

IBM IMS Recovery Solution Pack for z/OS  
Version 2 Release 1

*IMS Database Recovery Facility User's  
Guide and Reference*



**Note:**

Before using this information and the product it supports, read the information in [“Notices” on page 359](#).

**Third Edition (December 2020)**

This edition applies to Version 2 Release 1 of IBM® IMS Recovery Solution Pack for z/OS®: IMS Database Recovery Facility (program number 5655-ISR) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-8441-01.

© **Copyright International Business Machines Corporation 2000, 2020.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

|                                                                            |           |
|----------------------------------------------------------------------------|-----------|
| <b>About this information.....</b>                                         | <b>ix</b> |
| <b>Part 1. IMS Database Recovery Facility overview.....</b>                | <b>1</b>  |
| Chapter 1. IMS Database Recovery Facility overview.....                    | 3         |
| What's new in IMS Database Recovery Facility.....                          | 3         |
| IMS Database Recovery Facility features and benefits.....                  | 4         |
| IMS Database Recovery Facility terminology.....                            | 6         |
| Product architecture.....                                                  | 7         |
| Product architecture with integrated auxiliary utilities.....              | 9         |
| Service updates and support information.....                               | 9         |
| Product documentation and updates.....                                     | 9         |
| Accessibility features.....                                                | 10        |
| Chapter 2. Product requirements and restrictions.....                      | 13        |
| Product requirements.....                                                  | 13        |
| Hardware and software requirements.....                                    | 13        |
| Operational requirements.....                                              | 13        |
| Product restrictions.....                                                  | 14        |
| Seasonal time change timestamp restriction.....                            | 14        |
| Database and area restrictions.....                                        | 14        |
| Esoteric name tape drive restriction.....                                  | 15        |
| Migration considerations.....                                              | 16        |
| Coexistence.....                                                           | 16        |
| Fallback recommendations.....                                              | 16        |
| Region size considerations.....                                            | 16        |
| Chapter 3. Integrated auxiliary utilities overview.....                    | 19        |
| Terminology for integrated auxiliary utilities processing.....             | 19        |
| Integrated auxiliary utilities usage.....                                  | 20        |
| Processing address space architecture and configuration.....               | 21        |
| Control flow and the RAUX control task .....                               | 25        |
| SYSOUT reports, WTO messages, and return codes.....                        | 27        |
| Write to operator (WTO) messages.....                                      | 27        |
| SYSOUT reports.....                                                        | 28        |
| Viewing the integrated auxiliary utilities reports.....                    | 28        |
| WTO capture services.....                                                  | 29        |
| Additional data set processing (RPTTYPE=APP and RPTTYPE=SEP only).....     | 29        |
| <b>Part 2. Configuring IMS Database Recovery Facility.....</b>             | <b>31</b> |
| Chapter 4. Configuring the product environment.....                        | 33        |
| Creating the environmental control member: FRXDRFxx.....                   | 33        |
| Processing System Level Backup (SLB) image copies.....                     | 34        |
| Setting up the environment to support SLB image copies.....                | 35        |
| Command reference for SLB API activation utility.....                      | 36        |
| Example BSYIN control statements.....                                      | 37        |
| Environmental control statements.....                                      | 40        |
| Chapter 5. Configuring the integrated auxiliary utilities environment..... | 63        |

|                                                                                     |            |
|-------------------------------------------------------------------------------------|------------|
| Specifying utility control statements.....                                          | 63         |
| Reference: Utility control statement parameters.....                                | 63         |
| Reference: Utility control statements.....                                          | 69         |
| Global environmental control statement (UTILGBL).....                               | 70         |
| Integrated auxiliary utility parameters for IMS Index Builder.....                  | 70         |
| Integrated auxiliary utility parameters for the Build Index function of FPA.....    | 75         |
| Integrated auxiliary utility parameters for IMS HP Image Copy.....                  | 78         |
| Integrated auxiliary utility parameters for IMS HP Pointer Checker.....             | 87         |
| Chapter 6. Configuring JCL statements and procedures.....                           | 93         |
| Master address space JCL.....                                                       | 93         |
| DDNAME reference for master address space.....                                      | 93         |
| Report DDNAME statement reference.....                                              | 96         |
| Recovery sort subordinate address space JCL.....                                    | 98         |
| IMS Index Builder, DFSPRECO, and Build Index function of FPA address space JCL..... | 100        |
| IMS High Performance Pointer Checker JCL.....                                       | 101        |
| <b>Part 3. Using IMS Database Recovery Facility.....</b>                            | <b>103</b> |
| Chapter 7. Using batch recovery control statements.....                             | 105        |
| Batch recovery control statement commands.....                                      | 105        |
| Reference: ADD command.....                                                         | 105        |
| ADD command examples.....                                                           | 109        |
| Reference: REMOVE command.....                                                      | 111        |
| Reference: START command.....                                                       | 113        |
| Reference: Timestamp format.....                                                    | 119        |
| Chapter 8. Using the integrated auxiliary utilities.....                            | 123        |
| Utility control statement specification.....                                        | 123        |
| Reference: ADD IB() command.....                                                    | 124        |
| ADD IB() command for IMS Index Builder.....                                         | 124        |
| ADD IB() command for the Build Index function of FPA.....                           | 129        |
| Reference: ADD IC() command.....                                                    | 131        |
| Reference: ADD PC() command.....                                                    | 140        |
| Return code handling.....                                                           | 144        |
| Chapter 9. Reference: DBRC commands for database recovery.....                      | 147        |
| Using the INIT.DBDSGRP command.....                                                 | 147        |
| Using the CHANGE.DBDSGRP command.....                                               | 147        |
| Using the LIST.DBDSGRP command.....                                                 | 148        |
| Using the NOTIFY.RECOV command.....                                                 | 148        |
| Chapter 10. Performing a database recovery.....                                     | 149        |
| 1. Defining recovery scope.....                                                     | 149        |
| 2. Taking databases offline.....                                                    | 151        |
| 3. Building a recovery list.....                                                    | 151        |
| 4. Defining physical data sets.....                                                 | 152        |
| 5. Starting recovery.....                                                           | 152        |
| 6. Using integrated auxiliary utility processing.....                               | 152        |
| Calling IMS HP Image Copy.....                                                      | 153        |
| Calling IMS HP Pointer Checker.....                                                 | 153        |
| Calling IMS Index Builder or DFSPRECO.....                                          | 153        |
| Message and report output handling.....                                             | 154        |
| 7. Performing post-recovery activities.....                                         | 154        |
| <b>Part 4. Administration and management.....</b>                                   | <b>157</b> |

|                                                                                |            |
|--------------------------------------------------------------------------------|------------|
| Chapter 11. Administering IMS Database Recovery Facility.....                  | 159        |
| Database administration checklist.....                                         | 159        |
| Managing sources of input.....                                                 | 159        |
| Input data sets from earlier product releases.....                             | 160        |
| DBRC controls.....                                                             | 160        |
| Image copy input types.....                                                    | 160        |
| Recovering with a nonstandard image copy.....                                  | 160        |
| Change accumulation input data sets.....                                       | 161        |
| Log data sets usage for recovery.....                                          | 161        |
| Image copy frequency.....                                                      | 162        |
| Recovery from secondary sources.....                                           | 162        |
| Managing output.....                                                           | 162        |
| Stacked image copy output.....                                                 | 162        |
| Managing the environment.....                                                  | 162        |
| Data integrity.....                                                            | 163        |
| How security is assigned.....                                                  | 163        |
| Database availability during recovery.....                                     | 163        |
| Starting databases and areas.....                                              | 163        |
| DEDBs with MADS.....                                                           | 163        |
| Restart considerations.....                                                    | 164        |
| Auditability considerations.....                                               | 164        |
| Spill data space management.....                                               | 164        |
| Chapter 12. Managing performance.....                                          | 165        |
| Performance improvement.....                                                   | 165        |
| Control of input parallelism.....                                              | 166        |
| Virtual tape system management.....                                            | 166        |
| Tape device availability management.....                                       | 166        |
| Chapter 13. Managing time stamp recovery.....                                  | 169        |
| Time stamp recovery without PITR.....                                          | 169        |
| Time stamp recovery with PITR.....                                             | 169        |
| RECOVGRP.....                                                                  | 170        |
| Time stamp recovery with PITR and UOR.....                                     | 170        |
| Time stamp recovery with PITR and database updates on multiple UOR.....        | 171        |
| <b>Part 5. Usage scenarios for optimization.....</b>                           | <b>173</b> |
| Chapter 14. Setting up a basic IMS Database Recovery Facility environment..... | 175        |
| Setting up the IMS Recovery Solution Pack.....                                 | 175        |
| Installing IMS Recovery Solution Pack libraries.....                           | 175        |
| Reviewing the IMS Recovery Solution Pack installation checklist.....           | 176        |
| Customizing the procedures and JCL members.....                                | 176        |
| Tips and best practices.....                                                   | 177        |
| Chapter 15. Recover the database to the current time.....                      | 179        |
| Creating and running an IMS Database Recovery Facility job.....                | 179        |
| Reviewing the output.....                                                      | 180        |
| Chapter 16. Validating recovery by using IMS HP Pointer Checker .....          | 181        |
| Creating and running an IMS Database Recovery Facility job.....                | 181        |
| Reviewing the output.....                                                      | 182        |
| Chapter 17. Taking an incremental image copy of offline databases.....         | 185        |
| Identifying the databases to perform incremental image copy.....               | 185        |
| Creating and running an IMS Database Recovery Facility job.....                | 186        |

|                                                                                                              |            |
|--------------------------------------------------------------------------------------------------------------|------------|
| Reviewing the output.....                                                                                    | 187        |
| Common errors and problems.....                                                                              | 188        |
| Optional: Verifying the ICR in DBRC.....                                                                     | 188        |
| Next steps.....                                                                                              | 189        |
| Chapter 18. Taking an incremental image copy of an online database to the current time or point-in-time..... | 191        |
| Identify the databases to perform incremental image copy.....                                                | 191        |
| Creating and running your IMS Database Recovery Facility job.....                                            | 192        |
| Reviewing the output.....                                                                                    | 193        |
| Common errors and problems.....                                                                              | 194        |
| Optional: Verifying the ICR in DBRC.....                                                                     | 195        |
| Next steps.....                                                                                              | 196        |
| Chapter 19. Using IMS Index Builder to rebuild indexes during recovery.....                                  | 197        |
| Creating and running an IMS Database Recovery Facility job.....                                              | 197        |
| Reviewing the output.....                                                                                    | 198        |
| Common errors and problems.....                                                                              | 199        |
| Chapter 20. Recovering HALDB databases that use online reorganization.....                                   | 201        |
| Recovering HALDB databases "into an OLR" .....                                                               | 201        |
| Recovering HALDB databases "prior" to an OLR.....                                                            | 202        |
| Recovering HALDB databases "after" an OLR .....                                                              | 203        |
| <b>Part 6. Troubleshooting.....</b>                                                                          | <b>205</b> |
| Chapter 21. Runtime messages (FRD).....                                                                      | 207        |
| Chapter 22. Abend codes.....                                                                                 | 263        |
| Reason codes for service error abend ABENDU0384.....                                                         | 263        |
| Reason codes for logic error abend ABENDU0385.....                                                           | 269        |
| Chapter 23. Batch condition codes.....                                                                       | 273        |
| Chapter 24. Gathering diagnostic information.....                                                            | 275        |
| <b>Part 7. Reference: Utilities and reports.....</b>                                                         | <b>277</b> |
| Chapter 25. Recovery job statistics report.....                                                              | 279        |
| Chapter 26. Sample reports.....                                                                              | 283        |
| Output report content.....                                                                                   | 283        |
| Sample reports processing.....                                                                               | 284        |
| Example report: Image copy or incremental image copy output.....                                             | 284        |
| Example report: Point-in-time recovery.....                                                                  | 290        |
| Example report: Full point-in-time recovery.....                                                             | 295        |
| Example report: Verify of the point-in-time recovery.....                                                    | 298        |
| Example report: VERIFY(ALLOC) recovery.....                                                                  | 302        |
| Example report: VERIFY(OPEN) recovery for a point-in-time.....                                               | 305        |
| Example report: Full recovery using environment control statements.....                                      | 308        |
| Example report: Full recovery with IC() and PC() utilities.....                                              | 311        |
| <b>Part 8. Reference: Base Primitive Environment (BPE).....</b>                                              | <b>317</b> |
| Chapter 27. BPE commands.....                                                                                | 319        |
| Specifying BPE command parameters.....                                                                       | 319        |
| BPE TRACETABLE commands.....                                                                                 | 319        |

|                                              |            |
|----------------------------------------------|------------|
| DISPLAY TRACETABLE command.....              | 319        |
| DISPLAY TRACETABLE command output.....       | 321        |
| UPDATE TRACETABLE command.....               | 322        |
| UPDATE TRACETABLE command output.....        | 323        |
| Chapter 28. BPE messages and codes.....      | 325        |
| BPE messages.....                            | 325        |
| BPE service return codes.....                | 342        |
| BPE user abend codes.....                    | 350        |
| <b>Part 9. Reference: General.....</b>       | <b>355</b> |
| Chapter 29. How to read syntax diagrams..... | 357        |
| <b>Notices.....</b>                          | <b>359</b> |
| <b>Index.....</b>                            | <b>363</b> |





## About this information

---

IBM IMS Recovery Solution Pack for z/OS: IMS Database Recovery Facility (also referred to as IMS Database Recovery Facility) is an IMS tool that allows you to simultaneously recover multiple database data sets, HALDB partitions, and Fast Path areas.

This user's guide provides instructions for using IMS Database Recovery Facility.

To use the procedures in this user's guide, you must have already installed IBM IMS Recovery Solution Pack for z/OS by completing the SMP/E installation process that is documented in the *Program Directory for IMS Recovery Solution Pack for z/OS*, GI13-4315, which is included with the product. You must also perform the post-installation steps as outlined in the *IMS Recovery Solution Pack: Overview and Customization*, SC27-8440.

These topics are designed to help database administrators, system programmers, application programmers, and system operators perform the following tasks:

- Understand the capabilities of the functions that are associated with IMS Database Recovery Facility
- Install and operate IMS Database Recovery Facility
- Customize your IMS Database Recovery Facility environment
- Diagnose and recover from IMS Database Recovery Facility problems
- Use IMS Database Recovery Facility with other IMS products

To use these topics, you should have a working knowledge of:

- The z/OS operating system
- ISPF
- SMP/E
- IMS

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



---

# Part 1. IMS Database Recovery Facility overview

IBM IMS Recovery Solution Pack for z/OS: IMS Database Recovery Facility (also referred to as IMS Database Recovery Facility) is an IMS tool that allows you to simultaneously recover multiple database data sets, HALDB partitions, and Fast Path areas.

The topics in this section provide you with an overview of the IMS Database Recovery Facility product.

**Topics:**

- [Chapter 1, “IMS Database Recovery Facility overview,” on page 3](#)
- [Chapter 2, “Product requirements and restrictions,” on page 13](#)
- [Chapter 3, “Integrated auxiliary utilities overview,” on page 19](#)



---

# Chapter 1. IMS Database Recovery Facility overview

IBM IMS Recovery Solution Pack for z/OS: IMS Database Recovery Facility (also referred to as IMS Database Recovery Facility) is an IMS tool that allows you to simultaneously recover multiple database data sets, HALDB partitions, and Fast Path areas.

IMS Database Recovery Facility is one of several IBM tools that can help you manage backup and recovery operations for your IMS databases in the event of a system outage or application failure.

## Topics:

- [“What's new in IMS Database Recovery Facility” on page 3](#)
- [“IMS Database Recovery Facility features and benefits” on page 4](#)
- [“IMS Database Recovery Facility terminology” on page 6](#)
- [“Product architecture” on page 7](#)
- [“Product architecture with integrated auxiliary utilities” on page 9](#)
- [“Service updates and support information” on page 9](#)
- [“Product documentation and updates” on page 9](#)
- [“Accessibility features” on page 10](#)

---

## What's new in IMS Database Recovery Facility

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.

### SC27-8441-02 - December 2020

- APAR PI96280 - update to the MAXSEC parameter (environmental control statement)
- APAR PI96280 - updates to messages FRD3302I and FRD6124A
- APAR PH12756 - new KEYLABEL parameter (utility control statement)
- APAR PH26679 - update to the READNUM parameter (environmental control statement)
- APAR PH31569 - IMS OSAM data set encryption support

### SC27-8441-01 - November 2018

- New recovery job statistics report chapter.
- Corrected syntax and descriptions for ADD IB() INPUT.
- APAR PI92958 - timestamp format updates.
- Updates to Configuring the integrated auxiliary utility environment.
- APAR PI86870 - new SAS option is added to the RPTTYPE keyword on the REPORT parameter.
- APAR PI86870 - updates to: SYSOUT reports, WTO messages, and return codes.
- APAR PI86870 - change to ASPREF and NUM options of SORTPARM parameter (environmental control statements).
- APAR PI86870 - update to PCPREF (Integrated auxiliary utility parameters for IMS HP Pointer Checker).
- APAR PI86870 - new HPIO parameter (environmental control statements).
- APAR PI86870 - new abend codes ABENDU0384-033, ABENDU0384-034.
- APAR PI86870 - updates to messages FRD4301I and FRD4221I.

## SC27-8441-00 - October 2016

- Maintenance updates for IMS Recovery Solution Pack V2.1.
- Information pertaining to the inclusion of IMS Recovery Expert as a product belonging to the IMS Recovery Solution Pack V2.1.
- Information pertaining to the inclusion of IMS Solution Packs: Data Sensor as a product belonging to the IMS Recovery Solution Pack V2.1.
- Information about using IMS Tools Setup for initial product installations, first-time users, and product evaluations.

## IMS Database Recovery Facility features and benefits

---

The IMS Database Recovery Facility allows you to simultaneously recover multiple database data sets, HALDB partitions, and fast path areas in a single job step. IMS Database Recovery Facility also integrates with several other IMS Tools products as integrated auxiliary utilities to allow you to create new image copies, rebuild indexes, and validate recovered databases.

Running IMS Database Recovery Facility does not require the presence of an active IMS DB/TM or DBCTL subsystem.

The phrases "database data set" or "database data sets" refer to full function database data sets, HALDB partitions, and fast path area data sets, unless otherwise noted.

By using the IMS Database Recovery Facility you can reduce operational complexity and the impact of database recovery on system resources. It simplifies your database recovery process by eliminating the need to run a separate recovery job for each database data set that requires recovery.

Recovery using the IMS Database Recovery Facility reduces the time that broken databases are unavailable by processing input data in parallel and recovering multiple database data sets simultaneously. Further time saving is possible because running change accumulation before recovery of a shared database is optional when you use this product.

In addition, IMS Database Recovery Facility provides the capability to automatically invoke the following utilities during the recovery process:

- IMS High Performance Image Copy to create image copies
- IMS Index Builder to build the primary and all secondary indexes for a database.
- IMS HALDB DFSPRECO utility to rebuild the HALDB Primary Index data set and Index List data set
- IMS High Performance Pointer Checker and the IMS High Performance Fast Path Utilities to hash check database pointers
- IMS Library Integrity Utility to verify the consistency of DBD Library used in a recovery
- IMS Tools Knowledge Base to enable IMS High Performance Pointer Checker, IMS Index Builder, and IMS Fast Path Advanced Tool to store reports in the IMS Tools KB Repository.
- The Build Index function of IMS Fast Path Advanced Tool (also referred to as FPA), a tool within the IMS Fast Path Solution Pack for z/OS. You can invoke this utility to build Fast Path secondary indexes.

IMS Database Recovery Facility is invoked by creating JCL statements and product control statements and submitting the job for execution. It can also be invoked using JCL statements and product control statements specified in a started task.

IMS Database Recovery Facility provides you with the ability to create and store recovery scenarios before they are needed. By doing so, you can audit recovery processes and jobs to ensure that as many database failure scenarios as possible are covered. This saves valuable time during a database outage by having the necessary recovery jobs already defined before they are required.

### Supported database types

You can use the IMS Database Recovery Facility to recover these database types:

- Full Function: HDAM, HIDAM, HISAM, SHISAM, and INDEX

- HALDB: PHIDAM, PHDAM, and PSINDEX
- Fast Path: DEDB: ADS and MADS

The phrases database data set or database data sets refer to full function database data sets, HALDB partitions, and fast path area data sets, unless otherwise noted.

For a list of database types that you cannot recover with IMS Database Recovery Facility, see [“Product restrictions”](#) on page 14.

## Recovery features

You can perform the following recovery tasks with the IMS Database Recovery Facility:

- Simultaneously recover multiple full-function databases, HALDB partitions, and DEDB areas in a single pass of the change accumulation and log data sets.
- Perform time stamp recovery by using time stamps that are not restricted to allocation boundaries.

The ability to recover to any point in time allows you to recover IMS databases to the same point in time as other databases in your environment.

- Recover database data sets and areas without needing to preprocess IMS log data with a change accumulation utility.
- Maintain efficient control of input to the IMS Database Recovery Facility by specifying how many log, change accumulation, and image copy data sets can be read in parallel.
- Specify whether or not recovery is to continue for the set of database data sets being processed when recovery for one of them fails.
- Integrate recovery tasks into a single job step, which can include:
  - Creating image copies
  - Validating recovery through pointer checker
  - Rebuilding your primary and secondary index data sets
  - Rebuilding your HALDB index list data sets

## Database management features

You can also perform the following database management tasks by using the IMS Database Recovery Facility:

- Automatic /DBR of databases in the recovery list prior to performing recovery.
- Automatic startup of databases in the recovery list on all applicable IMS systems or on the IMS that performs recovery after recovery completes successfully.
- Create image copy data sets during the recovery process to eliminate the need to follow recovery with a separate image copy process.
- Create index data sets as part of the recovery process, including primary, secondary, and HALDB index list data sets, in order to eliminate the need to follow recovery with a separate index build process.
- Create offline, duplicate copies of a database that can be used for tasks such as application program testing, audit, and database recovery.
- Validate data during the recovery process, by running a pointer checker against the database, which eliminates the need to follow a recovery with a separate validation process invocation.
- Verify that the DBD being used for recovery and for the integrated auxiliary utilities is the one that was used to create the database.
- Automatically delete and define database data sets as part of the recovery process, which eliminates the need to perform these tasks before recovery.
- Generate new incremental image copy data sets by applying database updates from a log data set or change accumulation data set to existing image copy data. This allows a new image copy to be created without needing to access the database.

- Verify the existence and availability of recovery input data sets by attempting to dynamically allocate and open the input data sets without needing to run an actual recovery.

## IMS Database Recovery Facility terminology

---

IMS Database Recovery Facility includes several unique terms that you should understand before you begin to use the tool.

### **Change Accumulation (CA) data set**

Data set created by a database Change Accumulation utility run that contains all of the database updates between the last image copy and the time the CA utility completed execution.

There are two basic kinds of CA data sets:

- Complete CA data sets (based on available system log data sets (SLDS)).
- Incomplete CA data sets that are created at the time that CA utility was generated and when the required SLDS are not yet available.

### **Disaster recovery site (DR)**

Location at which data sets necessary to recover IMS databases are located in the event of a disastrous IMS outage such as natural calamities, fires, power failures, or other disasters.

### **Full recovery**

Recovery of a database using the most current information available: image copy (IC), CA, Logs.

### **Point-in-time recovery (PITR)**

Timestamp input to which a database can be restored. The database was typically allocated and being accessed at the selected recovery timestamp.

### **Recovery point**

Timestamp that can be used as input to database recovery utilities such as IMS Database Recovery Facility or the standard IMS Recovery utility DFSURDB0, as a time to which a database can be restored.

### **System Level Backup (SLB)**

The System Level Backup (SLB) is created by IMS Recovery Expert and is an instantaneous backup of the IMS system including the IMS database data sets.

The SLB can be used to recover all or part of the IMS system and can create image copies that are used as input to IMS Database Recovery Facility recovery functionality.

### **Timestamp recovery (TSR)**

Timestamp input to either IMS Database Recovery Facility or another standard database recovery program such as DFSURDB0.

The database is typically offline at this timestamp.

In the RPID documentation, TSR refers to non-PITR timestamps, though PITR is actually a special case of TSR.

### **Verify recovery assets (VERIFY)**

The VERIFY function interrogates the RECON data sets and first generates a list of the recovery assets that are needed in order to recover one or more databases. When this list is generated, this feature provides options to perform the following tasks:

- Print the list.
- Allocate each asset to ensure that it exists.
- Open each recovery asset that exists to ensure that it is valid.

VERIFY uses information from the RECON data sets, the IMS Recovery Expert repository for System Level Backups (SLBs), and the MVS™ catalog.



## Product architecture

---

The IMS Database Recovery Facility product consists of several components that operate together to provide the product functionality.

This information describes the architecture of the basic product only. For the architectural description that includes the integrated auxiliary utilities, see [“Product architecture with integrated auxiliary utilities” on page 9](#).

The components related to the IMS Database Recovery Facility are:

- BPE (base primitive environment)

A set of system services that are needed by the IMS Database Recovery Facility.

BPE provides the infrastructure upon which the IMS Database Recovery Facility, is built.

- DBRC

A facility that maintains records of recovery-related resources in the RECON data sets.

- IMS Database Recovery Facility master address space (MAS)

An MVS address space in which the controlling components of IMS Database Recovery Facility reside.

It might also be referred to as the master address space.

The log and change accumulation data sets are read in this address space.

- IMS Database Recovery Facility sort subordinate address space or spaces (RSS)

One or more MVS address spaces which are started by the master address space.

These address spaces sort log and change accumulation data and restore image copy data.

This diagram shows the components and the processes of the IMS Database Recovery Facility for the basic product. It does not include information related to the integrated auxiliary utilities.

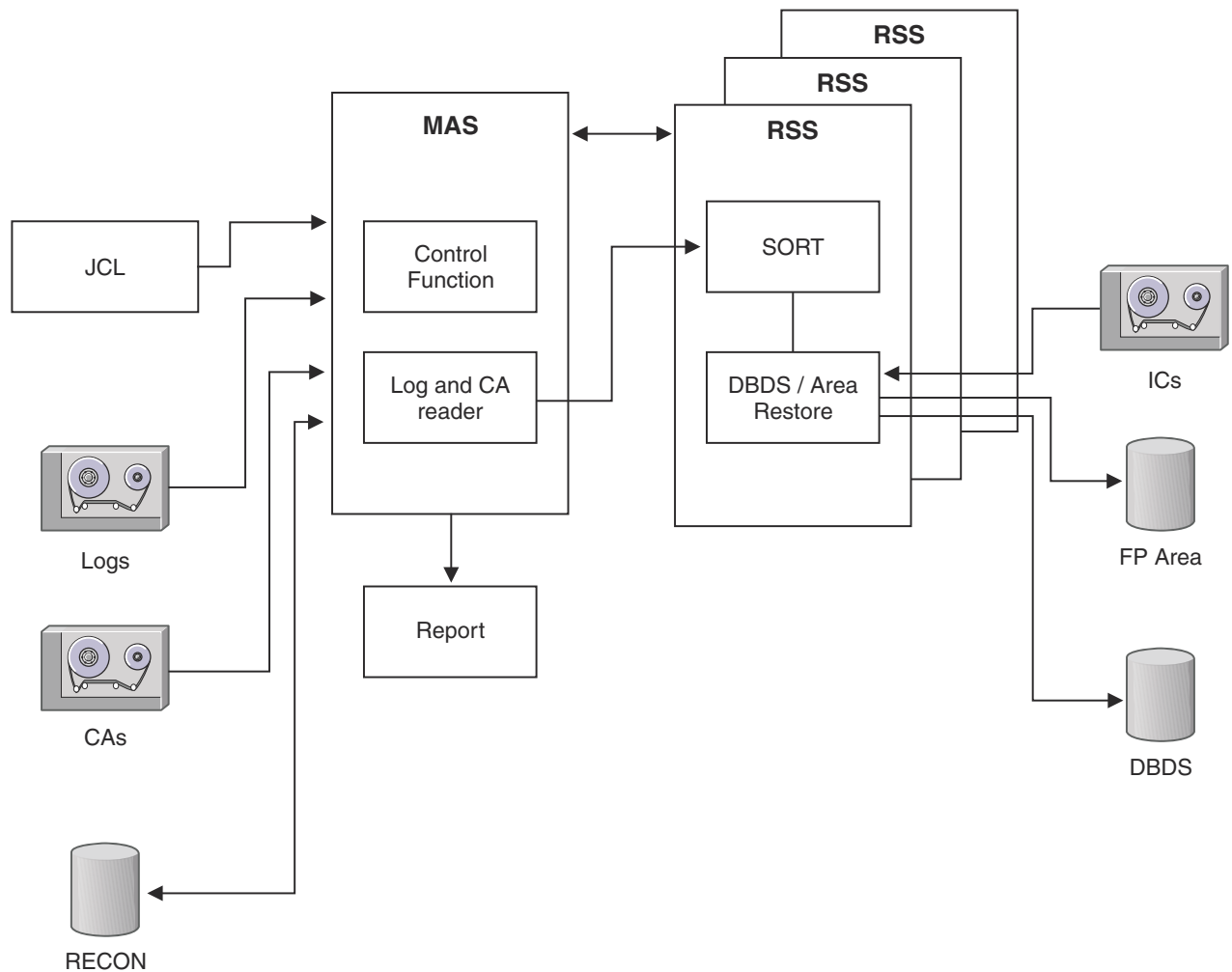


Figure 1. Components and processes of the IMS Database Recovery Facility for the basic product

Component descriptions:

#### JCL

The Job Control Language stream that initiates the IMS Database Recovery Facility.

