE COPY COMPRESSION ROUTINE *comprtn*

Explanation

The DEDB Pointer Checker could not delete the image copy compression routine *comprtn*.

System action

The DEDB Pointer Checker ends with an abend code of 3854.

User response

Contact IBM Software Support.

FABA3901E	UNKNOWN FUNCTION REQUESTED

Explanation

The module FABAIC20 was called with an unknown function code.

System action

The DEDB Pointer Checker issues a U3901 abend.

User response

Correct all errors and rerun the job. If the problem remains, save the entire run listing (including the dump, JCL and all FPB reports) and contact IBM Software Support.

FABA3902E	"GET" FAILED FOR DDNAME *ddname*

Explanation

The DEDB Pointer Checker tried to issue a GET macro for the file associated with the DD statement. The SYNAD exit routine was called a GET macro for the file associated with the DD statement. The SYNAD exit routine was called during the GET processing.

System action

The DEDB Pointer Checker issues a U3902 abend.

User response

Make sure that the DD statement is specifying the correct data set. Correct the error, and rerun the job.

FABA3903E	"GETMAIN" FAILURE OCCURRED (RC=*XX*) FOR IMAGE COPY 2 BUFFER

Explanation

The DEDB Pointer Checker issued a GETMAIN macro. The return code specifies that the attempt to do so was unsuccessful.

System action

The DEDB Pointer Checker issues a U3903 abend.

User response

Correct all errors and rerun the job. If the problem persists, save the entire run listing (including the dump, JCL, and all FPB reports) and contact IBM Software Support.

FABA3904E	"FREEMAIN" FAILURE OCCURRED (RC=*XX*) FOR IMAGE COPY 2 BUFFER

Explanation

The DEDB Pointer Checker issued a FREEMAIN macro. The return code specifies that the attempt to do so was unsuccessful.

System action

The DEDB Pointer Checker issues a U3904 abend.

User response

Correct all errors and rerun the job. If the problem remains, save the entire run listing (including the dump, JCL and all FPB reports) and contact IBM Software Support.

FABA3911E   FIRST RECORD IS NOT A TAPE HEADER RECORD

Explanation

The first record of the image copy taken with the Database Image Copy 2 utility is not a tape header record, but the DEDB Pointer Checker expected it to be.

System action

The DEDB Pointer Checker issues a U3911 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3912E   INITIALIZATION FOR IMAGE COPY 2 READING INCOMPLETE - DATA SET HEADER RECORD NOT FOUND - VOLUME DEFINITION RECORD NOT FOUND

Explanation

The DEDB Pointer Checker could not find the record specified. The DEDB Pointer Checker collects the information necessary to analyze the data set while performing initialization processing, but processing was incomplete.

System action

The DEDB Pointer Checker issues a U3912 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3913E   UNEXPECTED LOGICAL END OF FILE ON IMAGE COPY 2 DATA SET

Explanation

An unexpected trailer record was encountered while the DEDB Pointer Checker was reconstructing the logical record image from the image copy taken with the Database Image Copy 2 utility. The image copy has two trailer records and the DEDB Pointer Checker regards them as the end-of-data.

System action

The DEDB Pointer Checker issues a U3913 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3914E   INTERNAL RECORD LENGTH ERROR ON IMAGE COPY 2 DATA SET

Explanation

The DEDB Pointer Checker checked the logical record length of the dumped data set in the image copy and found an invalid length.

System action

The DEDB Pointer Checker issues a U3914 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3915E   SPECIFIED IMAGE COPY 2 DATA SET IS EMPTY

Explanation

The DEDB Pointer Checker attempted to get a record from the specified image copy taken with the Database Image Copy 2 utility, but the data set was empty.

System action

The DEDB Pointer Checker issues a U3915 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3916E   **UNEXPECTED PHYSICAL END OF FILE ON IMAGE COPY 2 DATA SET**

Explanation

The DEDB Pointer Checker attempted to get a record from the specified image copy taken with the Database Image Copy 2 utility, but an unexpected physical end-of-file was encountered. The image copy has no trailer record. DEDB Pointer Checker normally ends image copy reading on the first trailer record instead of the physical end-of-file.

System action

The DEDB Pointer Checker issues a U3916 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3921E   **UNSUPPORTED DUMP FORMAT. LOGICAL DUMP REQUIRED**

Explanation

The image copy taken with the Database Image Copy 2 utility must be a DFSMSdss logical dump format, but it was not.

System action

The DEDB Pointer Checker issues a U3921 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3922E   **MORE THAN ONE DATA SET IS DUMPED IN THE IMAGE COPY**

Explanation

Input data set contained more than one database data set. It will be an error because the image copy taken with the Database Image Copy 2 utility contains only one database data set.

System action

The DEDB Pointer Checker issues a U3922 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3923E   **DATA SET ORGANIZATION CONFLICT**
-EXPECTED : *xxxx*
-DUMPED DATA SET : *yyyy*

Explanation

The dumped data set organization conflicted with the parameter which was passed by the caller. (*xxxx*/*yyyy*: V SAM/OSAM/ESDS/KSDS)

System action

The DEDB Pointer Checker issues a U3923 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3924E   **VSAM KSDS DATA SET WAS DUMPED WITHOUT VALIDATION**

Explanation

VSAM KSDS must be dumped with validation by the IMS Database Image Copy 2 utility, but the input data set was not.

System action

The DEDB Pointer Checker issues a U3924 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3925E   **DUMPED DATA SET IS UNSUPPORTED FORMAT**
-FOR OSAM DATA SET, BLOCKED RECORD
-FOR VSAM DATA SET, NOT ESDS/ KSDS
-FOR VSAM DATA SET, SPANNED RECORD
-FOR DEDB/HD AM/HIDAM DATA, BLOCKED RECORD

Explanation

The dumped database data set organization is not supported. Details are shown by sub-messages.

System action

The DEDB Pointer Checker issues a U3925 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DD statement identifies the correct data set. Correct the error, and rerun the job.

FABA3926E   **LOAD FAILED FOR ADRCOMP LOAD MODULE (ABEND CODE S*xxx*/ REASON CODE *yyyyyyyy*)**

Explanation

FABAIC20 issued an SVC 8(LOAD) to bring a copy of the data decompression module ADRCOMP into the core. The return code from OS ("S*xxx* ") specified that the attempt was unsuccessful.

System action

The DEDB Pointer Checker issues a U3926 abend.

FABB messages

The following information is about messages and codes that begin with FABB.

FABB0100I   **FABBME1 ENDED NORMALLY**

Explanation

All processing has successfully completed.

System action

FABBME1 ends normally with a return code of zero.

User response

None. This message is informational.

FABB0100W   **FABBME1 ENDED WITH WARNINGS**

Explanation

Warning conditions were detected.

System action

Program FABBME1 ends with a return code of 4.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABA3927E   **ERROR IN CALL TO ADRCOMP - ERROR CODE: *xx***

Explanation

The data decompression module ADRCOMP encountered an error while processing compressed Image Copy 2 data set.

System action

The DEDB Pointer Checker issues a U3927 abend.

User response

You may have specified an incorrect or broken data set. Make sure that the DFSUDUMP statement specifies the correct data set. If the problem persists, contact IBM Software Support.

User response

Examine the FABB01*xx* messages generated to determine the cause of these warning conditions. Correct the problem and rerun this job or continue with the execution as specified.

FABB0100E   **FABBME1 ENDED WITH ERRORS**

Explanation

Errors were detected.

System action

Program FABBME1 ends with a return code of 8.

User response

Examine the FABB01*xx* messages generated to determine the cause of these errors. Correct the problem and rerun this job or continue with the execution as specified.

FABB0101I   **PROCESSING COMPLETE FOR DBDNAME: *dbdname***

Explanation

All DEDB Pointer Checker SRDA13 data for the specified DEDB has been processed.

System action

Program FABBME1 continues normal processing.

User response

None. This message is informational.

FABB0102I   RECORDS READ FROM MEDARI: *zzz,zzz,zz9*
(TYPE "1,0, ": *zzz,zzz,zz9*)
(TYPE "1,1, ": *zzz,zzz,zz9*)
(TYPE "3 RESOLVED": *zzz,zzz,zz9*)
(OTHER: *zzz,zzz,zz9*)
RECORDS WRITTEN TO MEDARO: *zzz,zzz,zz9*
(TYPE "1,0, ": *zzz,zzz,zz9*)
(TYPE "1,1, ": *zzz,zzz,zz9*)
(TYPE "3, , ": *zzz,zzz,zz9*)

Explanation

This message is generated at end of job to detail the number of records read from the MEDARI DD statement, and the number written to the MEDARO DD statement. (The "counts by record type" information is provided only when RPT=YES is specified on the user-supplied control statement.)

System action

Program FABBME1 continues normal processing.

User response

None. This message is informational.

FABB0103I   DEDB ANALYZER RECORDS READ FOR AREAS:
zzzzz zzzzz zzzzz zzzzz zzzzz zzzzz zzzzz zzzzz

Explanation

This message is generated during end-of-job processing to specify the area, or areas, for which FABBME1 encountered DEDB Pointer Checker SRDA13 data in the file(s) associated with the DD statement MEDARI. *zzzzz* is the area number.

System action

Program FABBME1 continues normal processing.

User response

None. This message is informational.

FABB0104I   DEDB MODELER RECORDS WRITTEN FOR AREAS:
zzzzz zzzzz zzzzz zzzzz zzzzz zzzzz zzzzz zzzzz

Explanation

This message is generated during end-of-job processing to specify the area, or areas, for which FABBME1 wrote DEDB Tuning Aid records to the MEDARO file. *zzzzz* is the area number.

System action

Program FABBME1 continues normal processing.

User response

None. This message is informational.

FABB0105I   SORT CONTROL CARDS:
SORT
FIELDS=(*xx,xx***,CH,A),SIZE=***xxxxx*
RECORD
TYPE=V,LENGTH=(,,,*xx,xx***)**

Explanation

This message is generated during end-of-job processing to show the SORT control statement records written to the data set associated with the DD statement MESORTCD.

System action

Program FABBME1 continues normal processing.

User response

None. This message is informational.

FABB0110W   NO. OF AREAS MISMATCH
- NO. AREAS IN ORIGINAL DATABASE: *zzzz9*
- NO. AREAS IN DEDB ANALYZER DATA SET: *zzzz9*

Explanation

Program FABBME1 determined that the file(s) associated with the DD statement MEDARI did not contain DEDB Pointer Checker SRAD13 data for all the areas defined in the original DMB.

System action

FABBME1 sets an end-of-job return code of 4, and continues processing.

User response

Verify that the file(s) associated with the DD statement MEDARI correctly specify the data set(s) containing the DEDB Pointer Checker SRDA13 for all required areas of the database being analyzed. See "Input for DEDB Tuning Aid" on page 100 and Table 28 on page 100 in this manual for further information about the input requirements of FABBME1.

FABB0112E **NO RECORDS WRITTEN TO DEDB MODELER RECORDS D/S**

Explanation

Program FABBME1 did not write out any DEDB Tuning Aid records to the file associated with the MEDARO DD statement.

System action

FABBME1 ends with a return code of 8.

User response

Correct any data set or control statement errors noted in other FABB01*xx* error messages. Verify that the Randomizer module being used will, for the data in the file(s) associated with the DD statement MEDARI, select RAPs in at least one of the area's selected in the AREA= parameter of the control statement provided. Correct the error. Rerun the job.

FABB0120E **CONTROL CARD DATA SET IS EMPTY**

Explanation

The file associated with the DD statement SYSIN was empty, or a DUMMY data set.

System action

Program FABBME1 ends with a user abend 3620.

User response

Make sure that the SYSIN file contains a valid control statement. Rerun the job.