#### Logs

IMS RLDS (recovery log data set) or SLDS (system log data set) that are used as input to IMS Database Recovery Facility

#### CAs

Change accumulation data sets that are used as input to IMS Database Recovery Facility

#### RECON

DBRC recovery control data sets that are used as input to and output from IMS Database Recovery Facility

#### MAS

The IMS Database Recovery Facility master address space in which the control function and log and CA reader processes are performed

#### Report

Output recovery summary and statistical reports from IMS Database Recovery Facility

#### RSS

One or more recovery sort subordinate address spaces in which log records are sorted and recovery processes are performed.

#### ICs

Image copy data sets that are used as input to the recovery process

**FP Area**

Fast Path area that is the output from the recovery process

**DBDS**

Full function or HALDB database data set that is the output from the recovery process

## Product architecture with integrated auxiliary utilities

---

IMS Database Recovery Facility provides the functionality to perform a recovery and execute the integrated auxiliary utilities in a single-step batch job.

By using a single-step batch job, CPU, and I/O Service Unit are reduced, JCL and control statements are simplified, and the Integrated Auxiliary Utility reports, messages, and return codes are integrated into a single output.

The integrated auxiliary utilities include:

- IMS High Performance Image Copy to create image copies
- IMS Index Builder to build the primary and all secondary indexes for a database
- IMS High Performance Pointer Checker and the IMS High Performance Fast Path Utilities to hash check recovered databases
- IMS HALDB DFSPRECO utility to rebuild the HALDB Primary Index data set and Index List data set
- IMS Library Integrity Utility to verify that the DBD used for the recovery is the same as the one that was used to create the database
- The Build Index function of FPA to build Fast Path secondary indexes

When any of the integrated auxiliary utilities are invoked additional address spaces can also be started. Here is a list of the address spaces that can be started by the IMS Database Recovery Facility to process the integrated auxiliary utilities:

- Utility address space for IMS Index Builder
- Utility address space for DFSPRECO
- Utility address space for the Build Index function of FPA.
- Utility address space for IMS High Performance Pointer Checker

These address spaces guide the processing of the various integrated auxiliary utilities. The integrated auxiliary utilities can also initiate other utility-specific address spaces. For more information about the address spaces, see the *IMS Recovery Solution Pack: Overview and Customization*.

For more detailed information about the architecture and flow for the integrated auxiliary utilities, refer to Chapter 3, “Integrated auxiliary utilities overview,” on page 19.

## 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:

[http://www.ibm.com/support/entry/portal/Overview/Software/Information\\_Management/IMS\\_Tools](http://www.ibm.com/support/entry/portal/Overview/Software/Information_Management/IMS_Tools)

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

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

<http://www.redbooks.ibm.com>

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

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

### Receiving documentation updates automatically

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to <http://www.ibm.com/support/mysupport>
2. Enter your IBM ID and password, or create one by clicking **register now**.
3. When the My Notifications page is displayed, click **Subscribe** to select those products that you want to receive information updates about. The IMS Tools option is located under **Software > Information Management**.
4. Click **Continue** to specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

### How to send your comments

Your feedback helps IBM to provide quality information. Send any comments that you have about this book or other IMS Tools documentation to [comments@us.ibm.com](mailto:comments@us.ibm.com). Include the name and version number of the product and the title and number of the book. If you are commenting on specific text, list the location of the text (for example, a chapter, topic, or section title).

## Accessibility features

---

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

The major accessibility features in this product enable users to perform the following activities:

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

– *z/OS TSO/E User's Guide*

These guides describe how to use the ISPF interface, including the use of keyboard shortcuts or function keys (PF keys), include the default settings for the PF keys, and explain how to modify their functions.



---

## Chapter 2. Product requirements and restrictions

The following topics discuss the product requirements and restrictions for IMS Database Recovery Facility.

### Topics:

- [“Product requirements” on page 13](#)
- [“Product restrictions” on page 14](#)
- [“Migration considerations” on page 16](#)
- [“Coexistence” on page 16](#)
- [“Fallback recommendations” on page 16](#)
- [“Region size considerations” on page 16](#)

---

### Product requirements

The topics in this section discuss the specific software, hardware, operational, and storage area requirements for an IMS Database Recovery Facility installation.

### Topics:

- [“Hardware and software requirements” on page 13](#)
- [“Operational requirements” on page 13](#)

### Hardware and software requirements

IMS Database Recovery Facility is installed by using SMP/E and standard RECEIVE, APPLY, and ACCEPT processing.

Complete information about installation requirements, prerequisites, and procedures for IMS Recovery Solution Pack for z/OS is located in the *Program Directory for IMS Recovery Solution Pack for z/OS*, GI10-8819.

IMS Database Recovery Facility can run on any z/OS hardware environment that supports the required software.

IMS Database Recovery Facility is designed to run with IMS V14 and IMS V15.

### Operational software prerequisites

IMS Database Recovery Facility requires that the latest version of IBM IMS Tools Base for z/OS be installed and that the following components be installed and configured:

- IMS Tools Knowledge Base
- IMS Tools Online System Interface
- IMS Tools Generic Exits

### Operational requirements

The topics in this section discuss operational requirements for IMS Database Recovery Facility.

Before you use the IMS Database Recovery Facility, you must satisfy the following operational requirements:

- The following IMS Database Recovery Facility data set components must be registered with DBRC:
  - All database data sets to be recovered
  - Log data sets
  - Change accumulation data sets

- Image copy data sets
- If you use a security management system to protect database data sets, the user ID that is associated with the IMS Database Recovery Facility job or started task must have the appropriate authorization to these data sets.

## Product restrictions

---

IMS Database Recovery Facility has certain restrictions regarding databases, areas, XRF, RSR, and the use of esoteric name tape drives.

### Topics:

- [“Seasonal time change timestamp restriction” on page 14](#)
- [“Database and area restrictions” on page 14](#)
- [“Esoteric name tape drive restriction” on page 15](#)

## Seasonal time change timestamp restriction

Recovery times **MUST** be specified in UTC format when databases have to be recovered to a recovery point timestamp prior to a seasonal time change and recovery is being performed after a seasonal time change.

This is because the UTC offset value when the recovery job is run is different from the UTC offset value of the recovery point timestamp.

## Database and area restrictions

Ensure that the databases with elements that are being recovered by the IMS Database Recovery Facility are unavailable to IMS processing.

You must ensure database and area unavailability by issuing the /DBRECOVERY command to any online IMS with active allocations of the databases being recovered. You must also ensure that there are no active batch jobs that are accessing the databases that are being recovered.

IMS Database Recovery Facility can be used in combination with IMS High Performance Image Copy to create a fuzzy IC at a point in time where the database is online. The fuzzy IC can be used in a subsequent recovery, though some of the logs before the IC stop time would be used to account for the uncommitted updates that were not on the fuzzy IC.

You cannot use the IMS Database Recovery Facility to recover the following database types:

- GSAM, SHSAM, HSAM, and MSDB
- HALDB PHIDAM primary indexes\*
- ILDSs (Indirect List Data Sets)\*

\* HALDB PHIDAM primary indexes and ILDSs must be recovered with the IMS HALDB Index and ILDS Rebuild Utility, DFSPREC0, or an equivalent tool.

However, you can have the IMS Database Recovery Facility invoke DFSPREC0 after recovery has completed in order to rebuild the PHIDAM primary index and ILDS.

Also, IMS Index Builder V3.1 has the ability to rebuild these indexes when IMS Index Builder V3.1 is invoked by IMS Database Recovery Facility.

If any database data sets that are being recovered have DFSMSDSS SAMEDS format image copies, the image copy is first restored and then log and change accumulation data are read and applied.

Restrictions on recovering encrypted databases, and how the IMS Database Recovery Facility environment control statements affect whether the KEYLABEL information is inherited to the recovered data sets:

- If the VSAM data set (including VSAM LDS for OSAM of IMS FF databases (HALDB and non-HALDB)) to be recovered is encrypted with KEYLABEL defined in the data class, the new recovered data set is



encrypted with the same KEYLABEL since the IMS Database Recovery Facility creates a new data set with the same data class.

- On the other hand, if the data class does not have a KEYLABEL setting and is encrypted with the KEYLABEL parameter specified by the DEFINE CLUSTER command, the KEYLABEL information is not inherited and the newly created recovery data set is not encrypted.

See the table below for how the IMS Database Recovery Facility environment control statements affect whether the KEYLABEL information is inherited to the recovered data sets:

*Table 1. Relationship between the environmental control statements and whether or not the KEYLABEL information is inherited*

| Environmental control statements |              | DBDS to be recovered        | Recovered DBDS           |
|----------------------------------|--------------|-----------------------------|--------------------------|
| OUTPUT                           | DBDSL (DISP) | Data set encryption         | Data set encryption      |
| PRO                              | OLD          | Y (predefined PRO DBDS)     | Y                        |
|                                  | NEW          | Y with KEYLABEL             | N (see Note)             |
|                                  |              | Y with data class           | Y                        |
|                                  | PDS          | Y                           | Y with IDCAMS statements |
| DUP                              | OLD          | Y (predefined DUP DBDS)     | Y                        |
|                                  | NEW          | Y with KEYLABEL             | N (see Note)             |
|                                  |              | Y with data class           | Y                        |
|                                  | PDS          | Y                           | Y with IDCAMS statements |
| BOTH                             | OLD          | Y (predefined PRO/DUP DBDS) | Y                        |
|                                  | NEW          | Y with KEYLABEL             | N (see Note)             |
|                                  |              | Y with data class           | Y                        |
|                                  | PDS          | Y                           | Y with IDCAMS statements |

**Note:** When you execute the IMS Database Recovery Facility, KEYLABEL information can be inherited by specifying the data class in which KEYLABEL is defined in the DATAC option of the DBDSL $_{nnn}$  parameter of the environmental control statement.

## Esoteric name tape drive restriction

When IMS log archive JCL specifies an esoteric name in the UNIT= parameter, the tape addresses to be used are defined by the definition of that name in the HCD (hardware configuration definition).

The specified esoteric name is used to search for eligible devices and the device type is recorded to the RECON for each LOG data set. Device type is an IBM-supplied name that identifies a device by its machine type and model. For example: 3480 or 3400-5.

When IMS Database Recovery Facility dynamically allocates a cataloged data set, the device type is used for allocation. The set of devices defined for this device type name can be fewer or more devices than the set described by the original esoteric name.

For example, if the esoteric name LOGTAPES includes just the five 3490 device addresses 3B0-3B4, LOGTAPES will be searched and the resulting device type 3490 is recorded in the RECON. Device type 3490 can include many more devices than just 3B0-3B4.

When these archived logs are specified to IMS Database Recovery Facility for use in the recovery process, other similar tape device addresses might be called for in the mount requests. Allocation with UNIT=3490 will select available 3490 units that might or might not be within the LOGTAPES esoteric group.

## Migration considerations

The topics in this section discuss conditions for migrating IMS Database Recovery Facility.

In order to call the index building, image copy creation, pointer checking, and DBD library integrity checking functions of IMS Database Recovery Facility, you must install the appropriate product specified in the following table.

For information on how to install the products, refer to the product's installation documentation.

*Table 2. Products required for IMS Database Recovery Facility index building, image copy, and pointer checking functions*

| Function                                            | Product                                                                                                                                     |
|-----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Index building - IB() keyword                       | Index Builder, IMS DFSPRECO Utility, and the Build Index function of FPA.                                                                   |
| Image copy creation - IC() keyword                  | High Performance Image Copy                                                                                                                 |
| Pointer checking - PC() keyword                     | High Performance Pointer Checker (full function and HALDB) High Performance DEDB Pointer Checker from the Basic Fast Path Tools (fast path) |
| DBD library integrity checking - LIU@GOPT() keyword | IMS Library Integrity Utility                                                                                                               |

## Coexistence

The IMS Database Recovery Facility provides coexistence by handling back-level log, change accumulation, and image copy records according to the following list.

- IMS Database Recovery Facility processes IMS log data sets in the format as they are created by any currently supported IMS version.
- IMS Database Recovery Facility processes IMS change accumulation data sets in the format as they are created by any currently supported IMS version.
- IMS Database Recovery Facility processes IMS image copy data sets in the format as they are created by any currently supported IMS version.

## Fallback recommendations

If IMS Database Recovery Facility does not operate satisfactorily, fallback by uninstalling it and reverting to your previous recovery tool or utility.

## Region size considerations

The region size used in production environments is typically larger than the size shown in the examples.

For recoveries that process large amounts of data, you might need to increase the region size to avoid abnormal or early termination that might result from storage shortages.

When setting the region size parameter in the startup JCL for both MAS and RSS, use the following procedures. These procedures will determine the probable number of log records to be processed during recovery and will set a region size for the master and subordinate address spaces:

1. Check the summary report for a prior recovery of the database data sets currently being recovered.

If a summary report is not available, you can use reports from recoveries of other databases.

If you use such alternate reports, they should reflect recoveries with similar update patterns as the databases currently being recovered.

You should also use reports from recoveries of other HIDAM, PHDAM, or Fast Path areas, for example, rather than using databases with organizations that are very different from the ones being recovered.

2. Add records processed for log data sets listed in the summary report.
3. Determine the average number of the processed records.
4. Run a recovery with the VERIFY function to determine the number of log data sets to be processed.
5. Multiply the average number of records found in Step 3 by the number of log data sets found in Step 4.
6. Use the following table to determine the value for the region size:

| <i>Table 3. Guidelines for determining region size for master and subordinate address spaces</i> |                    |
|--------------------------------------------------------------------------------------------------|--------------------|
| <b>Number of records processed</b>                                                               | <b>Region size</b> |
| <35 million log records                                                                          | 32M - 256M         |
| Between 35 million and 50 million log records                                                    | At least 256M      |
| >50 million log records                                                                          | At least 512M      |



---

## Chapter 3. Integrated auxiliary utilities overview

The integrated auxiliary utilities (IAUs) work in combination with IMS Database Recovery Facility. The following topics provide diagrams, processing descriptions, and configuration recommendations.

IMS Database Recovery Facility has a 'single-step' solution that allows you to perform a recovery and include in the recovery the activation of the integrated auxiliary utilities (IAUs). Traditionally, the activities of the IAUs have been done post-recovery, in separate batch jobs or multiple job steps subsequent to the recovery.

IMS Database Recovery Facility is now able to include these activities in a single-step batch job. There are several advantages to the single-step solution, including CPU and I/O Service Unit reduction, JCL and control statement simplification, and integration of Integrated Auxiliary Utility reports, messages and return codes.

### Topics:

- [“Terminology for integrated auxiliary utilities processing” on page 19](#)
- [“Integrated auxiliary utilities usage” on page 20](#)
- [“Processing address space architecture and configuration” on page 21](#)
- [“Control flow and the RAUX control task ” on page 25](#)
- [“SYSOUT reports, WTO messages, and return codes” on page 27](#)

---

### Terminology for integrated auxiliary utilities processing

IMS Database Recovery Facility includes several unique terms used in describing integrated auxiliary utilities processing.

The following acronyms are used in describing Integrated Auxiliary Utility processing.

#### **IBSS**

IMS Index Builder sort subordinate address space

#### **ILDS**

Indirect list data set. In a HALDB, an IMS system index data set.

#### **MAS**

IMS Database Recovery Facility master address space

#### **RAUX**

Recovery auxiliary utility control task

#### **RSS**

IMS Database Recovery Facility recovery sort subordinate address space

#### **SAS**

Subordinate address space, a generic term that refers to any address space that might have been started by IMS Database Recovery Facility

#### **IB-UAS**

IMS Database Recovery Facility utility address space for Index Builder

#### **PR-UAS**

IMS Database Recovery Facility utility address space for DFSPRECO

#### **PC-UAS**

Address space for IMS High Performance Pointer Checker DMB Analyzer (FABPATHx)

#### **FS-UAS**

Utility address space for the Build Index function of FPA

## Integrated auxiliary utilities usage

---

IMS Database Recovery Facility operates with the Integrated Auxiliary Utilities under the listed constraints.

### **IMS High Performance Image Copy**

Used for database image copy generation, including incremental image copy (ICR). Taking image copies is a usual part of backup and recovery procedures.

Use the IC keyword of the ADD command to take an IC of a recovered database data set, including those in a CAGROUP or RECOVGRP. Only IMS High Performance Image Copy image copies are generated.

The following types of image copies are not generated:

- IMS Standard Image Copy Utility image copies
- IMS Online Image Copy Utility image copies
- IMS Image Copy 2 Utility image copies
- IMS HPIC Flash Copy image copies

### **IMS utility DFSPRECO**

Used for rebuilding HALDB partitions PHIDAM primary indexes and ILDS data sets.

These entities are not recoverable and DFSPRECO must be used to rebuild them. Use the HALDB keyword of the UTILGBL control statement to activate DFSPRECO.

See the next topic, [“Rebuilding indexes and ILDSs” on page 21](#).

### **Build Index function of FPA**

Used for rebuilding Fast Path secondary indexes.

Use the IB keyword of the ADD command or the BLD\_SECONDARY keyword on the UTILGBL control statement to activate the Build Index function of FPA. This is useful in cases where the indexes are not recoverable or not marked as recoverable in the RECONS.

### **IMS Index Builder**

Used for rebuilding Full Function primary indexes and all secondary indexes, including those of HALDB partitions.

Use the IB keyword of the ADD command or the BLD\_PRIMARY and BLD\_SECONDARY keywords on the UTILGBL control statement to activate IMS Index Builder. This is useful in cases where the indexes are not recoverable or not marked as recoverable in the RECONS.

For more information, see [“Rebuilding indexes and ILDSs” on page 21](#).

### **IMS High Performance Pointer Checker and IMS Fast Path Basic Tools DEDB Pointer Checker**

Used for hash pointer checking of full function databases and fast path databases, respectively. These ensure the integrity of the recovered database data sets.

Use the PC keyword of the ADD command to activate hash pointer checking. Ensure that all logically-related data sets are available to Pointer Checker by including the PC keyword on the corresponding ADD commands; otherwise, Pointer Checker returns multiple pointer errors to the MAS.

However, this might be acceptable for your purposes; for example, if you only want the Free Space Report from Pointer Checker. Note that Pointer Checker does not allow for hash pointer checking of HALDB primary indexes or ILDS data sets and therefore is not called, even if they are rebuilt by DFSPRECO as part of the recovery.

### **IMS Library Integrity Utility**

Used for verifying the database descriptor (DBD) in the DBDLIB used for recovery and any invoked Integrated Auxiliary Utilities.

The DBD in the DBDLIB is the same as the DBD used for building the database. This prevents database corruption due to recovering a database using the wrong DBD, such as a DBD that is used for application testing or one that is outdated.

## Rebuilding indexes and ILDSs

IMS Index Builder IMS Index Builder processing, DFSPREC0 processing, or processing for the Build Index function of FPA is invoked only after database recovery is complete because the entirely recovered database is required in order to rebuild the indexes or ILDSs.

You need only rebuild indexes or ILDSs if the media containing them fails or if a database is recovered to a particular time stamp (either PITR or TSR). This means that all indexes do not need to be rebuilt for all recoveries.

Image copies of indexes are taken only when the indexes are marked recoverable in the RECONs. If your installation chooses to rebuild indexes after database recovery, then taking an image copy of an index is not necessary.

For Integrated Auxiliary Utility processing, additional JCL procedures must be defined in the procedure library. Please see the installed SFRXSAMP library for sample JCL procedures and instructions for their definition.

When the MAS job JCL is submitted, it reads the PROCLIB member FRXDRFxx (where xx is given by the DRFMBR=xx parameter) and the SYSIN control statements to determine the scope of the recovery, including any integrated auxiliary utility processing requested. Integrated auxiliary utility processing is activated when any of the following is encountered in the input stream:

- ADD command keywords IB, PC or IC.

The parameters in these keywords are passed by IMS Database Recovery Facility to the corresponding integrated auxiliary utility.

The parameters are shown in the syntax diagrams for the ADD command, but the individual Integrated Auxiliary Utility user's guides should be consulted for their meaning and usage.

- UTILGBL statement keywords HALDB, BLD\_PRIMARY or BLD\_SECONDARY.
- LIU@GOPT control statement, to activate Library Integrity Utility.

**Note:** The REPORT and DRFIAX control statements influence integrated auxiliary utility processing, but their existence alone does not cause integrated auxiliary utility invocation.

## Processing address space architecture and configuration

---

The address space architecture when IMS Database Recovery Facility is running with all the integrated auxiliary utilities active is described here.

Shown here is a comprehensive view of integrated auxiliary utility processing with address spaces shown.

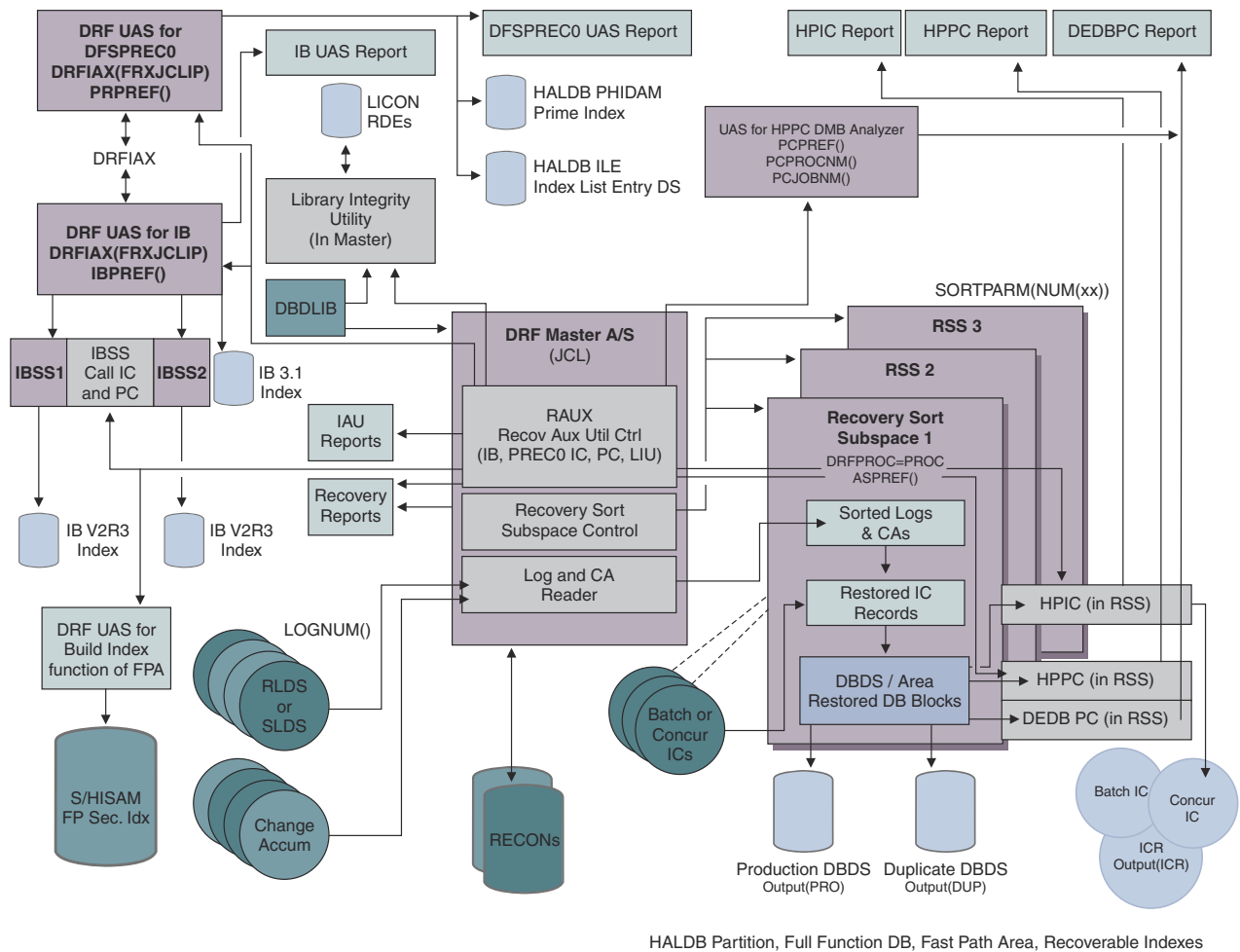


Figure 2. Integrated auxiliary utility processing with address spaces shown

Diagram Color Key:

- Purple = separate address space
- Grey = control function within an Address Space
- Lime Green Lines = Recovery auxiliary utility control task (RAUX) in MAS, reaching throughout all the address spaces participating in the recovery
- Black Lines = data flow
- Green = recovery asset data, either external or internally staged
- Bright Blue = recovery or utility output
- Pale Blue = SYSOUT reports

Below is a list of the address spaces shown in the diagram. The associated configuration parameters are also described. The arrows leading from the RAUX box in the master address space illustrate this task as the central point of control of integrated auxiliary utility processing throughout all the address spaces participating in the recovery.

### Master Address Space (MAS)

Created when the JCL for the job is submitted.

### Recovery Sort Subordinate (RSS) Address Space (and multiples)

- Created by the MAS for restoring image copies and recovering database data sets.
- The procedure for this address space is given by the DRFPROC parameter.

The name of the started task is prefixed with the contents of the ASPREF keyword.



- Subtasks related to IMS High Performance Image Copy, IMS High Performance Pointer Checker and DEDB Pointer Checker are active in the RSS when the IC or PC keywords are present on the ADD command.

### **FS-UAS**

- Created by the MAS when the IMS Index Builder integrated auxiliary utility is requested by using the IB() keyword on the ADD command or in UTILGBL control statement parameter BLD\_SECONDARY.
- The procedure for this address space is given by the DRFIAX control statement.  
The name of the started task is prefixed with the contents of the FSPREF keyword.
- The FS-UAS is created after all RSS address spaces have completed.

### **PR-UAS**

- The PR-UAS is created by the MAS when the DFSPREC0 integrated auxiliary utility is requested via the HALDB keyword on the UTILGBL control statement or in the IB() keyword of the ADD command.
- The procedure for this address space is given by the DRFIAX control statement.  
The name of the started task is prefixed with the contents of the PRPREF keyword.
- This address space is created after all RSS address spaces have completed, but before recovery is ended.
- A new PR-UAS is started for each HALDB partition for which a PHIDAM primary index or ILDS data set rebuild is required, but only one is active at any given time.

### **IB-UAS**

- Created by the MAS when the IMS Index Builder integrated auxiliary utility is requested by using the IB() keyword on the ADD command or in UTILGBL control statement parameters BLD\_PRIMARY or BLD\_SECONDARY.
- This is also known as the IMS Index Builder master address space because IMS Index Builder is started as a subtask in this address space.
- The procedure for this address space is given by the DRFIAX control statement.  
The name of the started task is prefixed with the contents of the IBPREF keyword.
- The IB-UAS is created after all RSS address spaces have completed.
- If the version of IMS Index Builder is V2R3, then all the indexes are built in the IMS Index Builder sort subordinate address space (IBSS).

If the version of IMS Index Builder is V3R1 all the indexes are built here in the IB-UAS.

However, one or more IBSS address spaces may be started for the purpose of running Image Copy and Pointer Checker for the rebuilt indexes.

- The first time you run either IMS Index Builder V3R1 or V2R3, the client should first do this as a native batch job to ensure its procedures and parameters are installed correctly.  
It should be able to build indexes for IMS Database Recovery Facility restored databases.  
This test ensures that IMS Index Builder successfully runs as an Integrated Auxiliary Utility.
- IMS Database Recovery Facility will not prevent you from both recovering and rebuilding an index in the same one-step recovery job.  
However this results in redundant processing and should be avoided.
- IMS Database Recovery Facility passes internally to IMS Index Builder V2R3 some parameters that cannot be changed because they are required in this environment.

This restriction applies only in this environment and stand-alone IMS Index Builder continues to operate as usual. These include:

- SORT35(YES) and SORT15(YES): Though externally available, these options are ignored if specified on either the UTILGBL control statement or the IMS Index Builder keyword of the ADD command.

- ALTER(NO)
- STEPLIB(TRANS): This parameter causes the transfer of the STEPLIB DD concatenation from the IB-UAS to the IBSS.
- Some of the IMS Index Builder parameter IMS Database Recovery Facility ADD command keywords that are supported in IMS Index Builder V2R3 are not supported in IMS Index Builder V3R1.

If used in IMS Index Builder V3R1, it results in a warning message that they keyword is no longer supported, but IMS Index Builder V3R1 continues processing:

- ALTER
- NDXIOUBF
- SORTE15
- SORTE35
- SORTID
- SORTOPT
- SORTOUT
- SORTSTAT
- **Note:** IMS Index Builder 3.1 has more functions than IMS Index Builder 2.3, yet the number of keyword parameters has been reduced in order to reduce complexity.

The first APAR to support IMS Index Builder V3R1 as an integrated auxiliary utility was designed to simulate V2R3 functions. If you run IMS Index Builder V3R1 as an integrated auxiliary utility using the JCL procedures from IMS Index Builder V2R3, you will get the same results, yet with a few warning messages about parameters that are no longer supported in IMS Index Builder V3R1.

The new features of IMS Index Builder V3R1, such as the ability to recover HALDB PHIDAM primary indexes, are yet to be exploited by IMS Database Recovery Facility.

### **IMS Index Builder Sort Subordinate (IBSS) Address Space (and multiples):**

- The IBSS is created by the IB-UAS.
- The JCL procedure used to start the IBSS depends on the installed version of IMS Index Builder.

In IMS Index Builder V3R1, the procedure used is always IIUBSRT and the started task names are IIUAPIFC and IIUSORTS.

With V3R1, the procedure name and started task name can be overridden by using IIURPRMS module SIIULMOD.

In IMS Index Builder V2R3, the procedure used defaults to IIUBSRTT.

With V2R3 IMS Database Recovery Facility is able to override IIUBSRTT as the procedure name and started task name by passing the SORTID(x) parameter to IMS Index Builder.

In that case, the procedure and started task names are IIUBSRTx.

- Subtasks for IMS High Performance Image Copy and IMS High Performance Pointer Checker may be active in the IBSS when image copy and pointer checker processing has been requested for the rebuilt indexes.
- If IMS Index Builder V2R3 is installed, all indexes are built here in the IBSS.

If IMS Index Builder V3R1 is installed, then these subordinate address spaces are built for the purpose of running IMS High Performance Image Copy and IMS High Performance Pointer Checker when requested by IMS Database Recovery Facility.

The indexes are built in the IB-UAS (IMS Index Builder Master Address Space).

## PC-UAS

- The PC-UAS, which runs the IMS High Performance Pointer Checker DMB Analyzer, is created by the MAS when the IMS High Performance Pointer Checker integrated auxiliary utility is requested for full-function databases via the PC keyword on the ADD command.
- The procedure for this address space is given by the PCPROCNM keyword and the stored task name is given by the PCJOBNM keyword. The PCPREF() control statement sets the prefix for the name of the PC-UAS started task.

## Library Integrity Utility (LIU)

Although it is depicted separately in the diagram, Library Integrity Utility does not run in a separate address space. It is run in the MAS when the LIU@GOPT(yyyy) control statement is coded.

As a technique for correlating all the address spaces for a recovery, code the same three characters as the first part of the four character prefix for ASPREF, FSPREF, IBPREF, PCPREF, PRPREF and PCJOBNM.

For example: ASPREF(DR3S), FSPREF(DR3F), IBPREF(DR3I), PCPREF(DR3D), PRPREF(DR3P) and PCJOBNM(DR3C).

Also the job name for the MAS JCL should begin with these same three characters. This will make it easier to find the job output for the related address spaces.

## Control flow and the RAUX control task

---

During initialization, the master address space (MAS) calls upon DBRC to identify from the RECONs the assets required for database data set recovery, including the images copies (ICs), change accumulations (CAs), and logs.

### RAUX control task overview

If Integrated Auxiliary Utility processing has been requested, the recovery auxiliary utility (RAUX) control task is started. The MAS then starts the number of recovery sort subordinate (RSS) address spaces corresponding to the SORTPARM(NUM(xx)) parameter.

The RSS restores the ICs, sorts the log and CA records sent to it by the MAS, and applies the database updates to create a restored database data set or, alternatively, an incremental image copy (ICR).

As the database blocks are updated and written to the recovered database, they can also be processed in parallel by the IC and pointer checker (PC) tasks. Blocks need only be read once, but can be processed by both IC and PC in parallel. This is more efficient than running IC and PC as separate job steps after the recovery job step.

The MAS attaches the number of log and CA read tasks that corresponds to the LOGNUM parameter. These tasks:

- Determine the correct RSS to send the database update records to,
- Buffer them,
- Send them to the RSS where they are sorted,
- Apply the records as updates to the database blocks restored from the ICs.

For offsite storage, remote site recovery, or testing purposes, recovered databases can be written to:

- The production database,
- An ICR, or
- A duplicate copy database.

If ICs of the restored database are requested, the blocks for them are written in parallel. Any ICs created are registered in the RECONs during MAS termination.

Instead of ICs, ICRs can be created by restoring the most recent IC and applying all log and CA database update records. The difference in this case is that the output of the recovery is an ICR rather than a restored database.

The resulting IC may be a Batch-IC or Concurrent-IC, depending on whether the database was online at the time of the ICR. The IC created by the ICR process is registered in the RECONS and is usable in subsequent database recovery.

### **RAUX control task results**

The RAUX control task in the MAS serves to initialize, monitor, drive processing for, and collect results from the integrated auxiliary utilities running throughout the address spaces participating in the recovery.

Results include:

- SYSOUT reports,
- Return codes, and
- Write To Operator (WTO) messages issued to consoles or job logs.

RAUX services are provided via z/OS cross-memory services to all address spaces that require them. The RAUX allocates its major control block to store the results of the integrated auxiliary utilities. Updates to that control block may be done from any address space involved in the recovery.

When the MAS arrives at its termination logic, the RAUX organizes all of the Integrated Auxiliary Utility results and stores them as directed by the MAS JCL and its parameters.

### **Image copy and pointer checker processing**

If pointer checking was selected for a full function database data set, then the RAUX starts the IMS High Performance Pointer Checker DMB Analyzer (PC-UAS) procedure FABPATHx.

If image copy or pointer processing was requested for any database data set, then the RAUX control task starts IMS High Performance Image Copy or IMS High Performance Pointer Checker in the RSS.

In the RSS, IMS HP Image Copy, IMS HP Pointer Checker, and DEDB Pointer Checker run under separate tasks so that their services can be multi-threaded for the purpose of efficiency. For example, IMS HP Pointer Checker can be in the process of collecting information on the database pointers in a block at the same time that the database block is being written by IMS HP Image Copy to the IC data set.

### **HALDB support**

IMS Database Recovery Facility also has the capability of rebuilding the PHIDAM primary index and ILDS data set for a HALDB partition.

This is accomplished by starting the procedure identified by the DRFIAX control statement. This PR-UAS is started just before MAS termination and after all recovery processing for the HALDB partitions has completed.

A new PR-UAS is started for each HALDB partition database data set. This is because DFSPREC0 is not able to operate on more than one of these HALDB partition database data sets in a given execution.

Once again, the services of the RAUX are used to capture the reports and return codes from DFSPREC0, and they are communicated back to the master address space for inclusion on the reports.

### **Index build processing**

IMS Database Recovery Facility can also invoke IMS Index Builder to build the primary and secondary indexes for Full Function databases and secondary indexes for HALDB partitioned databases.

Both IMS Index Builder V2.3 and V3.1 can build HALDB primary indexes, and V3.1 can additionally build ILDSs. To rebuild these HALDB components, IMS Database Recovery Facility uses DFSPREC0, and not IMS Index Builder.

You can use the Build Index function of FPA to rebuild secondary indexes for Fast Path DEDB databases.

When index rebuilding has been requested, the MAS starts either the utility address space for IMS Index Builder (IB-UAS) or the utility address space for the Build Index function of FPA (FS-UAS) just before its own ending.

The function of the IB-UAS depends on the version of IMS Index Builder:

- In IMS Index Builder V2R3, the IB-UAS starts the IBSS to rebuild the full function indexes.
- In IMS Index Builder V3R1, the IB-UAS starts the IBSS for running IMS HP Image Copy and IMS HP Pointer Checker/ DEDB Pointer Checker when requested by the MAS.

The indexes themselves are built in the IB-UAS.

Again, the RAUX is active to capture reports, return codes, and WTO messages so they can be communicated back to the master for organization.

Note that IMS Database Recovery Facility is able to recover most types of indexes, provided an image copy has been kept and the indexes are registered as RECOV in DBRC. If an image copy has been kept, it is more efficient to let IMS Database Recovery Facility recover the index during the recovery in the RSS than it is to run IMS Index Builder or the Build Index function of FPA at the end of all recovery processing as a separate process.

However, if a corrupted index was copied or the copy was destroyed, running IMS Index Builder or the Build Index function of FPA can be the only option.

### **Verification by IMS Library Integrity Utilities**

IMS Library Integrity Utilities verifies that the DBD library being used in the recovery (the one allocated to the IMS DD in the MAS JCL) is correctly matched to the database data sets being recovered and processed by the other integrated auxiliary utilities.

This verification prevents database corruption that can happen by allocating the wrong DBD library, such as one being used for new application testing.

IMS Library Integrity Utilities runs in the MAS and dynamically allocates its library control (LICON) data set based on information in the IMS Library Integrity Utilities load library concatenated to the STEPLIB DD.

All IMS Library Integrity Utilities messages are written to the MAS job log. There is no specific IMS Library Integrity Utilities SYSOUT report data set written or appended to the MAS REPORT DD.

However, there is an IMS Library Utilities Final Return / Reason Code in the Utility section of that report. If you see a non-zero return code, you should examine the Master job log for FABLxxxxE messages. If IMS Library Integrity Utilities determines that an incorrect DBD library has been allocated, the recovery is halted to prevent database corruption.

## **SYSOUT reports, WTO messages, and return codes**

---

One of the capabilities provided by the activation of the integrated auxiliary utilities in the IMS Database Recovery Facility 'single-step' recovery solution is the integration of integrated auxiliary utility reports, messages, and return codes.

### **Topics:**

- [“Write to operator \(WTO\) messages” on page 27](#)
- [“SYSOUT reports” on page 28](#)
- [“Viewing the integrated auxiliary utilities reports” on page 28](#)
- [“WTO capture services” on page 29](#)
- [“Additional data set processing \(RPTTYPE=APP and RPTTYPE=SEP only\)” on page 29](#)

### **Write to operator (WTO) messages**

Write to operator messages (WTO) are generated by IMS Index Builder, DFSPREC0, the Build Index function of FPA, IMS High Performance Image Copy, Library Integrity Utilities (LIU), IMS High Performance Pointer Checker, and DEDB Pointer Checker.

For the RPTTYPE=SEP and RPTTYPE=APP options, write to operator messages generated by IMS Index Builder, DFSPREC0, IMS High Performance Image Copy, Library Integrity Utility, IMS High Performance

Pointer Checker, and DEDB Pointer Checker in the various address spaces are collected by the RAUX control task in the MAS.

- If the RPTTYPE=SEP option is specified, then all messages from the integrated auxiliary utilities are written to the FRXWTO DD in the MAS.
- If the RPTTYPE=APP option is specified, they are written to the REPORT DD in the MAS.
- If the RPTTYPE=SAS option is specified, then all WTO messages from the integrated auxiliary utilities are simply left on the job log in the subordinate address space (SAS) in which they were generated.

RPTTYPE=SAS reduces the overall processing time and I/O for IMS Database Recovery Facility.

- In all cases, the WTO messages are separated by DD Name and DD/Area Name. Additionally, the address space in which they originated is listed.

## **SYSOUT reports**

The same DBD Name and DD/Area Name separation also applies to the various integrated auxiliary utilities SYSOUT reports.

- For the RPTTYPE=SEP and RPTTYPE=APP options, the RAUX control task in the MAS gathers all integrated auxiliary utilities reports from all address spaces and distributes them.
- If RPTTYPE=SEP is specified, the reports are written to the integrated auxiliary utilities standard report DDs in the MAS.

The standard integrated auxiliary utility SYSOUT report DD names can be added to the MAS JCL to contain any separate report data. If the report DD names are not added to the MAS JCL, then they are dynamically allocated.

- If the RPTTYPE=APP option is specified, the integrated auxiliary utility SYSOUT reports are instead appended to the MAS REPORT DD file.
- If RPTTYPE=SAS is specified, the reports are written to the integrated auxiliary utilities standard report DDs in the SAS in which they were generated.

RPTTYPE=SAS reduces the overall processing time and I/O for IMS Database Recovery Facility.

- As with the MAS, if these DDs are not in the JCL, they are dynamically allocated in the SAS.

## **Viewing the integrated auxiliary utilities reports**

The most convenient way to view the integrated auxiliary utility reports is to choose the RPTTYPE=SEP option, then use spool display and search facility to select the MAS job log so that the SYSOUT data sets can be viewed separately by DD Name.

The RPTTYPE=APP is provided as a convenience to those clients that do not have spool display and search facility or who prefer to store all IMS Database Recovery Facility and integrated auxiliary utility reports in the single file defined by the MAS REPORT DD.

The MAS REPORT DD has a separate section that provides a summary of integrated auxiliary utility processing. The summary shows whether an integrated auxiliary utility has been involved in the recovery of a given database data set and the results of that involvement in terms of the return and reason code.

The end of the utility report shows a final return and reason code for each integrated auxiliary utility that was invoked.

- If the final return code from an integrated auxiliary utility is greater than 4, the MAS return code is set to 8.
- If an integrated auxiliary utility final return code is 4, then the MAS return code is set to 4.

## WTO capture services

When the integrated auxiliary utilities run in one of the address spaces, there are times when z/OS or the integrated auxiliary utilities themselves issue an SVC 35 (WTO) to write a message to the console or the job log.

The RAUX is aware of and manages the collection of WTO messages and SYSOUT reports on behalf of the integrated auxiliary utilities.

WTO messages are managed as if they were a separate integrated auxiliary utility SYSOUT report, even though the integrated auxiliary utilities do not themselves create a separate WTO messages report.

In order to capture the WTO messages, the RAUX has a facility known as WTO capture services. This routine receives control when an SVC 35 is issued by any task in the address space.

If the message applies to one of the integrated auxiliary utility tasks, it is quickly copied to a RAUX memory structure, and separated by DBD Name and DD/Area Name. Then the RAUX allows the normal SVC 35 process to continue.

During address space termination of the subordinate address spaces, the captured SVC 35 messages are written from the memory structures to data sets allocated specifically for this purpose. In this way, WTO messages are associated with a specific DBD Name and DD/Area Name.

These names of these data sets are stored in the RAUX main control block and left to be managed by the RAUX control task in the MAS during its termination. Note that when RPTTYPE=SAS is specified, the WTO messages are simply left on the job log in the SAS.

## Additional data set processing (RPTTYPE=APP and RPTTYPE=SEP only)

SYSOUT reports written by the integrated auxiliary utilities in the various address spaces are retained in cataloged data sets for the duration of the recovery job.

The MAS RAUX controller copies these data sets, as directed by the REPORT control statement options, to various DD of the MAS JCL. These data sets are then, under normal conditions, deleted and uncataloged by the RAUX control task.

You can see evidence of this activity in the MAS job log by searching for the string “SAS Copy ->:” The integrated auxiliary utility SYSOUT report data set names are included in these messages.

Here is an example of what can be searched for in the MAS job log to see evidence of this data set management:

```
ICUST1A.ICUST1AA IBSS=DR3P0001 DD=WTOs captured
-----
SAS COPY->: DD=FRXWTO DSN=DRFHILEV.FRXWTOIB.ICUST1A.ICUST1AA.T0048339
```

In this example:

- The RAUX has captured WTO messages associated with building the index for the ICUST1A partition's ICUST1AA database data set in the PR-UAS, under the started task name of DR3P0001.
- The name of the WTO data set is DRFHILEV.FRXWTOIB.ICUST1A.ICUST1AA.T0048339.
- The captured messages that were contained in the file were written, along with the identifying address space name, to either the MAS FRXWTO DD or the REPORT DD, depending on the RPTTYPE option.
- The file was then deleted and uncataloged.

Notice that the heading contains the *dbdname.ddname* string. This string can be used to find all SYSOUT data gathered by the RAUX for that particular recovered database data set. This is especially useful when RPTTYPE=APP is coded on the REPORT control statement and you are viewing the entire REPORT DD of appended integrated auxiliary utility reports and messages.

These integrated auxiliary utility report data set names are patterned as follows:

```
&DRFHILEV.FRXxxxx.&dbdname.&ddname.Thhmmssm
```

where:

- &DRFHILEV is taken from the REPORT control statement.
- *xxxxx* corresponds to the particular type of integrated auxiliary utility report.
- *&dbdname* and *&ddname* correspond to the database data set being processed (including indexes).
- *hhmmssm* is the hour, minute, second, and millisecond of the time the recovery was started.

The time stamp portion can be used to correlate all integrated auxiliary utility report data sets associated with a given recovery job.

Because of the 44-character DSN length constraint, the possibility exists that the RAUX attempts to allocate, as new, a DSN that already exists.

If the RAUX attempts and fails to dynamically allocate an integrated auxiliary utility report data set, message FRD9003A is written and ABEND U0384-02C is issued. This can occur when an earlier recovery job was:

- Stopped (by **/P** command), or
- Canceled (by **/CANCEL** command) by a z/OS operator, or
- When the RPTRET=Y option is set on the REPORT control card.

To minimize this possibility, set RPTRET=N so that RAUX termination logic deletes the integrated auxiliary utility report data sets.

RAUX termination logic is not driven when the job is stopped or canceled. In this event, it is advisable to delete all integrated auxiliary utility report data sets named with the pattern &DRFHILEV.FRX\*.\*.\*.T\* before running the next recovery job.

Furthermore, if RPTRET=N is set, IMS Database Recovery Facility deletes, during initialization, any utility report data sets left from prior executions that would otherwise cause allocation errors due to duplicate DSNs.



---

## Part 2. Configuring IMS Database Recovery Facility

You must complete IMS Database Recovery Facility installation and configure the tool to meet your environment.

After completing the post-installation tasks and configuration, IMS Database Recovery Facility can begin to assist you in managing backup and recovery operations for your databases.

Before continuing with configuration, you must first have completed the SMP/E installation as outlined in the *Program Directory for IMS Recovery Solution Pack for z/OS*, GI10-8819 and the post-installation steps as outlined in the *IMS Recovery Solution Pack: Overview and Customization*, SC27-8440.

The topics in this section covers the steps needed to configure IMS Database Recovery Facility in your environment:

### Topics:

- [Chapter 4, “Configuring the product environment,” on page 33](#)
- [Chapter 5, “Configuring the integrated auxiliary utilities environment,” on page 63](#)
- [Chapter 6, “Configuring JCL statements and procedures,” on page 93](#)



---

## Chapter 4. Configuring the product environment

You must install and configure IMS Database Recovery Facility before it can be used. The information in this section provides you with support for configuring IMS Database Recovery Facility.

For information on configuring the product for use with the integrated auxiliary utilities, see [Chapter 5, “Configuring the integrated auxiliary utilities environment,”](#) on page 63.

### Topics:

- [“Creating the environmental control member: FRXDRFxx”](#) on page 33
- [“Processing System Level Backup \(SLB\) image copies”](#) on page 34
- [“Setting up the environment to support SLB image copies”](#) on page 35
- [“Environmental control statements”](#) on page 40

---

### Creating the environmental control member: FRXDRFxx

You can use the FRXDRFxx PROCLIB member to provide the IMS Database Recovery Facility with the parameters to establish its environment.

It is specified using the DRFMBR=xx EXEC parameter on the execution JCL where xx is a 1 or 2-character suffix appended to FRXDRFxx. This optional member can be used to set up default processing parameters for multiple jobs.

To create a shared environmental control member, create the FRXDRFxx PROCLIB member and place it in a data set included in the PROCLIB DD concatenation of the IMS Database Recovery Facility master address space JCL.

The FRXDRFxx member must have a record length of 80. Of this 80 character length, you can only use columns 1 through 72 for specifying control statements. Columns 73 through 80 are ignored.

You can enter the environment control statement parameters in a free format. Keywords can start in any valid column, as long as parameters are not split in an invalid manner.

You can code continuation statements by typing a dash after the last parameter on one line and resuming the listing of parameters on the following line.

The following example shows a continuation statement coded incorrectly because the keyword parameter MAINSIZE has been split between two lines.

```
SORTPARM(NUM(5),HIPRMAX(OPTIMAL),MAIN-  
SIZE(30),ASPREF(FRXI),AVGLEN(256),FILSZ(210000))
```

The following example shows a continuation statement coded correctly.

```
SORTPARM(NUM(5),HIPRMAX(OPTIMAL),MAINSIZE(30),-  
ASPREF(FRXI),AVGLEN(256),FILSZ(210000))
```

You can also specify comments. Comments can be included on lines which contain valid statements, or they can inhabit their own lines. You must enclose comments between /\* (starting) and \*/ (ending) delimiters. The two delimiters must be on the same physical line; comments cannot be continued across multiple lines.

The following example shows 3 lines of comments coded incorrectly because the starting /\* and the ending \*/ are not on the same line.

```
/* This is my first line of comments.  
This is my second line of comments.  
This is my third line of comments. */
```

The following example shows 3 lines of comments coded correctly.

```
/* This is my first line of comments. */  
/* This is my second line of comments. */  
/* This is my third line of comments. */
```

## Processing System Level Backup (SLB) image copies

---

IMS Database Recovery Facility has the capability to use IMS Recovery Expert System Level Backups (SLB) as recovery assets.

When IMS Recovery Expert creates an SLB, it can keep track of all DBDS that are backed up in the SLB. The SLB contains the equivalent of fast replication image copies for each DBDS in the SLB.

A dynamic application programming interface (API) is provided that allows IMS Database Recovery Facility jobs to utilize these SLBs for image copies when those SLBs can meet the recovery criteria better than an image copy recorded in the RECON data sets. By using IMS Recovery Expert SLBs as image copies, you can reduce the need for creating image copies.

### Setting up the environment to use SLB image copies

The IMS Recovery Expert SLB API allows IMS Database Recovery Facility jobs to use SLBs as image copies. The API requires information relating to the IMS Recovery Expert environment. This information includes the names of the IMS Recovery Expert repository data sets and the execution load libraries.

This information is provided in one of two ways:

- Modify the IMS Database Recovery Facility JCL to include DD statements for each of these data sets, or
- Run the SLB API utility to dynamically build the API environment that allows IMS Database Recovery Facility jobs to access the required information without the need for JCL changes.

The recommended method is to use the dynamic SLB API activation utility so that IMS Database Recovery Facility JCL does not have to be changed.

The dynamic API activation utility is discussed in the following topic:

[“Setting up the environment to support SLB image copies” on page 35](#)

### Using SLBs in the recovery process

The use of IMS Recovery Expert SLBs by IMS Database Recovery Facility jobs is controlled by the USESLBIC() keyword.

- When USESLBIC(Y) is specified, IMS Database Recovery Facility jobs use the SLB API and SLBs can be used as sources for image copies.
- When USESLBIC(N) is specified, SLBs are not used.
- When USESLBIC(Y) is set, and a given database is not contained in the SLB used for processing the recovery list, IMS Database Recovery Facility attempts to use non-SLB image copies as sources for image copies.

These image copies must have time stamps greater-than or equal-to the time stamp of the SLB.

When IMS Database Recovery Facility jobs run with USESLBIC(Y) and determine that there is an SLB that best matches the recovery criteria, the SLB API issues a NOTIFY.IC command to create an image copy record in the RECON data sets for that SLB image copy.

The format of the SLB image copy record data set name is:

```
SLB.Ixxxx.Dxxxx.dbdname.ddname
```

The SLB image copy data set name is symbolic and no physical data set exists with that name. When a SLB image copy record exists in the RECON data sets, and an IMS Database Recovery Facility job is run with USESLBIC(N), and the job determines that this record should be used for recovery, then the job fails

when attempting to allocate the symbolic data set. Therefore, once IMS Database Recovery Facility jobs execute with USESLBIC(Y), the jobs should continue to run with this setting.

When USESLBIC(Y) is specified, IMS Database Recovery Facility is able to use SLBs as recovery assets:

- SLBs are created by IMS Recovery Expert and written to fast-replication devices.
- SLBs can also be offloaded to other DASD or tape devices. Therefore SLBs can exist on fast-replication devices, offload data sets, or both.
- If the SLB exists on both, the SLB on the fast-replication device is used in recovery.

If there are offload data set(s), they are displayed on the VERIFY(ALLOC) and VERIFY(OPEN) reports. This does not necessarily indicate that the offload data set(s) are used for recovery. It just indicates that they are available if needed. If the offload data set(s) are used, the recovery report indicates that fact.

When an SLB is used for recovery, the SLB API issues a NOTIFY.IC to register the image copy record in the RECON data sets. These image copy records are the symbolic records discussed earlier. IMS Recovery Expert can be used to delete the SLB from the fast replication devices.

As long as the SLB still exists on offload data sets, the SLB, and any symbolic image copy records associated with the SLB, are still usable. However, if the offload data sets are also deleted, the symbolic image copy records are no longer valid. If you want to ensure that these records are not used, you can delete them from the RECON data sets.

The VERIFY function can be used to determine the correlation between the SLB's fast-replication volume(s) and the offload data sets.

When USESLBIC(Y) is specified, and either databases are being recovered or their recovery assets are being verified, it is a best practice to examine the job log for FRD4312E and FRD4320I error messages.

These messages are displayed when there is an error in the application programming interface between IMS Database Recovery Facility and IMS Recovery Expert or when the database is not contained in the SLB.

## Setting up the environment to support SLB image copies

---

The IMS Recovery Expert System Level Backup (SLB) API activation utility is used to activate, deactivate, or list the dynamic API environments that exist on an z/OS system.

This API allows IMS Database Recovery Facility and IMS Database Recovery Facility Extended Functions jobs to use any existing IMS Recovery Expert SLBs when performing functions that access image copies. The API identifies the IMS Recovery Expert environment and eliminates the need to change the job JCL.