FABB0122E **INVALID CONTROL CARD ENCOUNTERED (CARD NO. *zz9*)**
```
1...5...10...15...20...25..
.30...35
```

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxx
```

Explanation

A control statement was found to contain syntax errors. (The control statement record is shown.)

System action

Program FABBME1 ends with a user abend 3620.

User response

See the other messages generated for information about the error detected. Correct the control statement. Rerun the job.

FABB0123W **BLANK CONTROL CARD ENCOUNTERED - IGNORED**

Explanation

Self-explanatory.

System action

Program FABBME1 sets an end-of-job return code of 4, and continues processing.

User response

Remove the blank statement in future runs.

FABB0124E **INVALID CHARACTER NEAR COLUMN *xx***

Explanation

Program FABBME1 encountered an invalid character while parsing the control statement.

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0125E **UNKNOWN KEYWORD NEAR COLUMN *xx***

Explanation

Program FABBME1 detected an unknown "keyword=value" specification on the control statement.

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement. Rerun the job.

FABB0126E **["DBDNAME=" | "AREA="] SPECIFICATION NOT FOUND**

Explanation

Self-explanatory.

System action

Program FABBME1 ends with a user abend 3620.

User response

Make sure that the control statement in the SYSIN file has, at least, a valid specification for the DBDNAME= and AREA= parameters. Rerun the job.

FABB0127E **ERROR IN SPECIFICATION OF THE "DBDNAME=" PARM - VALUE FOR DATA BASE NAME NOT PRESENT**

Explanation

Program FABBME1 found an error in the specification of the specified control statement "keyword."

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0127E **INVALID SPECIFICATION FOR THE "DBDNAME=" PARM - DATA BASE NAME EXCEEDS 8 CHARACTERS**

Explanation

Program FABBME1 found an error in the specification of the specified control statement "keyword."

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0127E **ERROR IN SPECIFICATION OF THE "AREA=" PARM - VALUE FOR AREA NO.(S) NOT PRESENT**

Explanation

Program FABBME1 found an error in the specification of the specified control statement "keyword."

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0127E **INVALID SPECIFICATION FOR THE "AREA=" PARM - INVALID SYNTAX NEAR COLUMN** *xx*

Explanation

Program FABBME1 found an error in the specification of the specified control statement "keyword."

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0127E **ERROR IN SPECIFICATION OF THE "RMOD=" PARM - VALUE FOR RANDOMIZER NAME NOT PRESENT**

Explanation

Program FABBME1 found an error in the specification of the specified control statement "keyword."

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0127E **INVALID SPECIFICATION FOR THE "RMOD=" PARM - RANDOMIZER NAME EXCEEDS 8 CHARACTERS**

Explanation

Program FABBME1 found an error in the specification of the specified control statement "keyword."

System action

FABBME1 ends with a user abend 3620.

User response

Correct the control statement, and rerun the job.

FABB0130W   AREA *zzzzz* REQUESTED; NOT DEFINED IN *dbdname*

Explanation

Program FABBME1 determined that the DMB *dbdname* does not contain a definition for an area, as specified in the message, which was specified on the control statement provided. *zzzzz* is the area number.

System action

FABBME1 sets an end-of-job return code of 4, and continues processing.

User response

Verify that the specification of the AREA= parameter on the control statement provided is correct. Correct the error and rerun the job, or continue execution of the DEDB Tuning Aid procedure, as desired.

FABB0135I   CONTROL CARD RANDOMIZER OVERRIDE DETECTED - RANDOMIZER *rmname-1* WILL BE USED (*rmname-2* SPECIFIED IN DMB *dbdname*)

Explanation

RMOD=*rmname-1* was found on the user-supplied control statement. This module will be loaded from RMODLIB and used for this execution of program FABBME1.

The DMB *dbdname* (from ACBLIB) was generated with RMNAME=*rmname-2*.

System action

FABBME1 continues normal processing.

User response

None. This message is informational.

FABB0300I   UTILITY FABBME3 ENDED NORMALLY

Explanation

All processing has successfully completed.

System action

Program FABBME3 ends normally with a return code of zero.

User response

None. This message is informational.

FABB0300W   FABBME3 ENDED WITH WARNINGS

Explanation

Warning conditions were detected during this execution of program FABBME3.

System action

FABBME3 ends with a return code of 4.

User response

Examine the FABB03*xx* messages generated to determine the cause of these warning conditions. Correct the problem and rerun this job, or continue with the execution, as specified.

FABB0300E   FABBME3 ENDED WITH ERRORS

Explanation

Errors were detected during this execution of program FABBME3.

System action

FABBME3 ends with a return code of 8.

User response

Examine the FABB03*xx* messages generated to determine the cause of these errors. Correct the problems and rerun this job, or continue with the execution, as specified.

FABB0301I   PROCESSING STARTED FOR DBDNAME: *dbdname* AREA: *areaname*

Explanation

Program FABBME3 has started the tuning process for area *areaname* in database *dbdname*.

System action

FABBME3 continues execution.

User response

None. This message is informational.

FABB0302I	NUMBER OF RECORDS WRITTEN FOR THE AREA: DDNAME MEDAR12O *zzz,zzz,zz9* (1,0: *zzz,zzz,zz9*) (1,1: *zzz,zzz,zz9*) (2,T: *zzz,zzz,zz9*) (2,F: *zzz,zzz,zz9*) (2,R: *zzz,zzz,zz9*) DDNAME MEDAR3O *zzz,zzz,zz9* (3,S: *zzz,zzz,zz9*)

Explanation

This message is generated to show the number of records written to the data sets associated with the DD statements MEDAR12O and MEDAR3O as a result of processing the DEDB Tuning Aid data for the area named in the message.

System action

Program FABBME3 continues normal processing.

User response

None. This message is informational.

FABB0303I	SORT CONTROL CARDS WRITTEN TO DDNAME: *ddname* SORT FIELDS=(*xx,xx*,CH,A),SIZE=*xxxxx* RECORD TYPE=V,LENGTH=(,,,*xx,xx*)

Explanation

This message is generated during end-of-job processing to show the SORT control statement records written to the data sets associated with the DD statements SORT12, and SORT3.

System action

Program FABBME3 continues normal processing.

User response

None. This message is informational.

FABB0310I	NO RECORDS IN AREA

Explanation

No DEDB Tuning Aid segment data records were read for the specified area.

System action

Program FABBME3 continues processing.

User response

None. This message is informational.

FABB0390W	MISSING OR INVALID PARM - LOAD ASSUMED

Explanation

PARM= was not specified on the EXEC statement for program FABBME3, or was specified with a value other than L or R.

System action

FABBME3 assumes a value of L ("load" mode processing), and continues execution.

User response

Make sure that PARM= is coded with either of the valid values (L or R) in future runs.

FABB0391E	INPUT FILE IS EMPTY

Explanation

The MEDARI file was empty, or a DUMMY data set.

System action

Program FABBME3 ends with a return code of 8.

User response

Verify that DEDB Tuning Aid records were written by FABBME1. Verify that the MEDARI DD statement for FABBME3 specifies the data set created in step SORT13 of the DEDB Tuning Aid procedure. Correct any errors, and rerun the job.

FABB3600E	SEQUENCE ERROR DETECTED IN DEDB ANALYZER RECORDS (AT RECORD *zzz,zzz,zz9*)

Explanation

Program FABBME1 determined that the DEDB Pointer Checker SRDA13 data in the file associated with the ddname MEDARI was not in the proper order.

System action

FABBME1 ends with a User Abend 3600.

User response

Make sure that the MEDARI DD statement specifies the data set created by the SORTR13 step of the DEDB Pointer Checker run for the database being tuned, and that the SORT was successful. Correct any errors, and rerun the job.

FABB3601E   DEDB ANALYZER DATA SET UNUSABLE - NOT GENERATED IN "TYPRUN=MODEL" MODE

Explanation

Program FABBME1 determined that the SRDA13 data in the MEDARI file was not created with TYPRUN=MODEL specified on the FABADA1 control statement in the DEDB Pointer Checker run.

System action

FABBME1 ends with a User Abend 3601.

User response

Rerun the DEDB Pointer Checker procedure with TYPRUN=MODEL specified on the FABADA1 control statement. Then rerun the DEDB Tuning Aid job.

FABB3602E   NO RECORDS FOUND IN DEDB ANALYZER RECORDS D/S

Explanation

Program FABBME1 determined that the file associated with the DDname MEDARI was empty, or DUMMY.

System action

FABBME1 ends with a User Abend 3602.

User response

Make sure that the MEDARI DD statement specifies the data set created by step SORTR13 of the DEDB Pointer Checker run for the database being tuned. Correct the error, and rerun the job.

FABB3605E   DEDB ANALYZER "AREA DESCRIPTION" RECORDS NOT FOUND (AT RECORD *zzz,zzz,zz9*)

Explanation

Program FABBME1 determined that the file associated with the DDname MEDARI did not contain the required DEDB Pointer Checker "area Description" records.

System action

FABBME1 ends with a User Abend 3605.

User response

Make sure that the DD statement provided for MEDARI correctly specifies the proper data set—that created by the SORTR13 step of the DEDB Pointer Checker procedure. Correct any errors, and rerun the job.

FABB3606E   DEDB ANALYZER "SEG. DESCRIPTION" RECORDS NOT FOUND (AT RECORD *zzz,zzz,zz9*)

Explanation

Program FABBME1 determined that the file associated with the DDname MEDARI did not contain the required DEDB Pointer Checker "Segment Description" records.

System action

FABBME1 ends with a User Abend 3606.

User response

Make sure that the DD statement provided for MEDARI correctly specifies the proper data set—that created by the SORTR13 step of the DEDB Pointer Checker procedure. Correct any errors, and rerun the job.

FABB3607E   DEDB ANALYZER RECORD FOR ROOT SEGMENT NOT FOUND (AT RECORD *zzz,zzz,zz9*)

Explanation

Program FABBME1 determined that the MEDARI file did not contain a segment data record for the root segment of the segment in the current SRDA13 record.

System action

FABBME1 ends with a User Abend 3607.

User response

Make sure that the DD statement provided for MEDARI correctly specifies the data set created by step SORTR13 of the DEDB Pointer Checker procedure, and that the area processed is not empty. Correct any errors, and rerun the job.

FABB3610E **"OPEN" FAILED FOR DDNAME**
ddname **- NOT A FIXED LENGTH RECORD DATA SET - NOT A VARIABLE LENGTH RECORD DATA SET - NOT AN 80 BYTE RECORD DATA SET - RECORD LENGTH (LRECL) TOO SMALL (** *xxxxx* **SPECIFIED) (** *xxxxx* **REQUIRED) - BLOCK SIZE (BLKSIZE) TOO SMALL (** *xxxxx* **SPECIFIED) (** *xxxxx* **REQUIRED)**

Explanation

Program FABBME1 attempted to OPEN the file associated with the specified *ddname*. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME1 ends with a User Abend 3610.

User response

Make sure that a DD statement has been provided for *ddname*, and that it correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3612E **"DEVTYPE" FAILED FOR DDNAME**
ddname **(RC =** *xx* **)**

Explanation

Program FABBME1 issued an SVC 24 (DEVTYPE) to obtain information about the input/output device associated with *ddname*. The return code, as shown in the generated message, specified that the attempt to do so was unsuccessful.

System action

FABBME1 ends with a User Abend 3612.

User response

Make sure that a DD statement has been provided for *ddname*, and that it correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3620E **CONTROL CARD ERRORS DETECTED - PROCESSING TERMINATED**

Explanation

Program FABBME1 detected one or more errors in the user-specified control statements.

System action

FABBME1 ends with a User Abend 3620.

User response

See the FABB01*xx* messages generated to determine the nature of the problem. Correct the errors, and rerun the job.

FABB3621E **NO VALID AREA SELECTION CRITERIA FOUND**

Explanation

The user-specified control statements did not contain a valid specification of the AREA= parameter. (Either there were syntax errors, or the area(s) requested were not defined in the DMB read from ACBLIB.)

System action

Program FABBME1 ends with a User Abend 3621.

User response

See the FABB01*xx* messages generated to determine the nature of the problem. Make sure that a valid specification of the AREA= parameter is provided, and that the area(s) are defined in the DMB in ACBLIB data set. Correct the errors, and rerun the job.

FABB3625E **DBDNAME MISMATCH - DBDNAME FROM ACBLIB MEMBER** *xxxxxxxx* **:** *dbdname1* **- DBDNAME IN DEDB ANALYZER DATA SET :** *dbdname2*

Explanation

The DEDB Pointer Checker SRDA13 record from the file associated with the MEDARI DD statement was not for the database named in the DMB read from ACBLIB (that is, the database named in the DBDNAME= parameter of the user-specified control statement).

System action

Program FABBME1 ends with a User Abend 3625.

User response

Make sure that the DBDNAME= parameter on the user-supplied control statement is correct. Verify that the MEDARI DD statement correctly specifies the data set(s) containing the DEDB Pointer Checker SRDA13 data for the database to be analyzed. Correct any errors. Rerun the job.

FABB3626E   **DEDB ANALYZER DATA SET CONTAINS MULTIPLE DATABASES - PREVIOUS DBDNAME: *dbdname1* - "NEW" DBDNAME: *dbdname2* (AT RECORD: *zzz,zzz,zz9*)**

Explanation

Program FABBME1 read DEDB Pointer Checker SRDA13 records for a database other than that currently being processed.

System action

FABBME1 ends with a User Abend 3626.

User response

Verify that the MEDARI DD statement specifies data set(s) containing DEDB Pointer Checker SRDA13 data only for the database being analyzed. Correct any errors, and rerun the job.

FABB3627E   **DATA FOUND IN DEDB ANALYZER DATA SET FOR PREVIOUSLY PROCESSED AREA *zzzzz* (REC: *zzz,zzz,zz9*)**

Explanation

Program FABBME1 encountered DEDB Pointer Checker SRDA13 data for an area which had already been processed. *zzzzz* is the area number.

System action

FABBME1 ends with a User Abend 3627.

User response

Verify that the MEDARI DD statement is specified correctly. Correct the error. Rerun the job.

FABB3630E   **NO. SEGMENTS DEFINED MISMATCH - NO. DEFINED IN ACBLIB MEMBER *acbmbr*: zz9 - NO. DEFINED IN DEDB ANALYZER DATA SET: *zz9***

Explanation

Program FABBME1 determined that the number of segments defined in the database described by ACBLIB member *acbmbr* was not the same as the number defined in the database for which DEDB Pointer Checker SRDA13 data was provided in the file(s) associated with the DD statement MEDARI.

System action

FABBME1 ends with a User Abend 3630.

User response

Make sure that the DBDNAME= parameter on the user-supplied control statement is correct, and that the specification of the ACBLIB DD statement identifies the correct IMS ACB library data set. Verify that the MEDARI DD statement specifies the correct data set(s). Correct any errors. Rerun the job.

FABB3631E   **SEGMENT NAME MISMATCH FOR SEG-CODE *xx* - NAME IN ACBLIB MEMBER *acbmbr* : *segname1* - NAME IN DEDB ANALYZER DATA SET: *segname2***

Explanation

Program FABBME1 determined that the segment name definitions in the DMB contained in the ACBLIB member *acbmbr* differ from those in the DEDB Pointer Checker SRDA13 data records.

System action

FABBME1 ends with a User Abend 3681.

User response

Make sure that the DBDNAME parameter on the user-supplied control statement is correct, and that the specification of the ACBLIB DD statement identifies the correct IMS ACB library data set. Verify that the MEDARI DD statement specifies the correct data set(s). Correct any errors, and rerun the job.

FABB3632E   **ROOT SEGMENT KEY POSITION/ LENGTH MISMATCH SOURCE "START=" "BYTES=" ACBLIB MEMBER *acbmbr zzzz9 zz9* DEDB ANALYZER DATA SET *zzzz9 zz9* (REC: *zzz,zzz,zz9*)**

Explanation

Program FABBME1 determined that the root segment key field position, and length definitions in the DMB in the ACBLIB member *acbmbr* differ from those in the DEDB Pointer Checker SRDA13 records.

System action

FABBME1 ends with a User Abend 3632.

User response

Make sure that the value for the DBDNAME parameter on the user-supplied control statement is correct, and that the specification of the ACBLIB DD statement identifies the correct IMS ACB library data set. Check that the DBDGEN and ACBGEN were properly specified and performed. Verify that the MEDARI DD statement specifies the correct data sets. Correct any errors, and rerun the job.

FABB3649E	INSUFFICIENT STORAGE AVAILABLE

Explanation

Program FABBME1 issued a GETMAIN to acquire storage for internal control blocks. The return code specifies that the attempt was unsuccessful.

System action

FABBME1 ends with a User Abend 3649.

User response

Increase the region size parameter on the EXEC statement for FABBME1. Rerun the job.

FABB3650E	"DEVTYPE" FAILED FOR DDNAME *ddname* (RC = *xx*)

Explanation

Program FABBME3 issued an SVC 24 (DEVTYPE) to obtain information about the input/output device associated with *ddname*. The return code, as shown in the generated message, specifies that the attempt to do so was unsuccessful.

System action

FABBME3 ends with a User Abend 3650.

User response

Make sure that a "ddname" DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3651E	"OPEN" FAILED FOR DDNAME "SYSPRINT"

Explanation

Program FABBME3 attempted to OPEN the file associated with the indicated DD. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME3 ends with a User Abend corresponding to the message number.

User response

Make sure that the DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3652E	"OPEN" FAILED FOR DDNAME "MEDARI"

Explanation

Program FABBME3 attempted to OPEN the file associated with the indicated DD. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME3 ends with a User Abend corresponding to the message number.

User response

Make sure that the DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3653E	"OPEN" FAILED FOR DDNAME "MEDAR12O"

Explanation

Program FABBME3 attempted to OPEN the file associated with the indicated DD. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME3 ends with a User Abend corresponding to the message number.

User response

Make sure that the DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3654E	"OPEN" FAILED FOR DDNAME "SORT12"

Explanation

Program FABBME3 attempted to OPEN the file associated with the indicated DD. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME3 ends with a User Abend corresponding to the message number.

User response

Make sure that the DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3655E	"OPEN" FAILED FOR DDNAME "SORT3"

Explanation

Program FABBME3 attempted to OPEN the file associated with the indicated DD. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME3 ends with a User Abend corresponding to the message number.

User response

Make sure that the DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3656E	"OPEN" FAILED FOR DDNAME "MEDAR3O" - DD STATEMENT NOT FOUND - RECFM INCORRECT (S/B: VB) - LRECL TOO SMALL (DO NOT SPECIFY) - BLKSIZE TOO SMALL (GIVE US A BREAK...) - FAILED BY OPERATING SYSTEM

Explanation

Program FABBME3 attempted to OPEN the file associated with the indicated DD. The attempt was unsuccessful. (If possible, a reason for the failure is also given.)

System action

FABBME3 ends with a User Abend corresponding to the message number.

User response

Make sure that the DD statement has been provided that correctly specifies the proper data set. Correct any errors, and rerun the job.

FABB3658E	INCORRECT INPUT SEQUENCE

Explanation

The DEDB Tuning Aid records in the file associated with the MEDARI DD statement were not in the correct sequence.

System action

Program FABBME3 ends with a User Abend 3658.

User response

Make sure that the MEDARI DD statement is correctly specified, and that the SORT in step SORT13 performed successfully. Correct any errors, and rerun the job.

FABB3659E	INSUFF. STORAGE FOR: $aaaaaaaa$ - INCREASE REGION SIZE

Explanation

Program FABBME3 issued a GETMAIN to acquire storage for internal control blocks $aaaaaaaa$. The return code specified that the attempt was unsuccessful.

System action

FABBME3 ends with a User Abend 3659.

User response

As indicated in the message, increase the region size parameter on the EXEC statement for FABBME3. Rerun the job.

FABB3660E	OUT OF SPACE WHILE ATTEMPTING TO ALLOCATE AN IOVF CI

Explanation

Program FABBME3 attempted to pseudo insert a segment into the area model. The UOW being processed required an additional IOVF CI. None were available; the area is full.

System action

FABBME3 ends with a User Abend 3660.

User response

Reevaluate the database specifications, and re-tune the database.

FABC messages

The following information is about messages and codes that begin with FABC.

FABC0100I   **FABCUR1 ENDED NORMALLY**

Explanation

This message is generated when all requested processing has been completed without errors.

System action

Program FABCUR1 ends with a return code of 0.

User response

None. This message is informational.

FABC0100W   **FABCUR1 ENDED WITH WARNINGS**

Explanation

This message is generated when trivial error conditions were encountered by program FABCUR1.

System action

FABCUR1 ends with a return code of 4.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR1. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0100E   **FABCUR1 ENDED WITH ERRORS**

Explanation

This message is generated when nontrivial error conditions were encountered by program FABCUR1.

System action

FABCUR1 ends with a return code of 8.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR1. Correct the problem, and rerun the job.

FABC0101I   **DATA SET UNLOADED FOR AREA *zzzzz* (AREA NAME: *areaname*) - DD NAME: *ddname* DS NAME: *dsname***

Explanation

This message is generated when program FABCUR1 selects the area data set specified for unloading the area.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0105I   **PROCESSING COMMENCES FOR AREA *zzzzz* (DDNAME: *ddname*) [(BUFND = *zzz*) | (ACCESS = FAST)]**

Explanation

This message is generated when program FABCUR1 dispatches an unload subtask to process the specified area. BUFND = parameter is the value used when the area is opened. If ACCESS = FAST was specified, ACCESS = FAST is displayed instead of BUFND = parameter value.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0106I   **PROCESSING COMPLETED FOR AREA *zzzzz* (DDNAME: *ddname*) [AREA IS EMPTY | (*n1 / n2*)]**

Explanation

This message is issued when an unload subtask notifies program FABCUR1 that the specified area has been successfully unloaded. If the area is empty, the message text that specifies this is issued. *n1* is the user-task ID. *n2* is the number of times that FABCUR1

is put into a wait state while the database segment records are written.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0107W	NO SEGMENTS WILL BE RELOADED TO AREA *zzzzz* (AREANAME: *areaname*)

Explanation

No segment will be reloaded to the specified area.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

None.

FABC0108W	UNLOADED DATA FOR AREA *zzzzz* (AREA NAME: *areaname*) MAY BE WRONG - DUE TO *reason*

Explanation

When unload operation completed for the area specified, DBRC gives one of the following reasons that implies that the unloaded data may be wrong.

- RECOVERY NEEDED STATUS: By the IMS online system or by the DBRC batch command the area was made recovery needed during unload processing. Existence of EEQE: An EEQE has been created by the IMS online system during unload processing.

- CONCURRENT UPDATE: The IMS online system accessed the area with update intent during unload processing.

- AREA DROPPING FROM DBRC: The area registration was dropped by the DBRC batch command from DBRC during unload processing.

- ADS UNAVAILABLE STATUS: The ads were made unavailable during unload processing by the IMS online system or by the DBRC batch command.

- ADS DROPPING FROM DBRC: During unload processing, the ads registration was dropped by the DBRC batch command from DBRC.

System action

Program FABCUR1 sets an end-of-job return code 4 and continues processing.

User response

Make sure that there was no update operation by the IMS online system during unload operation. If there is a possibility that the area was updated during unload processing, then rerun the job for the area specified.

FABC0109I	AREA *zzzzz* (AREANAME: *areaname*) IS NOT REGISTERED IN DBRC

Explanation

Program FABCUR1 found that the specified area was not registered in DBRC.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0110W	NO RECORDS WRITTEN TO DDNAME DURD*zzz*O / XD*zzzzz*O

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Attempt to determine if there should have been any segment data records written to the specified output file. Verify that the DD statement NEWACB correctly identifies the proper data set, and that the DBDGEN and ACBGEN for the database being processed were performed correctly. Check that the randomizer module is specified correctly. Review the FILECTL specifications, if any. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0111I	DBRC=Y IS SPECIFIED

Explanation

DBRC=Y is specified in the EXEC parameter of the FABCUR1 JCL. Program FABCUR1 will establish DBRC interface and obtain area information from DBRC.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0112I   **DBRC=N IS SPECIFIED - EEQE DETECTION NOT PERFORMED**

Explanation

DBRC=N is specified in the EXEC parameter of the FABCUR1 JCL. Program FABCUR1 does not establish DBRC interface and does not check the existence of EEQEs.

System action

FABCUR1 continues processing.

User response

If the area is registered in DBRC, get DBRC RECON list and make sure there are no EEQEs registered in DBRC for the area. If there is an EEQE for the area, recover the area and rerun the job.

FABC0113I   **AREA *zzzzz* (AREANAME: *areaname*) DDNAME: *ddname* IS UNAVAILABLE IN DBRC**

Explanation

Program FABCUR1 found that the specified area data set was unavailable in DBRC.

System action

FABCUR1 ignores the area data set and continues processing.

User response

Get a LIST.RECON output report, and specify an unused area data set name. Then, specify the name in the DARVSAM DD statement, and rerun the job.

FABC0114I   **AREA *zzzzz* (AREANAME: *areaname*) DDNAME: *ddname* NOT SAME DSNAME BETWEEN DD STATEMENT AND DBRC**

Explanation

Program FABCUR1 found that the area data set dsname specified in ddname DD statement was not same one registered in DBRC.

System action

FABCUR1 ignores the area data set and continues processing.

User response

Get a LIST.RECON output report, and specify an unused area data set name. Then, specify the name in the DARVSAM DD statement, and rerun the job.

FABC0115I   **SCHEDULING PARAMETERS: NO. UTASK'S : *zz9* UOW BFR SIZE: *z,zzz,zz9* BYTES**

Explanation

This message is generated, when the STATS keyword is specified on the DBDNAME control statement, to detail the parameters being used to dispatch and manage the unload subtasks.

System action

Program FABCUR1 continues processing.

User response

None. This message is informational.

FABC0116W   **UTASK *zzz* TERMINATED DUE TO STORAGE CONSTRAINTS**

Explanation

Program FABCUR1 issued an OPEN for the ACBs associated with the next area to be processed, when preparing to dispatch an unload subtask to unload it. The return code from VSAM specified that the request failed because of insufficient storage being available for the required buffers and control blocks.

System action

FABCUR1 sets an end-of-job return code of 4, dispatches the specified unload subtask with a control code indicating that it should terminate itself, and continues processing.

User response

Review the unload region-size calculations (especially if BUFND overrides are being used on the area data

sets). Check that the REGION= parameter is coded correctly on the EXEC statement for FABCUR1.

FABC0117I	EXIT ROUTINE *exitname* "END" CALL FINISHED -first 80 bytes characters of the message that user exit routine returned -subsequent 48 bytes characters of the message that user exit routine returned

Explanation

Program FABCUR1 called the user exit routine *exitname* with "END" call and the exit routine returned the message specified.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0120I	CARD *xx: zzzz...zzzz*

Explanation

This message is generated to show the control statement currently being processed.

System action

Program FABCUR1 continues processing.

User response

None. This message is informational.

FABC0121W	ERROR DETECTED NEAR COLUMN *xx*

Explanation

Program FABCUR1 detected an error in the control statement currently being processed. (See the immediately preceding FABC0120I message.)

System action

FABCUR1 continues processing, and issues one or more other FABC01*xx* messages.

User response

To determine the nature and causes of the errors detected, see the other messages generated by

FABCUR1. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0121E	ERROR DETECTED NEAR COLUMN *xx*

Explanation

See message number FABC0121W.

System action

See message number FABC0121W.

User response

See message number FABC0121W.

FABC0122W	BLANK/INVALID CONTROL CARD

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an end-of-job return code of 4, and continues processing.

User response

Remove the specified control statement in subsequent executions of FABCUR1.

FABC0123E	UNKNOWN KEYWORD

Explanation

Program FABCUR1 encountered a control statement with a value starting in column one that is not one of the valid control statements.

System action

FABCUR1 ends with an abend code of 3728.

User response

Correct, or remove, the specified control statement.

FABC0123W	UNKNOWN KEYWORD

Explanation

Program FABCUR1 encountered a control statement with a value starting in column one that is not one of the valid control statement types.

System action

FABCUR1 ends with an abend code of 3728.

User response

Correct, or remove, the specified control statement. Rerun the job.

FABC0125E   **1ST CONTROL CARD NOT DBDNAME= CARD**

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3728.

User response

The control statement stream must include one DBDNAME control statement, and it must be the first statement in the stream. Correct the control statement stream. Rerun the job.

FABC0126E   **MULTIPLE DBDNAME= CARDS ENCOUNTERED**

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3728.

User response

The control statement stream must include only one DBDNAME control statement, and it must be the first statement in the stream. Correct the control statement stream. Rerun the job.

FABC0127I   **- FOLLOWING VALUES ARE DEFINED BY SITE DEFAULT TABLE (*xxxxxxxx*)**
                  - *keyword=value*
                  - *keyword=value*

Explanation

This message is generated to show the site default table (FABCOP1D/FABCOP3D/FABCOP6D/FABCOP9D) being processed.

System action

Program FABCUR1/FABCUR3/FABCUR6/FABCUR9 uses the values as the default values for the control statement, and continues processing.

User response

None. This message is informational.

FABC0130E   **INVALID DBDNAME= CONTROL CARD SYNTAX ERROR DETECTED**

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the DBDNAME control statement. Correct the errors, and rerun the job.

FABC0131E   **INVALID DBDNAME= CONTROL CARD DBDNAME MISSING/INVALID**

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the DBDNAME control statement. Correct the error. Rerun the job.

FABC0133E   **INVALID DBDNAME= CONTROL CARD "REORG" AND "HIERCHNG"/ "RMODTYPE"/"NEWDBDNM" KEYWORDS ARE MUTUALLY EXCLUSIVE**

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the DBDNAME control statement. Correct the error. Rerun the job.

FABC0135E	INVALID AREACTL= CONTROL CARD AREA NO(S) SPECIFICATION MISSING/INVALID

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the AREACTL control statement. Correct the error. Rerun the job.

FABC0136W	INVALID AREACTL= CONTROL CARD AREA *xxxxx* PREVIOUSLY SPECIFIED

Explanation

Self-explanatory. *xxxxx* is the area number.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Remove the duplicate specification for the specified area in subsequent executions of FABCUR1.

FABC0137W	INVALID AREACTL= CONTROL CARD HIERCHNG=YES SPECIFIED; AREACTL=ALL REQUIRED

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Specify AREACTL=ALL or remove HIERCHNG=YES specification on the control statement in subsequent executions of FABCUR1.

FABC0140E	INVALID FILECTL= CONTROL CARD FILE NO(S) SPECIFICATION MISSING/INVALID

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the FILECTL control statement. Correct the control statement stream, and rerun the job.

FABC0141E	INVALID FILECTL= CONTROL CARD AREA *zzzzz* PREVIOUSLY SPECIFIED

Explanation

Self-explanatory. *zzzzz* is the area number.

System action

Program FABCUR1 discards the specification for the specified area, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the FILECTL control statement. Correct the error, and rerun the job.

FABC0142E	INVALID FILECTL= CONTROL CARD FILE *zzzzz* PREVIOUSLY SPECIFIED

Explanation

Self-explanatory. *zzzzz* is the file number.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the FILECTL control statement. Correct the error, and rerun the job.

FABC0143E	INVALID FILECTL= CONTROL CARD FILECTL=[(*) | ALL] PREVIOUSLY SPECIFIED

Explanation

Program FABCUR1 detected a FILECTL control statement after having received the specified FILECTL specification.

System action

FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the FILECTL control statement. Correct the error, and rerun the job.

FABC0145E	INVALID TASKCTL= CONTROL CARD ERROR IN [NO. SUB-TASKS | NO.IOVF BUFFERS | UOW BUFFER SIZE] SPECIFICATION

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the TASKCTL control statement. Correct the error, and rerun the job.

FABC0146W	TASKCTL= CONTROL CARD PREVIOUSLY PROCESSED PREVIOUS [NO. SUB-TASKS | NO.IOVF BUFFERS | UOW BUFFER SIZE] VALUE DISCARDED

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Remove the duplicate specification in subsequent executions of FABCUR1.

FABC0147E	INVALID FORMAT= CONTROL CARD SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, set an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for syntactical details of the FORMAT control statement. Correct the error, and rerun the job.

FABC0148E	INVALID EXITRTN= CONTROL CARD SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, set an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for syntactical details of the EXIT control statement. Correct the error, and rerun the job.

FABC0149E	INVALID EXITRTN= CONTROL CARD EXITRTN CONTROL CARD PREVIOUSLY PROCESSED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0150E	INVALID LOADCTL= CONTROL CARD SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the LOADCTL control statement. Correct the error, and rerun the job.

FABC0151W	LOADCTL= CONTROL CARD PREVIOUSLY PROCESSED PREVIOUS LOADCTL SPECIFICATION FOR SEGMENT *segname* DISCARDED

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Remove the duplicate control statement in subsequent executions of FABCUR1.

FABC0152W	LOADCTL SPECIFICATION FOR [ROOT	SDEP] SEGMENT (*segname*) IGNORED

Explanation

LOADCTL cannot be specified for ROOT or SDEP segments.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Remove the specified control statement in subsequent executions of FABCUR1.

FABC0153E	INVALID LOADCTL= CONTROL CARD TOO MANY LOADCTL SPECIFICATIONS ENCOUNTERED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for details on the syntax of the LOADCTL control statement. Correct the errors, and rerun the job.

FABC0154I	FABCUR1 NOT APF AUTHORIZED PROGRAM; ACCESS=FAST IGNORED

Explanation

Program FABCUR1 found that the IMS HP Fast Path Utilities load module library data set specified on the JOBLIB/STEPLIB DD statement was not authorized by APF. To invoke the ACCESS=FAST function, the load module FABCUR1 must be on the APF authorized library.

System action

FABCUR1 ignores the ACCESS=FAST request and continues processing with the ACCESS=VSAM option.

User response

Make the IMS HP Fast Path Utilities load module library APF authorized for future processing.

FABC0155W   *xxxxxxxxx*; NEWDBDNM= KEYWORD IGNORED

Explanation

Program FABCUR1 found a DB name change requirement, but the NEWACB DD statement was not present or the IMSCATACB_OUTPUT keyword is not specified. An additional message FABC0164I is printed.

System action

FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Correct the control statement stream in subsequent executions of FABCUR1.

FABC0156E   INVALID PTRERROR= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for the syntactical details of the PTRERROR control statement. Correct the error, and rerun the job.

FABC0157W   PTRERROR= CONTROL CARD PREVIOUSLY PROCESSED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Unload" on page 130 for the syntactical details of the PTRERROR control statement. Correct the error, and rerun the job.

FABC0158E   INVALID KEYSEQERROR= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

Correct the error, and rerun the job.

FABC0159W   KEYSEQERROR= CONTROL CARD PREVIOUSLY PROCESSED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal error flag, and continues processing.

User response

Correct the error, and rerun the job.

FABC0160W   *xxxxxxxx*; HIERCHNG=YES/ YESFORCE IGNORED

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Correct the control statement stream in subsequent executions of FABCUR1. The HIERCHNG keyword of the DBDNAME control statement should be specified with a value of YES/YESFORCE only when:

1. A change to the segment hierarchy is desired, *and*

2. A DMB reflecting that change is provided in the file associated with the NEWACB DD statement or the pending ACB definition in the IMS directory.

FABC0161W HIERCHNG=YES/YESFORCE SPECIFIED; NO STRUCTURE CHANGES DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Correct the control statement stream in subsequent executions of FABCUR1. The HIERCHNG keyword of the DBDNAME control statement should be specified with a value of YES/YESFORCE only when:

1. A change to the segment hierarchy is desired, *and*
2. A DMB reflecting that change is provided in the file associated with the NEWACB DD statement.

FABC0162W NEWDBDNM= SPECIFIED; NO DB NAME CHANGES DETECTED

Explanation

Program FABCUR1 found a DB name change requirement, but the new DB name was equal to the old one.

System action

FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

Correct the control statement stream in subsequent executions of FABCUR1. The NEWDBDNM keyword of the DBDNAME control statement should be specified only to change the name of the DB and to reload.

FABC0163W *xxxxxxxx*; IGNORED DUE TO "REORG" KEYWORD

Explanation

Program FABCUR1 detected the presence of a NEWACB DD statement or an IMSCATACB_OUTPUT keyword. When REORG is specified for the DBDNAME control statement, NEWACB DD statement and IMSCATACB_OUTPUT keyword are ignored.

System action

FABCUR1 continues processing in "reorg" mode (see).

User response

Review your JCL and control statements to verify that "reorg" mode is what you want. If you really have a new DBD, you want "change" mode. In this case, remove *REORG* from your control statement, and rerun the job.

FABC0164I *xxxxxxxx*; "REORG" MODE ASSUMED

Explanation

Program FABCUR1 issued an SVC 24 (DEVTYPE) specifying the ddname NEWACB. The return code specified that such a DD statement was not present in the JCL stream, hence REORG mode processing was assumed. If IMS managed ACBs are used and the IMSCATACB_OUTPUT keyword is not specified, the program runs the job in REORG mode.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

**FABC0165I STRUCTURE CHANGE DETECTED FOR SEGMENT *segname*
- "OLD" SEG-CD: *zz9*
- "NEW" SEG-CD: *zz9***

Explanation

Program FABCUR1 determined that the specified segment was not defined in the same place in the hierarchical structure in the DMB from the NEWACB ACB library as it was in the DMB from the OLDACB file.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

**FABC0166I DBD SPECIFICATION CHANGE(S) DETECTED
- NO. AREAS IN "OLD" DMB: *zz9*
- NO. AREAS IN "NEW" DMB: *zz9***

Explanation

Program FABCUR1 determined that the number of areas defined in the DMB from the NEWACB ACB library differed from the number defined in the DMB from the OLDACB file.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0168I   **DATABASE NAME CHANGED**

Explanation

Program FABCUR1 found DB name change requirement. FABCUR1 creates the unload data sets with the DMB from the NEWACB library.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0169I   **AUTHORIZED STRUCTURE CHANGE(S) DETECTED**

Explanation

HIERCHNG=YES/YESFORCE was specified on the DBDNAME control statement and program FABCUR1 detected one or more structure changes.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0170W   **I/O ERROR FOR OUTPUT DATA SET DDNAME *ddname1*, UNLOAD PROCESS CONTINUES WITH DATA SET DDNAME *ddname2***

Explanation

Program FABCUR1 issued a PUT for the *ddname1* specified. The PUT operation failed.

System action

FABCUR1 continues processing with the *ddname2* specified.

User response

None.

FABC0171W   **HIERCHNG=YESFORCE SPECIFIED; AREA(S) NOT SPECIFIED MUST BE UNLOADED BY ANOTHER JOB(S).**

Explanation

HIERCHNG=YESFORCE and RMODTYPE=S are specified but AREACTL=ALL is not specified so that only AREAs specified on the AREACTL control statement will be unloaded. AREAs of the DEDB which are not specified must be unloaded subsequently by other jobs.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing to unload AREAs specified on the AREACTL control statement.

User response

After finishing the job normally, another unload job for remaining AREAs of the DEDB which are not unloaded should be run subsequently. Otherwise the integrity of the DEDB between the AREAs will be lost.

FABC0172W   **NEWDBDNM=*acbname* SPECIFIED; AREA(S) NOT SPECIFIED SHOULD BE UNLOADED BY ANOTHER JOB(S)**

Explanation

NEWDBDNM=*acbname* and RMODTYPE=S are specified but AREACTL=ALL is not specified so that only areas specified on the AREACTL control statement are unloaded. Other jobs should unload the areas of the DEDB that are not specified.

System action

Program FABCUR1 sets on end-of-job return code of 4, and continues processing to unload areas specified on the AREACTL control statement.

User response

When you finish the job normally, run another unload job for remaining Areas of the DEDB.

FABC0173W   **LRECL FOR DDNAME** *ddname* **IS OVERRIDDEN BY SYSTEM DETERMINED VALUE - SPECIFIED BLKSIZE TOO SMALL OR TOO LARGE - LRECL SPECIFIED:** *xxxxx* **- LRECL OVERRIDDEN:** *xxxxx*

Explanation

This message is issued when LRECL is specified in JCL but its value is either smaller than the minimum or larger than the maximum tolerance level.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing.

User response

None.

FABC0174E   **- INCORRECT SEGMCTL= CONTROL CARD - SYNTAX ERROR DETECTED - "ALL" AND "SEGMENT NAME" PARAMETERS ARE MUTUALLY EXCLUSIVE - SEGMENT NAMES IN EXCESS OF 127 ARE SPECIFIED**

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0174E   **- INCORRECT SEGMCTL= CONTROL CARD - SDEP SEGMENT** *segname* **IS SPECIFIED AGAINST SDEP=NO| PHYSICAL ON DBDNAME CONTROL STATEMENT - SEGMENT:** *segname* **IS NOT DEFINED IN DMB** *xxxxxxxx* **FROM** *nnnnnnnn*

Explanation

Program FABCUR1 found an incorrect *segname* in the SEGMCTL statement.

System action

FABCUR1 ends with an abend code 3728.

User response

Correct the *segname* in the SEGMCTL statement, and rerun the job.

FABC0175W   **SEGMCTL= CONTROL CARD PREVIOUSLY PROCESSED FOR SEGMENT** *segname*

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing. The specified segment is extracted.

User response

Remove the duplicate *segname* in the subsequent executions of FABCUR1.

FABC0176W   **SEGMENT** *segname* **IS IGNORED BECAUSE OF SDEP=NO|PHYSICAL ON DBDNAME CONTROL CARD**

Explanation

Self-explanatory.

System action

Program FABCUR1 sets an end-of-job return code of 4, and continues processing. The specified segment is not extracted.

User response

Correct the *segname* or the SDEP= option on the DBDNAME control statement in the subsequent executions of FABCUR1.

FABC0179I   **USER EXIT FABC1IE0 IS CALLED**

Explanation

User exit routine FAB11IE0 is called.

System action

Program FABCUR1 continues processing.

User response

None. This message is informational.

FABC0180E   **- INVALID LOADPLACE= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal flag, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0181E   **- LOADPLACE= CONTROL CARD PREVIOUSLY PROCESSED**

Explanation

Self-explanatory.

System action

Program FABCUR1 discards the control statement, sets an internal flag, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0182E   **INCORRECT LRECL= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

An incorrect LRECL was detected.

System action

Program FABCUR1 discards the control statement, sets an internal flag, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0183W   **LRECL= CONTROL CARD PREVIOUSLY SPECIFIED**

Explanation

The LRECL= control card has been already specified.

System action

Program FABCUR1 discards the control statement, sets an internal flag, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0185I   **RMODTYPE=S BUT RANDOMIZED TO ANOTHER AREA**

Explanation

RMODTYPE=S is specified, but the randomizer randomized a record to another area that is different from the original one.

System action

Program FABCUR1 continues processing.

User response

None. This message is informational.

FABC0186E   *keyword1=value1* **IS NOT ALLOWED**
- *keyword1=value1* **AND** *keyword2=value2* **ARE EXCLUSIVE**

Explanation

Keywords *keyword1* and *keyword2* cannot be specified together.

System action

FABCUR1 ends with an abend code of 3728.

User response

Correct or remove the indicated control statement.

FABC0187I   **INFORMATION OF THE DB DEFINITION WAS OBTAINED FROM** *resource*

Explanation

This message indicates the resource (ACB library or IMS directory) where FABCUR1 obtained DMB definitions from.

System action

FABCUR1 continues processing.

User response

None. This message is informational.

FABC0188E   *keyword*= **CONTROL CARD PREVIOUSLY PROCESSED**

Explanation

The indicated keyword cannot be specified more than once in a control statement.

System action

FABCUR1 ends with an abend code of 3728.

User response

Remove the duplicate specification and rerun the job.

FABC0189E   **INVALID** *keyword*= **CONTROL CARD**

Explanation

Program FABCUR1 encountered an invalid specification while parsing the user-supplied control statement.

System action

FABCUR1 ends with an abend code of 3728.

User response

Correct the error and rerun the job.

FABC0300I   **PROCESSING STARTED FOR AREA** *zzzzz* **(AREANAME** *areaname*)

Explanation

This message is generated as reload processing starts for each area.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0301I   **PROCESSING COMPLETE FOR AREA** *xxxxx* **(AREANAME** *areaname*) **[AREA IS EMPTY]**

Explanation

This message is generated as reload processing successfully completes for the area. When this message is generated, the area has been successfully reloaded, and the data set has been closed. If the area continues no segment, the message text that specifies this is issued.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0302I   **AREA INITIALIZATION STATISTICS - DBDNAME:** *dbdname* **AREANAME:** *yyyy*
- ROOT PORTION:
CI'S/UOW ADDRESSABLE BY RMOD: *zzzz9*
CI'S/UOW USED AS OVERFLOW: *zzzz9*
TOTAL CI'S/UNIT OF WORK: *zzzz9*
TOTAL UOW'S: *zzzzzzzz9*
- INDEPENDENT OVERFLOW PORTION:
TOTAL CI'S: *zzzzzzzz9*
- SEQUENTIAL DEPENDENT PORTION:
TOTAL CI'S: *zzzzzzzz9*
- SEQUENTIAL DEPENDENT PORTION: (ADS*n*: ads*xxx*)
- TOTAL CI'S: *zzzzzzzz9*

Explanation

Program FABCUR3 does the area initialization function normally performed by the DEDB Initialization utility (DBFUMIN0). This message provides the area statistics in the same format as the statistics provided by DBFUMIN0. For DDNAME control statements on the area*xxx* DD, the message, 'SEQUENTIAL DEPENDENT PORTION: (ADS*n*: ads*xxx*)' for each area data set is issued.

System action

FABCUR3 continues processing.

User response

None. This message is informational.

FABC0303I   **VSAM "BUFND" VALUES:**
- AREA (SEQ): *zz9* **(OVERRIDDEN FROM** *zz9*)

- ADS*n* (SEQ): *zz9* (OVERRIDDEN FROM *zz9*)
- WORK (SEQ): *zz9* (OVERRIDDEN FROM *zz9*)
- WORK (DIR): *zz9*
- DATA SPACE USED FOR WORK

Explanation

This message provides statistics on the BUFND values used for the area and work data sets. Program FABCUR3 calculates default values for the number of sequential buffers. The number of direct buffers is provided by the IOVFBUF= keyword on the control statement (default = 4).

System action

FABCUR3 continues processing.

User response

None. This message is informational.

FABC0304I	NUMBER OF UNUSED IOVF CI'S: *nnnnnnnnn*

Explanation

This message provides statistics of unused IOVF CIs. The total number of unused IOVF CIs is specified by *nnnnnnnnn*.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0305I	RELOAD SUCCESSFULLY COMPLETED FOR ADS*n*: ads*xxx*

Explanation

This message is issued as reload processing successfully completes for the area data set on the ads*xxx* DD card. This message is generated only when area data sets control statements are specified on the area*xxx* DD card.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0305W	RELOAD BYPASSED FOR ADS*n*: ads*xxx*

Explanation

This message is issued when one of the area data sets ads*xxx* in area*xxx* was not reloaded successfully. The processing for the specified area data set has been bypassed, but other ADS for the same area area*xxx* is continuing.

System action

Program FABCUR3 continues processing.

User response

Copy data to the area data set on the specified ads*xxx* DD from the area data set that was reloaded successfully.

FABC0306I	AREA *zzzzz* (AREANAME: *areaname*) IS NOT REGISTERED IN DBRC

Explanation

Program FABCUR3 found that the specified area was not registered in DBRC.

System action

FABCUR3 continues processing.

User response

None. This message is informational.

FABC0307W	DATA FOR AREA *zzzz9* BYPASSED

Explanation

This message is only generated when the STARTAREA keyword is present (that is, in restart situations). The message is issued for each area with data that is being bypassed. *zzzz9* is the area number.

System action

Program FABCUR3 sets an end-of-job return code of 4, and continues processing.

User response

None.

FABC0308I	SDEP SEGMENTS ARE RELOCATED WITH SDEPRELOCATE=YES OPTION

Explanation

This message is generated when SDEP Relocation starts for each area.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0309W   **IOVF INTERVAL AT RBA:** *xxxxxxxx* **IS FULL**

Explanation

Program FABCUR3 attempted to allocate an IOVF CI from an overflow unit, but found that all CIs within the overflow unit were in use. *xxxxxxxx* is the RBA of the associated directory entry.

System action

FABCUR3 issues a warning message, sets an end- of-job return code of 4, and sequentially searches the overflow directory entries to find an overflow unit with an available IOVF CI.

User response

This situation can seriously impact the online performance of the database. The database probably requires expansion and performance tuning.

FABC0310I   **FABCUR3 ENDED NORMALLY**

Explanation

This message is generated on completion of processing by program FABCUR3. See message number FABC0310W. Also, review the other generated messages.

System action

FABCUR3 ends with a return code of 0 or 4.

User response

None. This message is informational.

FABC0310W   **FABCUR3 ENDED WITH WARNINGS**

Explanation

This message is generated on completion of processing by program FABCUR3. The "W" level

message denotes that trivial errors were encountered. Review the other generated messages, especially message number FABC0310I.

System action

FABCUR3 ends with a return code of 0 or 4.

User response

None.

FABC0310E   **FABCUR3 ENDED WITH ERRORS - RELOADED FOR ALL AREAS SUCCESSFULLY BUT SOME ADS(S) BYPASSED**

Explanation

This message is issued when the reload processing is completed successfully, but at least one process for area data set is bypassed. The detail of the cause is shown in the other error message for the area data set.

System action

Program FABCUR3 ends with return code of 8.

User response

Copy data to the bypassed area data set from the area data set that was reloaded successfully.

FABC0311I   **DBRC=Y IS SPECIFIED**

Explanation

DBRC=Y is specified in the EXEC parameter of the FABCUR3 JCL. Program FABCUR3 will establish a DBRC interface and obtain area information from DBRC.

System action

FABCUR3 continues processing.

User response

None. This message is informational.

FABC0312I   **DBRC=N IS SPECIFIED**

Explanation

DBRC=N is specified in the EXEC parameter of the FABCUR3 JCL. Program FABCUR3 does not establish DBRC interface.

System action

FABCUR3 continues processing.

User response

None. This message is informational.

FABC0313I   **CARD** *xx: zzzz...zzzz*

Explanation

This message is shows the control statement currently being processed.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0315I   *zzz,zzz,zz9* **SEGMENT RECORDS OF INSERT LIMIT COUNT IN UNLOADED FILE**

Explanation

The Insert Limit Count (ILC) records were detected. These records are ignored in the key field sequence check.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0316W   **SUMMARY OF KEY SEQUENCE ERRORS
- NUMBER OF RELATED DB RECORDS:** *zzz,zzz,zz9*
- NUMBER OF THE SEGMENTS DETECTED AS KEY SEQUENCE ERROR: *zzz,zzz,zz9*
- ERROR SEGMENTS SUM TOTAL INCLUDING CHILD SEGMENTS: *zzz,zzz,zz9*

Explanation

This message means the number of the error segments of key field sequence check. This message is issued only when KEYSEQERROR=BYPASS is specified in the SYSIN DD control statement, and when

FABCUR3 detected the error segments of key field sequence check.

System action

Program FABCUR3 sets an end-of-job return code of 4, and continues processing.

User response

Verify the unloaded file in the DURDATA DD statement. Correct the problem, and if necessary, rerun the unload job.

FABC0317I   **EXIT ROUTINE** *exitname* **"END" CALL FINISHED
- first 80 bytes characters of the message that user exit routine returned
- subsequent 48 bytes characters of the message that user exit routine returned**

Explanation

Program FABCUR3 called the user exit routine *exitname* with "END" call and the exit routine returned the message specified.

System action

FABCUR3 continues processing.

User response

None. This message is informational.

FABC0318I   **SDEPRELOCATE=YES OPTION IS IGNORED**

Explanation

Program FABCUR3 has been ignored for one of the following reasons:

- The unloaded file that was used as the input of Reload processing was not unloaded with the SDEP=PHYSICAL option.
- The SDEP Logical Begin and Logical End pointers in the second CI of the original area were not physically inverted.
- The target area has insufficient SDEP space for SDEP Relocation.

System action

FABCUR3 will not relocate SDEP pointers, and continues processing.

User response

Check the condition for SDEP relocation; if necessary, correct the condition; rerun the job.

FABC0320E **UNKNOWN KEYWORD (NEAR COLUMN *xx*)**

Explanation

Program FABCUR3 was searching for the start of a "keyword=value" specification on the control statement. At column *xx* of the statement, an unknown keyword was encountered.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0321E **"STARTAREA=" VALUE INVALID**

Explanation

Program FABCUR3 encountered a STARTAREA keyword whose associated parameter value is missing or not numeric.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0322E **"IOVFBUF=" VALUE INVALID**

Explanation

Program FABCUR3 encountered an IOVFBUF= keyword whose associated parameter value is missing or not numeric.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0323W **"IOVFBUF=" LESS THAN MIN. REQUIRED - DEFAULT ASSUMED**

Explanation

Program FABCUR3 encountered an IOVFBUF= keyword whose associated parameter value was less than the minimum acceptable value (default = 4).

System action

FABCUR3 assumes the default minimum, sets an end-of-job return code of 4, and continues processing.

User response

None.

FABC0324E **"TBLENTRY=" VALUE INVALID**

Explanation

Program FABCUR3 encountered a TBLENTRY= keyword whose associated parameter value is missing or not numeric.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0325E **"EXITRTN" VALUE INVALID**

Explanation

Self explanatory.

System action

Program FABCUR3 discards the control statement, sets an internal error flag, and continues processing.

User response

See "Input for DEDB Reload" on page 172 for details on the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0326E **"EXITRTN=" PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

| FABC0327I | - "SDEPRELOCATE=YES" IS IGNORED DUE TO AREC=N |

Explanation

Program FABCUR3 found SDEPRELOCATE=YES in the SYSIN DD statement when AREC=N was specified on the EXEC parameter.

System action

FABCUR3 ignores this keyword and will not relocate SDEP pointers. The reload process continues.

User response

Check the condition for SDEP relocation. If necessary, correct the condition and rerun the job.

| FABC0328I | SDEP SEGMENTS ARE RELOCATED DUE TO SDEP=PHYSICAL SPECIFIED WITH *xxxxxxxx* |

Explanation

Locations of each SDEP segments are changed from the original RBA because the segments were unloaded with SDEP=PHYSICAL accompanied by a DEDB change with NEWACB or the IMSCATACB_OUTPUT keyword.

System action

Program FABCUR3 will relocate the RBA of each SDEP segments, associating RBA of PCF pointer in their root segments, and PTF pointer of their twin segments.

User response

The RBA value to identify SDEP marker is no longer used because absolute value of RBA of each SDEP segments at unload were changed in reloaded area.

| FABC0330E | *area*xxx CONTROL STATEMENT DATASET IS EMPTY |

Explanation

Self-explanatory.

System action

Program FABCUR3 ends with an abend code of 3761.

User response

Verify the correctness of the control statement on the *area*xxx DD. Correct the error, and rerun the job.

| FABC0331E | UNKNOWN KEYWORD ENCOUNTERED IN *area*xxx DATASET |

Explanation

Program FABCUR3 found that there were invalid string on the *area*xxx DD control statement.

System action

FABCUR3 ends with an abend code of 3761.

User response

Verify the correctness of the control statement on the *area*xxx DD. Correct the error, and rerun the job.

| FABC0332E | NO VALID DDNAME CONTROL STATEMENT SPECIFIED IN *area*xxx DATASET |

Explanation

Program FABCUR3 could not find valid DDNAME control statement on the *area*xxx DD.

System action

FABCUR3 ends with an abend code of 3761.

User response

Verify the correctness of the control statement on the *area*xxx DD. Correct the error, and rerun the job.

| FABC0333E | MORE THAN 7 DDNAME CONTROL STATEMENTS SPECIFIED IN *area*xxx DATASET |

Explanation

Program FABCUR3 found that there were more than 7 DDNAME control statements on the *area*xxx DD.

System action

FABCUR3 ends with an abend code of 3761.

User response

Verify the correctness of the control statement on the *area*xxx DD. Correct the error, and rerun the job.

FABC0334E   DUPLICATE DDNAME ads*xxx* SPECIFIED IN area*xxx* DATASET

Explanation

Program FABCUR3 found that duplicate DDNAME control statements were specified on one area*xxx* DD.

System action

FABCUR3 ends with an abend code of 3761.

User response

Verify the correctness of the control statement on the area*xxx* DD. Correct the error, and rerun the job.

FABC0335E   NO DDNAME SPECIFIED IN area*xxx* DATASET

Explanation

Program FABCUR3 found that neither VSAM data set nor DDNAME control statement was specified on the area*xxx* DD.

System action

FABCUR3 ends with an abend code of 3761.

User response

Verify the correctness of the control statement on the area*xxx* DD. Correct the error, and rerun the job.

FABC0336E   "IMGCPY=" VALUE INVALID

Explanation

Program FABCUR3 encountered an IMGCPY= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0342E   "ICCOMPRESS=" VALUE INVALID

Explanation

Program FABCUR3 encountered an ICCOMPRESS= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0343E   "ICHASH=" VALUE INVALID

Explanation

Program FABCUR3 encountered a ICHASH= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0344E   "IMGCPY=" PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0345E   "ICCOMPRESS=" PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0346E   "ICHASH=" PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0347E	ICHASH=YES AND/OR ICCOMPRESS=YES SPECIFIED BUT IMGCPY=YES|DUAL NOT SPECIFIED

Explanation

Program FABCUR3 found that there was no IMGCPY=YES|DUAL keyword parameter specified even though ICHASH=(YES), ICCOMPRESS=(YES), or both were specified. To invoke the ICHASH=YES option and/or the ICCOMPRESS=YES option, IMGCPY=YES|DUAL must also be specified.

System action

FABCUR3 ends with an abend code of 3761.

User response

Correct the control statement, and rerun the job.

FABC0348W	"OPEN" FAIL FOR DDNAME [HDI*xxxy*/]XI*xxxxxy* - DD STATEMENT NOT FOUND OR DUMMY SPECIFIED

Explanation

Program FABCUR3 found that a DD statement was not specified or DD DUMMY was specified for the ddname specified to create an image copy.

System action

FABCUR3 ignores image copy processing for the associating area with an area number specified by *xxx* or *xxxxx*, and continues unload operation for the succeeding areas.

User response

If image copy is required for the area, run the IMS Image Copy utility or the IBM IMS Image Copy Extensions for z/OS utility after the job is finished.

FABC0350I	IMAGE COPY PROCESSING STARTED

Explanation

Self-explanatory.

System action

Program FABCUR3 starts an image copy subtask.

User response

None. This message is informational.

FABC0351I	- IMAGE COPY PROCESSING ENDED NORMALLY

Explanation

This message is generated when all requested image copy processing are completed without errors.

System action

If there were no other FABC03*xx*W/E messages, program FABCUR3 will end with the return code of 0.

User response

None. This message is informational.

FABC0351W	- IMAGE COPY PROCESSING ENDED WITH WARNINGS RC=*xx*

Explanation

This message is generated when one or more requested image copy processing has been completed with errors. *xx* shows the highest return codes that the IBM IMS Image Copy Extensions for z/OS utility returned in message FABC0353W.

System action

If there were no other FABC03*xx*E messages, program FABCUR3 will end with a return code of 4.

User response

Follow the programmer action for message FABC0353W.

FABC0351E	- IMAGE COPY PROCESSING ENDED ABNORMALLY

Explanation

This message is generated when the program FABCUR3 main task detected critical image copy process errors, including image copy subtask abends, when finishing the reload operation for all areas.

System action

If there were no other FABC03*xx*E messages, FABCUR3 will end with a return code of 4.

User response

None.

FABC0352I	- IMAGE COPY COMPLETED FOR AREA *zzzzz* (AREANAME *areaname*) TIME STAMP - *xx...xx* COPY 1 DATASET NAME - *dd...dd* COPY 2 DATASET NAME - *dd...dd*

Explanation

Image copy processing for the area specified completed normally. The time stamp of Image copy and the image copy data set(s) are shown.

System action

Program FABCUR3 continues.

User response

None. This message is informational.

FABC0353W	- IMAGE COPY COMPLETED WITH ERRORS FOR AREA *zzzzz* (AREA NAME:*areaname*) RC=*xx*

Explanation

Image copy processing for the area specified completed, and the IBM IMS Image Copy Extensions for z/OS utility returned error return code *xx*.

System action

Program FABCUR3 continues.

User response

Follow the programmer action of any messages issued by the IBM IMS Image Copy Extensions for z/OS utility.

FABC0354E	"KEYSEQERROR=" PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0355E	"KEYSEQERROR=" VALUE INCORRECT

Explanation

Program FABCUR3 encountered a KEYSEQERROR= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0356E	"SDEPRELOCATE= OR SDEPRE=" PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0357E	"SDEPRELOCATE= OR SDEPRE=" VALUE INCORRECT

Explanation

Program FABCUR3 encountered an SDEPRELOCATE= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing

User response

Correct the control statement, and rerun the job.

FABC0360I   **USER EXIT FABC3IE0 IS CALLED**

Explanation

User exit routine FABC3IE0 is called.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0361I   **INFORMATION OF THE DB DEFINITION WAS OBTAINED FROM** *resource*

Explanation