The SLB API activation utility should be enabled after each IPL on any z/OS system where IMS Database Recovery Facility and IMS Database Recovery Facility Extended Functions jobs can run. Once activated, the API remains active until the next IPL or until it is removed or deactivated through the API activation utility.

If the SLB API activation utility is executed multiple times to activate different environments, only the last environment activated is used. All previous environments remain defined but are inactive.

Changes are not allowed to the dynamic API environments if the utility detects that there are active users of the API interface.

### Sample JCL

The sample JCL for activating, deactivating, and listing the API environments can be found in SAMPLIB(BSY#UTIL). Customize this job to your environment and set it up to run after each IPL on any LPAR where IMS Database Recovery Facility and IMS Database Recovery Facility Extended Functions jobs need to run.

The EMCLOAD and FDRLOAD DDs are required if either tool is used for data movement.

The BSYGROUP DD is required only if the combined SLB feature is active.

All other DDs should match the data sets specified in your IMS Recovery Solution Pack CLIST, which invokes the IMS Recovery Expert ISPF interface.

## Command reference for SLB API activation utility

The following commands can be specified in the BSYIN DD control statement to enable IMS Recovery Expert SLB API activation utility functions.

### ACTIVATE

The ACTIVATE command is used to either build a new dynamic API environment and make it active, or convert an existing environment to be the active environment.

If other environments exist when the ACTIVATE is performed, those environments remain intact but become inactive and are not used by the API. Only one dynamic API environment can be active at a time. Once the activation is completed successfully, the API can be called dynamically without specifying the load library and repository data sets in the JCL that uses the API.

If you want to use the API for an environment that is different from the active environment, you must specify the required DD statements in the IMS Database Recovery Facility or IMS Database Recovery Facility Extended Functions JCL.

ACTIVATE performs the following operations:

1. Ensure that all required DD statements are specified in the JCL.

The utility terminates otherwise.

2. Check to see if there are any active users of the API.

If there are active users, then no changes are allowed and the utility terminates.

No changes are allowed (for example, ACTIVATE or DEACTIVATE) when there are active users of the API.

3. Check to see if the BSY#API interface module is loaded into LPA (Link Pack Area).

If the module is not loaded, a copy is loaded into LPA and messages are displayed indicating it was loaded.

Messages are also issued showing the maintenance level of the module and whether it was loaded or it already existed.

4. Check to see if there is a dynamic API environment that matches the DD statements allocated in the JCL.

If a matching environment exists, and it is not currently the active environment, then the environment is converted and designated as the active environment.

If a matching environment does not exist, then a new dynamic API environment is created and is designated the active environment.

### DEACTIVATE

The DEACTIVATE command is used to remove one or more dynamic API environments. The DEACTIVATE parameter can be specified using the following formats:

#### DEACTIVATE

This form of the command deactivates only the dynamic API environment that matches the DD statements allocated in the JCL.

DEACTIVATE performs the following operations:

1. Ensure that all required DD statements are specified in the JCL.

The utility terminates otherwise.

2. Check to see if there are any active users of the BSY#API interface module loaded in LPA (Link Pack Area).

If there are active users, then no changes are allowed and the utility terminates.

No changes are allowed (for example, ACTIVATE or DEACTIVATE) when there are active users of the API.

3. Locate the dynamic API environment that matches the DD statements allocated in the JCL.

If none is found, a message is issued and the utility terminates.

4. If a dynamic API environment is found that matches the DD statements allocated in the JCL, the environment control blocks are unchained from the list and the storage is freed.

If the deactivated environment was the active environment, the next environment in the chain is made active.

If the last environment is deactivated, then the BSY#API interface module is also deleted from LPA.

**Note:** The DEACTIVATE command appears as DEACTIVATE(RVTE) in the output.

### DEACTIVATE(ALL)

This form of the command deactivates all dynamic API environments. If this form is used, the API can no longer be called unless the necessary data sets are allocated in the job that uses the API.

DEACTIVATE(ALL) performs the following operations:

1. Check to see if there are any active users of the API.

If there are active users, then no changes are allowed and the utility terminates.

No changes are allowed (for example, ACTIVATE or DEACTIVATE) when there are active users of the BSY#API module that is loaded into LPA (Link Pack Area).

2. Run the chain of all dynamic API environments and deletes each control block.
3. Remove the BSY#API module from LPA.

### LIST

The LIST command is used to display information about the dynamic API environment on a z/OS system.

Messages are issued showing information related to the BSY#API interface module and each dynamic API environment that exists.

## Example BSYIN control statements

The examples in this section show BSYIN control statements using commands to control the SLB API activation utility.

### Example 1

The following example lists any defined dynamic API environments and additionally shows the active API environment:

```
//BSYIN DD *  
LIST
```

### Example 2

The following example uses the DD statements allocated in the JCL to build a new API environment, or convert an existing environment, and activate that environment.

```
//BSYIN DD *  
ACTIVATE
```

### Example 3

The following example performs the following operations:

- List all dynamic API environments that currently exist and additionally show the active API environment
- Deactivate only the active dynamic API environment that matches the DD statements allocated in the JCL
- List again to ensure that the deactivated environment has been removed
- Deactivate all other existing dynamic API environments
- List again to ensure that the deactivated environments have been removed
- Build and activate a new API environment that matches the DD statements allocated in the JCL
- List again to ensure that the new dynamic API environment is now designated as active

```
//BSYIN DD *  
LIST  
DEACTIVATE  
LIST  
DEACTIVATE(ALL)  
LIST  
ACTIVATE  
LIST
```

### Example BSYIN control statement output

The following output was produced from a BSYIN control statement using commands to control the SLB API activation utility:

```
16:23:12 BSY0001I - IBM IMS Recovery Solution Pack for z/OS Starting.  
Version 02.02.000  
16:23:12 BSY0003I - Control Cards:  
16:23:12 BSY0004I - LIST  
16:23:12 BSY0004I - DEACTIVATE  
16:23:12 BSY0004I - LIST  
16:23:12 BSY0004I - DEACTIVATE(ALL)  
16:23:12 BSY0004I - LIST  
16:23:12 BSY0004I - ACTIVATE  
16:23:12 BSY0004I - LIST  
16:23:12 BSY0004I -  
16:23:12 BSY0004I -  
16:23:12 BSY0004I -  
16:23:12 BSY0004I -  
16:23:12 BSY0560I -  
16:23:12 BSY0560I - Processing LIST....  
16:23:12 BSY0560I - .... Querying LPA for BSY#API  
16:23:12 BSY0561I - BSY#API located in LPA  
16:23:12 BSY0568I - BSY#API ENTRYPT = 20E4D000  
16:23:12 BSY0568I - BSY#API LOADPT = 20E4D000  
16:23:12 BSY0568I - BSY#API MODLEN = 000038C8  
16:23:12 BSY0568I - BSY#API TIMESTAMP = APARV2R2M0  
16:23:12 BSY0568I - BSY#API VERSION = 03/17/20  
16:23:12 BSY0567I -  
16:23:12 BSY0567I - Active BSY#API environment:  
16:23:12 BSY0567I - Created by TSMXD on 2020-03-21 16:02:04.90467  
16:23:12 BSY0567I - BSYLOAD = IRS.DEV21.LOAD  
16:23:12 BSY0567I - EMCLOAD = RSRTE.EMC.LINKLIB.BETA760  
16:23:12 BSY0567I - FDRLOAD = RSRTE.VENDQR.FDR.LOAD  
16:23:12 BSY0567I - DB2PARMS= IRS.IRS21.CONTROL  
16:23:12 BSY0567I - BSYBPROF= IRS.IRS21.PROFILE  
16:23:12 BSY0567I - BSYBOFFL= IRS.IRS21.OFFOPTS  
16:23:12 BSY0567I - BSYBPMAP= IRS.IRS21.PROFILE.MAPS  
16:23:12 BSY0567I - BSYBPCAT= IRS.IRS21.PROFILE.CATS  
16:23:12 BSY0567I - BSYSBACK= IRS.IRS21.SYSBACK  
16:23:12 BSY0567I - BSYBBOBJ= IRS.IRS21.SYSBACK.OBJS.IMS  
16:23:12 BSY0567I - BSYSBVOL= IRS.IRS21.SYSBACK.VOLS  
16:23:12 BSY0567I - BSYSBSSD= IRS.IRS21.SYSBACK.SSIDS  
16:23:12 BSY0567I - BSYBREPT= IRS.IRS21.BREPORT  
16:23:12 BSY0567I - BSYPOBJS= IRS.IRS21.OBJECTS  
16:23:12 BSY0567I - BSYARCH = IRS.IRS21.ARCHIVES.IMS  
16:23:12 BSY0567I - BSY#PARM= IRS.DEV21.SAMPLE MEMBER = BSY#PARM  
16:23:12 BSY0560I -  
16:23:12 BSY0560I - Processing DEACTIVATE....  
16:23:12 BSY0570I - Deleting active BSY#API environment  
16:23:12 BSY0560I - .... Querying LPA for BSY#API
```



```

16:23:12 BSY0561I - BSY#API located in LPA
16:23:12 BSY0568I - BSY#API ENTRYPT = 20E4D000
16:23:12 BSY0568I - BSY#API LOADPT = 20E4D000
16:23:12 BSY0568I - BSY#API MODLEN = 000038C8
16:23:12 BSY0568I - BSY#API TIMESTAMP = APARV2R2M0
16:23:12 BSY0568I - BSY#API VERSION = 03/17/20
16:23:12 BSY0560I - .... BSY#API being deleted from LPA
16:23:12 BSY0561I - BSY#API deleted from LPA
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API not located in LPA
16:23:12 BSY0560I -
16:23:12 BSY0560I - Processing LIST....
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API not located in LPA
16:23:12 BSY0560I -
16:23:12 BSY0560I - Processing DEACTIVATE(ALL)....
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API not located in LPA
16:23:12 BSY0560I -
16:23:12 BSY0560I - Processing LIST....
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API not located in LPA
16:23:12 BSY0560I -
16:23:12 BSY0560I - Processing ACTIVATE....
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API not located in LPA
16:23:12 BSY0560I - .... BSY#API being added to LPA
16:23:12 BSY0561I - BSY#API added to LPA
16:23:12 BSY0568I - BSY#API TOKEN = 000009110000018E
16:23:12 BSY0568I - BSY#API ENTRYPT = A0E4D000
16:23:12 BSY0568I - BSY#API LOADPT = 20E4D000
16:23:12 BSY0568I - BSY#API MODLEN = 000038C8
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API located in LPA
16:23:12 BSY0568I - BSY#API ENTRYPT = 20E4D000
16:23:12 BSY0568I - BSY#API LOADPT = 20E4D000
16:23:12 BSY0568I - BSY#API MODLEN = 000038C8
16:23:12 BSY0568I - BSY#API TIMESTAMP = APARV2R2M0
16:23:12 BSY0568I - BSY#API VERSION = 03/17/20
16:23:12 BSY0560I - API environment added
16:23:12 BSY0567I -
16:23:12 BSY0567I - Active BSY#API environment:
16:23:12 BSY0567I - Created by TSMXD on 2020-03-21 16:23:12.36718
16:23:12 BSY0567I - BSYLOAD = IRS.DEV21.LOAD
16:23:12 BSY0567I - EMCLOAD = RSRTE.EMC.LINKLIB.BETA760
16:23:12 BSY0567I - FDRLOAD = RSRTE.VENDOR.FDR.LOAD
16:23:12 BSY0567I - DB2PARMS= IRS.IRS21.CONTROL
16:23:12 BSY0567I - BSYBPROF= IRS.IRS21.PROFILE
16:23:12 BSY0567I - BSYBOFFL= IRS.IRS21.OFFOPTS
16:23:12 BSY0567I - BSYBPMAP= IRS.IRS21.PROFILE.MAPS
16:23:12 BSY0567I - BSYBPCAT= IRS.IRS21.PROFILE.CATS
16:23:12 BSY0567I - BSYSBACK= IRS.IRS21.SYSBACK
16:23:12 BSY0567I - BSYSBOBJ= IRS.IRS21.SYSBACK.OBJS.IMS
16:23:12 BSY0567I - BSYSBVOL= IRS.IRS21.SYSBACK.VOLS
16:23:12 BSY0567I - BSYSBSSD= IRS.IRS21.SYSBACK.SSIDS
16:23:12 BSY0567I - BSYBREPT= IRS.IRS21.BREPORT
16:23:12 BSY0567I - BSYPOBJS= IRS.IRS21.OBJECTS
16:23:12 BSY0567I - BSYARCH = IRS.IRS21.ARCHIVES.IMS
16:23:12 BSY0567I - BSY#PARM= IRS.DEV21.SAMPLE MEMBER = BSY#PARM
16:23:12 BSY0560I -
16:23:12 BSY0560I - Processing LIST....
16:23:12 BSY0560I - .... Querying LPA for BSY#API
16:23:12 BSY0561I - BSY#API located in LPA
16:23:12 BSY0568I - BSY#API ENTRYPT = 20E4D000
16:23:12 BSY0568I - BSY#API LOADPT = 20E4D000
16:23:12 BSY0568I - BSY#API MODLEN = 000038C8
16:23:12 BSY0568I - BSY#API TIMESTAMP = APARV2R2M0
16:23:12 BSY0568I - BSY#API VERSION = 03/17/20
16:23:12 BSY0567I -
16:23:12 BSY0567I - Active BSY#API environment:
16:23:12 BSY0567I - Created by TSMXD on 2020-03-21 16:23:12.36718
16:23:12 BSY0567I - BSYLOAD = IRS.DEV21.LOAD
16:23:12 BSY0567I - EMCLOAD = RSRTE.EMC.LINKLIB.BETA760
16:23:12 BSY0567I - FDRLOAD = RSRTE.VENDOR.FDR.LOAD
16:23:12 BSY0567I - DB2PARMS= IRS.IRS21.CONTROL
16:23:12 BSY0567I - BSYBPROF= IRS.IRS21.PROFILE
16:23:12 BSY0567I - BSYBOFFL= IRS.IRS21.OFFOPTS
16:23:12 BSY0567I - BSYBPMAP= IRS.IRS21.PROFILE.MAPS
16:23:12 BSY0567I - BSYBPCAT= IRS.IRS21.PROFILE.CATS
16:23:12 BSY0567I - BSYSBACK= IRS.IRS21.SYSBACK
16:23:12 BSY0567I - BSYSBOBJ= IRS.IRS21.SYSBACK.OBJS.IMS
16:23:12 BSY0567I - BSYSBVOL= IRS.IRS21.SYSBACK.VOLS

```

```

16:23:12 BSY0567I - BSYBSSD= IRS.IRS21.SYSBACK.SSID5
16:23:12 BSY0567I - BSYBREPT= IRS.IRS21.BREPORT
16:23:12 BSY0567I - BSYPOBJS= IRS.IRS21.OBJECTS
16:23:12 BSY0567I - BSYARCH = IRS.IRS21.ARCHIVES.IMS
16:23:12 BSY0567I - BSY#PARM= IRS.DEV21.SAMPLE MEMBER = BSY#PARM
16:23:12 BSY0002I - IBM IMS Recovery Solution Pack for z/OS complete. RC=000.

```

## Environmental control statements

Environmental control statements can be specified to control the processing of the IMS Database Recovery Facility. Each control statement consists of parameters that contain information related to the statement.

The parameters associated with the control statements are listed in alphabetic order in the table, but you can specify them in any order. You can specify control statements in either the FRXDRFxx PROCLIB member or in the //SYSIN DD statement in the IMS Database Recovery Facility master address space JCL. Any parameters specified in //SYSIN DD override parameters specified in FRXDRFxx. The syntax is described in “Reference: Environmental control statements” on page 41.

| Parameter | Description                                                                                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CACHE     | Specifies that the image copy data sets are managed on a VTS (virtual tape server) device and they are to be preallocated to initiate staging to the cache before they are read. |
| CATDS     | Specifies the allocation option of image copies, logs, and change accumulation data sets from MVS catalog or RECON data set information.                                         |
| COMPRINA  | Specifies the aliases that you created for IMS High Performance Image Copy compression routines.                                                                                 |
| DBDSLnnn  | Specifies the data set characteristics for new data sets created by recovery.                                                                                                    |
| DBDSNnnn  | Specifies a template for naming new database data sets when duplicate databases are created.                                                                                     |
| DRFIAX    | Specifies the name of the procedure that is used to initiate the utility address spaces for IMS Index Builder, DFSPRECO, and the Build Index function of FPA.                    |
| DRFPROC   | Specifies the name of the procedure that is used to initiate the recovery sort subordinate address space.                                                                        |
| HPIO      | Specifies whether HPIO is to be used to write to any OSAM or VSAM ESDS DBDS during recovery.                                                                                     |
| ICNOTIFY  | Determines if the NOTIFY.IC commands for the generated image copies are issued by IMS HP Image Copy (HPIC) or IMS Database Recovery Facility (DRF).                              |
| ICNUM     | Specifies the maximum number of tape drives used for reading image copy data sets.                                                                                               |
| LBI       | Specifies the large block interface is in use.                                                                                                                                   |
| LCLTIME   | Specifies whether local or global time stamps are used in messages and reports.                                                                                                  |
| LIU@GOPT  | Specifies the LIU global option module LIU@xxxx, which is loaded from the STEPLIB by the Library Integrity Utility.                                                              |
| LOGNUM    | Specifies the maximum number of tape drives used for reading change accumulation data sets and the maximum number of read instances that can be initiated in parallel.           |
| MAXPRI    | Maximum primary allocations for dynamic SPACE allocation of image copy data sets.                                                                                                |

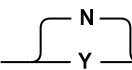
| Parameter | Description                                                                                                                                                                              |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAXSEC    | Maximum secondary allocations for dynamic SPACE allocation of image copy data sets.                                                                                                      |
| OPTION    | Specifies the optional parameters and overriding default values for your recovery environment.                                                                                           |
| OUTPUT    | Specifies the type of output processing to be performed by the IMS Database Recovery Facility.                                                                                           |
| READNUM   | Specifies the maximum number of log read tasks that are started in parallel for the tape only and the total number of concurrent read tasks.                                             |
| REPORT    | Specifies the options for producing reports when using the integrated auxiliary utilities.                                                                                               |
| SORTPARM  | Specifies the parameters that are passed to SORT as each task is started.                                                                                                                |
| SOURCE    | Specifies the source of image copies and log data sets for the recovery process.                                                                                                         |
| SPSIZE    | Specifies the size of each data space used by the IMS Database Recovery Facility.                                                                                                        |
| TAPECHK   | Specifies that the IMS Database Recovery Facility checks the availability status of tape devices before DBDS allocation.                                                                 |
| USESLBIC  | Specifies whether the IMS Recovery Expert System Level Backup (SLB) is used as a recovery asset. The SLB can be used for a full recovery or a recovery within the RCVTIME specification. |
| UTILGBL   | Specifies the parameters used to control the execution of the integrated auxiliary utilities.                                                                                            |
| XCFGROUP  | Specifies the XCF group name used by the IMS Tools Online System Interface to communicate with the IMS systems associated with this recovery environment.                                |

## Reference: Environmental control statements

### CACHE(Y | N)

Use this parameter specified as Y to indicate that the image copy data sets are managed on a VTS (virtual tape server) device and the data sets are to be preallocated to initiate staging to the cache before they are read.

#### CACHE parameter syntax

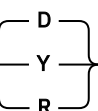
➤ CACHE(  ) ➤

The default value is N, signifying that no image copy data sets are located on a VTS device.

### CATDS(Y | D | R)

Use this parameter to control allocation of image copies, logs, and change accumulation data sets from either the MVS catalog or the RECON data sets.

#### CATDS parameter syntax

➤ CATDS — (  ) ➤

**D**

If a data set is cataloged, the catalog is used for allocation. If a data set is not cataloged, RECON data set information is used for allocation. This is the default value.

**Y**

Data sets must be cataloged, if not cataloged, abend U0384-00000029 is issued in the master address space, and allocation failed RSN: 1708 is issued in the subordinate address space.

**R**

The RECON data set information is used for allocation.

**COMPRTNA(<subparameters>)**

Use this parameter to specify the aliases that you created for IMS High Performance Image Copy compression routines. The options that you specify on this parameter become effective only when you use image copies that are compressed by IMS High Performance Image Copy.

**COMPRTNA subparameters syntax**

```
➔ COMPRTNA( FABJCOMP1(alias name), FABJCOMP2(alias name), FABJCOMP3(alias name), FABJCOMP4(alias name) ) ➔
```

**FABJCOMP1(alias name)**

Use this parameter to specify an alias name that overrides the IMS High Performance Image Copy compression 1 routine.

The default value is FABJCOMP1.

**FABJCOMP2(alias name)**

Use this parameter to specify an alias name that overrides the IMS High Performance Image Copy compression 2 routine.

The default value is FABJCOMP2.

**FABJCOMP3(alias name)**

Use this parameter to specify an alias name that overrides the IMS High Performance Image Copy compression 3 routine.

The default value is FABJCOMP3.

**FABJCOMP4(alias name)**

Use this parameter to specify an alias name that overrides the IMS High Performance Image Copy compression 4 routine.

The default value is FABJCOMP4.

The following example shows how you might use the COMPRTNA parameter to specify aliases.

```
COMPRTNA (FABJCOMP1 (COMPRES1) , FABJCOMP3 (COMPRES3) )
```

**DBDSLnnn**

Use this parameter to supply the data set characteristics for any data sets which are created by recovery. The DBDSLnnn parameter is used to specify allocation characteristics which are used to create new data sets.

This parameter is used if you specify DUP or BOTH on the OUTPUT parameter.

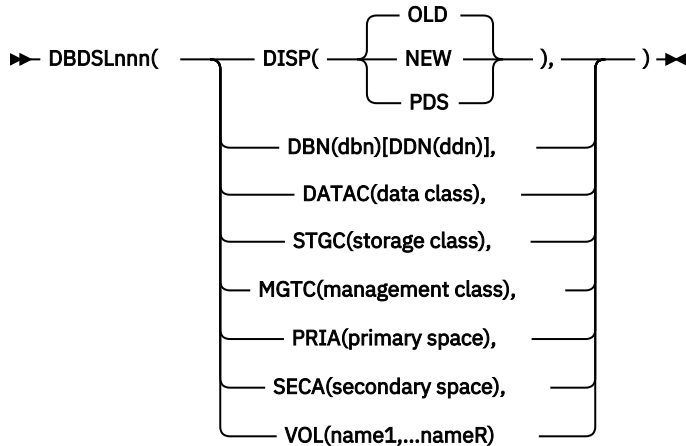
The DBDSLnnn parameters can also be specified on the ADD command using the DBATRB parameter for creating duplicate database data sets. Refer to the ADD command syntax for further details.

You can specify multiple DBDSLnnn control statements where nnn is a unique 1-to 3-digit alphanumeric value.

If you want the different data sets that are created to have unique data set characteristics, you can use multiple data set name (DBDSN $nnn$ ) and data set characteristic (DBDSL $nnn$ ) definitions. The data set characteristics you specify on the DBDSL $nnn$  environmental control statement override the data set characteristics obtained from the original data set.

The following section shows the syntax for specifying data set characteristics.

### DBDSL data set allocation specification syntax



### DISP(OLD | NEW | PDS)

If DISP(OLD) is supplied, an existing data set is used during recovery. If DISP(NEW) is supplied, a new data set is created during recovery. The DISP parameter is required if DBDSL $nnn$  is supplied. If DBDSL $nnn$  is not supplied, then the default disposition used is DISP(OLD).

**Note:** Database data sets that need to be automatically deleted and redefined must be included on the same ADD statement. The DBATRB() keyword must be specified on the ADD statement and must reference a DBDSL $nnn$  that indicates DISP(NEW).

### OLD

IMS Database Recovery Facility uses the characteristics of an existing data set. If you specify OLD, data set characteristics specified in DBDSL $nnn$  are rejected and error message FRD6124A is issued. This value is the default.

If you specify OLD with the OUTPUT(PRO) option, IMS Database Recovery Facility uses existing characteristics from the production database data set in the recovery list.

If you specify OLD with the OUTPUT(DUP) option, IMS Database Recovery Facility uses existing characteristics from the predefined duplicate database data set.

**Note:** For OUTPUT(PRO), DBDSN $nnn$  is ignored. For OUTPUT(DUP), DBDSN $nnn$ (dsn) is required to identify the name of the predefined duplicate database data set.

**Restriction:** When specifying DISP(OLD), you must ensure that your data sets are predefined and cataloged. Specifying DISP(OLD) when you are using data sets that are not predefined or cataloged results in allocation errors during recovery. FRD4302I messages are issued during recovery for this failure.

### NEW

IMS Database Recovery Facility creates a data set using data set characteristics identified in the DBDSL $nnn$  environmental control statement, if you provided it.

If you specified NEW with the OUTPUT(PRO) option, any data set with a matching name is deleted and redefined. IMS Database Recovery Facility deletes and redefines the production database data set using additional characteristics identified in the DBDSL $nnn$  control statement, if you provided it. DBDSN $nnn$  is ignored for OUTPUT(PRO).

If you specified NEW with the OUTPUT(DUP) option, IMS Database Recovery Facility uses DBDSN $nnn$  to generate the duplicate data set name. If a data set exists with that name, the

process ends with condition code 8. You can optionally provide characteristics with `DBDSLnnn` to define the new duplicate database data set. If you do not specify `DBDSLnnn`, characteristics are obtained from the production data set.

If you specified `NEW` with the `OUTPUT(BOTH)` option, IMS Database Recovery Facility deletes and redefines the production database data set and generates the duplicate data set. `DBDSNnnn` is ignored for the production database data set, but used to generate the name for the duplicate. The production database data set is deleted at the same time the duplicate data set is being generated.

If a failure occurs during the generation of the duplicate data set, recovery processing is stopped. You must then manually submit a job to redefine the production database data set before resuming the IMS Database Recovery Facility process.

Storage and management class information from the existing data set is collected and used in allocating the new data set. For SMS-managed data sets, ACS routines based on `STORCLAS` and `MGMTCLAS` can be used for `VOLUME` definition; otherwise, a threshold for 20 `VOLSERs` is set. If ACS routines do not manage `VOLUME` information, the `VOL()` parameter can be used to override the threshold limit for `VSAM` data sets.

**Note:** The following are the attributes and field names that are retrieved from an existing data set if not specified by `DBDSLnnn`:

- `DEVTYPE`
- `VOLSER`
- `STORCLAS`
- `DATACLAS`
- `MGMTCLAS`
- `NVSMATTR`
- `VSAMTYPE`
- `VSAMREUS`
- `ATTR2`
- `PRIMSPAC`
- `SCONSPAC`
- `SPACOPTN`
- `LRECL`
- `AMDKEY`
- `AMDCIREC`
- `PHYBLKSZ`
- `NOBLKTRK`
- `VSAMSTAT`
- `NOBYTTRK`

## **PDS**

Allows the use of a PDS library which contains the IDCAMS delete and define statements for the database data sets. The members within this PDS library must match the DD name of each individual database data set.

If `DISP(PDS)` is specified, IMS Database Recovery Facility searches through the `DDEFPPDS` library by using the DD name, invokes IDCAMS to perform the redefinition of the database data set by using the definitions from the member, and returns control back to IMS Database Recovery Facility for recovery.

A new DD name `DDEFPPDS` must be added as part of the IMS Database Recovery Facility `MAS` or `RSS JCL`. `DDEFPPDS` identifies the PDS which contains the IDCAMS commands and

definitions. If the DDEFPS DD name is specified in both the MAS and the RSS JCL, the DDEFPS DD specified in the MAS JCL prevails if they are different.

The PDS must be defined with LRECL=80 and fixed-block format.

The following example shows sample JCL that would accomplish this process:

```
//DDEFPS DD DSN=IMSTESTL.IDCAMS,DISP=SHR
```

### **DBN(*dbn*)**

The database name (DBN) of the entry to be added to the recovery list with the accompanying attributes until the next DBN is detected or the end of the DBDSL $nnn$  statement is reached.

If you did not specify DBN for a list of attributes, the attributes you specified in the list apply to all the recovery list entries being added with this DBDSL $nnn$  that do not have a DBN or DDN. *dbn* is the database name to which the DBDSL $nnn$  applies.

### **DDN(*ddn*)**

The DD name of the entry to be added to the recovery list with the accompanying attributes. If you did not supply *ddn*, the accompanying attributes apply to all data sets in the database.

*ddn* is the DD name of the database data set to which the DBDSL applies.

### **DATA(*data class*)**

The SMS data class for the output data set. *data class* must be a valid SMS data class.

*data class* is 1-to 8-character alphanumeric string.

### **STGC(*storage class*)**

The SMS storage class for the output data set. *storage class* must be a valid SMS storage class.

*storage class* is 1- to 8-character alphanumeric string.

### **MGTC(*management class*)**

The SMS management class for the output data set. *management class* must be a valid SMS management class.

*management class* is 1- to 8-character alphanumeric string.

### **PRIA(*primary space*)**

The primary space allocation to be used for the new data set. The values specified for primary space can be one of the following values:

#### **B(*xxx*)**

number of bytes

#### **R(*xxx*)**

number of records (VSAM/OSAM LDS) or blocks (OSAM)

#### **K(*xxx*)**

number of kilobytes

#### **M(*xxx*)**

number of megabytes

### **SECA(*secondary space*)**

The secondary space allocation to be used for the new data set. The values can be one of the following values:

#### **B(*xxx*)**

number of bytes

#### **R(*xxx*)**

number of records (VSAM/OSAM LDS) or blocks (OSAM)

#### **K(*xxx*)**

number of kilobytes

#### **M(*xxx*)**

number of megabytes

The volume serial numbers (VOLSER) for location of the new data set.

**DBDSNnnn**

The `DBDSNnnnn` parameter is used to form a template from which the new data set name is created. This parameter is required if you specify `OUTPUT(DUP)` or `OUTPUT(BOTH)` parameter. The `DBDSNnnnn` parameters can also be specified on the `ADD` command using the `DBATRB` parameter for creating duplicate database data sets. Refer to the `ADD` command syntax for further details.

The following example is the syntax of the *DBDSN* parameter:

### Syntax Elements:

## DBDSN name template symbolic qualifiers syntax



- $q = 1$  through 22; no default
- literal = a string that you define of up to 8 characters

Each qualifier in the DBDSN`nnnn` parameter equates to the corresponding qualifier in the new data set that is created. If you specified a null qualifier then the value used for that qualifier is the corresponding qualifier from the original source data set name. Only the qualifier levels that you want to change from the original source data set name need be included in the DBDSN parameter.



You can use periods or commas to separate the qualifier values. Qualifiers that you do not want to change can be omitted using a comma. For example, if you want to change only the first and third qualifiers, specify the following syntax: DBDSN001 ( ' HIGHQUAL ' , , %TIME ). More examples of the use of this parameter follow the descriptions.

If the new data set name is greater than 44 characters or otherwise invalid, an error message is issued and generation of the copy is skipped or the process is stopped, depending on the error handling option, ERROR(STOP | CONT | ABORT) that you chose.

There are two types of values allowed in the data set name qualifiers; literals and predefined keywords. A literal is a static and unchangeable value that you specify and is used directly in the data set name. An example of a literal is %P(TEST) which would result in a prefix of TEST being added to the new data set name.

For a predefined keyword, the value is set by the system according to the current system attributes. An example of a predefined keyword is %TIME which would result in the current timestamp being inserted in the new data set name.

**Keyword Definitions:** The following list shows the qualifiers which allow a literal to be specified.

**%P(literal)**

Generates a prefix to your data set name. The literal is a string of up to eight alphanumeric characters, where the first character must be an alphabetic character.

If used, %P(literal) must be the first qualifier specified.

**%S(literal)**

Generates a suffix to your data set name. The literal is a string of up to eight characters of your choice that is allowable in a data set name.

If used, %S(literal) must be the last qualifier keyword you specify, except for %END.

**literal**

Generates a static character string in your data set name. The literal is a string of up to eight alphanumeric characters that are allowable in a data set name.

The literal must be contained within single quotation marks. If not, error message FRD6124A INVALID DATA ENCOUNTERED is issued, and the subsequent command or control statement report is blank.

The SYSIN DD statements are not displayed.

**< predefined keyword >**

The following qualifiers allow predefined keywords to be defined.

**%TIME =**

Generates a qualifier consisting of a system timestamp in the format of *Thhmmssm*.

Where *T* is a literal character, *hh* is hour in 24 hour format, *mm* is minutes, *ss* is seconds, and *m* is milliseconds.

**%DATE =**

Generates a qualifier that consists of a system date stamp in the format of *Dyyyyddd*.

Where *D* is a literal character, *yyyy* is year and *ddd* is the number of days.

**%JOB =**

Generates a qualifier that consists of the IMS Database Recovery Facility job name.

**%STEP =**

Generates a qualifier that consists of the IMS Database Recovery Facility step name.

**%MDBC =**

Generates a qualifier that consists of the HALDB master database name for a HALDB or the database name for a non-HALDB.

**%DBD =**

Generates a qualifier that consists of the HALDB partition name for a HALDB or the database name for a non-HALDB.

**%DDN =**

Generates a qualifier that consists of the DD name for the database data set.

**%SDSN(q) =**

Generates a qualifier from the source data set name. This element requires a value (*q*) that specifies the number of the qualifier referenced from the source data set name. The valid values of *q* are 1 - 22.

This qualifier is only needed if you want to use a qualifier from the source data set name but want it to appear in a different position in the new data set name.

If %SDSN(*q*) refers to a qualifier in the source data set name that is greater than the number of qualifiers in the data set, then the last encountered qualifier is used. For example if DBDSN101( ' TESTDB ' ,%SDSN(4) ) is used as the data set name template and the original data set name is 'PRODDB.PARTS.INV', then the generated data set name is 'TESTDB.INV'.

**%END =**

No further qualifiers are to be included. Use this qualifier to truncate the new name.

**Rules for specifying qualifiers:**

Each DBDSN $nnn$  parameter can specify any number of qualifiers to a maximum of 22. If the source data set name consists of fewer qualifiers than are specified in the parameter, then the extra qualifiers correspond to the later qualifiers in the source.

For example, if the source data set name is PRODDB.PARTS.IMSA.INV and DBDSN001(,,,%,DATE,%,TIME) is specified, the resulting new data set name would be PRODDB.PARTS.Dyyyyddd.Thhmmss. As you can see, the source data set name consisted of only four qualifiers but the new data set name consisted of six possible qualifiers. The last two qualifiers of the source data set name were replaced by a date stamp and a time stamp.

If you specifying %P(literal), it must be the first qualifier in the list.

If you specifying %S(literal), it must be the last qualifier in the list, just before the %END.

If the source data set name consists of more qualifiers than are specified in the DBDSN $nnn$  parameter, then the additional qualifiers from the source data set name are included in the new data set name in their corresponding positions.

**Examples of specifying DBDSN $nnn$ :**

The following are examples of specifying DBDSN $nnn$  with positional notation:

**DBDSN001()**

Use the source data set name for the new data set name.

**DBDSN001(%P(BKU))**

Prefix the first database data set name with BKU. For example, if the source data set name is:

```
IMS1.REGTST.SAMPLIB.V345
```

Your new data set name would be:

```
BKU.IMS1.REGTST.SAMPLIB.V345
```

**DBDSN002(,,,%,TIME)**

Use the system time as the fifth qualifier in the new data set name. The first four qualifiers are copied from the source data set name and the time stamp is used as the fifth qualifier.

If the source data set name has fewer than five qualifiers, the time stamp is substituted for the last qualifier. For example, if the source database data set name is:

```
IMS1.REGTST.SAMPLIB.V345
```

Your new data set name would be:

```
IMS1.REGTST.SAMPLIB.Thhmmss
```

Even though the DBDSN $nnn$  indicated that %TIME is to be used as the fifth qualifier, because the input data set has only four qualifiers, %TIME was substituted as the last qualifier in the new data set name.

#### **DBDSN001(,,,%,DATE,%,TIME)**

Use the system date as the fourth qualifier and the system time as the fifth qualifier in the new data set name.

The first three qualifiers are copied from the source data set name. For example, if the source database data set name is:

```
IMS1.REGTST.SAMPLIB.V345.TEST
```

Your new data set name would be:

```
IMS1.REGTST.SAMPLIB.Dyyyyddd.Thhmmss
```

If there are 5 or fewer qualifiers in the source data set name, date and time are substituted for the last two qualifiers.

#### **DBDSN001('HIGHQUAL')**

Use the literal 'HIGHQUAL' in place of the high-level qualifier and use the remaining qualifiers from the source data set name to create the new data set name.

The commas for null values are not necessary in this case.

#### **DBDSN001('HIGHQUAL',,,,%,TIME)**

Use the literal 'HIGHQUAL' as the high-level qualifier and time as the fifth qualifier for the new data set name. If there were four or fewer qualifiers in the source data set name, time is substituted for the last qualifier.

The shortest possible generated name in this case would consist of two qualifiers.

#### **DBDSN001('HIQUAL',%SDSN(1),,%,TIME,%END)**

Use the literal 'HIQUAL' as the high-level qualifier, use the existing high-level qualifier as the second-level qualifier, use the existing third qualifier, and use time as the fourth qualifier to create the new data set name.

End the name after the time qualifier.

#### **DBDSN002(%S(UPDT))**

Suffix the new data set name with UPDT. For example, if your source data set name is:

```
IMS2.REGTST.SAMPLIB.V346
```

Your new data set name would be:

```
IMS2.REGTST.SAMPLIB.V346.UPDT
```

Specifying DBDSN002(%S(UPDT),%END) gives you the same result.

#### **DRFIAX(procname)**

Use this parameter to specify the name of the procedure that is used to initiate the IMS Database Recovery Facility Utility Address Space (UAS/IX) for IMS Index Builder, the image copy function of IMS HP Image Copy, DFSPRECO, and the Build Index function of FPA.

The default for the DRFIAX parameter is FRXJCLIP.

The procedure must reside in a valid z/OS PROCLIB data set.

Samples of this procedure are located in SFRXSAMP(FRXIAX|FRXJCLIP).

**Tip:** Customize the FRXIAX procedure and rename to FRXIAXxx, where xx is the maintenance level used (for example, TEST or PROD).

#### DRFIAX parameter syntax

➔ DRFIAX( — { FRXIAXxx  
FRXJCLIP  
procname } — ) ➔

#### DRFPROC(procname)

Use this parameter to specify the name of the procedure that is used to initiate the IMS Database Recovery Facility recovery sort subordinate address space (RSS).

The procedure must reside in a valid z/OS PROCLIB data set. If you specify this parameter in the SYSIN control statements, the procedure name that is defined by the SET statements in your execution JCL is overridden.

Samples of this procedure are located in SFRXSAMP(FRXRSS|FRXJCLSB).

**Tip:** Customize the FRXRSS procedure and rename to FRXRSSxx, where xx is the maintenance level used (for example, TEST or PROD).

#### DRFPROC parameter syntax

➔ DRFPROC( — { FRXRSSxx  
FRXJCLSB  
procname } — ) ➔

#### HPIO

Use this parameter to specify whether HPIO writes to any OSAM or VSAM ESDS DBDS during recovery.

HPIO write is not supported for HISAM Overflow DBDS.

HPIO write is not supported for OUTPUT(BOTH) and is dynamically disabled when OUTPUT(BOTH) is specified.

#### HPIO parameter syntax

➔ HPIO( — WRITE( — { N  
Y } — , — { 5  
nn } — ) — ) ➔

The WRITE option sets HPIO write either ON (Y) or OFF (N) for writing to the DBDS during recovery.

```
WRITE([Y|N],[nn])
```

#### WRITE(Y,[nn])

The optional *nn* subparameter specifies a 1-2 digit integer value ranging from 5-15 that sets the number of HPIO write threads.

The default value for *nn* is 5.

#### WRITE(N)

WRITE(N) is the default setting.

For WRITE(N), *nn* is ignored.

#### ICNOTIFY(HPIC | DRF)

ICNOTIFY determines if IMS HP Image Copy (HPIC) or IMS Database Recovery Facility (DRF) causes the NOTIFY.IC commands to be issued to register the image copies generated by the DRF job, if any, to DBRC.

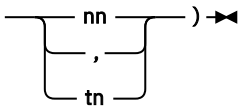
If ICNOTIFY is not specified, IMS HP Image Copy (HPIC) causes the NOTIFY.IC commands to be issued.

**Note:** ICNOTIFY(DRF) and ICNDX(Y) are mutually exclusive.

### ICNUM(*nn,tn*)

Use this parameter to specify the maximum number of tape drives used for reading image copy data sets (*nn*). For ICNUM, (*tn*) is ignored.

#### ICNUM parameter syntax

➤ ICNUM(  ) ➤

If you do not specify ICNUM, then the default values are those values that are used by READNUM.

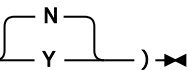
If ICNUM or READNUM are not specified, the default values for ICNUM are the same default values used by READNUM.

Refer to the description of READNUM for more details.

### LBI(Y | N)

Use this parameter to indicate whether the large block interface is in use at your location (Y), or not (N).

#### LBI parameter syntax

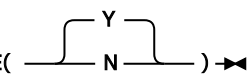
➤ LBI(  ) ➤

The default value of N is assumed if you do not make a specification.

### LCLTIME(Y | N)

Specifies whether local time stamps (Y) are to be used in messages and reports, or if GMT time stamps (N) are to be used.

#### LCLTIME parameter syntax

➤ LCLTIME(  ) ➤

The default is LCLTIME(Y).

### LIU@GOPT(*xxxx*)

Use this parameter to identify the LIU global option module LIU@xxxx, which is loaded from the STEPLIB by Library Integrity Utility. This allows the Library Integrity Utility to load global option member LIU@xxxx to accept user defined run time options that have been assembled therein.

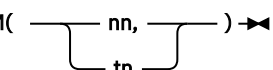
#### LIU@GOPT(*xxxx*) parameter syntax

➤ LIU@GOPT( *xxxx* ) ➤

### LOGNUM(*nn,tn*)

Use this parameter to specify the maximum number of tape drives used for reading change accumulation data sets (*nn*) and the maximum number of read instances which can be initiated in parallel (*tn*). LOGNUM operates in the same way that READNUM does, but is associated with reading change accumulation data sets. Refer to the description of READNUM for more details.

#### LOGNUM parameter syntax

➤ LOGNUM(  ) ➤

If you do not specify LOGNUM then the default values are those that are used by READNUM.

If LOGNUM or READNUM are not specified, the default values for LOGNUM are the same default values used by READNUM. Refer to the description of READNUM for more details.

### MAXPRI(nnnn)

This optional parameter limits the space, in cylinders, of the primary allocation for the output image copy when the DYN option of the auxiliary parameter SPACE is specified.

Primary allocation is calculated based on the number of records in the input image copy used for recovery.

Range: 1-32767

Default: 6000

### MAXSEC(nnnn)

This optional parameter limits the space, in cylinders, of the secondary allocation for the output image copy when the DYN option of the auxiliary parameter SPACE is specified.

Secondary allocations are calculated as 10% of the primary allocation +1.

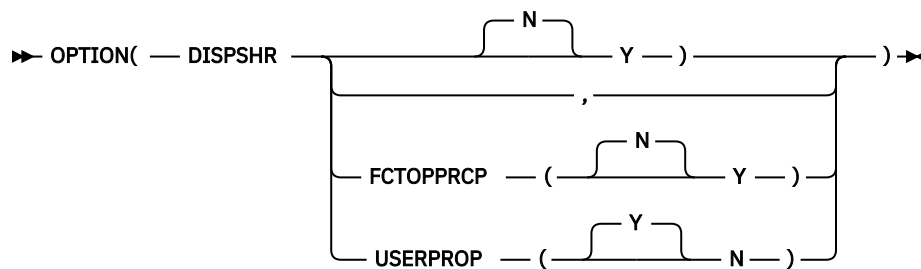
Range: 1-3276

Default: 600

### OPTION(<subparameters>)

Use this parameter to specify optional parameters and overriding default values for your recovery environment.

#### OPTION subparameters syntax



You can specify DISPSHR, FCTOPPRCP, USERPROP, or a combination of these sub parameters.

### DISPSHR(Y | N)

Specifies whether the default dynamic allocation for image copy, log and change accumulation data sets will be allocated with a disposition of share (Y) or not (N). The default is N which means that these data sets will be allocated using DISP=OLD.

Allocating these data sets with a DISP=OLD helps ensure that no other jobs attempt to update the contents of your input data sets while a recovery is in process.

The DISP=SHR setting gives users the flexibility of running multiple jobs in parallel if all that is expected by these jobs is READ access.

### FCTOPPRCP(Y | N)

Specifies whether the DFSMSdss FCTOPPRCPPrimary option is to be used (Y) or not (N) when restoring from a FlashCopy® image copy. The default is N.

For details, see the description about FCTOPPRCPPrimary in the *z/OS DFSMSdss Storage Administration Reference*.

**Note:** FCTOPPRCP is ignored when IMS Database Recovery Facility is not restoring from a FlashCopy or if the release level of DFSMSdss is lower than Version 1 Release 6. FCTOPPRCP is supported by DFSMSdss Version 1 Release 6, or higher.

## USERPROP(Y | N)

Specifies whether the userid that is associated with the IMS Database Recovery Facility master job is propagated to the subordinate address spaces that are directly started by IMS Database Recovery Facility.

These address spaces include the recovery sort subordinate address space, the IMS Index Builder address space, the DFSPREC0 utility address space, and the address space for the Build Index function of FPA.

Userid propagation allows the subordinate address space to run with the same level of security as the IMS Database Recovery Facility master job, eliminating the need for special entries in the RACF® STARTED class.

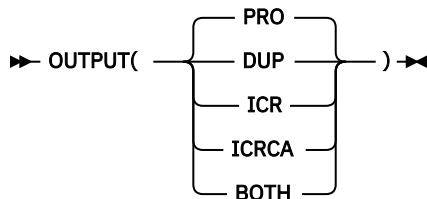
The default is Y, indicating that the userid propagation is performed.

This parameter is useful in cases where the TEMPDSN class is active and you need to manually code the SORTWORK DD statements in the DRFPROC() procedure.

## OUTPUT(DUP | ICR | ICRCA | PRO | BOTH)

Use this parameter to specify the type of output processing to be performed by the IMS Database Recovery Facility. The default is OUTPUT(PRO) which recovers all the production database data sets associated with the recovery list.

### OUTPUT parameter syntax



**Note:** The term **production DBDS** can be interpreted as the DBDS listed in the RECON data sets for the supplied DB name DD name pair. The term **non-production DBDS** can be interpreted as a copy of the DBDS listed in the RECON data sets for the supplied DB name DD name pair.

### DUP

Specifies Database Copy Generation, where a non-production duplicate of every DBDS in the recovery list is created using the recovery process. This results in creating a duplicate of each production DBDS with a new data set name. This DBDS duplicate can be processed in a test environment on a separate IMS system using the same DBD name, or it can be processed in the production environment if you provided a different DBD.

The DBDSN $nnn$  statement is used to create the data set name for the duplicate data sets and is required. The DBDSL $nnn$  statement can be used to specify the allocation attributes for creating the duplicate data sets.

DBDSL $nnn$  is optional and if it is not specified, the allocation attributes are taken from the original data sets. The DBATRB parameter must be specified on the ADD statement to specify the DBDSN $nnn$  and the DBDSL $nnn$  (if specified).

Sample JCL using OUTPUT(DUP):

```
DBDSN002('MYDATB')  
DBDSL004(DISP(NEW),DBNMYDB1))  
OUTPUT(DUP)  
ADD DB(DBOVLFPV) DBATRB(DBDSN(002) DBDSL(004))  
START
```

In this sample, duplicate database data sets are created for all database data sets associated with the DBOVLFPV database. The first level qualifier on the new data sets are 'MYDATB'. The allocation characteristics for the new data sets are taken from the original data sets.

### Restrictions:

- User-ICs or SAMEDS ICs are not supported with this function. IB and IC statements are ignored.
- If you are recovering to an OLR-capable database, OUTPUT(DUP) is not recommended.

Since OUTPUT(DUP) creates a copy of the "active" data sets for the current recons of the source system, it may not match the "active" data sets on the target system for the duplicated data sets.

You would be required to make sure the OLR status (active data sets) is the same on both systems and you should not use OUTPUT(DUP) for a copy of the database data sets (active or inactive) if either the source or target system is in the middle of an OLR.

## ICR

Specifies Image Copy, where a new image copy is created from an image copy and log, or change accumulation input data sets, for every DBDS in the recovery list by using the recovery process. The RCVTIME parameter can be specified to create the image copy to a specific point in time. Actual database recovery is not performed and the databases are not accessed.

If you specify OUTPUT(ICR) while the databases are offline, or specify RCVTIME to a time when the databases are offline, the image copy is registered to the RECON as BATCH.

If you specify OUTPUT(ICR) while the databases are online, or specify RCVTIME to a time when the databases are online, the image copy is registered to the RECON as CONCUR.

You cannot specify ICR with any other output option.

Sample OUTPUT(ICR) control statements:

```
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=DRFIC1)
UTILGBL(COMP(N),DSN(&ICHLQ..&DBD..&DDN.),UNIT(3390),-
VOLSER(222222))
OUTPUT(ICR)
ADD DB(DIVNTZ02,DHVNTZ02,DXVNTZ02) IC(ICHLQ=(DRFIC1),-
SPACE=(CYL,10,10))
START ERROR(CONT)
```

**Note:** Use the DSN() and DSN2() parameters of the UTILGBL() control statement to specify the name of the output image copy data set. You cannot use the DSN() or DSN2() parameters on the IC() control statement.

**Restriction:** The use of user-image copies, IC2, SAMEDS ICs, or FlashCopies are not supported with the ICR option. Use of a PITCA created by the IMS High Performance Change Accumulation is not allowed as input to the ICR process

## ICRCA

This option is like ICR, but it specifies that only a BATCH image copy is to be generated using only a prior image copy and a change accumulation as input. Log input is ignored.

IMS Database Recovery Facility locates and uses a timestamp where a clean recovery point or batch window exists. A clean recovery point or batch window, is defined as a period of time where the database was previously taken offline or where an OLDS switch occurred with logs archived.

IMS Database Recovery Facility uses this timestamp if a complete change accumulation also exists within this time frame. All databases that are included in the recovery list must belong to the same CAGRP. The resulting image copy is registered as a batch image copy and contains all updates up to this change accumulation.

If a complete change accumulation is not found within a usable batch window then a copy of the prior image copy is generated.

**Restriction:** The use of user-image copies, IC2, SAMEDS ICs, or FlashCopies are not supported with the ICRCA option. Use of a PIT CA created by the IMS High Performance Change Accumulation product is not allowed as input to the ICRCA process.

## PRO

Specifies recovery of one or more production databases, where every DBDS in the recovery list is recovered. This results in creating a recovered DBDS in place of the previous instance.



This is the default option for the OUTPUT keyword.

**Note:** If you specified OUTPUT(PRO), the DBATRB parameter is optional on the ADD statement.

## **BOTH**

Specifies that recovery performs the functions of PRO and DUP at the same time. Recovery creates a recovered DBDS in place of the previous instance and generate a non-production duplicate with a new data set name.

### **Restrictions:**

- If you specified OUTPUT(BOTH), you must also include the DBATRB parameter on the ADD statement.

The DBATRB parameter identifies which DBDSNnnn to reference for naming convention rules.

It can also specify a DBDSLnnn statement to identify allocation attributes for the new database data sets.

- If you are recovering to an OLR-capable database, OUTPUT(BOTH) is not recommended.

Because OUTPUT(BOTH) creates a copy of the active data sets for the current RECONS of the source system, it might not match the active data sets on the target system for the duplicated data sets. You would be required to make sure the OLR status (active data sets) is the same on both systems.

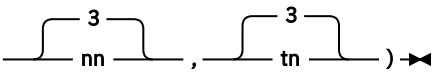
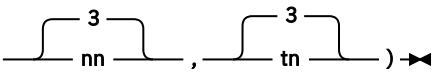
Do not use OUTPUT(BOTH) for a copy of the database data sets (active or inactive) if either the source or target system is in the middle of an OLR.

## **READNUM(nn,tn)**

Specify the maximum number of log read tasks that are started in parallel for tape only (*nn*) and the total number of concurrent read tasks (*tn*).

**Tip:** If possible, use the same system for recovery that was used to create the logs. If you are using a system with less capacity than the original system, consider reducing the READNUM (*nn*) value. If the DRF log read task (RVRD) abends with S878, consider reducing the READNUM (*tn*) value.

### **READNUM parameter syntax**

➡ READNUM(  ,  ) ➡

For tape input, specify this number as the maximum number of tape read devices (*nn*) to allocate to the recovery process.

The allowable range is 1 - 99. The default value for *nn* is 3.

The total number of concurrent read threads allowed is specified by *tn*. The actual number might be larger than *tn* if some input data sets exist that can be read from devices other than tape drives. The default maximum number of total concurrent read threads is *tn* or 3, whichever is greater.

If you omit or set the values of either *tn* or *nn* to 0, the default values are imposed.

The values specified in the FRXDRFxx member can be overridden using the READNUM parameter on the START control statement.

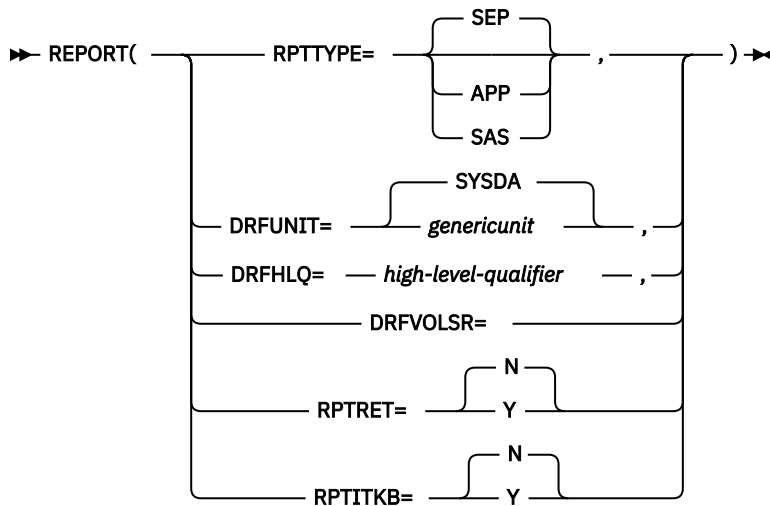
If you do not specify READNUM, the system default of (3,3) is used.

**Note:** If ICNUM or LOGNUM are not specified, the values used for READNUM are used.

## **REPORT(<subparameters>)**

IMS Database Recovery Facility and the integrated auxiliary utilities write messages to various report DD statements. The IMS Database Recovery Facility master address space collects all messages and reports from all IMS Database Recovery Facility subordinate address spaces and organizes them in either the master address space REPORT DD or in separate utility report DD statements, such as the STATIPRT for IMS High Performance Pointer Checker. The REPORT parameter is used to control how and where these reports are created.

## REPORT parameter syntax



### RPTTYPE=

Specifies where the messages generated by the integrated auxiliary utilities are written.

#### SEP

Indicates that the utilities reports and messages are to be written to the standard utility report DD statements, separate from the IMS Database Recovery Facility REPORT DD.

RPTTYPE=SEP is the default setting.

#### APP

Indicates that the utilities reports and messages are appended to the IMS Database Recovery Facility REPORT DD.

#### SAS

Indicates that the utilities reports and messages are not sent to the master address space. Instead, they are written to the standard integrated auxiliary utilities report DD statements in the subordinate address spaces in which they were generated.

RPTTYPE=SAS reduces overall processing time and I/O for IMS Database Recovery Facility.

### DRFUNIT=

Specifies a 1- to 7-character generic unit name passed to the subordinate address spaces in order to provide a unit type on which to dynamically allocate and catalog any report data sets. The default is SYSDA.