This message is to inform which resource (DURDBDFN DD, ACBLIB DD, or the IMS directory) is used to obtain the DEDB definition information.

System action

Program FABCUR3 continues processing.

User response

None. This message is informational.

FABC0362E   **"RAPERROR=" PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0363E   **"RAPERROR=" VALUE INCORRECT**

Explanation

Program FABCUR3 encountered a RAPERROR= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0364E   **"DBDNAME=" PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory.

System action

Program FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0365E   **"DBDNAME=" VALUE INCORRECT**

Explanation

Program FABCUR3 encountered a DBDNAME= keyword whose associated parameter value is missing or incorrect.

System action

FABCUR3 sets an end-of-job return code of 8, and continues processing.

User response

Correct the control statement, and rerun the job.

FABC0368E   **"*keyword*=" PREVIOUSLY SPECIFIED**

Explanation

The indicated keyword cannot be specified more than once in a control statement.

System action

FABCUR3 ends with an abend code of 3728.

User response

Remove the duplicate specification and rerun the job.

FABC0369E   **"*keyword*=" VALUE INCORRECT**

Explanation

Program FABCUR3 encountered an invalid specification while parsing the user-supplied control statement.

System action

FABCUR3 ends with an abend code of 3761.

User response

Correct the error and rerun the job.

FABC0370I	INSUFF. STORAGE FOR: *aaaa* - INCREASE REGION SIZE

Explanation

Program FABCUR3 issued a GETMAIN macro to allocate storage for the purpose of *aaaa*. The attempt was unsuccessful.

System action

FABCUR3 calculates HIGH ALLOCATE RBA of the multi-volume ADS without using the Catalog Search Interface (CSI) and continues processing.

User response

None. This message is informational.

FABC0371I	FAILURE READING DATA SET INFORMATION FROM CATALOG - DSN: *data_set* - REASON CODE: *rsn* RETURN CODE: *rc* - NOT FOUND DATA PORTION - INCONSISTENT ENTRY NUMBER. VOLSER: *nn xxxxxxxx*: *nn* - THE NUMBER OF VOLUME IS BEYOND THE LIMIT

Explanation

An inconsistency is found in the catalog information.

System action

FABCUR3 calculates HIGH ALLOCATE RBA of the multi-volume ADS without using the Catalog Search Interface (CSI) and continues processing.

User response

None. This message is informational.

FABC0500I	FABCUR5 ENDED NORMALLY

Explanation

This message is generated when all requested processing has been completed without errors.

System action

Program FABCUR5 ends with a return code of 0.

User response

None. This message is informational.

FABC0500W	FABCUR5 ENDED WITH WARNINGS

Explanation

This message is generated when trivial error conditions were encountered by program FABCUR5.

System action

FABCUR5 ends with a return code of 4.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR5. Correct the problem and rerun the job, or continue with the processing, as desired.

FABC0500E	FABCUR5 ENDED WITH ERRORS

Explanation

This message is generated when nontrivial error conditions were encountered by program FABCUR5.

System action

FABCUR5 ends with a return code of 8.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR5. Correct the problem and rerun the job.

FABC0501I	DATABASE DEFINITION RECORD FOR DBD: *dbdname* IS BUILT

Explanation

This message is generated when program FABCUR5 built the database definition record for the DBD specified.

System action

FABCUR5 continues processing.

User response

None. This message is informational.

FABC0502I **DBD DEFINITION INFORMATION FOR DBD:** *dbdname* **IS REPORTED**

Explanation

This message is generated when program FABCUR5 reports the DBD definition information for the DBD specified.

System action

FABCUR5 continues processing.

User response

None. This message is informational.

FABC0520I **CARD** *xx: zzzz…zzzz*

Explanation

This message is generated to show the control statement currently being processed.

System action

Program FABCUR5 continues processing.

User response

None. This message is informational.

FABC0521W **ERROR DETECTED NEAR COLUMN** *xx*

Explanation

Program FABCUR5 detected an error in the control statement currently being processed. (See the immediately preceding FABC0520I message.)

System action

FABCUR5 continues processing, and issues one or more other FABC05*xx* messages.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR5. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0521E **ERROR DETECTED NEAR COLUMN** *xx*

Explanation

Program FABCUR5 detected an error in the control statement currently being processed. (See the immediately preceding FABC0520I message.)

System action

FABCUR5 continues processing, and issues one or more other FABC05*xx* messages.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR5. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0522W **BLANK/INVALID CONTROL CARD**

Explanation

Self-explanatory

System action

Program FABCUR5 discards the control statement, sets an end-of-job return code of 4, and continues processing.

User response

Remove the specified control statement in subsequent execution of FABCUR5.

FABC0523E **UNKNOWN KEYWORD**

Explanation

Program FABCUR5 encountered a control statement with a value starting in column one that is not one of the valid control statements.

System action

FABCUR5 ends with an abend code of 3728.

User response

Correct, or remove, the specified control statement.

FABC0525E **1ST CONTROL CARD NOT DBDNAME= CARD**

Explanation

Self-explanatory.

System action

Program FABCUR5 ends with an abend code of 3728.

User response

The control statement stream must include one DBDNAME control statement, and it must be the first statement in the stream. Correct the control statement stream. Rerun the job.

FABC0526E	INVALID DBDNAME= CONTROL CARD - MULTIPLE DBDNAME= CARDS ENCOUNTERED

Explanation

Self-explanatory.

System action

Program FABCUR5 ends with an abend code of 3728.

User response

The control statement stream must include one DBDNAME control statement and it must be the first statement in the stream. Correct the control statement stream, and rerun the job.

FABC0527E	INVALID DBDNAME= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR5 discards the control statement, sets an internal error flag, and continues processing.

User response

See the topic "Input for the Database Definition Record Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details on the syntax of the DBDNAME control statement. Correct the error, and rerun the job.

FABC0528E	INVALID DBDNAME= CONTROL CARD - DBDNAME MISSING/INVALID

Explanation

Self-explanatory.

System action

Program FABCUR5 discards the control statement, sets an internal error flag, and continues processing.

User response

See the topic "Input for the Database Definition Record Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details on the syntax of the DBDNAME control statement. Correct the error, and rerun the job.

FABC0530E	INVALID FUNCTION= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

Program FABCUR5 discards the control statement, sets an internal error flag, and continues processing.

User response

See the topic "Input for the Database Definition Record Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details on the syntax of the FUNCTION control statement. Correct the error, and rerun the job.

FABC0531E	INVALID FUNCTION= CONTROL CARD - "FUNCTION=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR5 discards the control statement, sets an internal error flag, and continues processing.

User response

See the topic "Input for the Database Definition Record Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details on the syntax of the FUNCTION control statement. Correct the error, and rerun the job.

FABC0540I INPUT= CONTROL CARD IGNORED

Explanation

Self-explanatory.

System action

Program FABCUR5 discards the control statement and continues processing.

User response

None. This message is informational.

FABC0541E NO DD OR DUMMY SPECIFIED FOR BOTH ACBLIB DD AND DURDBDFN DD

Explanation

Program FABCUR5 found that no DD statement or DUMMY is specified for both ACBLIB DD and DURDBDFN DD data sets for FUNCTION=PRINT request.

System action

FABCUR5 ends with a return code of 8.

User response

Specify the correct data set for the ACBLIB DD, or the DURDBDFN DD statements, or both, and rerun the job.

FABC0600I FABCUR6 ENDED NORMALLY

Explanation

This message is generated when all requested processing has been completed without errors.

System action

Program FABCUR6 finished 'EOF' function with no errors.

User response

None. This message is informational.

FABC0600W FABCUR6 ENDED WITH WARNING

Explanation

This message is generated when trivial error conditions were encountered by program FABCUR6.

System action

FABCUR6 finished 'EOF' function with trivial errors.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR6. Correct the problem and rerun the job, or continue with the processing, as desired.

FABC0600E FABCUR6 ENDED WITH ERRORS

Explanation

This message is generated when nontrivial error conditions were encountered by program FABCUR6.

System action

FABCUR6 finished 'EOF' function with nontrivial errors.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR6. Correct the problem, and rerun the job.

FABC0601I OBTAINED DB DEFINITIONS FROM *resource*

Explanation

This message indicates the *resource* (ACB library or IMS directory) where FABCUR6 obtained database definitions from.

System action

FABCUR6 continues processing.

User response

None. This message is informational.

FABC0602W NO 'PUT' CALL PROCESSED

Explanation

This message is generated when the application program ended without a 'PUT' call request to program FABCUR6.

System action

FABCUR6 ends with FABC0600W message.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR6. Correct the problem and rerun the job.

FABC0603W   NO SEGMENTS WILL BE RELOADED TO AREA *zzzzz* **(AREANAME:** *yyyyyyyy*) **- EMPTY AREA WILL NOT BE RELOADED BECAUSE EMPTY=NO IS SPECIFIED - EMPTY AREA WILL NOT BE RELOADED BECAUSE FILECTL STATEMENT IS NOT SPECIFIED FOR THIS AREA - EMPTY AREA WILL BE RELOADED**

Explanation

There is no segment record written for area *zzzzz*. One of the three sub-texts follows the FABC0603W message to indicate that an empty area for area *zzzzz* will or will not be reloaded by the succeeding reload (FABCUR3) process.

System action

Program FABCUR6 continues processing.

User response

Attempt to determine if there should have been any segment data records written to the specified output file. Verify that the DD statement ACBLIB/IMSACBA/ IMSACBB correctly identifies the proper data set, and that the DBDGEN and ACBGEN for the database being processed were performed correctly. Check that the randomizer module is specified correctly. Review the FILECTL specifications, if any. If the condition described by the sub-text for the empty area is not an expected result, then check that the EMPTY= option and the FILECTL statement(s) are specified correctly. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0604W   NO RECORDS WRITTEN TO DDNAME DURD*zzz***O / XD***zzzzz***O EXCEPT AREA INFORMATION RECORD**

Explanation

Self-explanatory.

System action

Program FABCUR6 continues processing.

User response

Attempt to determine if there should have been any segment data records written to the specified output file. Verify that the DD statement ACBLIB/IMSACBA/ IMSACBB correctly identifies the proper data set, and that the DBDGEN and ACBGEN for the database being processed were performed correctly. Check that the randomizer module is specified correctly. Review the FILECTL specifications, if any. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0605W   EMPTY=NO IS FORCED BECAUSE AREA_INFORMATION_RECORD=NO IS SPECIFIED

Explanation

Self-explanatory.

System action

Program FABCUR6 continues processing.

User response

When AREA_INFORMATION_RECORD=NO is specified, EMPTY=YES is overridden by EMPTY=NO unless both FORMAT=TFMT and LRECL=SEGTFMT are specified with EMPTY=YES. Check that the EMPTY option and the AREA_INFORMATION_RECORD option are specified correctly. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0611I   EXIT ROUTINE *exitname* **"END" CALL FINISHED - first 80 bytes characters of the message that user exit routine returned - subsequent 48 bytes characters of the message that user exit routine returned**

Explanation

Program FABCUR6 called the user exit routine *exitname* with an "END" call and the exit routine returned the message specified.

System action

FABCUR6 continues processing.

User response

None. This message is informational.

FABC0620I   CARD *xx: zzzz…zzzz*

Explanation

This message is generated to show the control statement currently being processed.

System action

Program FABCUR6 continues processing.

User response

None. This message is informational.

FABC0621W   ERROR DETECTED NEAR COLUMN
                _xx_

Explanation

Program FABCUR6 detected an error in the control statement currently being processed. (See the immediately preceding FABC0620I message.)

System action

FABCUR6 continues processing, and issues one or more other FABC06_xx_ messages.

User response

To determine the nature and causes of the errors detected, see the other messages generated by FABCUR6. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0621E   ERROR DETECTED NEAR COLUMN
                _xx_

Explanation

See message number FABC0621W.

System action

See message number FABC0621W.

User response

See message number FABC0621W.

FABC0622W   BLANK/INVALID CONTROL CARD

Explanation

Self-explanatory.

System action

Program FABCUR6 discards the control statement, sets an end-of-job return code of 4, and continues processing.

User response

Remove the specified control statement in subsequent executions of FABCUR6.

FABC0623E   UNKNOWN KEYWORD

Explanation

Program FABCUR6 encountered a control statement with a value starting in column one that is not one of the valid control statements.

System action

FABCUR6 ends with an abend code of 3728.

User response

Correct, or remove, the specified control statement.

FABC0640E   INVALID FILECTL= CONTROL
                CARD
                - FILE NO(S) SPECIFICATION
                MISSING/INVALID

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FILECTL control statement. Correct the control statement stream, and rerun the job.

FABC0641E   INVALID FILECTL= CONTROL
                CARD
                - AREA _zzzzz_ PREVIOUSLY
                SPECIFIED

Explanation

Self-explanatory. *zzzzz* is the area number.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution*

Pack: Supplementary Utilities User's Guide for details about the syntax of the FILECTL control statement. Correct the error, and rerun the job.

FABC0642E   **INVALID FILECTL= CONTROL CARD**
- FILE *zzzzz* **PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory. *zzzzz* is the file number.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FILECTL control statement. Correct the error, and rerun the job.

FABC0643E   **INVALID FILECTL= CONTROL CARD**
- FILECTL=[(*) | ALL] PREVIOUSLY SPECIFIED

Explanation

Program FABCUR6 detected a FILECTL control statement after having received the specified FILECTL specification.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FILECTL control statement. Correct the error, and rerun the job.

FABC0644E   **INVALID EXITRTN= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details on the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0645E   **INVALID EXITRTN= CONTROL CARD**
- "EXITRTN=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0646E   **INVALID EXITRTN= CONTROL CARD**
- EXITRTN NAME SPECIFIED TOO LONG

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0647E   **INVALID IMSCOMP= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the IMSCOMP control statement. Correct the error, and rerun the job.

FABC0648E   **INVALID IMSCOMP= CONTROL CARD**
- "IMSCOMP=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the IMSCOMP control statement. Correct the error, and rerun the job.

FABC0649E   **INVALID USERCTL= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the USERCTL control statement. Correct the error, and rerun the job.

FABC0654E   **INVALID FORMAT= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FORMAT control statement. Correct the error, and rerun the job.

FABC0655E   **INVALID FORMAT= CONTROL CARD**
- "FORMAT=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FORMAT control statement. Correct the error, and rerun the job.

FABC0656E   **- INVALID AREA_INFORMATION_RECORD= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

For details on the syntax of the AREA_INFORMATION_RECORD control statement, see the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error, and rerun the job.

FABC0657E   **- INVALID AREA_INFORMATION_RECORD= CONTROL CARD**
- "AREA_INFORMATION_RECORD=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

For details on the syntax of the AREA_INFORMATION_RECORD control statement, see the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error, and rerun the job.

FABC0658E   **- INVALID LRECL= CONTROL CARD**
- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

Correct the control statement, and rerun the job.

FABC0659E   **- INVALID LRECL= CONTROL CARD**
- "LRECL=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

The LRECL= control card has been already specified.

System action

FABCUR6 ends with an abend code of 3728.

User response

Correct the control statement, and rerun the job.

FABC0660W   **- [BLKSIZE | LRECL] FOR DDNAME** *ddname* **IS OVERRIDDEN BY SYSTEM DETERMINED VALUE**

- SPECIFIED [BLKSIZE | LRECL] TOO SMALL OR TOO LARGE
- [BLKSIZE | LRECL] SPECIFIED: *mmmmm*
- [BLKSIZE | LRECL] OVERRIDDEN: *nnnnn*

Explanation

This message is issued when BLKSIZE/LRECL is specified in JCL but its value is either smaller than the minimum or larger than the maximum tolerance level.

System action

Program FABCUR6 continues processing.

User response

None.

FABC0661E   **WHEN LRECL=SEGTFMT IS SPECIFIED, FORMAT=TFMT HAS TO BE SPECIFIED**

Explanation

You have to specify FORMAR=TFMT when LRECL=SEGTFMT is specified.

System action

Program FABCUR6 sets an internal error flag, and continues processing.

User response

Correct the error, and rerun the job.

FABC0670W   **I/O ERROR FOR OUTPUT DATA SET DDNAME** *ddname1***, UNLOAD PROCESS CONTINUES WITH DATA SET DDNAME** *ddname2*

Explanation

Program FABCUR6 issued a PUT for the *ddname1* specified. The PUT operation failed.

System action

FABCUR6 continues processing with the specified *ddname2*.

User response

None.

FABC0672E   **INVALID OUTDD= CONTROL CARD**

- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the OUTDD control statement. Correct the error, and rerun the job.

FABC0673E	INVALID OUTDD= CONTROL CARD - "OUTDD=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the OUTDD control statement. Correct the error, and rerun the job.

FABC0674E	INVALID IMSCATHLQ= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the USERCTL control statement. Correct the error and rerun the job.

FABC0675E	INVALID IMSCATHLQ= CONTROL CARD - "IMSCATHLQ=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FILECTL control statement. Correct the error and rerun the job.

FABC0676E	INVALID IMSCATHLQ= CONTROL CARD - IMSCATHLQ SPECIFIED TOO LONG

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the EXITRTN control statement. Correct the error and rerun the job.

FABC0677E	INVALID IMSCATACB_INPUT= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details

about the syntax of the USERCTL control statement. Correct the error and rerun the job.

FABC0678E   **INVALID IMSCATHLQ_INPUT= CONTROL CARD - "IMSCATHLQ_INPUT=" CONTROL CARD PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory.

System action

FABCUR6 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Reload Segment Data Set Create utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the FILECTL control statement. Correct the error and rerun the job.

FABC0700I   **FABCUR7 ENDED NORMALLY**

Explanation

This message is generated when all requested processing has been completed without errors.

System action

Program FABCUR7 finished 'EOF' function with no errors.

User response

None. This message is informational.

FABC0700W   **FABCUR7 ENDED WITH WARNING**

Explanation

This message is generated when trivial error conditions were encountered by program FABCUR7.

System action

FABCUR7 finished 'EOF' function with trivial errors.

User response

See the other messages generated by FABCUR7 to determine the nature and causes of the errors detected. Correct the problem and rerun the job, or continue with the processing, as desired.

FABC0700E   **FABCUR7 ENDED WITH ERRORS**

Explanation

This message is generated when nontrivial error conditions were encountered by program FABCUR7.

System action

FABCUR7 finished 'EOF' function with nontrivial errors.

User response

See the other messages generated by FABCUR7 to determine the nature and causes of the errors detected. Correct the problem, and rerun the job.

FABC0701E   **INVALID EXITRTN= CONTROL CARD - SYNTAX ERROR DETECTED**

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0702E   **INVALID EXITRTN= CONTROL CARD - "EXITRTN=" CONTROL CARD PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0703E   **INVALID EXITRTN= CONTROL CARD**

- EXITRTN NAME SPECIFIED TOO LONG

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the EXITRTN control statement. Correct the error, and rerun the job.

FABC0704E   INVALID IMSCOMP= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the IMSCOMP control statement. Correct the error, and rerun the job.

FABC0705E   INVALID IMSCOMP= CONTROL CARD - "IMSCOMP=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the IMSCOMP control statement. Correct the error, and rerun the job.

FABC0706I   CARD *xx: zzzz...zzzz*

Explanation

This message is generated to show the control statement currently being processed.

System action

Program FABCUR7 continues processing.

User response

None. This message is informational.

FABC0707W   ERROR DETECTED NEAR COLUMN *xx*

Explanation

Program FABCUR7 detected an error in the control statement currently being processed. (See the immediately preceding FABC0706I message.)

System action

FABCUR7 continues processing, and issues one or more other FABC07*xx* messages.

User response

See the other messages generated by FABCUR7 to determine the nature and causes of the errors detected. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0708W   BLANK/INVALID CONTROL CARD

Explanation

Self-explanatory.

System action

Program FABCUR7 discards the control statement, sets an internal error flag, and continues processing.

User response

Remove the specified control statement in subsequent executions of FABCUR7.

FABC0709E   UNKNOWN KEYWORD

Explanation

Program FABCUR7 encountered a control statement with a value starting in column one that is not one of the valid control statement types.

System action

FABCUR7 ends with an abend code of 3728.

User response

Correct, or remove, the specified control statement, and rerun the job.

FABC0710W	NO RECORD PROVIDED FROM UR7DATA/UR7DATA1/UR7DATA2

Explanation

Program FABCUR7 found that there was no record read from the specified unloaded segment data set.

System action

FABCUR7 will issue FABC0700W message.

User response

Verify that the correct unloaded segment data set was specified.

FABC0711I	EXIT ROUTINE *exitname* "END" CALL FINISHED - first 80 bytes characters of the message that user exit routine returned - subsequent 48 bytes characters of the message that user exit routine returned

Explanation

Program FABCUR7 called the user exit routine *exitname* with "END" call and the exit routine returned the message specified.

System action

FABCUR7 continues.

User response

None. This message is informational.

FABC0714E	- INVALID AREA_INFORMATION_RECORD= CONTROL CARD - SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

For details on the syntax of the AREA_INFORMATION_RECORD control statement, see the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error, and rerun the job.

FABC0715E	- INVALID AREA_INFORMATION_RECORD= CONTROL CARD - "AREA_INFORMATION_RECORD" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

For details on the syntax of the AREA_INFORMATION_RECORD control statement, see the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error, and rerun the job.

FABC0717E	INVALID DBDNAME= CONTROL CARD - "DBDNAME=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the DBDNAME control statement. Correct the error, and rerun the job.

FABC0718E   **INVALID DBDNAME= CONTROL CARD**
- DBD NAME SPECIFIED TOO LONG

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide* for details about the syntax of the DBDNAME control statement. Correct the error, and rerun the job.

FABC0720I   **- SDEP=PHYSICAL UNLOADED SEGMENT RECORDS FOUND AND IGNORED**

Explanation

FABCUR7 found SDEP=PHYSICAL unloaded segment records. They were ignored because they are not actual SDEP image segment records.

System action

Program FABCUR7 discards them and continues processing.

User response

Check if the input file is correct.

FABC0721I   **OBTAINED DB DEFINITIONS FROM** *resource*

Explanation

This message indicates the resource (ACB library or IMS directory) where FABCUR7 obtained database definitions from.

System action

FABCUR7 continues processing.

User response

None. This message is informational.

FABC0722E   **INVALID IMSCATHLQ= CONTROL CARD**

- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error and rerun the job.

FABC0723E   **INVALID IMSCATHLQ= CONTROL CARD**
- "IMSCATHLQ=" CONTROL CARD PREVIOUSLY SPECIFIED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error and rerun the job.

FABC0724E   **INVALID IMSCATHLQ= CONTROL CARD**
- IMSCATHLQ SPECIFIED TOO LONG

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error and rerun the job.

FABC0725E   **INVALID IMSCATACB_INPUT= CONTROL CARD**

- SYNTAX ERROR DETECTED

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error and rerun the job.

FABC0726E   **INVALID IMSCATHLQ_INPUT= CONTROL CARD - "IMSCATHLQ_INPUT=" CONTROL CARD PREVIOUSLY SPECIFIED**

Explanation

Self-explanatory.

System action

FABCUR7 ends with an abend code of 3728.

User response

See the topic "Input for the DEDB Unloaded Segment Data Set Retrieve utility" in the *IMS Fast Path Solution Pack: Supplementary Utilities User's Guide*. Correct the error and rerun the job.

FABC0800I   **- FABCUR8 PROCESSING STARTED**

Explanation

This message is generated when FABCUR8 starts the requested processing.

System action

Program FABCUR8 continues processing.

User response

None. This message is informational.

FABC0801I   **- FABCUR8 ENDED NORMALLY**

Explanation

This message is generated when all requested processing have been completed without errors.

System action

Program FABCUR8 ended with a return code of 0.

User response

None. This message is informational.

FABC0801W   **- FABCUR8 ENDED WITH WARNINGS**

Explanation

This message is generated when trivial error conditions were encountered by program FABCUR8.

System action

FABCUR8 ends with a return code of 4.

User response

To determine the nature and causes of the errors detected, see the other messages that were generated by FABCUR8. Correct the problem and rerun the job, or continue with reload processing, as desired.

FABC0801E   **- FABCUR8 ENDED WITH ERRORS**

Explanation

This message is generated when nontrivial error conditions were encountered by program FABCUR8.

System action

FABCUR8 ends with a return code of 8.

User response

To determine the nature and causes of the errors detected, see the other messages that were generated by FABCUR8. Correct the problem, and rerun the job.

FABC0802I   **- HD DBD NAME IS NOT SPECIFIED IN THE EXEC PARAMETER**

Explanation

Program FABCUR8 detected that the HD DBD name is not specified in the EXEC parameter.

System action

FABCUR8 continues processing.

User response

FABCUR8 cannot verify the fixed-length segment and the compressed segment by the HD DBD, so the result may be unpredictable if HD unload file contains fixed-length or compressed segments. Make sure that HD unload files does not contain fixed-length or compressed segments.

FABC0803W   **- THE HD UNLOAD FILE BELONGS TO HALDB DBD:** *dbdname*

Explanation

Program FABCUR8 detected that the input HD unload file belongs to HALDB DBD specified by *dbdname*. To convert all records of the HALDB database to a DEDB database, HD unload files of all partitions of the HALDB must be specified.

System action

FABCUR8 sets a return code of 4 and continues.

User response

Make sure that HD unload files of all partitions of the HALDB database are concatenated as input. If the input is not for all partitions, then the FABCUR8 output unloaded segment records file must be sorted or merged with the output of another partitions before they are reloaded to the DEDB by using FABCUR3.

FABC0810E   **- SEGMENT:** *segname* **IN HD UNLOAD FILE IS NOT DEFINED IN HD DBD:** *dbdname*

Explanation

The segment specified by *segname* in the HD unload file is not defined in the HD DBD specified by *dbdname*.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0811E   **- SEGMENT CODE OF SEGMENT:** *segname* **BETWEEN HD UNLOAD FILE AND HD DBD:** *dbdname* **DOES NOT MATCH**
- HD UNLOAD FILE: *nnn* **(X'**xx**')**
- HD DBD: *nnn* **(X'**xx**')**

Explanation

The segment code of the segment specified by *segname* in the HD unload file and the HD DBD specified by *dbdname* did not match.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0812E   **- SEGMENT LEVEL OF SEGMENT:** *segname* **BETWEEN HD UNLOAD FILE AND HD DBD:** *dbdname* **DOES NOT MATCH**
- HD UNLOAD FILE: *nn* **(X'**xx**')**
- HD DBD: *nn* **(X'**xx**')**

Explanation

The segment level of the segment specified by *segname* in the HD unload file and the HD DBD specified by *dbdname* did not match.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0813E   **- SEGMENT:** *segname* **IN HD UNLOAD FILE DEFINED IN HD DBD:** *dbdname* **IS A FIXED-LENGTH SEGMENT**

Explanation

The segment specified by *segname* in the HD unload file that is defined in the HD DBD specified by *dbdname* is a fixed-length segment. Program FABCUR8 does not support fixed-length segment for input.

System action

FABCUR8 ends with a return code of 8.

User response

If you specify the incorrect HD unload file or incorrect DBD, specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0814E   **- SEGMENT:** *segname* **IN HD UNLOAD FILE DEFINED IN HD DBD:** *dbdname* **IS A COMPRESSED SEGMENT**

Explanation

The segment specified by *segname* in the HD unload file defined in the HD DBD specified by *dbdname* is a compressed segment. Program FABCUR8 does not support compressed segment for input.

System action

FABCUR8 ends with a return code of 8.

User response

If you had specified an incorrect HD unload file or an incorrect DBD, specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0815E   **- FIRST RECORD OF HD UNLOAD FILE IS NOT A HEADER RECORD**

Explanation

Program FABCUR8 found that the first record of the HD unload file that is specified by the DURINPT DD statement was not an HD unload header record.

System action

FABCUR8 ends with a return code of 8.

User response

Specify the correct HD unload file, and rerun the job.

FABC0816E   **- HD UNLOAD FILE SPECIFIED BY DURINPT DD IS EMPTY**

Explanation

Program FABCUR8 found that the HD unload file that is specified by the DURINPT DD statement was empty.

System action

FABCUR8 ends with a return code of 8.

User response

Specify the correct HD unload file, and rerun the job.

FABC0817E   **- SEGMENT:** *segname* **IN HD UNLOAD FILE IS A FIXED-LENGTH SEGMENT**

Explanation

The HD DBD name is not specified in the EXEC parameter but FABCUR8 found that the segment that is specified by *segname* in the HD unload file is a fixed-length segment. The first two bytes of the segment data is not a LL value. Program FABCUR8 does not support fixed-length segment for input.

System action

FABCUR8 ends with a return code of 8.

User response

Specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0818E   **- SEGMENT:** *segname* **DEFINED IN HD DBD:** *dbdname* **IS NOT FOUND IN THE HD UNLOAD HEADER RECORD**

Explanation