If *genericunit* is coded with a value that is not a valid generic unit in your system, messages FRD9003A and FRD4100I are issued and followed by abend U384-02C.

### DRFHLQ=

Specifies a 1- to 8-character high-level qualifier that is passed to the subordinate address spaces in order to provide a high-level qualifier used to allocate and catalog any report data sets.

### DRFVOLSR=

Specifies a 1- to 6-character volume serial number that is passed to the subspaces in order to provide a specific volume on which to allocate any report data sets. Use DRFVOLSR to specify a particular volume serial number on which to dynamically allocate and catalog any required data sets.

Specifying DRFVOLSR is optional. If you do not specify DRFVOLSR, the temporary data sets are written to local volumes. DRFVOLSR is required when you have insufficient local volumes or if the volumes are not SMS-managed.

### RPTRET=

Specifies the disposition of report data sets when the recovery job abends.

## N

Indicates that the REPORT work data sets are to be deleted in the event of an abend.

It also indicates that IMS Database Recovery Facility deletes any REPORT work data sets left from prior executions that would otherwise cause allocation errors due to duplicate names.

## Y

Indicates that the REPORT work data sets are to be retained in the event of an abend.

Retaining these data sets might be helpful for diagnostic purposes, however, if they are retained then it is up to you to manually delete them.

The default setting for RPTRET is N.

## RPTITKB=

Specifies whether or not IMS Tools Knowledge Base integration is active for the IMS Database Recovery Facility report.

**Important:** To use IMS Tools Knowledge Base integration, the ITKBSRVR keyword must be specified in the UTILGBL() environmental statement.

## N

Indicates that IMS Tools Knowledge Base integration is not active.

## Y

Indicates that IMS Tools Knowledge Base integration is active.

There are three reports types:

- DRF report
- WTO
- SYSPRINT

The default setting for RPTITKB is N.

After a batch recovery completes, examine, first, the REPORT DD output file in the master address space to determine the results of the auxiliary utilities.

The return codes and reason codes from each utility for each DBD and DD/AreaName are written under a new heading in the REPORT DD at the bottom of the file. If you invoked integrated auxiliary utilities and RPTTYPE=APP is specified, then reports and messages from the utilities appear in the REPORT DD after the IMS Database Recovery Facility report.

If RPTTYPE=SEP is specified, then reports and messages from the utilities are written separately to the standard utility report DD statements.

If RPTTYPE=APP is specified indicating the messages are appended to the REPORT DD, IMS Database Recovery Facility lists image copy messages followed by Pointer Checker messages under a separate heading by DBD and DD/AreaName, followed by messages for Index Builder, DFSPRECO, and the Build Index function of FPA.

The JOBNAME of the subordinate address space which created the utilities' reports and messages is also included in the heading.

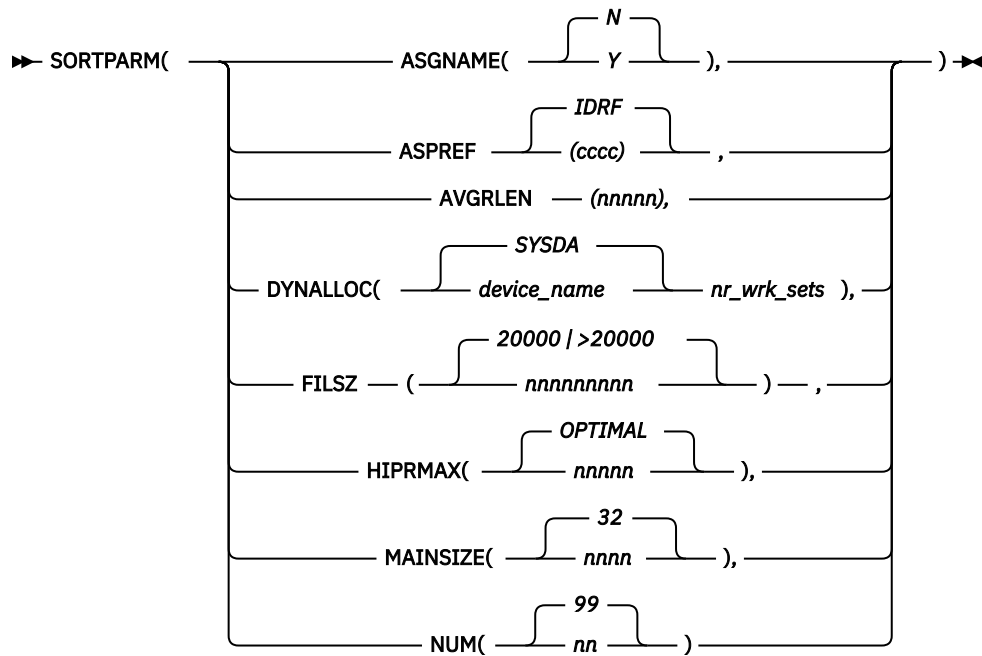
For a list of the utility report DD statements, refer to [Chapter 6, "Configuring JCL statements and procedures,"](#) on page 93.

If RPTTYPE=SAS is specified, the reports are written to the integrated auxiliary utilities standard report DDs in the SAS in which they were generated.

## **SORTPARM(<subparameters>)**

The subparameters listed here are values that you define that are passed to SORT as each task is started. Separate the *subparameters* with commas.

## SORTPARM subparameters syntax



### ASGNAME(*Y*|*N*)

Use this parameter to set a generic Started Task Control (STC) name for the sort subordinate address spaces created by the IMS Database Recovery Facility. This is an optional parameter.

#### **Y**

Specifies that IMS Database Recovery Facility overrides the job name for all sort subordinate address spaces in order to use DRFSORT as the started task name.

#### **N**

Specifies that the name used for all sort subordinate address spaces is generated using the information specified by the ASPREF parameter. This is the default.

See ASPREF for the method of naming subordinate address spaces.

### ASPREF(*cc* | *cccc* | **IDRF**)

Use this parameter to set the prefix of the started task name for the recovery sort subordinate address spaces (RSS) created by the IMS Database Recovery Facility. This is an optional parameter.

If ASGNAME(*N*) is specified and ASPREF() is not specified, the default prefix of IDRF is used.

#### **cc**

When a 2-character prefix is specified, the job name is influenced by the NUM parameter.

If the value of NUM is less than (<) 100, the job name is constructed by using the format *ccjjjj##*, where:

#### **cc**

The specified prefix.

#### **jjjj**

The last 4 digits of the JES job number associated with the master address space.

#### **##**

A number from 01-99, incremented with each new RSS.

If the value of NUM is greater than (>) 99, the job name is constructed by using the format *ccjjjj###*, where:

#### **cc**

The specified prefix.

**jjj**

The last 3 digits of the JES job number associated with the master address space.

**###**

A number from 1-250, incremented with each new RSS.

**cccc**

A 4-character prefix used to construct the STC name of the RSS.

When a 4-character prefix is specified, the job name is constructed by using the format *cccc####*, where:

**cccc**

The specified prefix.

**####**

A number from 0001-0250, incremented with each new RSS.

### **AVGLEN(nnnnn)**

Use this parameter to specify the average record length of the records to be sorted.

**nnnnn**

A 1-to 5-digit integer value of 4- through 32766 bytes that you use to specify the average record length of the records to be sorted.

The default value of *AVGLEN(nnnnn)* is half of the maximum record length.

The range includes the 4 -byte record descriptor word (RDW).

### **DYNALOC(device\_name | SYSDA , nr\_wrk\_sets)**

Use this parameter to specify the attributes for allocating the sort work data sets dynamically. The parameters that you specify is used in place of the SORT defaults.

**device\_name | SYSDA**

A valid device name on which the work data sets are dynamically allocated.

The default value of *device\_name* is SYSDA.

**nr\_wrk\_sets**

A 1-to 3-digit integer value from 1 to 255 that you use to specify the number of dynamically allocated work data sets that are used during the SORT.

The default for *nr\_wrk\_sets* is none, that defaults to either the IBM-supplied default (4 for block set technique), or as you specified in the installation ICEMAC DYNALOC option.

### **FILSZ(nnnnnnnnn | 20000 | > 20000)**

Use this parameter to specify the estimated number of records to be sorted.

**nnnnnnnnn**

A 1- to 9-digit integer value you use to specify the estimated number of records to be sorted.

This parameter is optional, and if you do not include it, the IMS Database Recovery Facility makes an estimate based on the RLDS record sequence number range provided by DBRC.

The minimum estimate that IMS Database Recovery Facility applies is 20000.

### **HIPRMAX(OPTIMAL | nnnnn)**

Use this parameter to specify the maximum size of hiperspace to be used in the sort process.

**OPTIMAL**

If you specify the OPTIMAL option, SORT is allowed to calculate and use as much hiperspace as possible.

The HIPRMAX default is OPTIMAL.

**nnnnn**

A 1- to 5-digit integer value from 0 MB to 32766 MB that you use to specify the maximum amount of hiperspace to be used.

**Note:** Specifying a value of 0 tells SORT to inhibit the use of hiperspace sorting.

### **MAINSIZE(*nnnn* | 32)**

The main storage size you set for the RSS SORT region.

#### ***nnnn***

A 1- to 4-digit integer value (4 through 2000; in megabytes) that you set for the RSS SORT region. The default is 32 MB.

**Note:** If the value you specify is less than the SORT product's MINLIM installation value, then the MINLIM value overrides MAINSIZE.

### **NUM(*nnn* | 250)**

A 1- to 3-digit integer value that you use to specify the maximum number of RSS spaces that are started in parallel.

The actual number of RSS spaces depends on the work required, so the number of subordinate address spaces might be less than the value you specified.

The number that is used is the smaller of the following values:

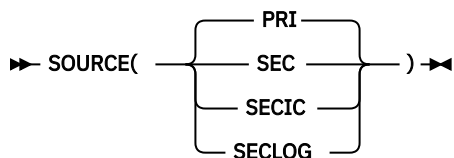
- The value *nnn* you specified in the NUM parameter.
- The number of unique collections of image copy data sets. These collections can be one or more of the following options:
  - An individual image copy data set on a DASD volume
  - Stacked image copy data sets on the same tape volume that do not span tape volumes
  - An image copy data set that spans volumes plus the image copy data sets stacked on the last volume of the spanning image copy
- The number 250 (the maximum value allowed).

If multiple image copy data sets that are required for recovery are stacked in the same volume or are part of the same data set (when they are created using the SAMEDS option for DFSMSdss - Concurrent Copy), they are processed by the same address space.

### **SOURCE(PRI | SEC | SECIC | SECLOG)**

Use this parameter to specify the source of image copies and log data sets for the recovery process.

#### **SOURCE parameter syntax**



#### **PRI**

The primary image copy and log data sets are used as the sources for the recovery process.

If any of the primary sources are marked as invalid or are unreadable, the secondary sources are used, if available.

PRI is the default option for SOURCE.

#### **SEC**

The secondary image copy and log data sets, either SECLOG or SECSTD, are used as the sources for the recovery process.

If a required data set is marked as invalid or it is unavailable, recovery fails for that database data set.

#### **SECIC**

The secondary image copy data sets are used for the recovery process.

Primary log data sets are used as the default data sets.

## SECLOG

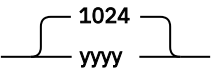
The secondary log data sets, either SECLOG or SECSLD, are used for the recovery process.

Primary image copy data sets are used as the default data sets.

## SPSIZE(yyyy | 1024)

Use this parameter to specify the integer value in megabytes, for the size of each data space used by the IMS Database Recovery Facility.

### SPSIZE parameter syntax

►► SPSIZE(  ) ►►

When data space usage reaches your specified value, another data space is obtained. The range of valid values is: 1 to 2047.

The default value for SPSIZE is 1024 MB.

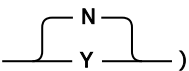
### yyy

The integer value in megabytes, for the size of each data space used by the spill manager.

## TAPECHK(Y | N)

Use this parameter (Y) to allow IMS Database Recovery Facility to check the availability status of tape devices before DBDS allocation.

### TAPECHK parameter syntax

►► TAPECHK(  ) ►►

If the number of image copies restore tasks on tape exceeds the number of available tape devices, then IMS Database Recovery Facility might stop scheduling parallel image copy restores when the number of tape devices that are required exceeds the number tape device that are available.

TAPECHK(Y) works in tandem with READNUM (or ICNUM, LOGNUM) to specify to IMS Database Recovery Facility the number of tape devices that are available.

TAPECHK(N) is the default.

## USESLBIC(Y | N)

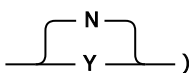
Specifies whether an IMS Recovery Expert System Level Backup will be used for recovery if it was created later than the image copy, whether for a full recovery or within the RCVTIME specification.

If IMS Recovery Expert is available and USESLBIC(YES) is specified, the Verify and Recovery functions cause IMS Recovery Expert to write an image copy record to the RECON which represents the System Level Backup. The image copy record receives a data set name in the form of SLB.Ixxxx.Dxxxx.dbdname.ddname, and this image copy record is used during the Verify and Recovery functions.

If there is already an image copy record for the System Level Backup or if no system level backup is available relative to the type of recovery specified then no image copy record will be written to the RECON.

**Note:** It is strongly suggested that USESLBIC is specified to match the same parameter in IMS Database Recovery Facility Extended Functions, especially when its Verify function is being used to determine the recovery time.

### USESLBIC parameter syntax

►► USESLBIC(  ) ►►

The default value of N is assumed if you do not make a specification.

### Reference: SLB offloaded data sets

SLBs exist either on a fast-replication capable device(s), or have been offloaded from the fast-replication devices onto sequential data sets on tape or DASD.

When on the fast-replication device, the data is an exact replica of the backed up data except that the data sets are not cataloged. During the offload process, IMS Recovery Expert creates one sequential data set per device and catalogs them on the offload volumes. An SLB can be used for recovery whether it resides on the fast-replication capable device(s) or the offload data sets.

For more detailed information, refer to the *IMS Recovery Expert User's Guide*.

### **UTILGBL(<subparameters>)**

Use this parameter is used to specify the control statements that are used to control the execution of the integrated auxiliary utilities.

This parameter, and all the associated subparameters are described in [Chapter 5, "Configuring the integrated auxiliary utilities environment,"](#) on page 63.

### **XCFGROUP(name)**

Use this parameter to specify the XCF group name used by the IMS Tools Online System Interface to communicate with the IMS systems associated with this recovery environment.

#### **XCFGROUP parameter syntax**

➤ XCFGROUP — ( — name — ) ➤

#### **name**

*name* is the 1-to 8-character alphanumeric name specified in the IMS Tools Online System Interface installation.

To find the IMS Tools Online System Interface XCF group name, look in the IMS PROCLIB member FOIxxxxP (where xxxx is the IMSID). In that member, the XCFGROUP= parameter specifies the last five characters of the XCF group name.

To construct the complete name, prefix this value with TOI. Or you can look in your IMS online job log for the FOI100I message; the value that is specified after XCF GROUP= indicates the complete IMS Tools Online System Interface XCF group name.

If this parameter is not specified, then automatic /DBR and /STA of databases will not be possible. There is no default for this parameter.



---

# Chapter 5. Configuring the integrated auxiliary utilities environment

Control statements are used by IMS Database Recovery Facility to specify the operation of the integrated auxiliary utilities.

**Topics:**

- [“Specifying utility control statements” on page 63](#)
- [“Reference: Utility control statement parameters” on page 63](#)
- [“Reference: Utility control statements” on page 69](#)

---

## Specifying utility control statements

Utility control statements are used to specify the operation of the integrated auxiliary utilities.

**Before you begin**

The integrated auxiliary utilities include IMS Index Builder, IMS High Performance Image Copy, IMS High Performance Pointer Checker, IMS High Performance Fast Path Utilities, the IMS DFSPRECO utility, and the Build Index function of FPA.

**Procedure**

- To set the default processing options for the utilities, use the UTILGBL() global environmental control statement.
- To override any options that are specified on UTILGBL() or to specify any additional options, use the appropriate ADD command:

**ADD IB()**

Use this ADD command to specify FPA and IMS Index Builder utility control statements.

**ADD IC()**

Use this ADD command to specify IMS High Performance Image Copy utility control statements.

**ADD PC()**

Use this ADD command to specify IMS High Performance Pointer Checker and IMS High Performance Fast Path Utilities utility control statements.

Any values that you specify on the ADD() command override the values that were specified on the UTILGBL() global environmental control statement.

---

## Reference: Utility control statement parameters

The following table lists the utility control statement parameters that are used to specify the operation of the integrated auxiliary utilities.

The parameters are listed in alphabetic order. Each row in the table indicates whether the parameter is valid on the UTILGBL() environmental statement and on the ADD IB(), ADD IC(), or ADD PC() commands. The table also indicates which utility the parameter is associated with.

| Utility control statement parameter | Integrated auxiliary utility                      | Valid on UTILGBL() | Valid on ADD(IB()) | Valid on ADD(IC()) | Valid on ADD(PC()) | Description                                                                                               |
|-------------------------------------|---------------------------------------------------|--------------------|--------------------|--------------------|--------------------|-----------------------------------------------------------------------------------------------------------|
| BITMAP                              | IMS HP Pointer Checker                            | Yes                | No                 | No                 | Yes                | Specifies whether to generate the Bit Map Display report for the HDAM, HIDAM, PHDAM, or PHIDAM database.  |
| BLD_PRIMARY                         | IMS Index Builder                                 | Yes                | Yes                | No                 | No                 | Specifies options for rebuilding the primary index of the full function database.                         |
| BLD_SECONDARY                       | Build Index function of FPA and IMS Index Builder | Yes                | Yes                | No                 | No                 | Specifies the options for rebuilding the secondary index.                                                 |
| COMP                                | IMS HP Image Copy                                 | Yes                | No                 | Yes                | No                 | Specifies whether the image copy is to be compressed.                                                     |
| COMPRTN                             | IMS HP Image Copy                                 | Yes                | No                 | Yes                | No                 | Specifies the name of the image copy compression routine.                                                 |
| DATACLAS                            | IMS HP Image Copy                                 | Yes                | No                 | Yes                | No                 | Specifies the SMS-managed data class that is used to allocate the image copy.                             |
| DSN                                 | IMS HP Image Copy                                 | Yes                | No                 | No                 | No                 | Specifies the template that is used to name the primary image copy data set.                              |
| DSN2                                | IMS HP Image Copy                                 | Yes                | No                 | No                 | No                 | Specifies the template that is used to name the secondary image copy data set.                            |
| DSNTYPE                             | IMS HP Image Copy                                 | Yes                | No                 | Yes                | No                 | Specifies whether the image copy can be allocated as Large Format data sets or as Basic Format data sets. |
| DUMPFORM                            | IMS HP Pointer Checker                            | Yes                | No                 | No                 | Yes                | Specifies the format of the internal control blocks that are printed.                                     |
| EXPDT                               | IMS HP Image Copy                                 | Yes                | No                 | Yes                | No                 | Specifies the expiration date for the output image copy data set.                                         |
| FABASNAP                            | IMS HP Pointer Checker                            | Yes                | No                 | No                 | Yes                | Specifies whether to generate the FABASNAP report when using DEDB Pointer Checker.                        |

| Utility control statement parameter | Integrated auxiliary utility                          | Valid on UTILGBL() | Valid on ADD(IB()) | Valid on ADD(IC()) | Valid on ADD(PC()) | Description                                                                                                      |
|-------------------------------------|-------------------------------------------------------|--------------------|--------------------|--------------------|--------------------|------------------------------------------------------------------------------------------------------------------|
| FSEMAP                              | IMS HP Pointer Checker                                | Yes                | No                 | No                 | Yes                | Specifies whether to generate the Free Space Map report for the HDAM, HIDAM, PHDAM, or PHIDAM database.          |
| FSNUM                               | Build Index function of FPA                           | Yes                | Yes                | No                 | No                 | Specifies the maximum number of address spaces for the Build Index function of FPA that are started in parallel. |
| FSPREF                              | Build Index function of FPA                           | Yes                | No                 | No                 | No                 | Specifies the prefix for creating the job name for the Build Index function of FPA.                              |
| HALDB                               | IMS Index Builder or DFSPRECO <sup>1</sup>            | Yes                | Yes                | No                 | No                 | Specifies the options for rebuilding HALDB ILDS and primary index.                                               |
| IBPREF                              | IMS Index Builder                                     | Yes                | No                 | No                 | No                 | Specifies the prefix that is used for creating the IMS Index Builder job name.                                   |
| ICBUF                               | IMS HP Image Copy                                     | Yes                | No                 | Yes                | No                 | Specifies the number of buffers that are used for referencing the image copy data set.                           |
| ICCAT                               | IMS HP Image Copy                                     | Yes                | No                 | Yes                | No                 | Specifies whether the output image copy is cataloged or not.                                                     |
| ICHLQ                               | IMS HP Image Copy                                     | Yes                | No                 | Yes                | No                 | Specifies the high-level qualifier for naming the primary image copy data set.                                   |
| ICHLQ2                              | IMS HP Image Copy                                     | Yes                | No                 | Yes                | No                 | Specifies the high-level qualifier for naming the secondary image copy data set.                                 |
| ICNDX                               | The Build Index function of FPA and IMS Index Builder | Yes                | Yes                | No                 | No                 | Specifies whether the secondary indexes are image copied.                                                        |
| ICNMRULE                            | IMS HP Image Copy                                     | Yes                | No                 | Yes                | No                 | Specifies the naming convention that is used for the output image copy data set.                                 |

| Utility control statement parameter | Integrated auxiliary utility | Valid on UTILGBL() | Valid on ADD(IB()) | Valid on ADD(IC()) | Valid on ADD(PC()) | Description                                                                                                              |
|-------------------------------------|------------------------------|--------------------|--------------------|--------------------|--------------------|--------------------------------------------------------------------------------------------------------------------------|
| INPUT                               | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies the type of input that is used to build indexes.                                                               |
| INTERVAL                            | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies the interval at which the Interval Statistics report and the Interval Free Space Summary report are produced.  |
| INTFS                               | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies whether to generate the Interval Free Space Summary report for the HDAM, HIDAM, PHDAM, or PHIDAM database.     |
| INTST                               | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies whether to generate the Interval Statistics report for the HDAM, HIDAM, PHDAM, or PHIDAM database.             |
| ITKBSRVR                            | Not Applicable               | Yes                | No                 | No                 | No                 | Specifies the name of the IMS Tools KB server XCF Group.                                                                 |
| KEYLABEL                            | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the name of the key label that is used to encrypt the image copy output data sets.                             |
| MAXFSD                              | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies whether to generate the Maximum Free Space Distribution report for the HDAM, HIDAM, PHDAM, or PHIDAM database. |
| MGMTCLAS                            | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the SMS-managed management class that is used to allocate the image copy.                                      |
| NDXIobuf                            | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies the number of buffers that VSAM uses when IMS Index Builder loads index data sets.                             |
| NOTIFY                              | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies if the image copy data set is registered to DBRC or not.                                                       |
| PCJOBNM                             | IMS HP Pointer Checker       | Yes                | No                 | No                 | No                 | Specifies the job name that is used for generating the IMS HP Pointer Checker job.                                       |

| Utility control statement parameter | Integrated auxiliary utility | Valid on UTILGBL() | Valid on ADD(IB()) | Valid on ADD(IC()) | Valid on ADD(PC()) | Description                                                                                                                                                                                  |
|-------------------------------------|------------------------------|--------------------|--------------------|--------------------|--------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PCPREF                              | IMS HP Pointer Checker       | Yes                | No                 | No                 | No                 | Specifies the prefix that is used for generating the IMS HP Pointer Checker job name.                                                                                                        |
| PCPROCNM                            | IMS HP Pointer Checker       | Yes                | No                 | No                 | No                 | Specifies the procedure name that is used to start the IMS HP Pointer Checker DMB Analyzer subordinate address space.                                                                        |
| PRINTDATA                           | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies the option for printing IMS HP Pointer Checker data.                                                                                                                               |
| PRPREF                              | DFSPREC0                     | Yes                | No                 | No                 | No                 | Specifies the prefix that is used for creating the DFSPREC0 job name.                                                                                                                        |
| RETPD                               | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the retention period for the output image copy data set.                                                                                                                           |
| RUNTM                               | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies whether to generate the separator page for DB/DSG reports with Runtime Option.                                                                                                     |
| SORTE35                             | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies whether IMS Index Builder can use the sort E35 exit to load sorted index records into the index data set.                                                                          |
| SORTFSZ                             | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies the estimated number of records to be sorted.                                                                                                                                      |
| SORTID                              | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies the ID of the SORT command that is used to start the address space for the sort program.                                                                                           |
| SORTOPT                             | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies the sort parameters to be appended to the SORT OPTION command.                                                                                                                     |
| SORTOUT                             | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies whether IMS Index Builder can copy sort messages to the data set that is defined in the IIUSOUT DD statement or leave them in the data set that is defined in the SORT DD command. |

| Utility control statement parameter | Integrated auxiliary utility | Valid on UTILGBL() | Valid on ADD(IB()) | Valid on ADD(IC()) | Valid on ADD(PC()) | Description                                                                                       |
|-------------------------------------|------------------------------|--------------------|--------------------|--------------------|--------------------|---------------------------------------------------------------------------------------------------|
| SORTSTAT                            | IMS Index Builder            | Yes                | Yes                | No                 | No                 | Specifies that sort count statistics are created for each index sorted.                           |
| SPACE                               | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the space parameters for allocating the image copy data set.                            |
| SSPCHECK                            | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies whether to perform subset pointer checking during the hash check process.               |
| STACK                               | IMS HP Image Copy            | Yes                | No                 | No                 | No                 | Specifies the option that is used for stacking image copies.                                      |
| STORCLAS                            | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the SMS-managed storage class that is used to allocate the image copy.                  |
| T2CHK                               | IMS HP Pointer Checker       | Yes                | No                 | No                 | Yes                | Specifies the T2 error processing option.                                                         |
| UNIT                                | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the unit that is used for allocating the primary image copy data set.                   |
| UNIT2                               | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the unit that is used for allocating the secondary image copy data set.                 |
| VIC                                 | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the virtual image copy processing option.                                               |
| VICDSN                              | IMS HP Image Copy            | Yes                | No                 | No                 | No                 | Specifies the data set name that is used for registering virtual image copies.                    |
| VOLCNT                              | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the maximum number of volumes for allocating the output image copy data set.            |
| VOLSER                              | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the volume serial numbers that are used for allocating the primary image copy data set. |

| Utility control statement parameter | Integrated auxiliary utility | Valid on UTILGBL() | Valid on ADD(IB()) | Valid on ADD(IC()) | Valid on ADD(PC()) | Description                                                                                                                                                                                                                              |
|-------------------------------------|------------------------------|--------------------|--------------------|--------------------|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VOLSER2                             | IMS HP Image Copy            | Yes                | No                 | Yes                | No                 | Specifies the volume serial numbers that are used for allocating the secondary image copy data set.                                                                                                                                      |
| WAITALOC                            | IMS HP Image Copy            | Yes                | No                 | No                 | No                 | Specifies whether to display the WAIT option in message IEF238D when dynamic allocation on the tape unit fails.                                                                                                                          |
| WAITMSG                             | IMS HP Image Copy            | Yes                | No                 | No                 | No                 | Specifies whether to display the FABJ3929A WTO message on the operator console, when dynamic allocation is being done on the tape unit. This keyword is effective for the dynamic allocation (SVC99) reason codes 214, 220, 228, or 484. |
| WAITTIME                            | IMS HP Image Copy            | Yes                | No                 | No                 | No                 | Specifies the wait time for repeating the dynamic allocation on the tape unit when the dynamic allocation reason code (SVC99) is 214, 220, 228, or 484.                                                                                  |

**Notes:**

1. If using Index Builder V3.1 or later and HALDB without the free space option, the utility is IMS Index Builder, otherwise the utility is DFSPRECO.

## Reference: Utility control statements

The utility control statements control the running of the integrated auxiliary utilities. This topic lists the utility control statements grouped by their usage.

**Topics:**

- [“Global environmental control statement \(UTILGBL\)” on page 70](#)
- [“Integrated auxiliary utility parameters for IMS Index Builder” on page 70](#)
- [“Integrated auxiliary utility parameters for the Build Index function of FPA” on page 75](#)
- [“Integrated auxiliary utility parameters for IMS HP Image Copy” on page 78](#)
- [“Integrated auxiliary utility parameters for IMS HP Pointer Checker” on page 87](#)

## Global environmental control statement (UTILGBL)

Use the optional UTILGBL() global environmental control statement to set the default processing options for the integrated auxiliary utilities.

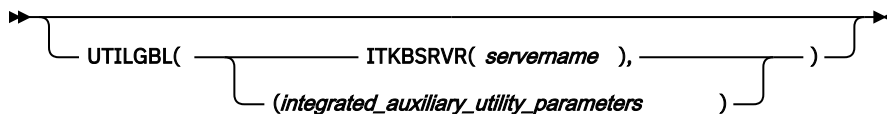
### Syntax diagram

The following syntax diagram for the UTILGBL global environmental control statement shows:

- The optional ITKBSRVR parameter that is specific to the UTILGBL control statement.
- The option to include specific integrated auxiliary utility parameters.

Any parameters specified at this global level function as default values, unless overridden by equivalent parameter specification in any of the ADD commands.

### Global environmental control statement



### Parameter reference

#### ITKBSRVR(servename)

*servename* specifies the name of the IMS Tools Knowledge Base server XCF group where the reports generated under IMS Database Recovery Facility in the IMS Tools Knowledge Base repository are stored.

The IMS Tools Knowledge Base server must be running.

The IMS Tools Knowledge Base client load library must be provided by concatenating in STEPLIB in both the IMS Database Recovery Facility master address space and applicable subordinate address space JCL procedures, including the RSS (recovery sort subordinate address space).

This parameter is optional. However, if not specified, no reports are written to the IMS Tools Knowledge Base.

The ITKBSRVR parameter is specific to the UTILGBL control statement and does not impact the integrated auxiliary utilities.

## Integrated auxiliary utility parameters for IMS Index Builder

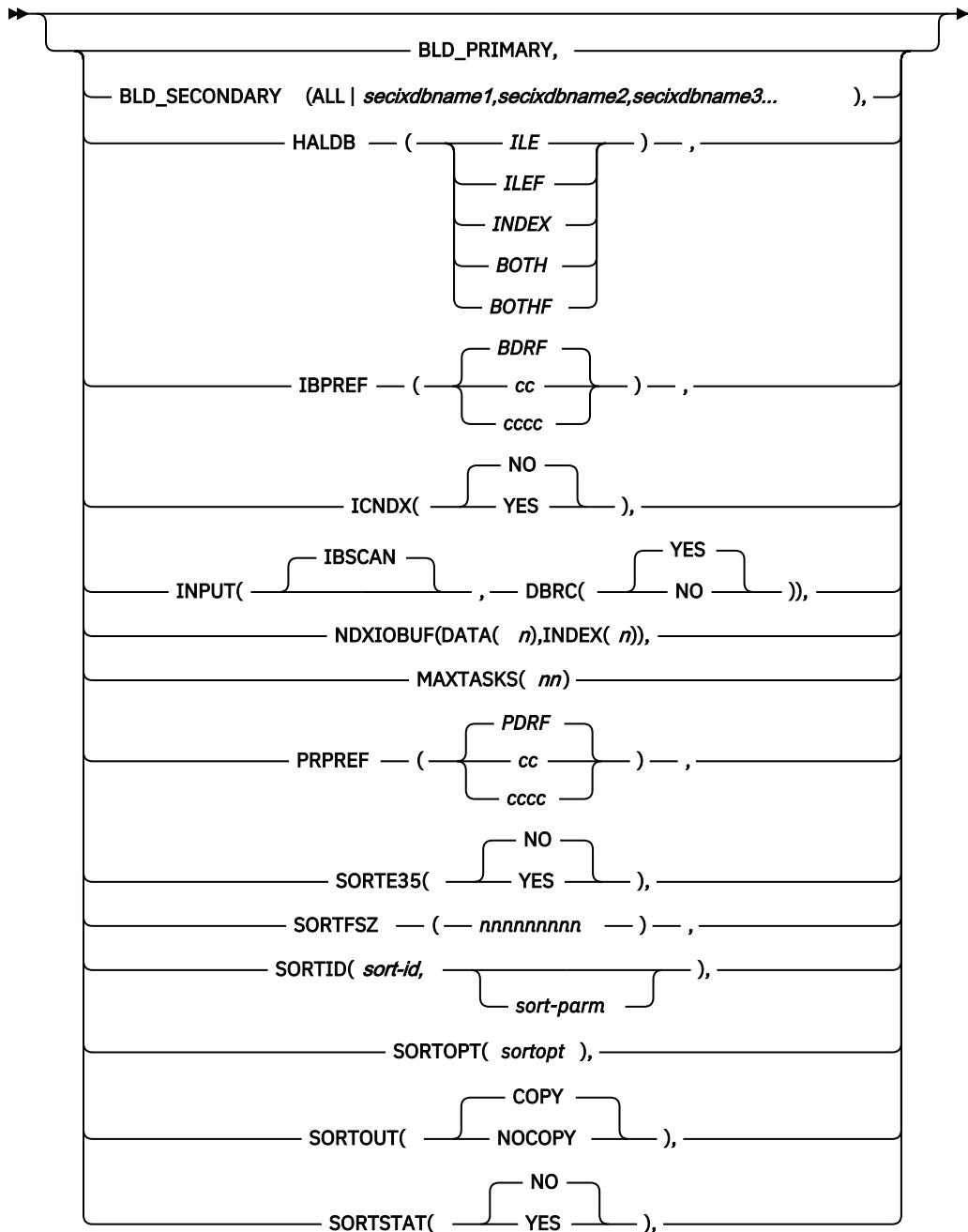
This topic describes the integrated auxiliary utility parameters for IMS Index Builder.

### Syntax diagram

The following syntax diagram shows the parameters which are associated with the IMS Index Builder auxiliary utility.



## Integrated auxiliary utility parameters for IMS Index Builder



### Parameter reference

#### BLD\_PRIMARY

Use this parameter to specify that IMS Database Recovery Facility should rebuild the primary index of the *dbname* specified on the ADD DB() command.

HALDB primary indexes are rebuilt using the DFSPREC0 utility. In IMS Database Recovery Facility, specifying the BLD\_PRIMARY command does not rebuild HALDB primary indexes. Use the parameters in the HALDB() statement to build HALDB primary indexes.

When the DFSPREC0 routine is needed for HALDB Primary Index and ILE processing, you must modify module DFSDAPLO.

**BLD\_SECONDARY(ALL | *secixdbname1,secixdbname2,secixdbname3...*)**

Use this parameter to rebuild the secondary index databases of the *dbname/areaname* specified on the ADD DB() command.

**Important:** When BLD\_SECONDARY(ALL) is specified, all secondary indexes are rebuilt. It is not necessary to specify BLD\_SECONDARY(*secixdbname1, secixdbname2, secixdbname3*).

**Note:** Only BLD\_SECONDARY(ALL) is supported when this parameter is specified on the UTILGBL() environmental statement.

**HALDB(ILE | ILEF | INDEX | BOTH | BOTHF)**

When the database is HALDB, use this parameter to specify whether to rebuild only the ILE data set, the Primary Index, or both. Specify this parameter only on the ADD statements for PHIDAM database types.

**ILE**

The indirect list entry (ILE) is to be rebuilt.

**ILEF**

This parameter, which is available with IMS Version 10 and later, rebuilds only the ILDS of the specified partition by using the free space option.

DFSPREC0 must have five z/OS data spaces available for processing if you select the free space option.

**INDEX**

The HALDB primary index is to be rebuilt.

**BOTH**

Specifies that both the specified ILE and the primary index are to be rebuilt.

**BOTHF**

This parameter, which is available with IMS Version 10 and later, rebuilds both the primary index and the ILDS of the specified partition.

Rebuilds the ILDS by using the free space option.

The utility must have five z/OS data spaces available for processing if you select the free space option.

**IBPREF(cc | *cccc* | **BDRF**)**

Use this parameter to set the prefix of the utility address space for the IMS Index Builder job name.

**cc**

A 2-character alphanumeric prefix used to construct the IB started task name.

When a 2-character prefix is specified, the job name is constructed by using the format *ccjjj###* where:

**cc**

The specified prefix.

**jjjj**

The JES job number associated with the master address space.

**##**

A number from 01 - 99, incremented with each new address space.

**cccc**

A 4-character alphanumeric prefix used to construct the IB started task name.

When a 4-character prefix is specified, the job name is constructed by using the format *cccc####*, where:

**cccc**

The specified prefix.

**####**

A number from 0001 - 0099, incremented with each new address space.

## **BDRF**

If IBPREF() is not specified, the job name is constructed by using the format BDRF####, where #### is a number from 0001 - 0099, incremented with each new address space.

## **ICNDX(NO | YES)**

This parameter specifies that indexes are to be image copied.

This parameter does not apply to HALDB primary index and ILE data sets because they are not recoverable.

This parameter also does not apply to secondary indexes that are not recoverable.

However, if you specify ICNDX(YES), secondary indexes for Full Function, Fast Path and HALDB databases are image copied. For Full Function databases, recoverable primary indexes are image copied.

NO is the default.

**Note:** If ICNDX(YES) is coded, it does not take effect unless the IC keyword is present on the ADD command.

## **INPUT(IBSCAN,DBRC(Y | N))**

The INPUT statement specifies the type of input used by IMS Index Builder to build indexes and specifies whether or not DBRC is used for both Index Builder and the Index Build function of FPA.

IBSCAN is the only option allowed in the first position.

IBSCAN does not apply to FPA and is ignored if specified. It is retained for consistency with IB control statements, but is also the default.

IBSCAN specifies that a scan of the physical data base is used as input. In IMS Database Recovery Facility, this option applies to secondary indexes for both HALDB and Full Function databases and to primary indexes for Full Function databases.

DBRC specifies whether DBRC is (DBRC(Y)) or is not (DBRC(N)) used. DBRC(Y) is the default.

## **MAXTASKS(nn)**

Use this parameter to specify that this is a MAXTASKS statement.

**nn**

Valid values are from 0 to 36, with leading zeros.

**Important:** MAXTASKS(2) has special meaning in that it instructs IMS Index Builder V3.1 to use the hierarchical HPU scan method for building indexes, such as in IMS Index Builder V2.3, instead of the sequential scan method. For more information, see the *IMS Index Builder User's Guide*.

## **NDXIOBUF(DATA(n),INDEX(n))**

Use this parameter to specify the number of buffers that VSAM uses when IMS Index Builder loads index data sets. Use this parameter when:

- IMS Index Builder loads index data sets without needing a sort.
- You specify the SORT E35 statement to use the E35 exit interface to sort.

## **PRPREF(cc | cccc | PDRF)**

Use this parameter to set the prefix of the utility address space for the DFSPREC0 job name.

**cc**

A 2-character alphanumeric prefix used to construct the DFSPREC0 started task name.

When a 2-character prefix is specified, the job name is constructed by using the format ccjjjj## where:

**cc**

The specified prefix.

**jjjj**

The JES job number associated with the master address space.

**##**

A number from 01 - 99, incremented with each new address space.

**cccc**

A 4-character alphanumeric prefix used to construct the IB started task name.

When a 4-character prefix is specified, the job name is constructed by using the format *cccc####*, where:

**cccc**

The specified prefix.

**####**

A number from 0001 - 0099, incremented with each new address space.

**PDRF**

If IBPREF() is not specified, the job name is constructed by using the format *PDRF####*, where *####* is a number from 0001 - 0099, incremented with each new address space.

**Note:** ALL is the only BLD\_SECONDARY() option that is allowed within the UTILGBL() statement.

**SORTE35(NO | YES)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces.

Use this parameter to specify whether IMS Index Builder can use the sort E35 exit to load sorted index records into the index data set.

When NO is specified, the sort program loads the sorted records directly into the index data set.

The default is NO unless you have coded IC() and ICNDX(YES) is intended. Then, the default is YES.

**SORTFSZ(nnnnnnnnn)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces.

Use this parameter to specify the estimated number of records to be sorted. *nnnnnnnn* is an integer from 1 to 999999999 that specifies the estimated number of records to be sorted.

The value is appended to the SORT option and is overridden if a non-zero index size is specified in the INDEX option for the DBD, or by the reused file size of the index.

If not specified, and not overridden, no FILSZ parameter is appended to the generated SORT option.

**SORTID(sort\_id,{sort\_parm,...})**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces.

Use this parameter to specify the ID of the SORT command used to start the address space for the sort program.

Only one *sort\_id* option is allowed. It must be a single alphanumeric character appended to the sort procedure name IIUBSRT. For example, if you specify the character A, IMS Index Builder uses the procedure IIUBSRTA to start address spaces used by the sort program.

The specified procedure must exist in one of the libraries in the SYS1.PROCLIB concatenation of the MVS primary subsystem. If a *sort\_id* is not specified, IMS Database Recovery Facility uses IIUBSRTT.

*sort\_parms* is a string of sort option parameters, separated by commas or blanks. The string is appended, unchanged, to the SORT OPTION command generated by IMS Index Builder.

The maximum length of the string is 34 characters, and the entire string must be on one line.

You can use this string to specify performance options or to override installation settings. For example:

```
MAINSIZE(100) FILSZ(E3000000)
```

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

### **SORTOPT(*sortopt...*)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces. Use this parameter to specify sort parameters to be appended to the SORT OPTION command.

Only one SORTOPT parameter is allowed. **Important:** The FILSZ specified in the SORTOPT parameter takes precedence.

*sortopt...* is a string of sort options, separated by commas or blanks. The string is appended, unchanged, to the SORT OPTION command generated by IMS Index Builder.

The maximum length of the string is 34 characters, and the entire string must be on one line.

You can use this string to specify performance options or to override installation settings. For example:

```
MAINSIZE(100) FILSZ(E30000000)
```

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

### **SORTOUT(COPY | NOCOPY)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate sort address spaces.

Use this parameter to specify whether IMS Index Builder can copy sort messages to the data set defined in the IIUSOUT DD statement or leave them in the data set defined in the SORT DD command, which is in the sort address space.

If you do not specify the SORTOUT parameter, COPY is assumed.

### **SORTSTAT(NO | YES)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate sort address spaces.

Use this parameter to tell IMS Index Builder to produce sort count statistics in the SYSPRINT data set for each index sorted.

If you specify this statement, the following counts are produced:

#### **SORT RECORD COUNT — E15**

The number of records passed to sort

#### **SORT BUFFER COUNT — E15**

The number of buffers processed

#### **SORT WAIT COUNT — E15**

The number of WAITs issued while waiting for a buffer

#### **SORT OUTPUT COUNT — E35**

The number of records sorted and loaded into the index data set

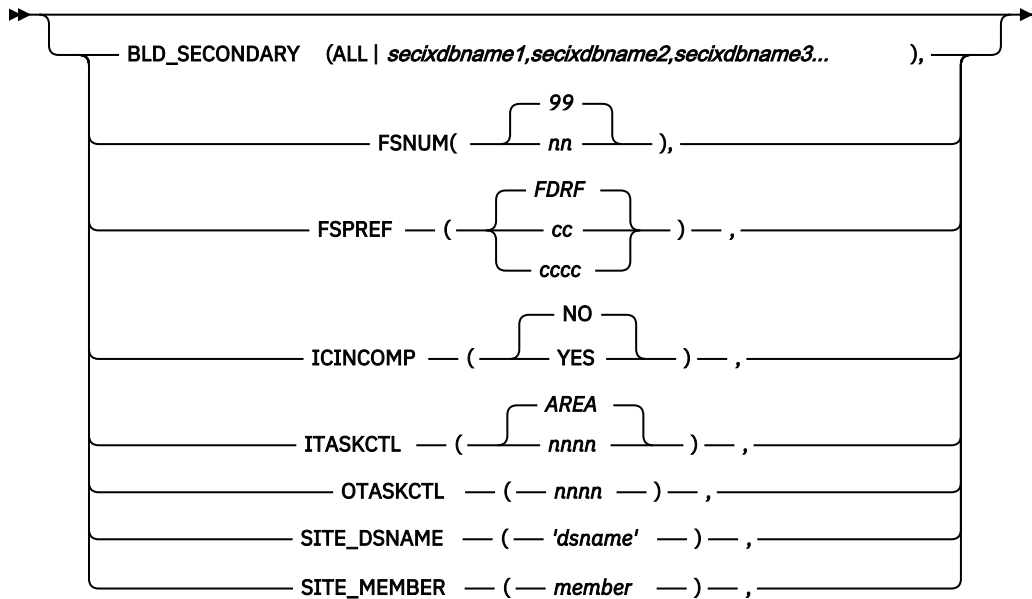
## **Integrated auxiliary utility parameters for the Build Index function of FPA**

This topic describes the integrated auxiliary utility parameters for the Build Index function of FPA.

### **Syntax diagram**

This syntax diagram shows the parameters which are associated with the Build Index function of FPA.

## Integrated auxiliary utility parameters for the Build Index function of FPA



### Parameter reference

#### BLD\_SECONDARY(ALL | *secixdbname1,secixdbname2,secixdbname3...*)

Use this parameter to rebuild the secondary index databases of the *dbname/areaname* specified on the ADD DB() command.

**Important:** When BLD\_SECONDARY(ALL) is specified, all secondary indexes are rebuilt. It is not necessary to specify BLD\_SECONDARY(*secixdbname1,secixdbname2,secixdbname3*).

**Note:** Only BLD\_SECONDARY(ALL) is supported when this parameter is specified on the UTILGBL() environmental statement.

#### FSNUM(*nn* | **99**)

A 1- to 2-digit integer value that you use to specify the maximum number of address spaces for the Build Index function of FPA that are started in parallel.

The actual number of FS-UAS address spaces depends on the work required, so the number of utility address spaces might be less than the value you specified.

If *nn* is sufficient, then one FS-UAS is started per DEDB for which index rebuild was requested.

#### FSPREF(*cc* | *cccc* | **FDRF**)

Use this parameter to set the prefix of the utility address space for the job name for the Build Index function of FPA.

##### **cc**

A 2-character alphanumeric prefix used to construct the FS started task name.

When a 2-character prefix is specified, the job name is constructed by using the format *ccjjj##* where:

##### **cc**

The specified prefix.

##### **jjj**

The JES job number associated with the master address space.

##### **##**

A number from 01 - 99, incremented with each new address space.

##### **cccc**

A 4-character alphanumeric prefix used to construct the FS started task name.

When a 4-character prefix is specified, the job name is constructed by using the format `cccc####`, where:

**cccc**

The specified prefix.

**####**

A number from 0001 - 0099, incremented with each new address space.

#### **FDRF**

If FSPREF() is not specified, the job name is constructed by using the format FDRF####, where #### is a number from 0001 - 0099, incremented with each new address space.

#### **ICINCOMP (No | Yes)**

Use this parameter to specify, when an image copy is used as an input data set, whether the image copy records are to be compressed by using the CSRCESRV MVS macro when they are stored into a data space storage.

IMS Fast Path Advanced Tool IMS Fast Path Advanced Tool stores all image copy records into a data space storage when an image copy is used as an input data set. Because the maximum size of one data space is 2 GB, if the total size of the records exceeds the limit, IMS Fast Path Advanced Tool ends with an error.

Try to reduce the total size of the records to under 2 GB by specifying ICINCOMP=YES. FPA compresses the image copy records by using the CSRCESRV macro before storing them into a data space and expands them when they are processed.

The data compression and expansion process increases the amount of CPU time.

##### **No**

The image copy records are stored into a data space storage without compression.

##### **Yes**

The image copy records are stored into a data space storage after compression.

#### **ITASKCTL(AREA | nnnn)**

Use this parameter to specify the maximum number of input areas to be processed concurrently.

##### **AREA**

The number of input areas.

##### **nnnn**

Specify a value from 1 to 2048. Specify a value less than or equal to the number of input areas.

If you specify a value that is greater than the number of input areas, it will be replaced by the number of input areas.

In case of the CHANGE command in the REORG mode, the following value is set: (ITASKCTL + OTASKCTL)/2 (the value is rounded off)

#### **OTASKCTL(nnnn)**

Use this parameter to specify the maximum number of output areas to be processed concurrently.

##### **nnnn**

Specify a value from 1 to 2048. Specify a value less than or equal to the number of output areas.

If you specify a value that is greater than the number of output areas, it will be replaced by the number of output areas.

In case of the CHANGE command in the REORG mode, the following value is set: (ITASKCTL + OTASKCTL)/2 (the value is rounded off)

#### **SITE\_DSNAME ('dsname')**

Use this parameter to specify the name of the data set that has the members of the site default tables.

***'dsname'***

Specify a data set name that is 1- to 44-characters and enclosed in single quotation marks.

The data set name identifies the data set containing the HFPCSITE global site default member.

**SITE\_MEMBER (*member*)**

Use this parameter to specify the name of the data set that has the members of the site default tables.

***'dsname'***

Use the SITE\_MEMBER parameter to specify the member name of the site default table.

## **Integrated auxiliary utility parameters for IMS HP Image Copy**

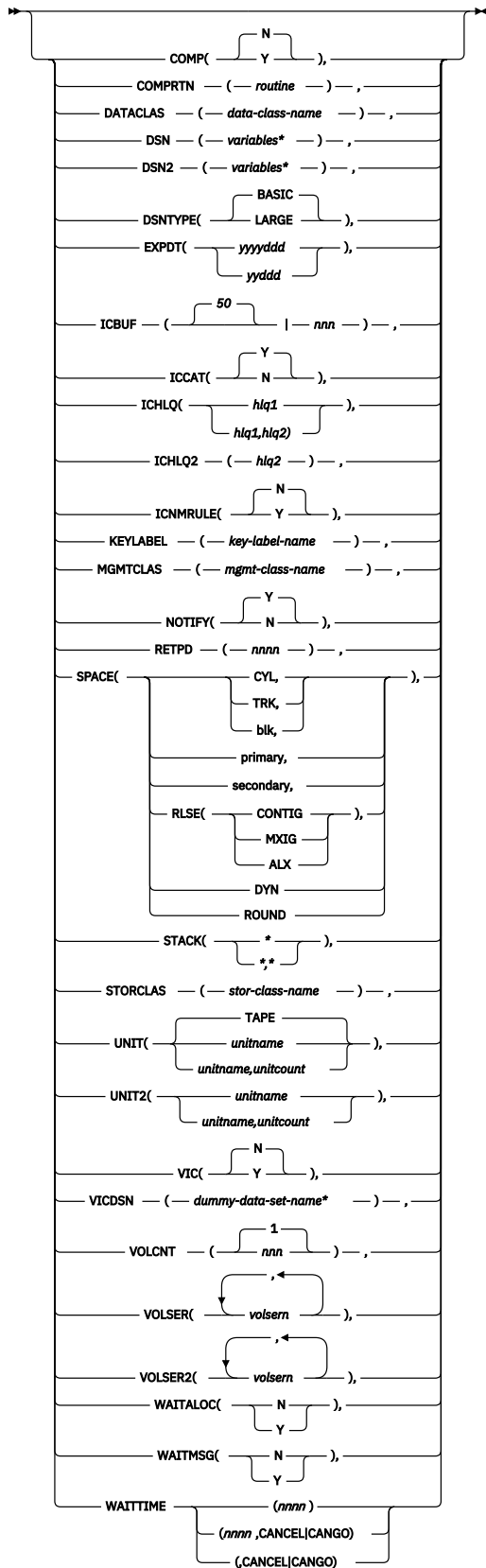
This topic describes the integrated auxiliary utility parameters for IMS High Performance Image Copy.

### **Syntax diagram**

This syntax diagram shows the parameters which are associated with the IMS High Performance Image Copy auxiliary utility.



## Integrated auxiliary utility parameters for IMS High Performance Image Copy



## Parameter reference

### COMP(Y | N)

This parameter specifies whether the Image Copy function is to compress the output image copy data.

#### **Y**

Specifies that the output image copy data is to be compressed by the Image Copy function.

#### **N**

Specifies that the output image copy data is not to be compressed by the Image Copy function.

The value N is the default.

### COMPRTN(*routine*)

This parameter specifies the name of the compression routine that the Image Copy function invokes to compress the output image copy data.

**Note:** When the IDRC feature is installed and is active on the native tape drive, software compression is suppressed for the IC1 and IC2 output data sets when the compression is not FABJCMP2.

The setting for *routine* can have the following meanings:

#### ***routine***

The name of the compression routine. IMS High Performance Image Copy provides four compression exit routines: FABJCMP1, FABJCMP2, FABJCMP3, and FABJCMP4.

Specify one of these compression routines to be called by the Image Copy function.

If the COMP=Y keyword is specified without the COMPRTN= keyword, FABJCMP1 is used as the default.

The same compression routine is automatically called by the Database Recovery function to extract the compressed data.



**Attention:** If you want to create a compressed image copy data set for a DEDB that has an SDEP segment, you must use FABJCMP4. FABJCMP1, FABJCMP2, and FABJCMP3 cannot be used for a DEDB that has an SDEP segment.

IMS High Performance Image Copy provides the following compression routines:

#### **FABJCMP1**

Does repeated character compression.

Free space can also be compressed.

#### **FABJCMP2**

Does only free space compression.

Segmented data is not compressed.

#### **FABJCMP3**

Does repeating-characters compression of the z/OS program compression method.

Free space might also be compressed.

#### **FABJCMP4**

Does repeating-characters compression of the z/OS program compression method.

This routine runs the block compression, which does not distinguish between the data portion, free space, and unused space in the database.

### DATACLAS(*data-class-name*)

This parameter specifies the name of the data class for the new SMS managed image copy output data sets that are dynamically allocated.

#### ***data-class-name***

The name of the data class to be used for allocating the data set.

The name that you define is one-to eight-characters.

This parameter does not have a default.

### **DSN(variables\*)**

Use this parameter to specify the template for the primary image copy data set that is dynamically allocated by IMS High Performance Image Copy where the following is true:

#### **variables\***

Template for the data set name.

The data set name that is generated by this specification must comprise no more than 44 characters in total.

For information on how to specify the values, see the *IMS High Performance Image Copy User's Guide*.

### **DSN2(variables\*)**

Use this parameter to specify the template for the secondary image copy data set that is dynamically allocated by IMS High Performance Image Copy where the following is true:

#### **variables\***

Template for the data set name.

The data set name that is generated by this specification must comprise no more than 44 characters in total.

For information on how to specify the values, see the *IMS High Performance Image Copy User's Guide*.

### **DSNTYPE(BASIC | LARGE)**

This parameter specifies whether the Image Copy can be allocated as Large Format data sets or Basic Format data sets.

More detail information for Large Format data sets is shown in the z/OS V1R7.0 DFSMS Using Data Sets.

#### **BASIC**

Specifies that the Image Copy function allocates Image Copy data sets as Basic Format data sets. BASIC is the default.

#### **LARGE**

Specifies that the Image Copy function allocates Image Copy data sets as Large Format data sets.

### **EXPDT(yyyyddd | yyddd)**

This parameter specifies the expiration date of the output image copy data set.

#### **yyyyddd or yyddd**

The expiration date of the output image copy.

The date format must comply with the DFSMS format (the Julian calendar format).

This parameter does not have a default.

### **ICBUF(nnn)**

This parameter specifies the number of buffers to be used to access the output image copy data set.

The value specified for ICBUF is used for BUFNO of QSAM DCB.

#### **nnn**

The left-aligned number of buffers that are used to access the output image copy data set.

Applicable values can range from 0 – 255.

The default value is 50.

### **ICCAT( Y | N)**

This parameter specifies whether the Image Copy function is to catalog the output image copy data set.

#### **Y**

Catalogs the output image copy data set.

This is the default.

## N

Does not catalog the output image copy data set.

### ICHLQ(*hlq* | *hlq1* , *hlq2*)

This parameter specifies the data set name prefix of the image copy data set.