The segment that is specified by *segname* defined in the HD DBD specified by *dbdname* is not found in the HD unload header record entry. Because this defines as a physical segment, the HD unload header record must have an entry for the segment.

System action

Program FABCUR8 ends with a return code of 8.

User response

Specify the correct HD unload file or the correct DBD, and rerun the job.

FABC0820E   **- SEGMENT:** *segname* **IN HD UNLOAD FILE NOT DEFINED IN DEDB DMB:** *dmbname*

Explanation

The segment specified by *segname* in the HD unload file is not defined in the DEDB DMD specified by *dmbname*.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DEDB DMD, and rerun the job.

FABC0821E   **- PARENT OF SEGMENT:** *segname* **IN HD UNLOAD FILE AND DEDB DMB:** *dmbname* **DOES NOT MATCH - HD UNLOAD FILE:** *parent-segname* **- DEDB DMB:** *parent-segname*

Explanation

Parent segment name of the segment specified by *segname* in the HD unload file and the DEDB DMB specified by *dmbname* did not match.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DEDB DMB, and rerun the job.

FABC0822E   **- HIERARCHY OF SEGMENT:** *segname* **UNDER THE PARENT IN HD UNLOAD FILE AND DEDB DMB:** *dmbname* **DOES NOT MATCH - PARENT:** *parent-segname*

Explanation

Hierarchy of the segment that is specified by *segname* under the parent specified by *parent-segname* in the HD unload file and that under the DEDB DMB that is specified by *dmbname* did not match.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DEDB DMB, and rerun the job.

FABC0823E   **- ROOT SEGMENT:** *segname* **IN HD UNLOAD FILE IS NOT A ROOT IN DEDB DMB:** *dmbname*

Explanation

The root segment specified by *segname* in the HD unload file is not a root in the DEDB DMB specified by *dmbname*.

System action

Program FABCUR8 ends with a return code of 8.

User response

Check the reason of the error, specify the correct HD unload file or the correct DEDB DMB, and rerun the job.

FABC0824E   **- SEGMENT:** *segname* **DEFINED IN DEDB DMB:** *dmbname* **IS A FIXED-LENGTH SEGMENT**

Explanation

The segment specified by *segname* defined in the DEDB DMB specified by *dmbname* is a fixed-length segment. Program FABCUR8 does not support fixed-length segment for output.

System action

FABCUR8 ends with a return code of 8.

User response

If you had specified an incorrect DEDB DMB, specify the correct DEDB DMB, and rerun the job.

FABC0900I   ddname: *<text>*

Explanation

The text displays the data contained on a record read from the file.

DDNAME is one of the following:

- CNTLCRDS
- SEGXREFI

System action

Processing continues.

User response

None. This message is informational.

Module

FABCUR9

FABC0901E   **NON-NUMERIC DATA IN** *<keyword>* **FIELD**

Explanation

The value associated with a keyword should be numeric; however, non-numeric data was found.

System action

Processing ends with return code 8. This message will be accompanied by message FABC0909E, which will display the control statement in error.

User response

Correct the statement in error, and resubmit job.

Module

FABCUR9P

FABC0902E   **SEGXREFI FILE IS NOT IN SEQUENTIAL, CONTIGUOUS ORDER AT RECORD # *xxxxxxx***

Explanation

The Segment Code variable contained in the records in this file must be in sequential ascending order. The message identifies the record which was detected to be out of order.

System action

Processing ends with return code 08.

User response

Correct the order of statements in the file and resubmit job.

Module

FABCUR9

FABC0903E   ***<file name>* FILE IS REQUIRED, BUT IS EMPTY OR INVALID**

Explanation

The invalid file may be one of the following:

- DURDBDFN

  The DURDBDFN DD statement was present, but the file was empty.

- SEGXREFI

  The CNTLCRDS file contained the keyword 'SEGXREFI', which specifies that records are to be processed from this file. The SEGXREFI file was found to be empty.

System action

Processing ends with return code 8.

User response

Take the following actions:

- DURDBDFN

  Make sure that a file containing a valid database definition record is referenced by the DURDBDFN DD statement.

- SEGXREFI

  Make sure that the correct records are in the file, or remove the SEGXREFI statement from the CNTLCRDS file and resubmit job.

Module

FABCUR9

FABC0904E   **INVALID SEG CODE *<segcode>* ENCOUNTERED AT RECORD # *xxxxxxx* COLUMN # *xxxxxxx***

Explanation

The UNLDREC file contained a record containing the specified segment code. This segment code was not determined to be valid for the database being processed.

System action

Processing ends with return code 8.

User response

Determine and correct the problem. One of the following may be in error:

- The DURDBDFN file may not match the database being processed, if this file is being used.
- The DURDBDFN file may not have been intended to be used; however, the file is not empty.
- The SEGXREFI file may not match the database being processed, if this file is being used.
- The SEGXREFI file may not have been intended to be used; however, a CNTLCRDS record specifies 'SEGXREFI'
- The UNLDREC file may contain an unsupported record format

Module

FABCUR9F

FABC0905E   **RECORD # : *xxxxxxx* SEG CODE : *xxx* KEY LEN : *xxxxxxx* APPEARS TO BE AN INVALID UNLOAD FORMAT RECORD**

Explanation

The UNLDREC file contained a record which FABCUR9 was not able to interpret.

System action

Processing ends with return code 8.

User response

Determine and correct the problem. One of the following may be in error:

- The UNLDREC file may contain an unsupported record format.

Module

FABCUR9F

FABC0906E   **PROBLEM RESOLVING INPUT <*input*>**

Explanation

A CNTLCRDS record specified an invalid INPUT value.

System action

Processing terminates with return code 8.

User response

Correct the invalid keyword and resubmit the job.

Module

FABCUR9

FABC0907E   **DBDNAME= PARAMETER HAS NOT BEEN PROVIDED**

Explanation

A value for DBDNAME was required; however, no control statement containing this value was supplied.

System action

Processing ends with return code 8.

User response

Include the DBDNAME= keyword in a CNTLCRDS control statement, and resubmit the job.

Module

FABCUR9

FABC0908E   **IF ANY DURDBDFN COMPONENTS ARE SPECIFIED, ALL MUST BE**

Explanation

There are three keywords related to overriding the default values for lengths of DURDBDFN components. Keywords were included in the CNTLCRDS file to override one or more of these values; however, at least one keyword was omitted.

System action

Processing ends with return code 8.

User response

Determine and correct the problem using one of the following procedures:

- Include all three parameters in the CNTLCRDS file. Consult the manual for the proper syntax.
- Remove any of the three parameters which are in the CNTLCRDS file.

Module

FABCUR9

FABC0909E   **INVALID ddname: <*text*>**

Explanation

The specified record contained an invalid statement.

DDNAME is one of the following:

- CNTLCRDS
- SEGXREFI

System action

Processing ends with return code 8.

User response

Correct the statement, and resubmit job.

Module

FABCUR9 FABCUR9P

FABC0913I   **DURDBDFN COMPONENT LENGTHS SUPPLIED IN CNTLCRDS**

Explanation

The default values for the lengths of the three DURDBDFN components was overridden by values contained in the CNTLCRDS file.

System action

Processing continues.

User response

None. This message is informational.

Module

FABCUR9

FABC0914E	INVALID INPUT *\<input\>* HAS BEEN SPECIFIED

Explanation

A CNTLCRDS record specified an invalid INPUT value.

System action

Processing terminates with return code 8.

User response

Correct the invalid keyword and resubmit the job.

Module

FABCUR9

FABC0915E	THE REQUESTED DBDNAME: *\<dbdname\>* IS NOT REFERENCED IN THE PSB

Explanation

The specified DBDNAME did not correspond to a PCB in the PSB.

System action

Processing ends with return code 8.

User response

Determine and correct the problem. One of the following errors may exist:

- The DBDNAME specified in the CNTLCRDS DBDNAME= statement does not correspond to a PCB in the PSB.
- The DBDLIB DD statement references a DBD which does not correspond to a PCB in the PSB.
- An incorrect PSB may be specified in the EXEC statement of DFSRRC00
- The intent may have been to run program FABCUR9 in test mode, which does not validate the DBD name

or issue DL/I calls; however, the 'TEST' keyword was not included in CNTLCRDS

Module

FABCUR9

FABC0916E	CHECKPOINT RESTART PROCESSING IS NOT SUPPORTED

Explanation

An attempt was made to checkpoint restart.

System action

Processing ends with return code 8.

User response

Do not attempt the check point restart. In many cases, such a restart is not required. Consult the manual for Recovery restrictions.

Module

FABCUR9

FABC0917E	UNABLE TO DETERMINE LANGUAGE OF PSB

Explanation

The utility was unable to determine the type of PSB which was in use.

System action

Processing ends with return code 8.

User response

This problem may be an internal error in the FABCUR9 Utility. Contact IBM Software Support.

Module

FABCUR9

FABC0918E	SEGXREFI FIXED FORMAT AREA FIELDS ARE IN INCORRECT COLUMNS

Explanation

Some of the fields within the first 43 positions of the fixed format area of the SEGXREFI input field are misaligned or in the wrong columns.

System action

Processing ends with return code 8.

User response

Correct the SEGXREFI file. Consult the user's guide for the proper position of fields within the segment cross-reference file records.

Module

FABCUR9

FABC0919W   **RECORD FORMAT *\<format>* INCORRECT FOR SEGMENT *\<segname>***

Explanation

A DL/I call returned a status code of 'LD' because the implied or specified record format ('V' or 'F') for the I/O area of the specified segment did not match the DBD specification for this segment.

System action

Processing continues.

The utility will use the alternative record format and will reattempt the DL/I call.

User response

If the Segment Cross-Reference Records are being used, the RECORD FORMAT value for this segment should be changed from 'V' to 'F' or vice versa.

Module

FABCUR9I

FABC0920E   **SEGXREFI FILE WITH VALID RECORD FORMAT VALUES REQUIRED FOR INPUT=*\<input>* PROCESSING**

Explanation

A value for record format is required for each segment in the Segment Cross-Reference File (SEGXREFI) for this unload file format type

The following INPUT types require RECORD FORMAT values:

- IMS High Performance Unload/Reload Format Records (FF)

System action

Processing terminates with return code 8.

User response

Create Segment Cross-Reference records which include the RECORD FORMAT value for each segment and resubmit the job.

Module

FABCUR9F

FABC0921E   **SEGXREFI RECORD FOR SEGMENT *segname* CONTAINS MISSING OR INVALID DATA FOR FIELD *fieldnme***

Explanation

The Segment Cross-Reference record for the specify segment was missing data for the specified field.

The specified fields are among the following:

- Segment ID
- Segment Level
- Parent
- Segment name
- Position (Blank or Zero is acceptable)
- Maximum Length (Blank or Zero is acceptable)
- Minimum Length (Blank or Zero is acceptable)

System action

Processing ends with return code 8.

User response

Specify values for the omitted fields and resubmit the job.

Module

FABCUR9

FABC0922E   **UNSUPPORTED DLI CALL: *calltype* ISSUED**

Explanation

The specified call type is not supported by the program.

System action

Processing will abend with code 3401.

User response

This problem may be an internal error in the FABCUR9 Utility. Contact IBM Software Support.

Module

FABCUR9I

FABC0923E   **OPEN ERROR ON** *dsname* **DATASET**

Explanation

The specified data set could not be opened.

System action

Processing ends with return code 8. This message will

User response

Supply a DD statement for the specified data set, or remove the request for the data set from the control cards. FABCUR9 Utility. Contact IBM Software Support.

Module

FABCUR9

FABC0924E   **UNABLE TO VALIDATE SEG CODE** *xxx* **WITH CODE** *xxx* **ENCOUNTERED AT RECORD #** *xxxxxxx* **COLUMN #** *xxxxxxx*

Explanation

A problem was encountered while determining the correct segment code for the record.

System action

Processing ends with return code 8.

User response

This problem may be an internal error in the FABCUR9 Utility. Contact IBM Software Support.

Module

FABCUR9F

FABC0925E   **MUTUALLY EXCLUSIVE SEG CROSS REF RECORDS ENCOUNTERED FOR SEGMENT** *segname*

Explanation

Mutually exclusive keywords were encountered in the Segment Cross-Reference File (SEGXREFI) for the specified segment.

The following combinations are not allowed:

- REPL and NOREP
- REPL and GHU
- BYPASS and REPL
- BYPASS and GHU

System action

Processing ends with return code 8.

User response

Resolve the conflicting parameters, and resubmit the job.

Module

FABCUR9

FABC0926E   **LOAD OF** *segname* **ATTEMPTED INTO A POPULATED DATABASE.**

Explanation

An attempt was made to perform updates in a database which was already populated. The REPL CNTLCRDS keyword was not specified

System action

Processing ends with return code 8.

User response

Determine whether load processing was intended

- If load processing was intended, scratch and reallocate the database
- If updates to the existing database were intended, specify the REPL CNTLCRDS keyword

Correct the situation, and resubmit the job

Module

FABCUR9U

FABC0927E   **BYPASS FOR SEGMENT** *segname1* **CONFLICTS WITH INTENT FOR SEGMENT** *segname2*

Explanation

The segment identified by segname2 may be processed; however, the Segment Cross-Reference table record for a segment higher in the hierarchy, identified by segname2, contains the BYPASS parameter. It is illegal for a dependent segment to be processed if a segment higher in the hierarchy is to be bypassed.

System action

Processing ends with return code 8.

User response

Change the designation of one of the involved segments:

- Add the BYPASS parameter to the dependent segment if processing is not required
- Change the parameter for the parent segment from BYPASS to NOREP if processing of the parent is not required.

Correct the situation and resubmit the job

Module

FABCUR9

**FABC0929E   REPL/GHU INTENT FOR SEGMENT
<segname> NOT ALLOWED WITH
NON-KEYED PARENT <parent-
segname>**

Explanation

The segment identified by *segname* has a non-keyed parent in the hierarchical chain. Keywords REPL or GHU were specified for this segment, but replace-related processes are not supported for segments where a parent does not have a key.

System action

Processing ends with return code 8.

User response

Specify BYPASS in the segment cross-reference table for these segments.

Correct the situation and resubmit the job

Module

FABCUR9

**FABC0930E   REPL/GHU INTENT NOT ALLOWED
FOR NON-KEYED SEGMENT
segname**

Explanation

The segment identified by *segname* is a non-keyed segment. Keywords REPL or GHU were specified for this segment, but replace-related processes are not supported for segments without keys.

System action

Processing ends with return code 8.

User response

Specify BYPASS in the segment cross-reference table for these segments.

Correct the situation and resubmit the job

Module

FABCUR9

**FABC0931E   STORAGE OBTAIN FAILURE ON
AREA: areaname**

Explanation

A GETMAIN of a storage area was attempted, but was unsuccessful.

System action

Processing ends with return code 8. This message will

User response

This problem may be an internal error in the FABCUR9 Utility. Contact IBM Software Support.

Module

FABCUR9

**FABC0932E   REPL/GHU INTENT NOT ALLOWED
FOR UNKNOWN KEY STATUS
SEGMENT segment**

Explanation

The segment cross-reference file does not contain information describing whether the segment is keyed, non-keyed or an SDEP. Keywords REPL or GHU were specified for this segment but replace-related processes are not supported for segments where the key status is not known. If the segment cross-

reference file was created during processing of a Full Function unload file, the unload file does not contain the key status information. In this situation, the key status will be initialized to hex zeros in the segment cross-reference file, and must be updated correctly before this file can be used to drive replace-related processes

System action

Processing ends with return code 8.

User response

Correct the key status information in the segment cross-reference table and resubmit the job.

Module

FABCUR9

FABC0933E   **INVALID VALUE: <value> FOR FIELD: <*field name*> IN FILE <*file name*>**

Explanation

The specified input file contains an invalid value for the specified field.

System action

Processing ends with return code 8.

User response

Determine the correct value for the field, and correct the input.

Module

FABCUR9

FABC0934E   **UNLOADED DATASET CONTAINING COMPRESSED SEGMENTS IS NOT SUPPORTED**

Explanation

The DEDB unloaded data set specified by the UNLDREC DD contained compressed unloaded segments.

System action

Processing ends with return code 8.

User response

Unload data set containing decompressed segments, and resubmit the job.

Module

FABCUR9F

FABC0935E   **UNLOADED DATASET CONTAINING SEGMENTS REACHED INSERT LIMIT COUNT IS NOT SUPPORTED**

Explanation

The DEDB unloaded data set specified by the UNLDREC DD contained segments which reached the ILC criteria.

System action

Processing ends with return code 8.

User response

Unload data set without ILC option, and resubmit the job.

Module

FABCUR9F

FABC0936I   **UNLOADED DATASET CONTAINED SEGMENTS WITH SUBSET POINTER(S) INFORMATION**

Explanation

The DEDB unloaded data set specified by the UNLDREC DD contained segments that contained subset pointer information. FABCUR9 Utility did not load subset pointer for the segment.

System action

Processing continues. The subset pointer for any segments with subset pointers will not be loaded.

User response

No action is required.

Module

FABCUR9F

FABC0937E   **UNSUPPORTED UNLOADED DATA SET DETECTED**

Explanation

The unload data set specified by the UNLDREC DD statement was in a format not supported by FABCUR9 Utility.

System action

Processing ends with return code 8.

User response

Make sure that only data sets in supported unload formats are specified. Consult the user's guide for a description of supported formats. Correct the errors, and resubmit the job.

Module

FABCUR9F

FABC0938I   **DEDB SDEP SEGMENTS WITH SDEP=PHYSICAL FORMAT DETECTED AND IGNORED**

Explanation

The DEDB unloaded data set specified by the UNLDREC DD contained segments with SDEP=PHYSICAL format. FABCUR9 Utility did not load SDEP physical records.

System action

FABCUR9 ignores all of SDEP segment records with SDEP=PHYSICAL format and continues processing.

User response

No action is required.

Module

FABCUR9F

FABC3700E   **>>>>> UNLOAD FAILED FOR AREA *nnnnn* (AREANAME: *areaname*) - ERROR OCCURRED DURING CALL TO RANDOMIZER *rmodname* FUNC: RANDOMIZING CALL**

Explanation

Program FABCUR1 determined that the unload subtask processing the specified area was unable to complete successfully. The first message is issued when an abend occurs during a call to the database Randomizer module. "FUNC: RANDOMIZING CALL"

is issued only when the unload subtask processing invoked the XCI randomizer.

System action

FABCUR1 ends with an abend code of 3700.

User response

Examine the other FABC37*xx* messages generated to determine the nature of the problem. Correct the problem, and rerun the job.

FABC3703E   **MEDIA MANAGER I/O ERROR - AREA *zzzz9* (DDNAME *ddname*) - REQUESTED RBA: *eeeeeeee* - MEDIA MANAGER RETURN CODE: *ccccffss***

Explanation

When an unload subtask issued the MMGRCALL to get access to the data set associated with the *ddname* specified, an unexpected Media Manager MMGRCALL error occurred. The variable *ccccffss* represents the Media Manager error return code used for problem determination. Media Manager return codes are described in the *DFSMS: DFSMSdfp Diagnosis Reference*. *zzzz9* is the area number.

System action

The unload subtask ends with an abend code of 3703.

User response

Check the Media Manager return code, correct the error, and rerun the job. If the error persists, contact IBM Software Support for additional analysis.

FABC3704E   **MEDIA MANAGER CONNECT| DISCONNECT ERROR - AREA *zzzz9* (DDNAME *ddname*) - MEDIA MANAGER RETURN CODE: *ccccffss***

Explanation

When an unload program issued the MMGRSRV to connect or disconnect the data set associated with the *ddname* specified, an unexpected Media Manager MMGRSRV error occurred.

The variable *ccccffss* represents the Media Manager error return code that can be used for problem determination. Media Manager return codes are described in the *DFSMS: DFSMSdfp Diagnosis Reference*. *zzzz9* is the area number.

System action

The unload subtask ends with the abend code of 3704.

User response

Check the Media Manager return code, correct the error, and rerun the job. If the error persists, contact IBM Software Support for additional analysis.

FABC3705E   **INSUFFICIENT STORAGE: INCREASE REGION SIZE (*aaaa*)**

Explanation

Programs FABCUR1/FABCUR5/FABCUR6/FABCUR7/FABCUR8 issued a GETMAIN macro to allocate storage for the purpose of *aaaa*. The attempt was unsuccessful.

System action

FABCUR1/FABCUR5/FABCUR6/FABCUR7/FABCUR8 ends with an abend code of 3705.

User response

Check the unload region size. Increase the REGION parameter on the EXEC statement for FABCUR1/FABCUR5/FABCUR6/FABCUR7/FABCUR8 as required. Rerun the job.

FABC3706E   **ERROR ATTACHING UNLOAD SUBTASK (RC = *xx*)**

Explanation

Program FABCUR1 issued an SVC 42 (ATTACH) to activate an unload subtask. The return code from OS specified that the attempt was unsuccessful.

System action

FABCUR1 ends with an abend code of 3706.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job.

FABC3707E   **CORRECT IMS RESLIB NOT CONCATENATED [- NO DFSBSCD0 MODULE FOUND] [- INVALID IMS LEVEL]**

Explanation

Correct IMS load module library was not concatenated to the JOBLIB/STEPLIB because JOBLIB/STEPLIB library has no DFSBSCD0 module or DFSBSCD0 module shows unsupported IMS level.

System action

Program FABCUR1/FABCUR3 ends with an abend code of 3707.

User response

Concatenate the correct IMS load module library to the JOBLIB/STEPLIB, and rerun the job.

FABC3708E   **IMS ONLINE SYSTEM IS ACCESSING AREA *zzzzz* (AREANAME: *areaname*) - WITH UPDATE/EXCLUSIVE INTENT**

Explanation

Program FABCUR1 found that the area specified was being used by an IMS online system with update intent.

System action

FABCUR1 ends with an abend code of 3708.

User response

Stop the area on the IMS online system(s) by entering / STOP AREA or /DBR AREA command or change the access intent of the area to read intent on the IMS system(s), and rerun the job if desired.

FABC3710E   **"OPEN" FAILED FOR DDNAME *ddname* - FAILED BY OS - DD STATEMENT NOT FOUND OR DUMMY/NULLFILE SPECIFIED - NOT A FIXED LENGTH RECORD DATASET - NOT A VARIABLE LENGTH RECORD DATASET - NOT AN 80 BYTE RECORD DATASET - RECORD LENGTH (LRECL) TOO SMALL (*xxxxx* REQUIRED) (*xxxxx* SPECIFIED) - BLOCK SIZE (BLKSIZE) TOO SMALL (*xxxxx* REQUIRED) (*xxxxx* SPECIFIED)**

Explanation

Program FABCUR1/FABCUR5/FABCUR6/FABCUR8 issued an OPEN for the ddname specified. The OPEN failed.

System action

FABCUR1/FABCUR5/FABCUR6/FABCUR8 ends with an abend code of 3710.

User response

Make sure that a DD statement is present for the ddname specified, and that it properly identifies the correct data set. Correct any errors, and rerun the job.

FABC3711E	IMS TOOLS CATALOG INTERFACE *function* FUNCTION (DEFINITION=[CURRENT | PENDING] FAILED - RETURN CODE: *rc*, REASON CODE: *rsn*

Explanation

The IMS Tools Catalog Interface ended with an error. *function* shows the function code of the IMS Tools Catalog Interface. The return code and reason code from the IMS Tools Catalog Interface are shown in *rc* and *rsn*, respectively.

System action

FABCUR6 or FABCUR7 ends with an abend code of U3711.

User response

If the function is OPEN, check if the correct high-level qualifier of the bootstrap data set is specified in the IMSCATHLQ keyword. Otherwise, contact IBM Software Support.

FABC3712E	MEMBER *acbname* NOT FOUND IN IMS CATALOG

Explanation

Program FABCUR6/FABCUR7 called an internal routine to obtain DMB information from the IMS directory. The return code from the routine indicates that the member does not exist.

System action

FABCUR6 or FABCUR7 ends with an abend code of 3712.

User response

Ensure that the high-level qualifier of the bootstrap data set of the IMS directory is correctly specified in the IMSCATHLQ keyword. Also, ensure that the IMS catalog population was correctly performed for the database being processed. Correct any errors, and rerun the job.

FABC3713E	MEMBER *member_name* FROM IMS CATALOG DEFINES DATABASE *dbdname*

Explanation

Program FABCUR6/FABCUR7 called an internal routine to obtain DMB information from the IMS directory. The return code from the routine indicates that the name of the DEDB DMB member does not match the name of the database.

System action

FABCUR6 or FABCUR7 ends with an abend code of 3713.

User response

Ensure that the high-level qualifier of the bootstrap data set of the IMS directory is correctly specified in the IMSCATHLQ keyword. Also, ensure that the IMS catalog population was correctly performed for the database being processed. Correct any errors, and rerun the job.

FABC3714E	MEMBER *member_name* FROM IMS CATALOG IS NOT A DEDB DMB

Explanation

Self-explanatory.

System action

FABCUR6 or FABCUR7 ends with an abend code of 3714.

User response

Ensure that the high-level qualifier of the bootstrap data set of the IMS directory is correctly specified in the IMSCATHLQ keyword. Also, ensure that the IMS catalog population was correctly performed for the database being processed. Correct any errors, and rerun the job.

FABC3715E	IMS LEVEL OF MEMBER *member_name* FROM IMS CATALOG IS NOT SUPPORTED

Explanation

Self-explanatory.

System action

FABCUR6 or FABCUR7 ends with an abend code of 3715.

User response

Ensure that the high-level qualifier of the bootstrap data set of the IMS directory is correctly specified in the IMSCATHLQ keyword. Also, ensure that the IMS catalog population was correctly performed for the database being processed. Correct any errors, and rerun the job.

FABC3719E **OPEN FAILED FOR VSAM DATASET AT DDNAME** *ddname* **(AREA** *zzzzz*)**
- VSAM ERROR DATA: RETURN-CODE: *rrr*
ACB "ERROR": *aaa* **(***bb***)**
- DD STATEMENT NOT FOUND

Explanation

Program FABCUR1 issued an OPEN for the ddname specified. The return code (*rrr*) from VSAM specified that the OPEN failed. The error code value from the ACB is shown in both decimal (*aaa*) and hexadecimal (*bb*) formats. *zzzzz* is the area number.

System action

FABCUR1 ends with an abend code of 3719.

User response

Make sure that an appropriate DD statement is present for the specified *ddname* and that it properly identifies the correct data set. See *DFSMS Macro Instructions for Data Sets*, which describes VSAM administration macros.

FABC3720E **"DEVTYPE" FAILED FOR DDNAME** *ddname* **(RC=***xx***)**

Explanation

Program FABCUR1/FABCUR5/FABCUR6/FABCUR7/ FABCUR8 issued an SVC 24 (DEVTYPE) to obtain information about the input/output device associated with ddname. The return code shows that the attempt was unsuccessful.

System action

FABCUR1/FABCUR5/FABCUR6/FABCUR7/FABCUR8 ends with an abend code of 3720.

User response

Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABC3721E **CALL TO "GET DEDB DMB" ROUTINE FAILED (RC =** *zz*)

Explanation

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6 called an internal routine to read and analyze a member from the ACB library data set. The return code from the routine (as shown in the message) shows that the attempt was unsuccessful.

System action

FABCUR1/FABCUR3/FABCUR5/FABCUR6 ends with an abend code of 3721.

User response

Contact IBM Software Support.

FABC3722E **TRKCALC FAILED FOR DDNAME** *ddname* **(RC =** *zz*)

Explanation

Program FABCUR1 invoked the TRKCALC macro to determine the 'number of CIs per track' value for the device on which the data set associated with the specified *ddname* resides. The return code from OS specified that the attempt was unsuccessful.

System action

FABCUR1 ends with an abend code of 3722.

User response

For further information, see *DFSMS DFSMSdfp Advanced Services*. Correct the errors, and rerun the job. If this situation persists, contact IBM software Support.

FABC3723E **LOAD FAILED FOR [COMPRESSION ROUTINE | EXIT ROUTINE | RANDOMIZER ROUTINE]** *xxxxxxxx* **(ABEND CODE S***xxxx* **/ REASON CODE** *xxxxxxxx***)**

Explanation

Program FABCUR1/FABCUR3/FABCUR6/FABCUR7 issued an SVC 8 (LOAD) to bring a copy of the randomizer routine or segment edit/compression

routine into the core. The return code from OS ('Abend Code') specifies that the attempt was unsuccessful.

System action

FABCUR1/FABCUR3/FABCUR6/FABCUR7 ends with an abend code of 3723.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3724E	*macro-name* **FAILED FOR DMB MEMBER** *member-name* **FOR DDNAME** *ddname* **(RC =** *rr***; REASON =** *zz***)**

Explanation

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6/FABCUR8 issued the specified macro (*macro-name*) to get access to the specified DMB member (*member-name*) in the data set specified (*ddname*). The return code and the reason code for z/OS specify that the attempt was unsuccessful.

System action

FABCUR1/FABCUR3/FABCUR5/FABCUR6/FABCUR8 ends with an abend code of 3724.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3725E	**CONTROL CARD DATASET IS EMPTY**

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3725.

User response

The control statement stream must contain one DBDNAME statement and at least one AREACTL statement. Correct the control statement stream, and rerun the job.

FABC3726E	**NO VALID DBDNAME= SPECIFICATION FOUND**

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3726.

User response

The control statement stream must contain one DBDNAME statement, and it must be the first control statement. Correct the control statement stream, and rerun the job.

FABC3727E	**NO VALID AREACTL= SPECIFICATION FOUND**

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3727.

User response

The control statement stream must contain at least one AREACTL statement. Correct the control statement stream, and rerun the job.

FABC3728E	**SEVERE CONTROL CARD ERROR(S) ENCOUNTERED**

Explanation

Program FABCUR1/FABCUR5/FABCUR6/FABCUR7 detected one or more severe errors during the analysis of the control statement.

System action

FABCUR1/FABCUR5/FABCUR6/FABCUR7 ends with an abend code of 3728.

User response

Examine the FABC01*xx*/FABC05*xx*/FABC06*xx*/FABC07*xx* messages generated to determine the nature of the problem(s). Correct the errors, and rerun the job.

FABC3729E	**DMB MEMBER** *member-name* **FOR** *xxxxxxxxxxxx* **HAS NO DATA**

Explanation

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6/
FABCUR8 tried to read all data of the DMB member
(*member-name*) but the member has no data.

System action

FABCUR1/FABCUR3/FABCUR5/FABCUR6/FABCUR8
ends with an abend code of 3729.

User response

Make sure that the DD statement (ddname) properly
specifies the correct data set. Correct the error, and
rerun the job.

FABC3730E   **MEMBER** *acbname* **NOT FOUND IN**
                acb-ddname DATASET

Explanation

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6/
FABCUR8 called an internal routine to read and
analyze a DMB in the specified ACBLIB data set.
The return code from the routine indicates that the
member does not exist.

System action

FABCUR1/FABCUR3/FABCUR5/FABCUR6/FABCUR8
ends with an abend code of 3730.

User response

Make sure that the DD statement specified properly
specifies the correct data set. If the ddname specified
is NEWACB, make sure that the required DMB is
present or remove the DD statement from the JCL
stream. Correct the errors, and rerun the job.

FABC3731E   **MEMBER** *member_name* **FROM**
                xxxxxxxxxx **DEFINES DATABASE**
                dbdname

Explanation

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6/
FABCUR8 called an internal routine to read and
analyze a DMB in the specified ACBLIB data set or
the IMS directory. The return code from the routine
indicates that the name of the DEDB DMB member
does not match the name of the database.

System action

FABCUR1/FABCUR3/FABCUR5/FABCUR6/FABCUR8
ends with an abend code of 3731.

User response

Make sure that the DD statement specified properly
specifies the correct data set, and that the DBDGEN
and ACBGEN, or IMS catalog population, have
completed successfully for the database being
processed. Correct the errors, and rerun the job.

FABC3732E   **MEMBER** *member_name* **FROM**
                xxxxxxxxxx **IS NOT A DEDB DMB**

Explanation

Self-explanatory.

System action

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6/
FABCUR8 ends with an abend code of 3732.

User response

Make sure that the DD statement specifies the correct
data set, and that the DBDGEN and ACBGEN, or IMS
catalog population, have completed successfully for
the database being processed. Correct the errors, and
rerun the job.

FABC3733E   **AREA** *xxxxx* **SPECIFIED ON**
                AREACTL= CARD NOT DEFINED IN
                DMB *xxxxxxxx* **FROM** *xxxxxxxx*

Explanation

Self-explanatory. *xxxxx* is the area number.

System action

Program FABCUR1 ends with an abend code of 3733.

User response

Verify the correctness of the user-supplied AREACTL
control statements. Make sure that the OLDACB DD
statement or the IMS directory is specified correctly.
Correct any errors, and rerun the job.

FABC3734E   **AREA** *xxxxx* **SPECIFIED ON**
                FILECTL= CARD NOT DEFINED
                IN DMB *xxxxxxxx* **FROM** *yyyyyyyy*
                DATASET

Explanation

Self-explanatory. *xxxxx* is the area number.

System action

Program FABCUR1/FABCUR6 ends with an abend code of 3734.

User response

Verify the correctness of the user-supplied FILECTL control statements. Make sure that the DD statement specified specifies the correct data set. If the ddname specified is NEWACB, make sure that the required DBDGEN and ACBGEN have been correctly performed. Correct the errors, and rerun the job.

FABC3735E	SEGMENT *segname* SPECIFIED ON LOADCTL= CARD NOT DEFINED IN "OLD" DMB

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3735.

User response

Verify the correctness of the user-supplied LOADCTL control statements. Make sure that the OLDACB DD statement specifies the correct data set. Correct the errors, and rerun the job.

FABC3736E	SEGMENT *segname* DEFINED IN "OLD" DMB; NOT FOUND IN "NEW" DMB

Explanation

Self-explanatory.

System action

Program FABCUR1 ends with an abend code of 3736.

User response

Verify that the OLDACB and NEWACB DD statements are specified correctly and the required DBDGEN and ACBGEN have been performed correctly. Correct the errors, and rerun the job.

FABC3737E	ROOT SEGMENT DEFINITION MISMATCH: [SOURCE SEGNAME | SOURCE KEY POSITION,LENGTH] "OLD" DMB *xxxxxxxx* "NEW" DMB *xxxxxxxx*

Explanation

Program FABCUR1 found that the definition for the root segment in the DMB from NEWACB data set did not match that in the DMB from the OLDACB data set.

System action

FABCUR1 ends with an abend code of 3737.

User response

Verify that the OLDACB and NEWACB DD statements are specified correctly, and the DBDGEN and ACBGEN have been performed correctly for the database being processed. Correct the errors, and rerun the job.

FABC3738E	IMS LEVEL OF MEMBER *member_name* FROM *xxxxxxxxxx* IS NOT SUPPORTED

Explanation

Self-explanatory.

System action

Program FABCUR1/FABCUR3/FABCUR5/FABCUR6/FABCUR8 ends with an abend code of 3738.

User response

Make sure that the correct data set is referenced, and that the DBDGEN and ACBGEN, or IMS catalog population have completed successfully for the database being processed. The ACB member may have been assembled with an unsupported release of IMS. Correct the errors, and rerun the job.

FABC3739E	MEMBER *member-name* FROM NEWACB DATASET IS LOWER IMS RELEASE LEVEL THAN ONE FROM OLDACB DATASET

Explanation

Program FABCUR1 found that IMS release of DMB member from NEWACB data set is lower than the DMB member from OLDACB data set.

System action

FABCUR1 ends with an abend code of 3739.

User response

Make sure that the correct data set is referenced, and that the DBDGEN and ACBGEN have been correctly

performed for the database being processed. Correct the errors, and rerun the job.

FABC3740E   **DATABASE STRUCTURE CHANGED; NOT AUTHORIZED**

Explanation

Program FABCUR1 determined that the hierarchical structure of the DMB from the NEWACB data set did not match that in the DMB from the OLDACB data set and HIERCHNG=YES/YESFORCE was not specified on the DBDNAME control statement.

System action

FABCUR1 ends with an abend code of 3740.

User response

Verify that the NEWACB DD statement is specified correctly and the required DBDGEN and ACBGEN have been performed correctly. If the structure change is desired, include HIERCHNG=YES/YESFORCE on the DBDNAME control statement. Correct the errors, and rerun the job.

FABC3741E   **INVALID DATABASE STRUCTURE CHANGE FOR SEGMENT** *segname*

Explanation

Program FABCUR1 determined that the hierarchical structure of the DMB from the NEWACB data set did not match that in the DMB from the OLDACB data set. The change detected was not one of those allowed by FABCUR1.

System action

FABCUR1 ends with an abend code of 3741.

User response

Verify that the NEWACB DD statement is specified correctly and the DBDGEN and ACBGEN have been performed correctly for the database being processed. For further information about the structure changes allowed by FABCUR1, see "Hierarchical structure changes" on page 117. Correct the errors, and rerun the job.

FABC3742E   **HIERCHNG=YES SPECIFIED W/O AREACTL=ALL**

Explanation

An AREACTL control statement with a value of ALL is required if database hierarchy changes are being performed.

System action

Program FABCUR1 ends with an abend code of 3742.

User response

Verify that the NEWACB DD statement is specified correctly and the required DBDGEN and ACBGEN have been performed correctly. If the structure change is desired, specify a value for the AREACTL control statement of ALL. Correct the errors, and rerun the job.

FABC3743E   **"MAX UOW BUFFER SPACE" SPECIFICATION IS TOO SMALL. MINIMUM SIZE MUST BE** *nnnn***K.**

Explanation

Program FABCUR1 determined that the BASE/DOVF buffer area size parameter on the user-supplied TASKCTL control statement was not large enough to contain at least the DOVF CIs and one (1) BASE CI for the areas being unloaded. The value specified by *nnnn* or larger value must be specified.

System action

FABCUR1 ends with an abend code of 3743.

User response

Review the control statement specifications. Correct the errors, and rerun the job.

FABC3744E   **RMODTYPE=G SPECIFIED W/O AREACTL=ALL**

Explanation

RMODETYPE=G is specified on the DBDNAME control statement and the NEWACB DD statement is present, but AREACTL=ALL is not specified.

System action

Program FABCUR1 ends with an abend code of 3744.

User response

Verify that the NEWACB DD statement is correct and the required DBDGEN and ACBGEN have been performed correctly. If the type of randomizer is

general, specify a value of `all` for the AREACTL control statement.

FABC3745E **DATASET AT DDNAME** *ddname* **IS NOT FOR AREA** *xxxxxxxx*
- CI-SIZE IN DMB *xxxxxxxx* **:** *xxxx*
- CI-SIZE OF DATASET: *xxxx*

Explanation

Program FABCUR1 opened the VSAM data set associated with the ddname specified. A comparison of certain key values extracted from the DMB from the OLDACB data set with the contents of the second CI in the VSAM data set specified that the VSAM data set was not the database described by the DMB.

System action

FABCUR1 ends with an abend code of 3745.

User response

Verify that the DD statement specified is correct and that the OLDACB DD statement specifies the correct data set. Correct the errors, and rerun the job.

FABC3745E **DATASET AT DDNAME** *ddname* **IS NOT FOR AREA** *xxxxxxxx*
- MISMATCH VALIDATING FIELD : *xxxxxxxx*
- VALUE IN DMB *xxxxxxxx* **:** *xxxx*
- VALUE IN "DMAC" CI : *xxxx*

Explanation

Program FABCUR1 opened the VSAM data set associated with the ddname specified. A comparison of certain key values extracted from the DMB from the OLDACB data set with the contents of the second CI in the VSAM data set specified that the VSAM data set was not the database described by the DMB.

System action

FABCUR1 ends with an abend code of 3745.

User response

Verify that the DD statement specified is correct and that the OLDACB DD statement specifies the correct data set. Correct the errors, and rerun the job.

FABC3746E **DETECT ERROR IN EQE LIST OF AREANAME:** *areaname* **(AREA NO:** *zzzzz***) - NUMBER OF EQE:** *nn*

Explanation

Program FABCUR1 found one or more error control intervals (CIs) extracted from the Error Queue Element (EQE) List in the second CI of the area data set specified.

System action

FABCUR1 ends with an abend code of 3746.

User response

Run the Full Recovery Utility, and rerun the job.

FABC3746E **AREANAME:** *areaname* **(AREA NO:***zzzzz***) CANNOT BE PROCESSED DUE TO X'80' SET IN EQE LIST**

Explanation

Program FABCUR1 found one or more error control intervals (CIs) extracted from the Error Queue Element (EQE) List in the second CI of the area data set specified.

System action

FABCUR1 ends with an abend code of 3746.

User response

Run the Full Recovery Utility, and rerun the job.

FABC3747E **EEQE DETECTED FOR AREA** *zzzzz* **(AREANAME:** *areaname***)**

Explanation

Program FABCUR1 found that the specified area has an EEQE.

System action

FABCUR1 ends with an abend code of 3747.

User response

Run the Full Recovery Utility, and rerun the job.

FABC3748E **AREA** *zzzzz* **(AREANAME:** *areaname***) IS RECOVERY NEEDED IN DBRC**

Explanation

Program FABCUR1 found that the specified area was recovery needed in DBRC.

System action

FABCUR1 terminates with an abend code of 3748.

User response

Make sure that correct area number was specified. If area number is correct, then run the Full Recovery Utility, and rerun the job.

FABC3749E **NO VALID DATA SET FOUND FOR AREA *zzzzz* (AREANAME: *areaname*)**

Explanation

Program FABCUR1 could not find a valid data set to receive the unloaded specified area.

System action

FABCUR1 terminates with an abend code of 3749.

User response

Get a LIST.RECON output report, identify unused area data set names, and rerun the job.

FABC3750E **"OPEN" FAILED FOR DDNAME *ddname subtext***

Explanation

OPEN processing failed for the file associated with the indicated DD statement. One of the following subtexts is issued:

- - DD STATEMENT NOT FOUND OR DUMMY/NULLFILE SPECIFIED
- - RECORD LENGTH (LRECL) TOO SMALL (*zzzz*9 REQUIRED) (*zzzz*9 SPECIFIED)
- - BLOCK SIZE (BLKSIZE) TOO SMALL (*zzzz*9 REQUIRED) (*zzzz*9 SPECIFIED)
- - FAILED BY OS

System action

Program FABCUR3 or FABCUR7 ends with an abend code of 3750.

User response

Based on the reason shown in the message, see the DD statement description for FABCUR3 or FABCUR7. Correct any errors, and rerun the job.

FABC3755E **"OPEN" FAILED - VSAM D/S: DDNAME: *ddname* (AREA *zzzz*9) - DD STATEMENT NOT FOUND**

Explanation

Program FABCUR3 found that there were no DD statements for the data set for the specified *ddname*. *zzzz*9 is the area number.

System action

FABCUR3 ends with an abend code of 3756.

User response

Provide a DD statement that identifies the correct data set for the area to be reloaded, and rerun the job.

FABC3756E **"OPEN" FAILED - VSAM D/S: DDNAME: *ddname* (AREA: *zzzz*9) - VSAM ERROR DATA: RETURN CODE: *aaa* (*bb*) ACB ERROR: *ccc* (*dd*)**

Explanation

Program FABCUR3 received a nonzero return code from VSAM when attempting to OPEN the data set for the ddname specified. The return code and ACB error code values are shown in both decimal (*aaa*, *ccc*) and hexadecimal (*bb*, *dd*) format. *zzzz*9 is the area number.

System action

FABCUR3 ends with an abend code of 3756. If duplicate data set is specified on more than two DD statements, this message might be issued.

User response

Make sure that a DD statement is present and that it properly identifies the correct data set for the area to be analyzed. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3757E **INSUFFICIENT SPACE DEFINED FOR "WORKFILE" DDNAME: DURIWRK**

Explanation

During IOVF work data set initialization, program FABCUR3 checked the HALCRBA and determined that insufficient space had been defined for the work data set; that is, an insufficient number of records had been defined.

System action

FABCUR3 abends with a user code of 3757.

User response

Calculate the number of records (that is, CIs) required for the work data set (largest UOW 1 times largest ROOT 2 value). Delete and redefine the work data set with sufficient space, and rerun the reload job.

FABC3758E	"STARTAREA=" VALUE GREATER THAN NO. AREAS DEFINED

Explanation

The STARTAREA= value provided on the control statement is greater than the number of areas defined in the DMB.

System action

Program FABCUR3 abends with a user code of 3758.

User response

Correct the value specified on the control statement, and rerun the reload job.

FABC3759E	"DEVTYPE" FAILED FOR DDNAME *ddname* (RC=*xx*)

Explanation

Program FABCUR3 issued an SVC 24 (DEVTYPE) to obtain information about the input/output device associated with *ddname*. The return code specified that the attempt was unsuccessful.

System action

FABCUR3 ends with an abend code of 3759.

User response

Make sure that a DD statement is present for *ddname*, and that it properly identifies the correct data set. Correct any errors, and rerun the job. If this situation persists, report it to systems operations personnel.

FABC3760E	INSUFF. STORAGE FOR: *aaaa* - INCREASE REGION SIZE

Explanation

Program FABCUR3 issued a GETMAIN macro to allocate storage for the purpose of *aaaa*. The attempt was unsuccessful.

System action

FABCUR3 ends with an abend code of 3760.

User response

Check the reload region size. Increase the REGION parameter in the EXEC statement for FABCUR3 as required, and rerun the job.

FABC3761E	CRITICAL CONTROL CARD ERROR ENCOUNTERED

Explanation

During parsing of the control statement, program FABCUR3 encountered a critical error. The critical error is described by another message.

System action

FABCUR3 ends with an abend code of 3761.

User response

Correct the control statement error, and rerun the job.

FABC3762E	"STARTAREA=" VALUE NOT FOUND IN INPUT FILE

Explanation

Data for the STARTAREA= value provided on the control statement was not found in the input file, or data was read for an area whose number is greater than that specified for the parameter.

System action

Program FABCUR3 abends with a user code of 3762.

User response

Correct the value specified on the control statement, and rerun the reload job. If input to FABCUR3 is a series of concatenated data sets, make sure that they are concatenated in ascending area number order.

FABC3763E	FILE DEFINED BY DDN "DURDBDFN" IS EMPTY

Explanation

Program FABCUR3/FABCUR5/FABCUR7 found that the file which should contain a formatted copy of the DMB (created by FABCUR1/FABCUR5/FABCUR6) is empty.

System action

FABCUR3/FABCUR5/FABCUR7 abends with a user code of 3763.

User response

The file must be re-created by FABCUR5. Be very careful and make sure that the correct dbd names are used for FABCUR5. Rerun the reload job, using the created DURDBDFN file.

FABC3764E   **DATA SET FOR AREA:** *zzzz9* **DDNAME:** *ddname* **NOT EMPTY**

Explanation

Program FABCUR3 found that the VSAM data set for the specified area was not empty. FABCUR3 examined the ENDRBA and found it to be greater than zero. DBFUMIN0 would issue message DFS2526I under the same conditions. *zzzz9* is the area number.

System action

FABCUR3 abends with a user code of 3764. When the same data sets are specified on more than two different area data set control statement (area*xxx*), this message might be issued. And if one data set is specified on the different DD statement as reloaded data sets, this message is also issued.

User response

Delete and redefine the VSAM cluster for the specified area. Do not run the DEDB Initialization utility (DBFUMIN0). Rerun the reload job.

FABC3765E   **DATA SET/DMB CI-SIZE CONFLICT FOR AREA:** *zzzz9* **DDNAME:** *ddname*

Explanation

Program FABCUR3 found that the CI size for the VSAM data set for the specified area did not match the CI size specified in the DBD. DBFUMIN0 would issue message DFS2509I under the same conditions. *zzzz9* is the area number.

System action

FABCUR3 abends with a user code of 3765. If same DD name is specified on more than two different area data statements (area*xxx*) as reloaded areas, this message might be issued.

User response

Delete and redefine the VSAM cluster for the specified area with the correct CI size. Rerun the reload job.

FABC3766E   **INSUFFICIENT SPACE DEFINED FOR AREA:** *zzzz9* **DDNAME:** *ddname*

Explanation

Program FABCUR3 found (by checking the HALCRBA) that insufficient space has been defined for the specified area. DBFUMIN0 would issue message DFS2510I under the same conditions. *zzzz9* is the area number.

System action

FABCUR3 abends with a user code of 3766.

User response

Delete and redefine the VSAM cluster for the specified area with more space. Rerun the reload job.

FABC3767E   **FILE DEFINED BY DDN "DURDATA" IS EMPTY**

Explanation

Program FABCUR3 found that the file which is supposed to contain the segment data for the area is empty.

System action

FABCUR3 abends with a user code of 3767.

User response

Make sure that the DD statement properly identifies the correct data set for the area to be reloaded. Correct any errors, and rerun the reload job. Empty areas cannot be initialized by FABCUR3.

FABC3768E   **DATASET/DMB CI-SIZE CONFLICT FOR WORKFILE DDNAME: DURIWRK**

Explanation

Program FABCUR3 found that the CI size of the work data set data set is smaller than the largest CI size specified in the DBD.

System action

FABCUR3 abends with a user code of 3768.

User response

Delete and redefine the work data set. The CI size of the work data set must be equal to the largest area CI size specified in the DBD. Rerun the reload job.

FABC3769E   INPUT DATA SEQUENCE ERROR (REC# *zzz,zzz,zz9*)

Explanation

Program FABCUR3/FABCUR7 found a record sequence error in the input segment data associated with ddname DURDATA.

System action

FABCUR3/FABCUR7 abends with a user code of 3769.

User response

Make sure that the sort for the data set is performed successfully. If input to FABCUR3/FABCUR7 is a series of concatenated data sets, make sure that they are concatenated in an ascending area number order. Correct the error, and rerun the reload job.

FABC3770E   TRKCALC FAILED (RC = *NN*)

Explanation

Program FABCUR3 issued a "TRKCALC" macro to determine if the control interval specified for the area data set or for the DURIWRK data set fits on the device to which the data set is allocated. The control interval size is too large for the device.

System action

FABCUR3 abends with a user code of 3770.

User response

Select a smaller control interval size or a DASD device with a longer track length. Then, rerun the reload job.

FABC3771E   VSAM I/O ERROR - "*zzz*" AREA *zzzz9* (DDNAME: *ddname*)
- REQUESTED RBA: *eeeeeeee*
- VSAM ERROR DATA: RETURN CODE: *aaa* (*bb*)
RPL FDBK: *ccc* (*dd*)

Explanation

Program FABCUR3 received a nonzero return code from VSAM when attempting to get access to either the work data set or area data set. The RBA of the CI being read is shown in hexadecimal format. The return

code and RPL FDBK code values are shown in both decimal (*aaa*, *ccc*) and hexadecimal (*bb*, *dd*) format. *zzzz9* is the area number.

System action

FABCUR3 ends with an abend code of 3771.

User response

See *DFSMS Macro Instructions for Data Sets* which describe VSAM administration macros. If this situation persists, contact IBM Software Support.

FABC3772E   DURDATA/DMB DEFINITION MISMATCH
- DURDATA AREA#: *zzzz9* MAX AREAS IN DMB: *zzzz9*

Explanation

The area number on the input segment data records is greater than the number of areas defined in the DMB.

System action

Program FABCUR3 abends with a user code of 3772.

User response

This is a serious error. Make sure that the data set associated with ddname DURDBDFN is correct. If it was incorrect, the areas that had already been reloaded (if any) should be deleted and reloaded. If in doubt, correct all JCL and rerun both the unload and reload jobs.

FABC3773E   ERROR "ATTACH"-ING WRITER SUB-TASK (RC = *xx*)

Explanation

Program FABCUR3 issued an ATTACH macro to attach the writer subtask. A nonzero return code specifies that the attempt was unsuccessful.

System action

FABCUR3 abends with a user code of 3773.

User response

Make sure that the module FABCUR3W is in the load library associated with STEPLIB. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3774E   ERROR "DETACH"-ING WRITER SUB-TASK (RC = *xx*)

Explanation

Program FABCUR3 issued a DETACH macro to detach the writer subtask. A nonzero return code specifies that the attempt was unsuccessful.

System action

FABCUR3 abends with a user code of 3774.

User response

Contact IBM Software Support.

FABC3775E	ERROR - MAXIMUM VSAM "RPL'S" EXCEEDED

Explanation

The value specified for IOVFBUF= exceeds 255, the maximum number of RPLs allowed.

System action

Program FABCUR3 abends with a user code of 3775.

User response

Reduce the value specified for the IOVFBUF= parameter, and rerun the reload job.

FABC3776E	ERROR - VSAM INCOMPATIBILITY PROBLEM

Explanation

Program FABCUR3 determined that the length of an RPL generated by a GENCB did not match the length of the DSECT.

System action

FABCUR3 abends with a user code of 3776.

User response

Contact IBM Software Support.

FABC3777E	SDEP FIRST CI RBA DISCREPANCY ERROR FOR AREA *zzzzz* (AREANAME: *areaname*)

Explanation

Program FABCUR3 found that the first SDEP CI RBA of the SDEP part of the target area was not the same as the unloaded area with SDEP=PHYSICAL keyword parameter.

System action

FABCUR3 terminates with an abend code of 3777.

User response

VSAM ESDS definition of the target area data set is not correct. Define the target area data set with the same CI size and space definition of the unloaded area data set, and rerun the job.

FABC3778E	AREA *zzzzz* IS FULL -PROCESSING TERMINATED

Explanation

Program FABCUR3 attempted to allocate an IOVF CI from an overflow unit, but found that all CIs within the overflow unit were in use. FABCUR3 then sequentially searched all overflow units in an attempt to locate an overflow unit that contains available IOVF CIs. All IOVF CIs in all overflow units were allocated. *zzzzz* is the area number.

System action

FABCUR3 abends with a user code of 3778.

User response

Increase the size of the area, and restart the reload job. Carefully examine the UOW and ROOT parameters.

FABC3779E	SEGCTL TABLE IS FULL - INCREASE NUMBER OF ENTRIES

Explanation

A segment control table is used to retain parentage information when the Insert Limit Count (ILC) feature of program FABCUR1 is used. There is one entry in the table for each ILC case encountered during the processing of a RAP. The default value is 500.

System action

FABCUR3 abends with a user code of 3779.

User response

Increase the number of entries by specifying the TBLENTRY= parameter on the control statement. Rerun/restart the reload job.

FABC3780E	VSAM "*xxxxxx*" ERROR - REG 15: *yy* REG 0: *zz*

Explanation

An error was encountered when performing one of the following VSAM functions: GENCB, MODCB, or SHOWCB. The values returned in registers 15 and 0 are shown (in hexadecimal format).

System action

The program abends with a user code of 3780.

User response

"VERIFY" the data set, and rerun/restart the reload job. If this situation persists, contact IBM software Support.

| FABC3781E | SEGMENT IN AREA *zzzzz* [AT RBA *xxxxxxxx*] (SEGCODE *yyy*) - RETURNED FROM [COMPRESSION ROUTINE *cmprname* | USER EXIT ROUTINE *exitname*] - EXCEEDS MAX DEFINED LENGTH |

Explanation

Program FABCUR1 unload subtask/FABCUR6 determined that the data in the specified segment was, when compressed/expanded by the specified segment edit/compression routine or by the specified user exit routine, longer than the length defined in the DBD member of the OLDACB or NEWACB ACB library. *zzzzz* is the area number.

System action

FABCUR1 unload subtask/FABCUR6 abends with a user code of 3781.

User response

Make sure that the segment edit/compression routine or the user exit routine is correct. Correct any errors, and rerun the job.

| FABC3782E | SEGMENT IN AREA *zzzzz* AT RBA *xxxxxxxx* (SEGCODE *yyy*) RETURNED FROM COMPRESSION ROUTINE *cmprname* TOO SHORT |

Explanation

An unload subtask determined that the data in the specified segment was, when compressed/expanded by the specified segment edit/compression routine, shorter than the length defined in the DBD member of the OLDACB or NEWACB ACB library. *zzzzz* is the area number.

System action

The unload subtask ends with an abend code of 3782.

User response

Make sure that the segment edit/compression routine is correct. Correct any errors, and rerun the job.

| FABC3783E | SEGMENT IN AREA *zzzzz* [AT RBA *xxxxxxxx*] (SEGCODE *yyy*) - RETURNED FROM [COMPRESSION ROUTINE *cmprname* | USER EXIT ROUTINE *exitname*] - KEY FIELD MODIFIED |

Explanation

FABCUR1 unload subtask/FABCUR6 determined that the key field in the segment was modified by the indicted segment edit/compression routine or by the specified user exit routine. *zzzzz* is the area number.

System action

FABCUR1 unload subtask/FABCUR6 abends with a user code of 3783.

User response

Make sure that the segment edit/compression routine or the user exit routine is correct. Correct any errors, and rerun the job.

| FABC3784E | INPUT DATA HAS INVALID COMPRESSION FLAG (REC# *zzz,zzz,zz9*) |

Explanation

Program FABCUR3 found one of the listed inconsistencies below among processing flag #1 (USRPFLG1, offset X'47') in the specified unloaded record, the database description table (DDT) flag byte for 'global information' (DDTFLG1, offset X'36'), and the segment description table (SDT) attribute flag byte (SDTFLG1, offset X'16') of the database definition record data set specified by the DURDBDFN DD statement.

- The SDTFCMP flag (X'01') on the SDTFLG1 flag byte in the SDT is off, but the USRPCOMP flag (X'02') on the USRPFLG1 flag byte in the unloaded record is on. That is, no segment edit/compression routine was defined for the segment but the data in the unloaded record was compressed.
- The SDTFCMP flag (X'01') on the SDTFLG1 flag byte in the SDT is on, and the DDTFCMY flag

(X'02') on the DDTFLG1 flag byte in the DDT is off, but the USRPCOMP flag (X'02') on the USRPFLG1 flag byte in the unloaded record is on. That is, segment edit/compression routine was defined for the segment, and the compression request was specified when the area was processed by the DEDB Unload/Reload program, but the unloaded record was not compressed.

- The SDTFCMP flag (X'01') on the SDTFLG1 flag byte in the SDT is on, and the DDTFCMY flag (X'02') on the DDTFLG1 flag byte in the DDT is on, but the USRPCOMP flag (X'02') on the USRPFLG1 flag byte in the unloaded record is off. That is, segment edit/compression routine was defined for the segment, and the compression request was not specified when the area was processed by the DEDB Unload/Reload program, but the unloaded record was compressed.

System action

FABCUR3 ends with an abend code of 3784.

User response

Verify that the unloaded data set specified by the DURDATA DD statement and the database definition record data set specified by the DURDBDFN DD statement are correct. Correct any errors, and rerun the job.

FABC3785E	SEGMENT AT REC#: *xxx,xxx,xx9* (AREA *zzzzz* SEGCODE *xxx*) - RETURNED FROM [USER EXIT ROUTINE *exitname* | COMPRESSION ROUTINE *comprname*] - EXCEEDS MAX DEFINED LENGTH

Explanation

Program FABCUR3/FABCUR7 determined that the data in the specified segment is, when compressed by the segment edit/compression routine or by the user exit routine, longer than that allowed by the definition for the segment in the database definition record specified by the DURDBDFN DD statement. *zzzzz* is the area number.

System action

FABCUR3/FABCUR7 abends with a user code of 3785.

User response

Check that the segment edit/compression routine or the user exit routine is correct for the specified segment and that the DURDBDFN DD statement

properly identifies the correct data set for the area to be reloaded. Correct any errors, and rerun the job.

FABC3786E	SEGMENT AT REC#: *zzz,zzz,zz9* (AREA *zzz* SEGCODE *xxx*) RETURNED FROM COMPRESSION ROUTINE *comprname* TOO SHORT

Explanation

Program FABCUR3 determined that the data in the specified segment was, when compressed by the specified segment edit/compression routine, shorter than allowed for the database definition record definition specified by the DURDBDFN DD statement.

System action

The unloaded subtask ends with an abend code of 3786.

User response

Check that the segment edit/compression routine is correct for the specified segment and that the DURDBDFN DD statement properly identifies the correct data set for the area to be reloaded. Correct any errors, and rerun the job.

FABC3787E	SEGMENT AT REC#: *zzz,zzz,zz9* (AREA *zzzzz* SEGCODE *xxx*) - RETURNED FROM [COMPRESSION ROUTINE *comprname* | USER EXIT ROUTINE *exitname*] - KEY FIELD MODIFIED

Explanation

Program FABCUR3/FABCUR7 determined that the key field in the specified segment was modified by the specified segment edit/compression routine or by the specified user exit routine. *zzzzz* is the area number.

System action

FABCUR3/FABCUR7 abends with a user code of 3787.

User response

Check that the segment edit/compression routine or the user exit routine is correct for the specified segment. Correct any errors, and rerun the job.

FABC3788E	SDEP=PHYSICAL SPECIFIED BUT - CI SIZE UNMATCH BETWEEN *xxxxxxxx* AND *yyyyyyyy* - RANDOMIZER RMODTYPE=G SPECIFIED/ASSUMED

- AREA NAME SPECIFIED IN *xxxxxxxx* **NOT FOUND IN** *yyyyyyyy*

Explanation

Program FABCUR1 found that the SDEP=PHYSICAL keyword parameter is specified, but there is one of the following errors:

- The CI size of an area does not match between OLDACB and NEWACB, or between the current active ACB and the pending ACB in the IMS catalog.
- The REORG keyword is not specified or NEWACB DD is provided with or without the HIERCHNG= keyword parameter.
- The area name defined in the OLDACB was not found in the NEWACB. If IMS managed ACBs are used, the area name defined in the current active ACB was not found in the pending ACB.

System action

FABCUR1 terminates with an abend code of 3788.

User response

Correct the error, and rerun the job.

FABC3789E	SDEP= CONTROL CARD SPECIFIED - SDEP SEGMENT NOT DEFINED IN DMB *xxxxxxxx* FROM *yyyyyyyy*

Explanation

Program FABCUR1 found that the SDEP=LOGICAL|PHYSICAL keyword parameter is specified, but the database specified by the DMB name does not define a SDEP segment.

System action

FABCUR1 terminates with an abend code of 3789.

User response

Make sure that the correct DBDNAME or ACB library, or the IMS catalog is specified. Correct the error, and rerun the job.

FABC3790E	ERROR IN CALL TO RANDOMIZER *xxxxxxxx* - AREA: *nnnnn* (AREANAME *areaname*) - RMODTYPE=S BUT RANDOMIZED TO ANOTHER AREA

Explanation

RMODTYPE=S is specified, but the randomizer randomized a record to another area that is different from the original one.

System action

Program FABCUR1 ends with an abend code of 3790.

User response

Check that the load library that contains the randomizer or the RMODTYPE=S option is correct. Correct the error and rerun the job.

FABC3790E	ERROR IN CALL TO RANDOMIZER *xxxxxxxx* FUNCTION: *function_name* - RETURN CODE: *xxx* [REASON CODE: X'*xxxxxxxx*'] - AREA: *zzzzz* (AREANAME *areaname*) - [SEG RBA: *xxxxxxxx* | USR REC#: *xxxxxxxx*]

Explanation

An unload subtask or a reload job called the specified randomizer module to calculate the new area and RAP number values for a root segment. The values returned were invalid. "FUNCTION" and "REASON CODE" are issued only when FABCUR1 or an unload subtask called the XCI randomizer routine.

System action

The unload subtask or a reload job ends with an abend code of 3790.

User response

Verify that the RMODLIB DD statement specifies the correct data set. Correct the errors, and rerun the job. If this situation persists, report it to database administration personnel.

FABC3791E	INVALID *xxx* POINTER IN AREA *zzzzz* (AREANAME *areaname*) - SOURCE SEG SEG-CD: *xxx* RBA: *xxxxxxxx* - TARGET SEG SEG-CD: *xxx* RBA: *xxxxxxxx*

Explanation

An IMS pointer in the specified area is in error. The pointer type (RAP, PCF, or PTF), the segment in which

the error was found, and the value of the pointer are shown in the message.

If PTRERROR=BYPASS is specified, program FABCUR1 issues message FABC3791W instead of FABC3791E.

System action

The unload subtask ends with an abend code of 3791.

User response

Consult database administration personnel about procedures for correcting the 'bad' pointer. If an error message was issued, correct the problem, and rerun the unload job.

FABC3791E	INVALID RAP POINTER IN AREA *zzzzz* (AREANAME *areaname*) - RAP AT RBA *xxxxxx* HAS A VALUE OF *xxxxxxxx*

Explanation

An IMS pointer in the specified area is in error. The pointer type (RAP, PCF, or PTF), the segment in which the error was found, and the value of the pointer are shown in the message.

If PTRERROR=BYPASS is specified, program FABCUR1 issues message FABC3791W instead of FABC3791E.

System action

The unload subtask ends with an abend code of 3791.

User response

Consult database administration personnel about procedures for correcting the 'bad' pointer. If an error message was issued, correct the problem, and rerun the unload job.

FABC3791W	INVALID *xxx* POINTER IN AREA *zzzzz* (AREANAME *areaname*) - SOURCE SEG SEG-CD: *xxx* RBA: *xxxxxxxx* - TARGET SEG SEG-CD: *xxx* RBA: *xxxxxxxx*

Explanation

An IMS pointer in the specified area is in error. The pointer type (RAP, PCF, or PTF), the segment in which the error was found, and the value of the pointer are shown in the message.

System action

FABCUR1 sets an end-of-job return code of 4 and continues processing.

User response

Consult database administration personnel about procedures for correcting the 'bad' pointer. If an error message was issued, correct the problem, and rerun the unload job.

FABC3791W	INVALID RAP POINTER IN AREA *zzzzz* (AREANAME *areaname*) - RAP AT RBA *xxxxxx* HAS A VALUE OF *xxxxxxxx*

Explanation

An IMS pointer in the specified area is in error. The pointer type (RAP, PCF, or PTF), the segment in which the error was found, and the value of the pointer are shown in the message.

System action

FABCUR1 sets an end-of-job return code of 4 and continues processing.

User response

Consult database administration personnel about procedures for correcting the 'bad' pointer. If an error message was issued, correct the problem, and rerun the unload job.

FABC3792E	I/O ERROR ATTEMPTING READ OF AREA *xxxxx* (AREANAME *areaname*) -REQUESTED RBA: *xxxxxxx* -VSAM ERROR DATA: RETURN CODE: *xx* : RPL "FDBK": *xxx* (*xx*)

Explanation

An unload subtask issued a GET for the data set associated with the ddname specified. The return code from VSAM specified that attempt was unsuccessful. The return code and the value of the FDBK field from the RPL are shown.

System action

The unload subtask ends with an abend code of 3792.

User response

See *DFSMS Macro Instructions for Data Sets* that describe VSAM administration macros. If this situation persists, contact IBM Software Support.

FABC3793E   **NO OUTPUT FILE PROVIDED FOR DATA FOR AREA** *zzzzz*

Explanation

Program FABCUR1 unload subtask/FABCUR6 was unable to write a segment data record for the specified area because no output File was specified to receive records for that area in the user-supplied FILECTL control statements. *zzzzz* is the area number.

System action

FABCUR1 unload subtask/FABCUR6 ends with an abend code of 3793.

User response

Review the FILECTL control statements. Correct any errors, and rerun the job.

FABC3794E   **INVALID "BLOCK TYPE ID" DETECTED IN AREA** *zzzzz* **(AREANAME** *areaname***)**
- VALUE EXPECTED: X"*xx***" (***xxxx* **CI)**
- VALUE FOUND: X"*xx***" (***xxxx* **CI)**

Explanation

An unload subtask read a CI from the specified area. The value in the IMS field known as DBLKBTID was not the value expected.

System action

The unload subtask ends with an abend code of 3794.

User response

Consult database administration personnel about procedures for fixing the bad data. When the problem has been corrected, rerun the unload job.

FABC3795E   **SEGMENT IN AREA** *zzzzz* **AT RBA** *xxxxxxxx* **(SEGCODE** *xxx***)** **EXCEEDS MAX DEFINED LENGTH**

Explanation

An unload subtask determined that the data in the specified segment was longer than allowed by the

definition for that segment in the "output" DMB. *zzzzz* is the area number.

System action

The unload subtask ends with an abend code of 3795.

User response

Correct the errors, and rerun the job.

FABC3796E   **"SUBSET" POINTER ERROR IN AREA** *zzzzz* **(AREANAME:** *areaname***)**

Explanation

An IMS pointer in the specified area is in error.

If PTRERROR=BYPASS is specified, program FABCUR1 issues message FABC3796W instead of FABC3796E.

System action

The unload subtask ends with an abend code of 3796.

User response

Consult database administration personnel about procedures for fixing the 'SUBSET' pointer error. If the error message was issued, rerun the unload job.

FABC3796W   **"SUBSET" POINTER ERROR IN AREA** *zzzzz* **(AREANAME:** *areaname***)**

Explanation

An IMS pointer in the specified area is in error.

System action

FABCUR1 sets end-of-job return code of 4, and continues processing.

User response

Consult database administration personnel about procedures for fixing the 'SUBSET' pointer error. If the error message was issued, rerun the unload job.

FABC3797E   **I/O ERROR FOR OUTPUT DATA SET DDNAME** *ddname1*

Explanation

Program FABCUR1/FABCUR6/FABCUR7 issued a PUT for the *ddname1* specified. The PUT operation failed.

System action

FABCUR1/FABCUR6/FABCUR7 ends with an abend code of 3797.

User response

Correct the errors, and rerun the job.

FABC3798E	AREA INFORMATION RECORD FOR AREA *zzzzz* (AREANAME: *areaname*) NOT FOUND

Explanation

Area information record for AREA *zzz* (DDNAME: ddname) could not be found in unloaded segment data records (DD name is DURDATA).

System action

Program FABCUR3 ends with an abend code of 3798.

User response

Correct the errors, and rerun the job.

FABC3799E	DURDBDFN RECORD IS INCORRECT

Explanation

The record format of the data set specified by the DURDBDFN DD statement is incorrect.

System action

Program FABCUR3, FABCUR5, FABCUR7, or FABCUR9 ends with an abend code of 3799.

User response

Correct the errors, and rerun the job. If the problem persists, save the entire run listing, including the dump, the JCL, and all the FPB reports, and contact IBM Software Support.

FABC3800E	GETMAIN FAILED DURING OPEN DCB/ACB

Explanation

A GETMAIN failed during an attempt to get storage for opening a DCB or an ACB. The return code means that the attempt was unsuccessful.

System action

Program FABCUR1 ends with an abend code of 3800.

User response

Increase the region size parameter on the JOB statement or the EXEC statement, and rerun the job.

FABC3801E	DSPSERV *xxxxxx* FAILURE OCCURRED: RETURN CODE : *yyyy* REASON CODE : *zzzzzzzz*

Explanation

Program FABCUR3 found that DSPSERV macro failed. The function code of the DSPSERV macro is shown in *xxxxxx* and the return code and reason code are shown in *yyyy* and *zzzzzzzz*.

System action

FABCUR3 ends with an abend code of 3801.

User response

For further explanation of the error, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job.

FABC3802E	ALESERV *xxxxxx* FAILURE OCCURRED: RETURN CODE : *yyyy*

Explanation

Program FABCUR3 found that the ALESERV macro failed. The function code of the DSPSERV macro is shown in *xxxxxx* and the return code is shown in *yyyy*.

System action

FABCUR3 ends with an abend code of 3802.

User response

For further explanation of the error, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job.

FABC3803E	ERROR OCCURRED WHEN PROCESSING DBRC RECON AREA=*areaname* FUNC=*function* RC=*nn*

Explanation

Program FABCUR3 was unable to successfully complete the DBRC call. The meanings of the functions are:

Function
  Meaning
SIGNON
  Sign-on call

SIGNOFF
Sign-off call

AUTH
Area authorization call

UNAUTH
Area unauthorization call

INIT
INIT function call

INIT0
INIT0 function call

INIT1
INIT1 function call

EOD
EOD function call

EOJ
EOJ function call

The preceding DBRC message explains the reason code.

System action

FABCUR3 ends with an abend code of 3803.

User response

Check the DBRC message preceding this message and follow the response in that message.

FABC3804E   NO DATA SET REGISTERED IN DBRC RECON ADS LIST FOR AREA *zzzzz* (AREANAME: *areaname*)

Explanation

Program FABCUR3 found that there was no area data set registered in DBRC for the specified area.

System action

FABCUR3 ends with an abend code of 3804.

User response

Get a LIST.RECON output report, specify an unused area data set name. Specify the name in the *adsname* DD statement, and rerun the job.

FABC3805E   AREA *zzzzz* (AREANAME: *areaname*) DDNAME: *ddname* - NOT SAME DS NAME BETWEEN DD STATEMENT AND DBRC

Explanation

Program FABCUR3 found that the area data set name specified in the ddname DD statement was not the same as the one registered in DBRC.

System action

FABCUR3 ends with ABEND 3805.

User response

Get a LIST.RECON output report and identify an unused area data set name, then specify the name in the DARVSAM DD statement, and rerun the job.

FABC3806E   NO VALID AREA DATA SET SPECIFIED FOR AREA *zzzzz* (AREANAME: *areaname*)

Explanation

Program FABCUR3 found that there was no valid area data set specified for the specified area.

System action

FABCUR3 ends with an abend code of 3806.

User response

IF DBRC=Y is specified, get a LIST.RECON output report and identify an unused area data set name, then specify it to the *adsname* DD statement, and rerun the job. If DBRC=N is specified, check and correct the content of the area*xxx* DD control statement and ads*xxx* DD data set(s), and rerun the job.

FABC3807E   DURDBDFN RECORD IS UNSUPPORTED DBT V2 OLD FORMAT

Explanation

Program FABCUR3, FABCUR5, FABCUR7, or FABCUR9 detected that the DURDBDFN record that was specified was an old level of the IMS DBT 2.x format. This old format record cannot be processed due to the lack of definition of minimum segment length.

System action

FABCUR3, FABCUR5, and FABCUR7 end with an abend code of 3807. FABCUR9 ends with a return code of 8.

User response

Re-create the FPB level of the DURDBDFN record file by using the FABCUR5 program with the correct

ACBLIB member. Rerun the job with the re-created DURDBDFN file.

FABC3808E   **SDEP CI FORMAT OLDER THAN IMS 6.1 DETECTED**

Explanation

Self-explanatory.

System action

Program FABCUR3 ends with an abend code of 3808, and continues processing.

User response

Re-unload the area whose CI is IMS 6.1 or higher, and rerun the job.

FABC3810E   **"OPEN" FAILED FOR DDNAME area*xxx* DD STATEMENT NOT FOUND**

Explanation

area*xxx* DD statement was not found.

System action

Program FABCUR3 ends with an abend code of 3810.

User response

Specify area*xxx* DD as reloaded VSAM data set, or area*xxx* DD that has DDNAME control statement as multi-area data sets. Rerun the job.

FABC3811E   **USER EXIT FABC1IE0 RETURNED WITH NON-ZERO RC**

Explanation

User exit routine FABC1IE0 sets nonzero to register 15 and returns to the caller.

System action

Program FABCUR1 ends with an abend code of 3811.

User response

Check the reason of the return code from FABC1IE0.

FABC3812E   **USER EXIT FABC3IE0 RETURNED WITH NON-ZERO RC**

Explanation

User exit routine FABC3IE0 sets nonzero to register 15 and returns to the caller.

System action

Program FABCUR3 ends with an abend code of 3812.

User response

Check the reason of the return code from FABC3IE0.

FABC3813E   **DBDNAME AND *xxxxxxx* ARE REQUIRED WHEN RAPERROR=ABEND IS SPECIFIED**

Explanation

Self-explanatory.

System action

Program FABCUR3 ends with an abend code of 3813.

User response

See "DEDB Reload SYSIN DD data set" on page 172 for details of the RAPERROR control statement. Correct the error, and rerun the job.

FABC3814E   **RAP DATA MISMATCH BETWEEN THE USR FILE AND THE RESULT OF RANDOMIZER**
- USR REC# : *xxx,xxx,xxx*
- AREA NO IN USR : *xxxxx*
- AREA NO FROM RANDOMIZER : *xxxxx*
- RAP RBA IN USR : *xxxxxxxx*
- RAP RBA FROM RANDOMIZER : *xxxxxxxx*

Explanation

Program FABCUR3 found that the RAP data (area number, RAP RBA) in the prefix of the unloaded segment record does not match the result of the randomizer.

System action

FABCUR3 ends with an abend code of 3814.

User response

Check whether the unloaded segment record file, the DEDB definition, or both that are obtained from the DURDBDFN DD or ACBLIB DD are correct. Correct the error, and rerun the job.

FABC3815E   **INCORRECT RAP RBA WAS DETECTED IN THE USR FILE [- RAP RBA IS NOT CI BOUNDARY | - RAP RBA IS NOT RAP CI RBA] - USR REC# :** *xxx,xxx,xxx* **- RAP RBA :** *xxxxxxxx*

Explanation

Program FABCUR3 found that the RAP RBA in the prefix of the unloaded segment record is not at a CI boundary or is not an RBA of a RAP CI.

System action

FABCUR3 ends with an abend code of 3815.

User response

Check whether the unloaded segment record file, the DEDB definition, or both that are obtained from the DURDBDFN DD or ACBLIB DD are correct. Correct the error, and rerun the job.

FABC3816E   **FIRST USR THAT HAS NEW RAP RBA WAS NOT FOR ROOT SEGMENT - USR REC# :** *xxx,xxx,xxx* **- RAP RBA :** *xxxxxxxx* **- SEGCODE :** *xxx*

Explanation

Program FABCUR3 found that the first unloaded segment record of a new RAP RBA is not a root segment. FABCUR3 expects that all unloaded segment records are sorted in the database hierarchical order and the first segment record of a new RAP RBA is a root segment.

System action

FABCUR3 ends with an abend code of 3816.

User response

Check whether the unloaded segment record file, the DEDB definition, or both that are obtained from the DURDBDFN DD or ACBLIB DD are correct. Correct the error, and rerun the job.

FABC3817E   **UNDEFINED SEGMENT CODE FOUND IN THE USR FILE RECORD FOR AREA NO:** *xxxxx***, AREANAME:** *xxxxxxxx* **- USR REC# :** *xxx,xxx,xxx* **- SEGCODE :** *xxx*

Explanation

Program FABCUR3 found that the segment code in the prefix of the unloaded segment record is not defined in the database.

System action

FABCUR3 ends with an abend code of 3817.

User response

Check whether the unloaded segment record file, the DEDB definition, or both that are obtained from the DURDBDFN DD or ACBLIB DD are correct. Correct the error, and rerun the job.

FABC3818E   **ROOT SEGMENT KEY LENGTH MISMATCH BETWEEN THE USR FILE AND DBD FOR AREA NO:** *xxxxx***, AREANAME:** *xxxxxxxx* **- USR REC# :** *xxx,xxx,xxx* **- USR KEY LEN :** *xxxxxxxxx* **- DBD KEY LEN :** *xxxxxxxxx*

Explanation

Program FABCUR3 found that the root segment key length in the prefix of the unloaded segment record does not match the database definition.

System action

FABCUR3 ends with an abend code of 3818.

User response

Check whether the unloaded segment record file, the DEDB definition, or both that are obtained from the DURDBDFN DD or ACBLIB DD are correct. Correct the error, and rerun the job.

FABC3819E   **SEGMENT LEVEL MISMATCH BETWEEN THE USR FILE AND DBD FOR AREA NO:** *xxxxx***, AREANAME:** *xxxxxxxx* **- USR REC# :** *xxx,xxx,xxx* **- SEGCODE :** *xxx* **- USR SEG LEVEL :** *xx* **- DBD SEG LEVEL :** *xx*

Explanation

Program FABCUR3 found that the segment hierarchical level of the segment code in the prefix of the unloaded segment record does not match the database definition.

System action

FABCUR3 ends with an abend code of 3814.

User response

Check the unloaded segment record file and/or the DEDB definition obtained from the DURDBDFN DD or ACBLIB DD is correct. Correct the error, and rerun the job.

FABC3820E   **UNSUCCESSFUL DYNALLOC REQUEST (SVC 99) RETURN CODE : *xx* REASON CODE : *yyyy***

Explanation

Program FABCUR3 issued an SVC 99 (DYNALLOC) to search information dynamically. The return code specified that the attempt was unsuccessful. The return code is shown in *xx*, and reason code is shown in *yyyy*.

System action

FABCUR3 ends with an abend code of 3820.

User response

For further explanation of the error, see the *MVS Programming: Authorized Assembler Services Reference*. Correct any errors, and rerun the job.

FABC3822E   **- NOT ENOUGH SDEP SPACE AVAIL IN AREA *zzzzz* (AREANAME: *areaname*) DUE TO DBD CHANGE - THE FIRST CI RBA IN THE NEW SDEP PART: X'*xxxxxxxx*' - REQUIRED SDEP SPACE: X'*xxxxxxxx*'**

Explanation

Program FABCUR1 unload subtask identified that the ESDS data set for the indicated area does not have enough space for the SDEP part (range between LB and LE) even if the ESDS data set is defined with the maximum size.

System action

The unload subtask ends with an abend code of 3822.

User response

Change the database definition so that enough space is available for the SDEP part, or delete the SDEP segments so that the required amount of SDEP space is reduced. Then rerun the unload job.

FABC3823E   **- THE FORMAT OF THE USR FILE IS OLD. REGENERATE THE USR FILE**

Explanation

Program FABCUR3 found that the SDEP flag field (USRSDEP) in the area information record of the input USR file is "PN". Such USR files are no longer supported.

System action

FABCUR3 ends with an abend code of 3823.

User response

Regenerate the input USR file by rerunning the unload job with SDEP=PHYSICAL accompanied by a DBD change. Then rerun the reload job.

FABC3825E   **THE SIZE OF THE RELOADED AREA DATA SET IS MORE THAN 4G BYTES - AREA NO: *nn*, AREA NAME: *areaname* - DDNAME: *ddname*, DSNAME: *dsname***

Explanation

The size of the reloaded area data set exceeds 4 GB.

System action

FABCUR3 ends with an abend code of 3825.

User response

Delete and redefine the area data set. Make sure that the size of the ADS does not exceed 4 GB.

FABC3830E   **IMAGE COPY LOAD MODULE NOT FOUND**

Explanation

There were no load modules of the IBM IMS High Performance Image Copy for z/OS in the JOBLIB/ STEPLIB DD library.

System action

Program FABCUR3 ends with an abend code of 3830.

User response

Concatenate the load module library data set of the IBM IMS High Performance Image Copy for z/OS, and rerun the job.

FABC3831E   INCORRECT LEVEL OF IMAGE COPY EXTENSIONS(ICE) LOAD MODULE DETECTED

Explanation

The load modules of the IBM IMS Image Copy Extensions for z/OS utility does not support the interface for Reload.

System action

Program FABCUR3 ends with an abend code of 3831.

User response

Specify the correct level of load module library of IBM IMS Image Copy Extensions for z/OS for the JOBLIB/ STEPLIB DD statement.

FABC3832E   LOAD FAILED FOR LOAD MODULE *modulename* (ABEND CODE S*xxxx* / REASON CODE *yyyyyyyy*)

Explanation

Program FABCUR8 issued an SVC 8 (LOAD) to load the module specified by *modulename* into the core. The return code from OS ('Abend Code') specifies that the attempt was unsuccessful.

System action

FABCUR8 ends with an abend code of 3832.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3833E   LOAD FAILED FOR DBD MEMBER *dbdname* (ABEND CODE S*xxxx* / REASON CODE *yyyyyyyy*)

Explanation

Program FABCUR8 issued an SVC 8 (LOAD) to load the DBD member specified by *dbdname* into the core. The return code from OS (S*xxxx*) specifies that the attempt was unsuccessful.

System action

FABCUR8 ends with an abend code of 3833.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3834E   DBDNAME CONTROL CARD/ [UR7DBDFN | DURDBDFN] MISMATCH CTL CARD DBDNAME *xxxxxxx* [UR7DBDFN | DURDBDFN] DBDNAME *xxxxxxxx*

Explanation

Program FABCUR3/ FABCUR7 found that the DBD name specified in the DBDNAME control statement and the one specified in the DURDBDFN/UR7DBDFN DD data set do not match.

System action

FABCUR3/FABCUR7 ends with an abend code of 3890.

User response

Check that the DBD name specified on the DBDNAME control statement and the DBD name of the DURDBDFN data set that is specified on the DURDBDFN/UR7DBDFN DD statement are correct. Correct the error, and rerun the job.

FABC3835E   IMS TOOLS CATALOG INTERFACE CANNOT BE USED - UNSUPPORTED IMS RELEASE

Explanation

The IMSCATHLQ=*bsdshlq* parameter is specified on SYSIN, but FABCUR1 or FABCUR3 could not use the IMS Tools Catalog Interface to read the ACB from the IMS directory because the version of IMS is lower than 14.

System action

FABCUR1 or FABCUR3 ends with an abend code of 3835.

User response

Rerun the job using a supported version of IMS.

FABC3836E   IMS TOOLS CATALOG INTERFACE *function* FUNCTION (DEFINITION=CURRENT | PENDING) FAILED

- RETURN CODE: *rc*, **REASON CODE:** *rsn*

Explanation

The IMS Tools Catalog Interface ended with an error. *function* shows the function code of the IMS Tools Catalog Interface. The return code and reason code from the IMS Tools Catalog Interface are shown in *rc* and *rsn*, respectively.

System action

FABCUR1 or FABCUR3 ends with an abend code of U3836.

User response

If the function is OPEN, check if the correct high-level qualifier of the bootstrap data set is specified in the IMSCATHLQ keyword. Otherwise, contact IBM Software Support.

FABC3890E **DBDNAME CONTROL CARD/ DURDBDFN MISMATCH CTL CARD DBDNAME** *xxxxxxxx* **DURDBDFN DBDNAME** *xxxxxxxx*

Explanation

Program FABCUR5 found that there is a discrepancy between the DBD name specified in the DBDNAME control statement and the one specified in the DURDBDFN DD data set.

System action

FABCUR5 abends with a user code of 3890.

User response

Verify the correctness of the DBD name specified on the DBDNAME control statement and the DBD name of the DURDBDFN data set specified on the DURDBDFN DD statement. Correct the error, and rerun the job.

FABC3900E *type* **SEGMENT KEY SEQUENCE ERROR IN AREA** *nnnnn* **(AREANAME** *areaname***)** **- SEG - CD:** *xxx* **RBA:** *xxxxxxxx*

Explanation

A segment key sequence in the specified segment of the specified area contains an error. The segment type, ROOT or DDEP, and the segment in which the error was found are shown in the message.

System action

Program FABCUR1 ends with an abend code of 3900.

User response

Consult database administration personnel about procedures for correcting the "bad" sequence field. Correct the problem, and rerun the unload job.

FABC3900E *type* **SEGMENT KEY SEQUENCE ERROR IN AREA** *nnnnn* **(AREANAME** *areaname* **)** **- SEG - CD:** *xxx* **- REC#:** *zzz,zzz,zz9*

Explanation

A segment key sequence in the specified segment of the specified area contains an error. The segment type, ROOT or DDEP, and the segment in which the error was found are shown in the message.

System action

Program FABCUR3 ends with an abend code of 3900.

User response

Verify the unloaded file in the DURDATA DD statement. Correct the problem, and rerun the reload job.

FABC3900W *type* **SEGMENT KEY SEQUENCE ERROR IN AREA** *nnnnn* **(AREANAME** *areaname* **)** **- SEG - CD:** *xxx* **RBA:** *xxxxxxxx*

Explanation

If PTRERROR=BYPASS is specified, program FABCUR1 issues message FABC3900W instead of FABC3900E.

System action

FABCUR1 sets and end-of-job return code of 4, and continues processing.

User response

None.

FABC3901E **FILE NUMBER ALREADY EXISTED IN AREA OUTPUT TABLE**

Explanation

Program FABCUR1/FABCUR6 tried to set a file number into the area output table (CCIAOUT) but it had been already set.

System action

FABCUR1/FABCUR6 ends with an abend code of 3901.

User response

Contact IBM Software Support.

FABC3902E   **LOAD FAILED FOR UNLOAD SUBTASK** *modname* **(ABEND CODE S***xxxx***/REASON CODE** *yyyyyyyy***)**

Explanation

Program FABCUR1 issued an SVC 8 (LOAD) to bring a copy of the unload subtask into the core. The return code received from OS (S*xxxx*) specifies that the attempt was unsuccessful. (ABEND CODE and REASON CODE are shown in hexadecimal format)

System action

FABCUR1 ends with an abend code of 3902.

User response

For further explanation of the error, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job.

FABC3903E   **FILE NUMBER IN AREA OUTPUT TABLE NOT FOUND IN FILE CONTROL TABLE**

Explanation

Program FABCUR1 found that the file number in the area output table (CCIAOUT) did not exist in the file control table.

System action

FABCUR1 ends with an abend code of 3903.

User response

Contact IBM Software Support.

FABC3905E   **AREA IN** *xxxxxxxx* **AREA** *nnnnn* **(AREANAME:** *areaname***) NOT DEFINED IN** *yyyyyyyy*

Explanation

The area specified is defined in OLDACB but not in NEWACB, or in the current active ACB but not in the pending ACB.

System action

Program FABCUR1 ends with an abend code of 3905.

User response

Make sure that the correct NEWACB data set or the pending ACB is specified. Correct the error, and rerun the job.

FABC3907E   **SUBTASK CONTROL ECB POSTED WITH UNEXPECTED REASON. GET UNEXPECTED COMPLETE CODE FROM SUBTASK**

Explanation

Program FABCUR1 received an unexpected termination code from the unload subtask.

System action

FABCUR1 ends with an abend code of 3907.

User response

Contact IBM Software Support.

FABC3908E   **INSUFFICIENT STORAGE: INCREASE REGION SIZE**

Explanation

An unload subtask could not open the VSAM data set for the area data set due to the storage shortage. Program FABCUR1 tried to dispatch the process to another unload subtask, but no subtasks were available.

System action

FABCUR1 ends with an abend code of 3908.

User response

Increase the REGION parameter on the EXEC statement and rerun the job.

FABC3909E   **UNLOCK FAILED FOR MESSAGE**

Explanation

Program FABCUR1 tried to unlock for message resources between subtasks, though they had not been locked.

System action

FABCUR1 ends with an abend code of 3909.

User response

Contact IBM Software Support.

FABC3910E   **UNEXPECTED STATUS CODE RETURNED FROM IMS**

Explanation

An unexpected IMS status code was returned during a DL/I call.

System action

This message is accompanied by message FABC3912E.

User response

Consult the description of the accompanying message

Module

FABCUR9A

FABC3911E   **CALL: <*call*> STATUS CODE: <*sc*>**

Explanation

An unexpected IMS status code was returned during a DL/I call.

System action

Processing ends with an abend code of 3911.

User response

Provide appropriate action in response to the status as described in *IMS Messages and codes Volume 1*, and resubmit the job.

Module

FABCUR9A

FABC3912E   **SEGMENT: <*segname*> CALL: <*call*> STATUS CODE: <*sc*> KEY: <*key*> REC#: <*record number*>**

Explanation

An unexpected IMS status code was returned during a DL/I call. REC# specifies the relative record within the UNLDREC file.

System action

Processing ends with an abend code of 3912.

User response

Provide appropriate action in response to the status as described in *IMS Messages and codes Volume 1*, and resubmit the job.

Module

FABCUR9A

FABC3914E   **- EXITRTN=FABCRPCX MUST BE SPECIFIED IN REORG MODE**

Explanation

The exit routine FABCRPCX was specified in the Change mode.

System action

Program FABCUR1 abends with user code of 3914.

User response

Remove the NEWACB DD statement, and rerun the job.

FABC3940E   **DIVISION FAILED - RPL POOL SIZE IS *zzzzzzzzz*: NUMBER OF IOVF BUFFERS IS 9*z***

Explanation

Program FABCUR3 found that the size of the RPL pool for the work data set obtained by GENCB macro was not divisible by the number of IOVF buffers. (The size of the RPL pool and the number of IOVF buffers are shown in decimal format.)

System action

FABCUR3 ends with an abend code of 3940.

User response

Contact IBM Software Support.

FABC3941E   **RDJFCB FAILED FOR DDNAME *ddname* (RC=*xx*)**

Explanation

Program FBCUR1/FABCUR3 issued an RDJFCB macro for the ddname specified. The macro failed. (Return code is shown in decimal format)

System action

FABCUR1/FABCUR3 ends with an abend code of 3941.

User response

To determine the cause of the problem specified by the RDJFCB ERROR reason code *yyyyyyyy*, see *DFSMS Macro Instructions for Data Sets* that explains the error and reason codes of the RDJFCB macro. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3942E	ROOT KEY NOT FOUND IN UTBL

Explanation

Program FABCUR3 found that a root key was not found in the UTBLs for disposing of a segment that had ICL FLAG (USRLCFG) on (X'FF'). The cause might be that there were no segments that had USERPFLG1 on (X'FF') in an unloaded file, and nothing was saved in UTBL.

System action

FABCUR3 ends with an abend code of 3942.

User response

If you create or modify the unloaded file, verify that the unloaded data set is correct. Then, rerun the job. If this situation persists, contact IBM Software Support.

FABC3943E	SEGMENT CODE *xxx* NOT FOUND IN UTBL

Explanation

Program FABCUR3 found different segment codes between the segment unloaded file that had USRPFLG1 on and the segment in UTBLs that had USRLCFLG on (X'FF'). (Segment code in UTBLs is shown in decimal format.)

System action

FABCUR3 ends with an abend code of 3943.

User response

If you created or modified the unloaded file, verify that the unloaded data set is correct. Then, rerun the job. If this situation persists, contact IBM Software Support.

FABC3944E	INSERT FAILED FOR DISPOSING SEGMENT AT THE BASE SECTION - INSERT POSITION NOT ANY OF BASE, DOVF OR IOVF

Explanation

Program FABCUR3 found that the insert position of the disposing segment at the base section was not BASE, DOVF or IOVF.

System action

FABCUR3 ends with an abend code of 3944.

User response

Contact IBM Software Support.

FABC3945E	PARENT SEGMENT CODE FIELD IN USR PREFIX IS INCORRECT FOR AREA NO: *nnnnn*, AREANAME: *areaname*. - USR REC# : *xxx,xxx,xxx* - SEGCODE : *xxx* - USR PARENT SEGCODE : *xxx* - DBD PARENT SEGCODE : *xxx*

Explanation

Program FABCUR3 detected that the segment code in the USRPSCD field of the unloaded segment record specified by the record number (REC#) was not correct. First child segment record must have its parent segment code in the USRPSCD field. Second or subsequent twin segment record must have its same segment code.

System action

FABCUR3 ends with an abend code of 3945.

User response

Check the content of the unloaded segment record specified by the record number (REC#), correct the value of the USRPSCD field, and rerun the job.

FABC3946E	INSERT FAILED FOR DISPOSING SEGMENT AT THE OVERFLOW SECTION - INSERT POSITION NEITHER A DOVF NOR IOVF

Explanation

Program FABCUR3 found that the insert position of the disposing segment at the overflow section was neither a DOVF nor IOVF.

System action

FABCUR3 ends with an abend code of 3946.

User response

Contact IBM Software Support.

FABC3948E **THE PLACE OF THE PARENT SEGMENT NOT ANY OF BASE, DOVF OR IOVF**

Explanation

Program FABCUR3 found that the place of the parent segment was not BASE, DOVF, or IOVF, when the insert segment's pointer was set into its parent segment.

System action

FABCUR3 ends with an abend code of 3948.

User response

Verify that the unloaded data set is correct. If you modified or created the unloaded file, the cause might be that the USRPSCD of the first occurrence of dependent segment is incorrect. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABC3950E **CALL FUNCTION "*function*" ERROR (FABCUR6/FABCUR7)**
- "*function*" CALL PROCESSED ALREADY
- "*function*" CALL NOT PROCESSED YET
- "INI*x*" CALL INITIATED BUT "GET*x*" CALL ISSUED
- STATUS "GB" RETURNED ALREADY
- "INIT" CALL DBNAME PARAMETER NOT PROVIDED
- "*function*" CALL UNKNOWN PARAMETER SPECIFIED
- "PUT" CALL NO I/O AREA SPECIFIED
- "PUT" CALL NO SEGMENT NAME FOUND
- "PUT" CALL SEGMENT DATA LENGTH TOO SHORT
- "PUT" CALL SEGMENT DATA LENGTH TOO LONG
- "PUT" CALL INVALID SEGMENT SUBSET POINTER
- "GET*x*" CALL NO STATUS CODE AREA SPECIFIED
- "GET*x*" CALL NO I/O AREA SPECIFIED
- "GET*x*" CALL NO EX. I/O AREA SPECIFIED
- "EOF" CALL UNKNOWN FUNCTION CODE PARAMETER SPECIFIED
- UNKNOWN FUNCTION CODE

Explanation

Program FABCUR6/FABCUR7 was called from the application but the call failed due to the reason described by the subtext.

System action

FABCUR6/FABCUR7 ends with an abend code of 3950.

User response

Correct the application program logic to call FABCUR6/7 correctly, and rerun the job.

FABC3951W **EXIT ROUTINE *exit-name* RETURNED STATUS CODE E1**
- first 80 bytes characters of the message that user exit routine returned
- subsequent 48 bytes characters of the message that user exit routine returned

Explanation

Program FABCUR1 unload subtask/FABCUR3/ FBCUR6/FABCUR7 got the status code E1 from the user exit routine exit-name specified by the EXITRTN= control statement.

System action

FABCUR1/FABCUR3/FABCUR6/FABCUR7 unload subtask sets an end-of-job return code of 4.

User response

Investigate why the user exit routine returned the status code E1. Correct the problem, and rerun the job.

FABC3951E **EXIT ROUTINE *exit-name* RETURNED STATUS CODE E2**

- **first 80 bytes characters of the message that user exit routine returned**
- **subsequent 48 bytes characters of the message that user exit routine returned**

Explanation

Program FABCUR1 unload subtask/FABCUR3/FBCUR6/FABCUR7 got the status code E2 from the user exit routine exit-name specified by the EXITRTN= control statement.

System action

FABCUR1/FABCUR3/FABCUR6/FABCUR7 unload subtask ends with an abend code of 3951.

User response

Investigate why the user exit routine returned the status code E2. Correct the problem, and rerun the job.

FABC3952E	EXIT ROUTINE *exit-name* RETURNED INVALID STATUS CODE *cc* (X'*xxxx*')

Explanation

Program FABCUR1/FABCUR3/FBCUR6/FABCUR7 unload subtask got the invalid status code specified by *cc* (X'*xxxx*') from the user exit routine exit-name specified by the EXITRTN= control statement.

System action

FABCUR1 unload subtask /FABCUR3/FABCUR6/FABCUR7 ends with an abend code of 3952.

User response

Investigate the logic of the user exit routine. Correct the exit routine, and rerun the job.

FABC3953E	ERROR IN CALL TO RANDOMIZER *xxxxxxxx subtext*

Explanation

Program FABCUR6 called the specified randomizer module to calculate the new area and RAP number values for a root segment. The values returned were invalid. One of the following subtexts is issued:

- FUNCTION: *function_name*
  - RETURN CODE: *xxxx* REASON CODE: X'*xxxxxxxx*'
- - INVALID AREA #, RAP # VALUES RETURNED FOR SEG AT RBA *xxxxxxxx*

(VALUES RETURNED: AREA #: *xxxxx*, RAP #: *xxx,xxx,xxx*)

"FUNCTION" and "REASON CODE" are issued only when FABCUR6 called the XCI randomizer routine.

System action

FABCUR6 ends with an abend code of 3953.

User response

Verify that the RMODLIB DD statement specifies the correct data set. Correct the errors, and rerun the job. If this situation persists, report it to database administration personnel.

FABC3954E	INPUT DATA ERROR (DDNAME: *ddname* REC# *zzz,zzz,zz9*)
	- SEGMENT CODE *zzzzz* NOT FOUND
	- HIERARCHY LEVEL INCORRECT (DURDBDFN: *nn* UNLOAD FILE RECORD: *nn*)
	- PARENT SEGMENT CODE INCORRECT (DURDBDFN: *nn* UNLOAD FILE RECORD: *nn*)
	- ROOT KEY LENGTH INCORRECT (DURDBDFN: *nn* UNLOAD FILE RECORD: *nn*)
	- DATA LENGTH ERROR (DURDBDFN MAX: *nnnnn* MIN: *nnnnn* UNLOAD FILE RECORD: *nnnnn*)
	- SUBSET POINTER INCORRECT
	- AREA *zzzzz* NOT DEFINED IN DMB
	- AREA INFORMATION RECORD FOR AREA *zzzzz* NOT FOUND
	- FIRST DATA NOT CORRECT AREA INFORMATION RECORD

Explanation

Self-explanatory. *zzzzz* is the area number.

System action

Program FABCUR7 ends with an abend code of 3954.

User response

Make sure that the correct data set is specified. If the data set is correct, make sure that the data set is sorted successfully. If input to FABCUR7 is a series of concatenated data sets, make sure that they are concatenated in an ascending area number order. Correct the error, and rerun the reload job.

FABC3955E   SEGMENT HIERARCHICAL SEQUENCE ERROR DETECTED

Explanation

Program FABCUR6 found that the segment provided by the application program was not in hierarchical sequence.

System action

FABCUR6 ends with an abend code of 3955.

User response

Verify that segment data used by the application program for FABCUR6 input is correct. Correct the errors, and rerun the job.

FABC3956E   EXIT ROUTINE *exit-name* RETURNED STATUS CODE *cc*(X'*zzzz*') BUT 'T2' EXPECTED

Explanation

Program FABCUR1 unload subtask got the status code specified by *cc* (X'*xxxx*') from the user exit routine *exit-name* specified on the EXITRTN= control statement. The subtask expected the status code T2 because the exit routine returned the T2 status code for the previous parent/twin segment.

System action

FABCUR1 unload subtask ends with an abend code of 3956.

User response

Investigate the logic of the user exit routine. Correct the exit routine, and rerun the job.

FABC3957E   - UNLOADED SEGMENT RECORD LENGTH EXCEEDS LRECL FOR DDNAME: *ddname*
        - USR LL: *nnnnn*
        - LRECL: *mmmmm*

Explanation

The length of the unloaded segment record is greater than LRECL of the output file specified by the *ddname*. A segment might be expanded by compression operation of the edit/compression routine and exceeded the maximum length that is defined in DBD.

System action

Program FABCUR1/FABCUR6 abends with user code of 3957.

User response

Increase the LRECL of the output data set specified by the *ddname* at least 10 bytes, and if specified, do the same for another output data set, and rerun the job.

FABC3958E   - AREA INFORMATION RECORD FOUND WHEN AREC=N WAS SPECIFIED.

Explanation

Program FABCUR3 found the area information record in an unloaded file when AREC=N was specified on the EXEC parameter.

System action

FABCUR3 abends with a user abend code of 3958.

User response

Specify a correct unloaded file or specify AREC=Y on the EXEC parameter, and rerun the job.

FABC3959E   - AREC=N IS NOT ALLOWED FOR SDEP=PHYSICAL WITH RELOCATION MODE

Explanation

AREC=N was specified with SDEP=PHYSICAL and RMODTYPE=S when both OLDACB DD and NEWACB DD were specified. Because it intends to relocate the SDEP segments at reload time, AREC=N is not allowed. An area information record is essential to relocate SDEP segments at reload.

System action

Program FABCUR1 abends with an abend code of 3959.

User response

Correct the combination of the EXEC parameter, control statements, and ACB DD statements, and rerun the job.

FABC3960E   - AREA INFORMATION RECORD FOUND WHEN AREA_INFORMATION_RECORD=NO WAS SPECIFIED

Explanation

Program FABCUR7 found the area information record in an unloaded file when the AREA_INFORMATION_RECORD=NO control statement was specified.

System action

FABCUR7 abends with user code of 3960.

User response

Specify a correct unloaded file or specify the AREA_INFORMATION_RECORD=YES control statement, and rerun the job.

FABC3961E	- SPECIAL RECORD FOUND WHEN AREC=Y WAS SPECIFIED

Explanation

Program FABCUR3 found the special record in unloaded segment file when AREC=Y was specified on EXEC parameter.

System action

FABCUR3 abends with user code of 3961.

User response

Specify a correct unloaded file or specify AREC=N on the EXEC parameter, and rerun the job.

FABC3962E	- SPECIAL RECORD FOUND WHEN AREA_INFORMATION_RECORD=YES WAS SPECIFIED

Explanation

Program FABCUR7 found the special record in unloaded segment file when AREA_INFORMATION_RECORD=YES was specified.

System action

FABCUR7 abends with user code of 3962.

User response

Specify a correct unloaded file or specify the AREA_INFORMATION_RECORD=NO control statement, and rerun the job.

FABC3989E	FIRST INSERTED SEGMENT NOT ROOT SEGMENT - AREA: *zzzzz* SEG-CD: *xxx* RBA: *xxxxxxxx*

Explanation

Program FABCUR3 found that the first inserted segment of the CI in the base section was not a root segment. If the unload file was not sorted before reloading, this abend could occur.

System action

FABCUR3 ends with an abend code of 3989.

User response

Verify that the UNLOAD FILE is sorted. If it is not, sort it and rerun the job.

FABC4095E	RECON ACCESS FAILED. *subtext*

Explanation

An error was detected in the RECON access processing. One of the following subtexts is issued:

- DBRC LIST COMMAND IS NOT COMPLETED. RC=*xxxxxxxx*
- SYSPRINT DD FOR DBRC LIST COMMAND IS SPECIFIED AS DUMMY
- INTERNAL ERROR OCCURRED
- FUNC=*ffffffff* RETURN CODE=*xxxxxxxx* REASON CODE=*xxxxxxxx*
  KEYS: DBD=*dbdname* DDN=*ddname*
  KEYTYPE=*xxxxxxxxxxxx*

System action

Program FABCUR1 or FABCUR3 ends with an abend code of 4095.

User response

Check the DBRC message preceding this message. Follow the response in that message, and rerun the job.

FABD messages

The following information is about messages and codes that begin with FABD.

FABD0100I   **OBTAINED DB DEFINITIONS FROM** *resource*

Explanation

This message indicates the resource (ACB library or IMS directory) where FABCRMIF or FABDRMIF obtained database definitions from.

System action

FABCRMIF or FABDRMIF continues processing.

User response

None. This message is informational.

FABD3650E   **DEVTYPE FAILED FOR DDNAME:** *ddname* **(RC =** *xx***)**

Explanation

Program FABADA1/FABCUR5/FABCUR7/FABCRMIF/ FABDRMI0 issued a DEVTYPE macro for the MODSTAT data set ddname specified. The macro failed. (Return code is shown in decimal format)

System action

FABADA1/FABCUR5/FABCUR7/FABCRMIF/FABDRMI0 ends with an abend code of 3650.

User response

To determine the cause of the problem specified by the DEVTYPE return code *xx*, see *DFSMS DFSMSdfp Advanced Services*, which explains the error return codes of the DEVTYPE macro. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABD3651E   **OPEN FAILED FOR DDNAME** *ddname*

Explanation

Program FABADA1/FABCUR5/FABCUR7/FABCRMIF/ FABDRMI0 issued an SVC 19 (OPEN) for the file associated with the MODSTAT data set DD statement specified. The OPEN was not successful.

System action

FABADA1/FABCUR5/FABCUR7/FABCRMIF/FABDRMI0 ends with an abend code of 3651.

User response

Make sure that the DD statement specified is present in the JCL stream being run, and that it properly specifies the correct data set. Correct any errors, and rerun the job.

FABD3652E   **I/O ERROR FOR INPUT DATA SET DDNAME** *ddname*

Explanation

Program FABADA1/FABCUR5/FABCUR7/FABCRMIF/ FABDRMI0 issued a GET for the MODSTAT data set ddname specified. The GET operation failed.

System action

FABADA1/FABCUR5/FABCUR7/FABCRMIF/FABDRMI0 ends with an abend code of 3652.

User response

Correct the errors, and rerun the job.

FABD3653E   **NO RECORD FOUND IN DATA SET DDNAME** *ddname*

Explanation

Program FABADA1/FABCUR5/FABCUR7/FABCRMIF/ FABDRMI0 found that the file for the MODSTAT data set ddname specified is empty.

System action

FABADA1/FABCUR5/FABCUR7/FABCRMIF/FABDRMI0 ends with an abend code of 3653.

User response

Make sure that the DD statement properly identifies the correct data set for the MODSTAT. Correct any errors, and rerun the job.

FABD3654E   **INCORRECT RECORD FOUND IN DATA SET DDNAME** *ddname*

Explanation

Program FABADA1/FABCUR5/FABCUR7/FABCRMIF/ FABDRMI0 found that the record for the MODSTAT data set ddname specified is incorrect.

System action

FABADA1/FABCUR5/FABCUR7/FABCRMIF/FABDRMI0 ends with an abend code of 3654.

User response

Make sure that the DD statement properly identifies the correct data set for the MODSTAT. Correct any errors, and rerun the job.

FABD3655E **INVALID CONTROL CARD ENCOUNTERED**
- UNKNOWN KEYWORD
- VALUE FOR THE *keyword* KEYWORD IS INVALID

Explanation

The user-supplied control statement was found to contain one or more errors.

System action

FABCRMIF or FABDRMIF ends with an abend code of 3655.

User response

Correct the control statement and rerun the job.

FABD3656E **IMS TOOLS CATALOG INTERFACE *function* FUNCTION (DEFINITION=CURRENT| PENDING) FAILED**
- RETURN CODE: *rc*, REASON CODE: *rsn*

Explanation

IMS Tools Catalog Interface ended with an error. *function* shows the function code of IMS Tools Catalog Interface. The return code and reason code from IMS Tools Catalog Interface are shown in *rc* and *rsn*, respectively.

System action

The job ends with an abend code of U3656.

User response

If the function is OPEN, check if the correct high-level qualifier of the bootstrap data set is specified for the IMSCATHLQ keyword. Otherwise, contact IBM Software Support.

FABD3661E **INVALID PARM LIST IN CALL TO FABCRMIF/FABDRMI0**

- UNKNOWN VALUE FOR "FUNCTION" PARAMETER
- 'FUNC' = "INIT"; "DBDNAME" PARAMETER NOT FOUND
-'FUNC' = "CALC"; TOO FEW PARM''S SPECIFIED

Explanation

Program FABCRMIF/FABDRMI0 determined that the parameter list specified by the calling program was incorrect.

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

Correct the CALL specifications in the program being used to invoke FABCRMIF/FABDRMI0. Rerun the job.

FABD3662E **PROCESSING FAILED FOR *xxxxxx* MEMBER**
- *xxxxxx* MEMBER IS NOT A DEDB DMB
- MEMBER NAME NOT EQUAL DEFINED DATABASE NAME
- *ddname* DD STATEMENT NOT FOUND
- MEMBER NOT FOUND IN *xxxxxx*
- INSUFFICIENT STORAGE
- INVALID PARAMETER LIST IN CALL TO FABCGDD
- IMS LEVEL OF ACB MEMBER *xxxxxx* NOT SUPPORTED
- ACB MEMBER VRSDSRF IS NOT SAME IMS LEVEL AS DBT LIBRARY
- NO. SEGS DEFINED EXCEEDS ALLOWED MAX.
- RC = *XX*

Explanation

To obtain DMB information of the database that is being processed, program FABCRMIF/FABDMRIF called either the FABAGDD program to obtain DMB information from the ACBLIB library or the FABAGDD2 program to obtain DMB information from the IMS directory. The return code indicates that the attempt to do so was unsuccessful.

System action

FABCRMIF/FABDMRIF ends with an abend code of 4011.

User response

Ensure that the files associated with the DD statements ACBLIB or the IMS directory are correctly specified. Also, ensure that the ACBGEN and DBDGEN or IMS catalog population were correctly performed for the database being analyzed. Correct any errors, and rerun the job.

If "- INVALID PARAMETER LIST IN CALL TO FABAGDD" is shown, contact IBM Software Support.

FABD3663E   **ERROR IN "CALC" CALL TO FABCRMIF/FABDRMI0 - "INIT"-IALIZATION HAS NOT BEEN PERFORMED**

Explanation

Self-explanatory.

System action

Program FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

Correct the call sequence in the program being used to invoke FABCRMIF/FABDRMI0, and rerun the job.

FABD3664E   *macro-name* **FAILED FOR DMB MEMBER** *member-name* **FOR DDNAME** *ddname* **(RC =** *rr* **: REASON =** *zz***)**

Explanation

Program FABCRMIF/FABDRMI0 issued the macro specified (*macro-name*) to access the DMB member (*member-name*) in the data set specified (*ddname*). The z/OS return code and reason code mean that the attempt was unsuccessful.

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

For further information, see the *MVS Programming: Assembler Services Reference*. Correct the problem, and rerun the job. If this situation persists, contact IBM Software Support.

FABD3665E   **OPEN FAILED FOR DDNAME** *ddname*

Explanation

Program FABCRMIF/FABDRMI0 issued an SVC 19 (OPEN) for the file associated with the DD statement specified. The OPEN was not successful.

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

Make sure that the DD statement specified is present in the JCL stream being run, and that it properly specifies the correct data set. Correct any errors, and rerun the job.

FABD3666E   **LOAD FAILED FOR RANDOMIZER ROUTINE** *rmodname* **(ABEND S***xxx* **/ REASON CODE** *xxxxxxxx***)**

Explanation

Program FABCRMIF/FABDRMI0 issued an SVC 8 (LOAD) to bring into storage a copy of the randomizer routine specified in the DMB for the database being processed. The return code received from OS means that the attempt failed. The return code (reason code) and abend code returned by OS are shown in the message.

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

For further explanation of the error, see the *MVS Programming: Assembler Services Reference*. Correct any errors, and rerun the job.

FABD3667E   **DEVTYPE FAILED FOR DDNAME:***ddname* **RC =** *xx***)**

Explanation

Program FABCRMIF/FABDRMI0 issued a DEVTYPE macro for the ddname specified. The macro failed. (Return code is shown in decimal format).

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

To determine the cause of the problem specified by the DEVTYPE return code *xx*, see *DFSMS DFSMSdfp Advanced Services*, which explains the error return codes of the DEVTYPE macro. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

FABD3668E	ERROR IN "INIT" CALL TO FABCRMIF "INIT" -IALIZATION HAS ALREADY BEEN PERFORMED FOR DMB MEMBER *member-name*

Explanation

Self-explanatory.

System action

Program FABCRMIF ends with an abend code of 4011.

User response

Correct the call sequence in the program that invokes FABCRMIF, and rerun the job.

FABD3669E	ERROR IN "INIT" CALL TO FABCRMIF. MORE THAN 16 DMB MEMBERS REQUESTED FOR DMB MEMBER *member-name*

Explanation

Program FABCRMIF found that 16 DMB members have already been initialized.

System action

FABCRMIF ends with an abend code of 4011.

User response

FABCRMIF supports up to 16 different DMB members. Correct the program invoking FABCRMIF, and rerun the job.

FABD3670E	ERROR IN CALL TO RANDOMIZER ROUTINE *rmodname* FUNCTION: *function_name* - RETURN CODE: *xxx* REASON CODE: X'*xxxxxxxx*'

Explanation

FABCRMIF/FABDRMI0 invoked the XCI randomizer routine with a function code of 'INIT', 'CALC', or 'TERM', and then returned with error. "FUNCTION"

and "REASON CODE" are issued only when FABCRMIF/FABDRMI0 invoked the XCI randomizer routine. Function is one of 'INITIALIZATION CALL', 'RANDOMIZING CALL', or 'TERMINATION CALL'.

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

Make sure that the RMODLIB DD statement properly identifies the correct data set, and that the randomizer routine has been correctly added, assembled, and link-edited. Correct any errors, and rerun the job.

FABD3671E	LOAD FAILED FOR LOAD MODULE FABCRMIX/FABCRMIZ (ABEND S*xxx*/REASON CODE *xxxxxxxx*)

Explanation

Program FABCRMIF/FABDRMI0/FABDRMIF issued an SVC 8 (LOAD) to bring into storage a copy of FABCRMIX/FABCRMIZ. The return code received from OS means that the attempt failed. The return code (reason code) and abend code returned by OS are shown in the message.

System action

FABCRMIF/FABDRMI0/FABDRMIF ends with an abend code of 4011.

User response

Make sure that the IMS HP Fast Path Utilities load module library is concatenated to the JOBLIB/STEPLIB DD statement. If it is concatenated, see the *MVS Programming: Assembler Services Reference* for a further explanation of the problem. Correct any errors, and rerun the job.

FABD3672E	INVALID *keyword*= KEYWORD SITE DEFAULT ERROR DETECTED

Explanation

An incorrect parameter was specified for the FABAOP1M, FABCOP1M, FABCOP3M, FABCOP6M, or FABCOP9M macro keyword.

System action

The assemble step ends with a return code of 8.

User response

Correct the error, and rerun the job.

FABD3673E   **TABLESET=DSECT/SYSTEM AND ANY OTHER KEYWORDS ARE MUTUALLY EXCLUSIVE**

Explanation

TABLESET=DSECT/SYSTEM cannot be specified with any other keyword parameters. TABLESET=DSECT/SYSTEM is for system use only. TABLESET=DSECT/SYSTEM must not be specified to define the site default table.

System action

The assemble step ends with a return code of 8.

User response

Correct the error, and rerun the job.

FABD3674E   **NO KEYWORD IS SPECIFIED FOR SITE DEFAULT TABLE**

Explanation

No keyword is specified for the FABAOP1M, FABCOP1M, FABCOP3M, FABCOP6M, or FABCOP9M macro.

System action

The assemble step ends with a return code of 8.

User response

Correct the error, and rerun the job.

FABD3675I   *keyword=* **PARAMETER IS IGNORED BECAUSE DEFAULT VALUE IS SPECIFIED**

Explanation

A *keyword=* parameter was specified that is the same as the system default value. The FABAOP1M, FABCOP1M, FABCOP3M, FABCOP6M, or FABCOP9M macro skips generating an entry of the site default table for *keyword=*.

System action

The assemble step continues normal processing.

User response

None. This message is informational.

FABD3676E   *macro-name* **MACRO SPECIFIED MORE THAN ONCE**

Explanation

The FABAOP1M, FABCOP1M, FABCOP3M, or FABCOP9M macro was specified more than once.

This macro must be specified only once when TABLESET=USER is specified (default value).

System action

The assemble step ends with a return code of 8.

User response

Correct the error, and rerun the job.

FABD3677E   **[IMSCOMP=/DLICOMP= | AREA_INFORMATION_RECORD=/ AIR= | INPUT=/FORMAT=] KEYWORDS ARE MUTUALLY EXCLUSIVE**

Explanation

The specified keywords cannot be used together.

System action

The assemble step ends with a return code of 8.

User response

Correct the error, and rerun the job.

FABD3678E   **WHEN LRECL=SEGTFMT IS SPECIFIED, FORMAT=TFMT HAS TO BE SPECIFIED**

Explanation

You have to specify FORMAR=TFMT when LRECL=SEGTFMT is specified.

System action

The assemble step ends with a return code of 8.

User response

Correct the error, and rerun the job.

FABD3690E   **INVALID MESSAGE NUMBER DETECTED MESSAGE NO.** *nnnn* **-ERROR NO.(HEX) IS IN REG15**

Explanation

While processing an error message, an invalid message number in register 15 was detected. This is an internal error.

System action

Program FABADA1/FABCUR1 ends with an abend code of 3690.

User response

Contact IBM Software Support.

FABD3693E **OBTAIN FAILED FOR VOL=SER=***aaaaaa*

Explanation

Program FABADA1 or FABCUR1 issued an OBTAIN macro to reserve the volume *aaaaaa*. The attempt was unsuccessful.

System action

FABADA1 or FABCUR1 ends with an abend code of 3693.

User response

Check the volume, fix the problem, and rerun the job.

FABD3694E **INSUFFICIENT STORAGE FOR** *aaaa* **- INCREASE REGION SIZE**

Explanation

Program FABADA1 or FABCUR1 issued a GETMAIN macro to allocate storage for the purpose of *aaaa*. The attempt was unsuccessful.

System action

FABADA1 or FABCUR1 ends with an abend code of 3694.

User response

Check the region size, increase the REGION parameter in the EXEC statement for FABADA1 or FABCUR1 as required, and rerun the job.

FABD3700E **TWO USABLE RECON DATA SETS ARE NOT PROVIDED FOR ERRORS RECON1 DD:** *subtext* **RECON2 DD:** *subtext* **RECON3 DD:** *subtext*

Explanation

For the explanation of the case for each subtext, see the corresponding explanation.

System action

For the system action of the case for each subtext, see the corresponding system action.

User response

For the user response of the case for each subtext, see the corresponding user response.

---

subtext **DEVTYPE FAILED (RC=***xx***)**

Explanation

Program FABADA1 or FABCUR1 issued a DEVTYPE macro for the DD name specified. The macro failed. (Return code is shown in hexadecimal format.)

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

To determine the cause of the problem, see *DFSMS DFSMSdfp Advanced Services*, which explains the error return codes of the DEVTYPE macro. Correct any errors and rerun the job. If this situation persists, contact IBM Software Support.

---

subtext **RDJFCB FAILED (RC=***xx***)**

Explanation

Program FABADA1 or FABCUR1 issued an RDJFCB macro for the DD name specified. The macro failed. (Return code is shown in hexadecimal format.)

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

To determine the cause of the problem, see *DFSMS DFSMSdfp Advanced Services*, which explains the error return codes of the RDJFCB macro. Correct any errors and rerun the job. If this situation persists, contact IBM Software Support.

---

subtext **SHOWCB FAILED (REG15=***xx* **REG0=***yy***)**

Explanation

Program FABADA1 or FABCUR1 issued a VSAM SHOWCB macro. The macro failed with return code *xx* and reason code *yy*. (The content of the register 15 and register 0 are shown in hexadecimal format.)

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

To determine the cause of the problem, see *DFSMS Macro Instructions for Data Sets*, which explains the error return codes of the SHOWCB macro. Correct any errors and rerun the job. If this situation persists, contact IBM Software Support.

subtext	OPEN FAILED (RC=*xx*(*XX*) REASON CODE=*yyy*(*YY*))

Explanation

Program FABADA1 or FABCUR1 issued a VSAM OPEN macro for the RECON data set specified. The macro failed with return code *xx* in decimal format (*XX* in hexadecimal format) and reason code *yyy* in decimal format (*YY* in hexadecimal format).

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

To determine the cause of the problem, see *DFSMS Macro Instructions for Data Sets*, which explains the error return codes of the VSAM OPEN macro. Correct any errors and rerun the job. If this situation persists, contact IBM Software Support.

subtext	MODCB FAILED (REG15=*xx* REG0=*yy*)

Explanation

Program FABADA1 or FABCUR1 issued a VSAM MODCB macro. The macro failed with return code *xx* and reason code *yy*. (The content of register 15 and register 0 are shown in hexadecimal format.)

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

To determine the cause of the problem, see *DFSMS Macro Instructions for Data Sets*, which explains the error return codes of the MODCB macro. Correct any errors and rerun the job. If this situation persists, contact IBM Software Support.

subtext	INCORRECT IMS RELEASE LEVEL RECON DATA SET IS USED

Explanation

The data set used for the DD name was not a correct IMS release level of the RECON data set.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Specify the correct IMS release level of the RECON data set and rerun the job.

subtext	NO RECON HEADER/EXTENSION RECORD FOUND

Explanation

The data set used for the DD name was not a RECON data set or a correct IMS release level of the RECON data set.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Specify the correct IMS release level of the RECON data set and rerun the job.

subtext	RECON DATA SET GET FAILED (RC=*xx*(*XX*) REASON CODE=*yyy*(*YY*))

Explanation

An attempt to GET a RECON record failed. The error return code and reason code are shown. The macro failed with return code *xx* in decimal format (*XX* in hexadecimal format) and reason code *yyy* in decimal format (*YY* as hexadecimal format).

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

To determine the cause of the problem, see *DFSMS Macro Instructions for Data Sets*, which explains the error return codes of the GET macro. Correct any errors, and rerun the job. If this situation persists, contact IBM Software Support.

subtext	OBTAIN FAILED FOR VOL=SER=*aaaaaa* (RC=*xx*)

Explanation

Program FABADA1 or FABCUR1 issued an OBTAIN macro to reserve volume *aaaaaa*. The attempt failed.

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Check the volume, fix the problem, and rerun the job.

subtext	INSUFFICIENT STORAGE FOR: *aaaa* - INCREASE REGION SIZE

Explanation

Program FABADA1 or FABCUR1 issued a GETMAIN macro to allocate storage for the purpose of *aaaa*. The attempt failed.

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Check the region size, increase the REGION parameter in the EXEC statement for FABADA1 or FABCUR1 as required, and rerun the job.

subtext	MINVERS LEVEL IS NOT CORRECT FOR RECON DATA SET

Explanation

The data set used for the DD name was not a correct IMS release level of the MINVERS mode RECON data set.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Specify the correct IMS release level of the MINVERS mode RECON data set, and rerun the job.

subtext	DYNALLOC FAILED (RC=*rrrr* RSN=*eeeeiiii*)

Explanation

DYNALLOC macro failed. Here, *rrrr* is the return code from SVC99, *eeee* is the SVC99 ERROR contents, and *iiii* is the SVC99 INFO contents. The return code and the reason code are described in the *MVS Programming: Authorized Assembler Services Reference*.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	DYNALLOC FAILED (DFSMDA MEMBER LOAD FAILED)

Explanation

The DFSMDA member was not found for the RECON data set.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	DYNALLOC FAILED (INCORRECT DFSMDA MEMBER)

Explanation

Member *ddname*1/2/3 was loaded as a DFSMDA member, but it does not have the correct DFSMDA format. The eye catcher 'MDA' is not found in the member.

System action

FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	DYNALLOC FAILED (BLDL FAILED (RC=*rrrr*))

Explanation

Program FABADA1 or FABCUR1 issued a BLDL macro for the RECON data set *ddname*1/2/3 specified in JOBLIB, STEPLIB, or SYSLIB. The macro failed. Here, *rrrr* is the return code from the macro. The return code is described in *DFSMS Macro Instructions for Data Sets*.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	DYNAMIC DEALLOCATION FAILED (RC=*rrrr* RSN=*eeeeiiii*)

Explanation

An attempt for dynamic deallocation of the ddname failed. Here, *rrrr* is the return code from SVC99, *eeee* is the SVC99 ERROR contents, and *iiii* is the SVC99 INFO contents. The return code and the reason code are described in the *MVS Programming: Authorized Assembler Services Reference*.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	DUPLICATED

Explanation

The same RECON data set is specified.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	USABLE AS COPY*n*

Explanation

This RECON data set was accepted as a COPY*n* (*n*=1 or 2), but the valid RECON data set used as a pair was not specified.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

subtext	NOT USED

Explanation

These are the possible reasons for the error:

- The RECON DD is not specified.
- The RECON data set was specified as DUMMY or NULLFILE.
- The RECON data set was empty.
- This is a spare RECON data set.

System action

Program FABADA1 or FABCUR1 ends with an abend code of 3700.

User response

Correct the error and rerun the job.

FABD3701E	ERROR IN "TERM" CALL TO FABCRMIF - FUNC="TERM"; "DBDNAME" PARAMETER NOT FOUND - TERMINATION HAS BEEN PERFORMED ALREADY OR NOT FOUND FOR DMB MEMBER *xxxxxxxx*

Explanation

Program FABCRMIF/FABDRMI0 determined that the parameter list specified by the calling program was incorrect.

System action

FABCRMIF/FABDRMI0 ends with an abend code of 4011.

User response

Correct the CALL specifications in the program being used to invoke FABCRMIF/FABDRMI0. Then rerun the job.

Gathering diagnostic information

Before you report a problem with IMS HP Fast Path Utilities to IBM Software Support, you need to gather the appropriate diagnostic information.

Provide the following information for all IMS HP Fast Path Utilities problems:

- A clear description of the problem and the steps that are required to re-create the problem
- The version of IMS that you are using and the version of the operating system that you are using
- A complete log of the job
- Snap dump generated in the HFPABEND data set

  The HFPABEND data set is generated only when the FPA process ends abnormally. If the HFPABEND DD is not specified in the JCL, FPA dynamically allocates the data set by using SYSOUT=*.

- A Load Module/Macro APAR Status report

Use the Diagnostics aid to create a Load Module/Macro APAR Status report. For more information, see the *IMS Fast Path Solution Pack: IMS High Performance Fast Path Utilities User's Guide*.

Notices

This information was developed for products and services offered in the US. This material might 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, MD-NC119
Armonk, NY 10504-1785
US

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

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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, MD-NC119
Armonk, NY 10504-1785
US

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.

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 actual people or business enterprises 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.

Programming interface information

This information is intended to help the customer use the functional capabilities of IMS High Performance Fast Path Utilities of IBM IMS Fast Path Solution Pack for z/OS.

However, this information also documents Product-Sensitive programming interface information and associated guidance information.

Product-Sensitive programming interfaces are provided to allow the customer installation to perform tasks such as tailoring, monitoring, modification, or diagnosis of this IBM product. Use of such interfaces creates dependencies on the detailed design or implementation of the IBM product. Product-Sensitive interfaces should be used only for these specialized purposes. Because of their dependencies on detailed design and implementation, it is to be expected that programs written to such interfaces may need to be changed in order to run with new product releases or versions, or as a result of service.

Product-Sensitive programming interface information is explicitly identified where it occurs, as an introductory statement to a chapter or section that is entirely Product-Sensitive programming interface information.

Product-Sensitive Programming Interface and Associated Guidance Information is identified where it occurs, either as an introductory statement to a chapter or section or by the following marking:

PSPI

Product-sensitive programming interface and associated guidance information...

PSPI

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml.

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.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

IBM.

Product Number: 5698-FPP