#### *hlq*

The data set name prefix of the output image copy data set for the Image Copy function or it symbolizes a value in &ICHLQ.

#### *hlq1*

The data set name prefix of the primary output image copy data set for the Image Copy function or it symbolizes a value in &ICHLQ.

#### *hlq2*

The data set name prefix of the secondary output image copy data set for the Image Copy function or it symbolizes a value in &ICHLQ2.

**Note:** The *hlq2* parameter and ICHLQ2= cannot be specified together.

When you use the default ICOUT naming (ICNMRULE=N), specify a data set name prefix containing 33 or fewer characters. This prefix is used as the high-level qualifier of the data set name.

If you use the default STACK naming, specify a data set name prefix containing 7 or fewer characters. MVS naming convention requires that a generated data set name contain 44 or fewer characters.

This prefix must conform to the normal data set naming standards; it can include periods but must not end with a period.

If this parameter is left blank, a data set name prefix is not assigned.

### ICHLQ2(*hlq2*)

This parameter specifies the data set name prefix of the image copy data set.

You can use a data set name prefix when you allocate an output image copy data set dynamically.

#### *hlq2*

The data set name prefix of the secondary output image copy data set for the Image Copy function or sets a symbolic value for &ICHLQ2.

ICHLQ2= parameter cannot be specified with ICHLQ=(*hlq1*,*hlq2*).

When you use the default ICOUT naming (ICNMRULE=N), specify a data set name prefix containing 33 or fewer characters.

This prefix is used as the high-level qualifier of the data set name.

If you use the default STACK naming, specify a data set name prefix containing 7 or fewer characters. MVS naming convention requires that a generated data set name contain 44 or fewer characters.

This prefix must conform to the normal data set naming standards; it can include periods but must not end with a period.

If this parameter is left blank, a data set name prefix is not assigned.

### ICNMRULE(Y | N)

This parameter is used to select a naming convention for the output image copy data set.

#### Y

Selects the naming convention of the output image copy data set as follows:

```
ichlq.ICn.dbdname.ddname.Dyyddd.Thmmss
```

#### N

Selects the naming convention of the image copy data set as follows:

```
ichlq.ICn.dbdname.ddname
```

Within these naming conventions, the following meanings apply:

***ichlq***

Is specified with the ICHLQ= keyword.

***n***

Is 1 or 2, indicating the primary or secondary copy.

***dbdname***

Is the DBD name of the database to be copied.

***ddname***

Is the DD name of the target data set group or area to be copied.

***yyddd***

Is the date.

***hhmmss***

Is the time stamp.

N is the default value.

**KEYLABEL(*key-label-name*)**

This parameter specifies the name of the key label used to encrypt the image copy output data sets.

***key-label-name***

The name of the key label to be used for allocating the encrypted image copy output data set.

The name that you define, is 1- to 64-characters long.

This parameter does not have a default.

**MGMTCLAS(*mgmt-class-name*)**

This parameter specifies the name of the management class for the new SMS managed image copy output data sets that are dynamically allocated.

***mgmt-class-name***

The name of the management class to be used for allocating the data set.

The name that you define, is one-to eight-characters long.

This parameter does not have a default.

**NOTIFY(*Y* | *N*)**

This parameter specifies whether the Image Copy function registers the image copy with DBRC.

***N***

Does not register the image copy with DBRC.

***Y***

Registers the image copy with DBRC.

Y is the default value.

**RETPD(*nnnn*)**

This parameter specifies the retention period, in days, for the output image copy data set.

***nnnn***

The retention period for the output image copy.

You can specify a value from 0 - 9999.

This parameter does not have a default value.

**SPACE({*CYL* | *TRK* | *blk*},*primary*[,*secondary*][,*RLSE*][,*CONTIG* | ,*MXIG* | ,*ALX*][,*DYN*][,*ROUND*])**

This parameter specifies the space specification of the output image copy data set for dynamic allocation.

***CYL* | *TRK* | *blk***

Specifies how the Image Copy function is to allocate the output image copy data set.

***CYL***

The allocation in cylinders.

**TRK**

The allocation in tracks.

**blk**

The allocation in blocks. For example: SPACE(1024,10,10).

**primary**

The number of tracks, cylinders, or blocks to be allocated.

**secondary**

The additional number of tracks or cylinders to be allocated, if more are needed.

If this option is specified, it must be located in the third position.

**RLSE**

Requests that space allocated to an output data set but not used, is to be released when the data set is closed.

**CONTIG**

Requests that space allocated to the data set must be contiguous.

**MXIG**

Requests that space allocated to the data set must be the largest area of available contiguous space on the volume and equal to, or greater than, the primary quantity.

**ALX**

Requests that space allocated to the data set is the largest 5 areas of available contiguous space on the volume.

**DYN**

Using this option causes IMS Database Recovery Facility to determine the space attributes for the output image copy from the space attributes of the input image copy being used for the recovery.

The dynamic space calculation is determined from the number of records in the input image copy:

- If the input image copy space attributes cannot be determined dynamically, the values specified on the SPACE parameter are used.
- If only SPACE(DYN) is specified and the space attributes cannot be determined, it is treated as if the SPACE parameter were omitted.

This option is only valid when the input image copy is on DASD. Otherwise dynamic space calculation is bypassed.

When DYN is specified, allocation is in units of TRK or CYL, RLSE is always appended and ROUND is disregarded.

If the input image copy has attribute CONTIG, MXIG, or ALX and propagation of this attribute to the output image copy is not required, then code a null argument at the end of the SPACE parameter. For example:

```
SPACE (CYL,10,1,DYN,)
```

**ROUND**

When the first option specifies the average block length, this option requests that space allocated to the data set must be equal to an integral number of cylinders.

More detail information for each of these values is shown in the z/OS JCL reference manual.

This parameter does not have a default.

- Output Device under SMS:

The SPACE parameter in the SMS definition is used as the default.

If you specify the SPACE keyword in the ICEIN control statement or the site default table, the SMS definition is overwritten by the specified SPACE parameters.

- Non-SMS Output Device:

The SPACE parameter is a required.

If you omit this keyword, dynamic allocation fails.

If you defined SPACE in the site default table, IMS High Performance Image Copy uses the SPACE parameter in the site default.

### **STACK(\* | \*,\*)**

This parameter places two or more output image copy data sets on the same tape volume or set of tape volumes.

If you specify STACK(\*), IMS High Performance Image Copy dynamically allocates the output image copy data set and requests a single copy of the stacked image copy output. Specify STACK(\*,\*) when you need dual copies.

For more information about stacking, see the *IMS High Performance Image Copy User's Guide*.

**Restriction:** STACK is supported only on the UTILGBL() statement. It is not a recognized option on the ADD command with the IC() parameter. Invoking STACK causes IMS Database Recovery Facility to process each DBDS in the recovery list serially within one RSS.

If the tape volser is required, you can specify the VOLSER option on the UTILGBL() statement to indicate which tape volumes to use. VOLSER is not allowed on the ADD command with the IC() parameter in this situation.

The following example includes a UTILGBL() statement:

```
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=DRFIC1)
UTILGBL(COMP=N,DSN(&ICHLQ..&DBD..&DDN.),STACK(*),UNIT(3390),-
        ICBUF(15),VOLSER(222222,333333))
ADD DB(DIVNTZ02) -
        IC(ICHLQ(DRFIC1),SPACE(CYL,1,1),COMPRTN(FABJCOMP1))-
ADD DB(DIVNTZ02,DXVNTZ02) -
        IC(ICHLQ(DRFIC3),SPACE(CYL,1,1),COMPRTN(FABJCOMP3))-
START ERROR(CONT)
```

### **STORCLAS(stor-class-name)**

This parameter specifies the name of the storage class for the new SMS managed image copy output data sets that are dynamically allocated.

#### **stor-class-name**

The name of a storage class to be used for allocating the data set.

The name that you define, is one-to eight-characters long.

An ACS routine can override the storage class that you specify in the STORCLAS keyword.

This parameter does not have a default.

### **UNIT(TAPE | unitname | unitname,unitcount)**

This parameter specifies the UNIT of the output image copy data set for allocation.

**Note:** When an image copy is requested, the UNIT parameter must be present either on the IC keyword or on the UTILGBL control statement; otherwise, an error results and no IC is created.

#### **TAPE | unitname | unitname,unitcount**

The option for allocating the output image copy.

For example, UNIT(TAPE) or UNIT(TAPE,2).

The default value is TAPE.

### **UNIT2(unitname | unitname,unitcount)**

This parameter specifies the UNIT of the secondary output image copy data set, which is used for allocation.

#### **unitname or unitname,unitcount**

The option for allocating the secondary output image copy.

For example, UNIT2(TAPE) or UNIT2(TAPE,2).

This parameter does not have a default.

If both UNIT2 and STORCLAS are specified, the value specified in the STORCLAS parameter is not applied to the secondary output image copy data set.

#### **VIC(Y | N)**

This parameter specifies whether the Image Copy function registers the dummy data set name that is specified on the VICDSN parameter with DBRC as a virtual image copy.

**Y**

Registers the image copy with DBRC.

**N**

Does not register the image copy with DBRC. N is the default.

#### **VICDSN(dummy-data-set-name\*)**

This parameter specifies a dummy data set name that is registered with DBRC as the data set name for NOTIRY.UIC command of virtual image copy process.

The dummy data set name must be not exceed 44 characters.

#### **VOLCNT(1 | nnn)**

This parameter specifies the number of volumes used for the output image copy data set.

**nnn**

The maximum number of volumes that an output image copy data set requires.

The volume count is a decimal number from 1 through 255 for a tape data set and from 1 through 59 for a DASD data set.

If you omit VOLCNT, 1 is used as the default.

#### **VOLSER(volsern | volsern, volsern...)**

This parameter specifies the volume serial number of the volume used for the primary data set of the output image copy.

**volser or (volser1, volser2,...)**

The volume serial number of the volume allocated to the primary data set of the output image copy.

This parameter does not have a default.

#### **VOLSER2(volsern | volsern, volsern...)**

This parameter specifies the volume serial number of the volume used for the secondary data set of the output image copy.

**volser or (volser1, volser2,...)**

The volume serial number of the volume allocated to the secondary data set of the output image copy.

This parameter does not have a default.

#### **WAITALOC(N | Y)**

The WAITALOC keyword specifies whether to display the WAIT option in message IEF238D when dynamic allocation on the tape unit fails.

**N**

Does not display the WAIT option in the message IEF238D.

**Y**

Displays the WAIT option in message IEF238D. If WAIT is replied to the message IEF238D, IMS High Performance Image Copy waits to do dynamic allocation until the required units are released.

**Important:** IMS Recovery Solution Pack does not set any defaults for any of the IMS High Performance Image Copy WAIT\* keywords. This behavior ensures that any values contained in the IMS High Performance Image Copy global site default table are not changed.



**WAITMSG(N | Y)**

This parameter specifies whether to display the FABJ3929A WTO message on the operator console, when dynamic allocation is being done on the tape unit.

This parameter is effective for the dynamic allocation (SVC99) reason codes 214, 220, 228, or 484.

**N**

Does not display the FABJ3929A WTO message on the operator console.

**Y**

Displays the FABJ3929A WTO message on the operator console until the dynamic allocation on the tape unit is successful.

**Important:** IMS Recovery Solution Pack does not set any defaults for any of the IMS High Performance Image Copy WAIT\* keywords. This behavior ensures that any values contained in the IMS High Performance Image Copy global site default table are not changed.

**WAITTIME(nnnn) | (nnnn, CANCEL | CANGO) | (,CANCEL | CANGO)**

This parameter specifies the wait time for repeating the dynamic allocation on the tape unit when the dynamic allocation reason code (SVC99) is 214, 220, 228, or 484.

**nnnn**

The amount of time allotted, in minutes, for repeating the dynamic allocation for the tape unit when the reason code is 214, 220, or 228. The minimum value is 0 and the maximum value is 9999. When 0 is specified, IMS High Performance Image Copy does not retry dynamic allocation and terminates with message U3916.

**CANCEL**

Specifies that IMS High Performance Image Copy does not retry dynamic allocation when the reason code is 484.

**CANGO**

Specifies that IMS High Performance Image Copy retries dynamic allocation when the reason code is 484.

**Important:** IMS Recovery Solution Pack does not set any defaults for any of the IMS High Performance Image Copy WAIT\* parameters. This behavior ensures that any values contained in the IMS High Performance Image Copy global site default table are not changed.

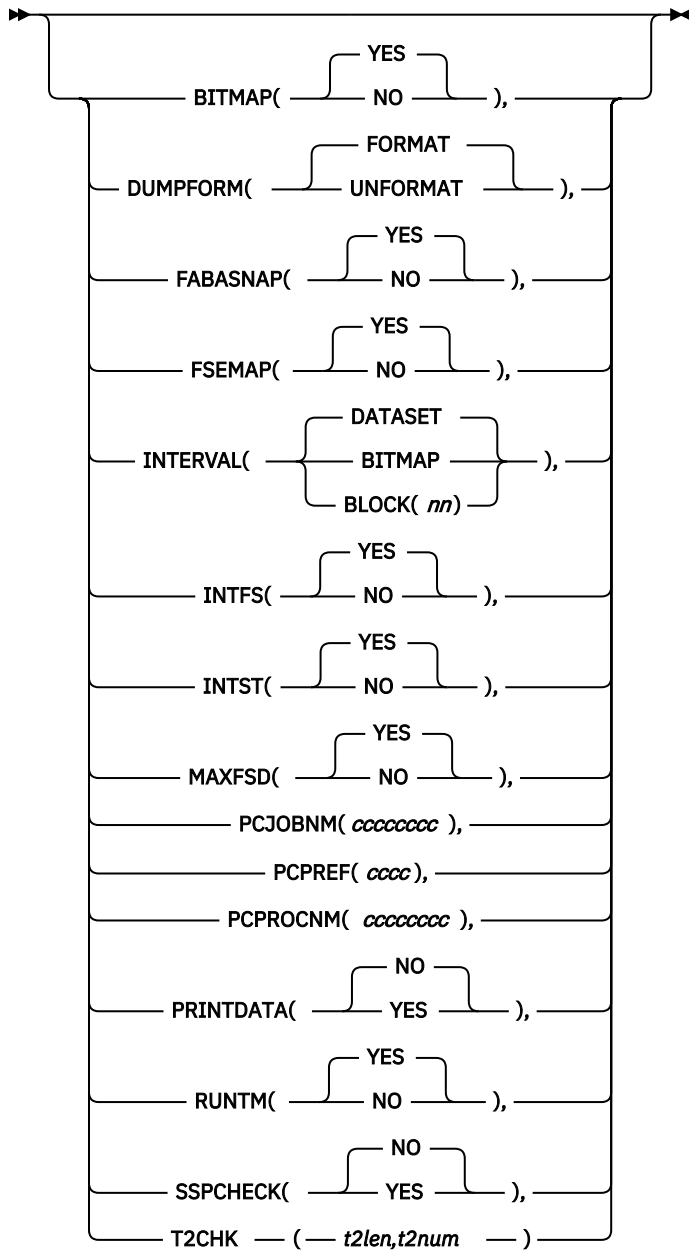
**Integrated auxiliary utility parameters for IMS HP Pointer Checker**

This topic describes the integrated auxiliary utility parameters for IMS High Performance Pointer Checker.

**Syntax diagram**

This syntax diagram shows the parameters which are associated with the IMS High Performance Pointer Checker auxiliary utility.

## Integrated auxiliary utility parameters for IMS High Performance Pointer Checker



### Parameter reference

#### **BITMAP(YES | NO)**

Specifies whether you want to generate the **Bit Map Display** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

This option can be specified when TYPE=ALL or SCAN is also specified.

#### **YES**

The report is generated.

This is the default value.

#### **NO**

The report is not generated.

#### **DUMPFORM(UNFORMAT | FORMAT)**

Specifies the dump format that you want to print block dumps with.

This option can be specified when DIAGDUMP=FIRST100 in the OPTION is also specified.

This option can be specified with any TYPE= specification.

**FORMAT**

Specifies that the formatted dumps are to be printed.

This is the default value.

**UNFORMAT**

Specifies that the unformatted dumps are to be printed.

**FABASNAP(YES | NO)**

Indicates whether you request the **FABASNAP** report when using DEDB Pointer Checker.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**FSEMAP(YES | NO)**

Specifies whether you want to generate the **Free Space Map** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

This option can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**INTERVAL(DATASET | BITMAP | BLOCK(nn))**

Specifies whether you want to define the interval at which the **Interval Statistics** report and the **Interval Free Space Summary** report are produced.

This option can be specified when TYPE=ALL or SCAN is specified on the PROC statement, and when INTFS=YES or INTST=YES is specified on the REPORT statement.

It is effective only for HDAM, HIDAM, PHDAM, or PHIDAM databases.

**DATASET**

The reports are produced for the entire database data set only once.

This is the default value.

**BITMAP**

The reports are produced each time a bitmap block is processed.

**BLOCK(nn)**

The number represented by *nn* times 100, is the number of blocks that is processed between statistics intervals.

To code this field, you must include two integer digits. Use leading zeros, if necessary.

**INTFS(YES | NO)**

Specifies whether you want to generate the **Interval Free Space Summary** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

The report is produced each time an interval is processed and the information in the report is added to the next report. That is, the *N*th report provides the total information of the 1st through *N*th reports.

This option can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**INTST(YES | NO)**

Specifies whether you want to generate the **Interval Statistics** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

The report is produced each time an interval is processed; the information in the report is added to the next report. That is, the *N*th report provides the total information of the 1st through *N*th reports.

This option can be specified when TYPE=ALL or SCAN is specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**MAXFSD(YES | NO)**

Specifies whether you want to generate the **Maximum Free Space Distribution** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

This option can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**PCJOBNM(ccccccc)**

Use this optional parameter to specify a job name for the IMS HP Pointer Checker DMB Analyzer subordinate address space.

PCJOBNM() is mutually exclusive with PCPREF().

When specified, IMS HP Pointer Checker starts the address space with either the FABPATHZ procedure or, if specified, the PCPROCNM(ccccccc) procedure, and assign this to the job.

If not specified, IMS HP Pointer Checker starts the address space with the standard FABPATH0 procedure and FABPATH0 is the name of the job.

**PCPREF(cccc)**

Use this optional parameter to specify the prefix of the pointer checker started task name.

PCPREF() is mutually exclusive with PCJOBNM().

**cccc**

A 4-character alphanumeric prefix used to specify the pointer checker job name.

When a 4-character prefix is specified, the job name is constructed by using the format cccc0001.

**PCPROCNM(ccccccc)**

Use this optional parameter to specify the procedure name to be used by IMS HP Pointer Checker to start the IMS HP Pointer Checker DMB Analyzer subordinate address space.

When specified, the procedure cccccc is used. If PCJOBNM is specified in conjunctions with PCPROCNM, then PCJOBNM(ccccccc) is used as the JOBNAME for the address space.

If this parameter is not specified, then the procedure used depends upon the PCJOBNM specification and either FABPATH0 or FABPATHZ is used.

**PRINTDATA(NO | YES)**

Specifies whether you want to print the pointer data that is extracted by the program.

If you specify YES, you might get a large report that is of little use.

This option can be specified when TYPE=ALL or SCAN is also specified.

This option must be used for debugging purpose only.

**NO**

Any extracted pointer data is not printed.

This is the default.

**YES**

The extracted pointer data is printed.

**RUNTM(YES | NO)**

Specifies whether you want to generate the separator page for DB/DSG reports with Run Time Option.

This option can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**SSPCHECK(YES | NO)**

Specifies whether you want to perform subset pointer checking during the hash check process.

**YES**

Subset pointer checking is performed.

**NO**

Subset pointer checking is not performed.

This is the default value.

**T2CHK(*t2len*,*t2num*)**

Use the T2CHK parameter to ignore the short and known T2 errors that are not really errors in HISAM, HDAM, HIDAM, PHDAM, and PHIDAM databases.

This parameter requires the following values:

***t2len***

The minimum T2 record length to be reported (for HDAM, HIDAM, PHDAM, and PHIDAM).

The T2 records whose lengths are shorter than the specified minimum value are not to be reported.

This setting allows you to ignore short T2s that might not be actual errors.

***t2num***

The maximum number of T2 records not to be reported (for HISAM, HDAM, HIDAM, PHDAM, and PHIDAM).

By specifying a T2 record threshold value for this specification, the known T2 records are not reported.

This setting allows you to ignore known T2s that might not be actual errors, but are reported until the database is reorganized.



---

## Chapter 6. Configuring JCL statements and procedures

To use IMS Database Recovery Facility, you must configure appropriate JCL and PROCLIB members for your environment.

The master address space is the main address space that drives IMS Database Recovery Facility and related processes. The recovery sort subordinate address space is used to sort log records and perform recovery processes.

The majority of the DD statements that are required for running in the subordinate address spaces are dynamically allocated and passed from the master address space to the recovery sort subordinate address space, the IMS Index Builder and DFSPREC0 address space, and the address space for the Build Index function of FPA.

### Topics:

- [“Master address space JCL” on page 93](#)
- [“Recovery sort subordinate address space JCL” on page 98](#)
- [“IMS Index Builder, DFSPREC0, and Build Index function of FPA address space JCL” on page 100](#)
- [“IMS High Performance Pointer Checker JCL” on page 101](#)

---

### Master address space JCL

The master address space is the main address space that drives IMS Database Recovery Facility and related processes.

Sample JCL is located in SFRXSAMP(FRXDRF/FRXDRFC) and SFRXSAMP(FRXIVPD1/D2) sample members. These JCL members run the SFRXSAMP(FRXMAS) procedure.

If you are using a consolidated IMS Tools load library, use the sample JCL located in SFRXSAMP(FRXDRFC) and FRXIVPD2.

**Important:** The IMS Database Recovery Facility master address space JCL cannot contain any hard-coded database data sets. Because IMS Database Recovery Facility dynamically allocates the database data sets for recovery in the subordinate address space, dynamic allocation will fail if any of the subsequent steps have these same data sets allocated by DD statements in the MAS. This is a z/OS restriction.

### Topics:

- [“DDNAME reference for master address space” on page 93](#)
- [“Report DDNAME statement reference” on page 96](#)

### DDNAME reference for master address space

The following master address space DDNAME statements are located in SFRXSAMP(FRXMAS|FRXMASC).

With the exception of the PSBLIB and SYSUDUMP DD statements, these DD statements are passed to the recovery sort subordinate address space, the address space for the Build Index function of FPA, as well as the address spaces for IMS Index Builder and DFSPREC0.

The DD statements are dynamically allocated if they do not already exist.

### STEPLIB

This required DD statement identifies the executable IMS Recovery Solution Pack load library (SFRXLOAD) and must not identify any other libraries.

All other required execution load libraries are passed from the master address space and dynamically allocated by FRXSDR00 to the TASKLIB execution load library.

**Important:** If the IMSDALIB DD statement is not coded and you are using dynamic allocation for the RECON and database data sets, you must also include your IMSDALIB data set.

## **IMS**

This statement is used by IMS Database Recovery Facility and the integrated auxiliary utilities to specify the IMS DBD library.

The IMS Index Builder load library, SIIULMOD, must also be included in the IMS DD statement concatenation to provide the IMS Index Builder with reusable program specification blocks (PSB) for processing.

## **IMSACB**

This DDNAME is used by the Build Index function of FPA to identify the ACB library.

Dynamic allocation of IMSACB is not supported.

**Important:** If IMSACB is not coded, minimally IMSACBA and MODSTAT must be coded.

## **IMSACBA**

If IMSACB is not coded, this DDNAME can be used in combination with MODSTAT and MODSTAT2 by the Build Index function of FPA to identify the ACB library.

However, if IMSACB is coded, it is used to identify the ACB library.

**Important:** If IMSACB is not coded, minimally IMSACBA and MODSTAT must be coded.

## **IMSACBB**

If IMSACB is not coded, this DDNAME can be used in combination with MODSTAT and MODSTAT2 by the Build Index function of FPA to identify the ACB library.

However, if IMSACB is coded, it is used to identify the ACB library.

**Important:** If IMSACB is not coded, minimally IMSACBA and MODSTAT must be coded.

## **MODSTAT**

If IMSACB is not coded, this DDNAME can be used in combination with IMSACBA and IMSACBB by the Build Index function of FPA to identify the ACB library.

However, if IMSACB is coded, it is used to identify the ACB library.

**Important:** If IMSACB is not coded, minimally IMSACBA and MODSTAT must be coded.

## **MODSTAT2**

If IMSACB is not coded, this DDNAME can be used in combination with IMSACBA and IMSACBB by the Build Index function of FPA to identify the ACB library.

However, if IMSACB is coded, it is used to identify the ACB library.

**Important:** If IMSACB is not coded, minimally IMSACBA and MODSTAT must be coded.

## **PSBLIB**

This statement identifies the library that contains the PSB members that are associated with the databases being processed.

This DD statement is required when performing a PITR recovery so that IMS Database Recovery Facility can report on any units of work that are active.

If this DD statement is not specified, message FRD6137W may be issued.

## **PROCLIB**

This statement identifies the set of libraries that IMS Database Recovery Facility uses to get the BPE configuration member (BPECFG=xxxxxxx) and the IMS Database Recovery Facility startup parameter member FRXDRFxx, (DRFMBR=xx).

**Important:** The PROCLIB statement is used only for configuration members. The procedures for the recovery sort subordinate, Build Index function of FPA, IMS Index Builder/DFSRECO, and IMS High Performance Pointer Checker address spaces must be located in a z/OS system procedure library.



**DFSRESLB**

This statement identifies the IMS RESLIB load library. Because the IMS Database Recovery Facility STEPLIB must be APF-authorized, having a DFSRESLB DD statement is optional if the IMS RESLIB is in the STEPLIB.

If specified, the DFSRESLB DD statement provides compatibility with functions for loading authorized IMS modules, using the IMS batch interface instead of STEPLIB/JOBLIB.

**IMSRESLB**

This DDNAME is for the Build Index function of FPA and contains, in part, the IMS load library.

IMSRESLB differs from DFSRESLIB in the fact that IMSRESLB is used to indicate that the IMS RESLIB also contains routines such as randomizer routines, segment compression/edit routines, index maintenance exits and user partition selection exits.

**IMSDALIB**

This statement identifies the libraries that contain dynamic allocation information for the database data sets and the RECON data sets.

**Important:** This statement is optional if the dynamic allocation data set is included in the STEPLIB DD concatenation.

**SORTLIB**

This statement identifies the location of the sort utility executable modules.

This DD statement is optional if your SORTLIB is located in the LINKLIST.

**DFSVSAMP**

This statement is used by IMS Index Builder and DFSPRECO to allocate the buffers that are needed to process VSAM and OSAM database data sets.

If this statement is not specified, one will be dynamically created for processing.

**RECONx**

These statements identify the RECON data sets.

The RECONx DDNAME statements are optional if they can be dynamically allocated by using the dynamic allocation members that are found in the STEPLIB or IMSDALIB DD statements.

**HFPCSITE**

This DDNAME identifies the global site default table data set for the Build Index function of FPA.

You can set the site default name by following any one of these three methods:

- Both keywords SITE\_DSNAME and SITE\_MEMBER are specified on the UTILGBL control statement.
- SITE\_MEMBER is specified in on the UTILGBL control statement and the HFPCSITE DD PDS is allocated and contains module HFPCSITE. HFPCSITE is mutually exclusive with SITE\_DSNAME.
- Neither SITE\_DSNAME nor SITE\_MEMBER is specified, but the HFPCSITE module exists in the load library for the Build Index function of FPA (HFPMAIN0).

**DDEFDPS**

This DDNAME is used by the Build Index function of FPA for dynamic delete or define of the secondary indexes that are being rebuilt.

**SYSUDUMP**

This statement identifies the data set definition where dump output is sent after an abnormal termination.

**Recommendation:** For debugging and diagnostic purposes, IBM support prefers an unformatted dump.

- If SYSUDUMP is coded, then a formatted dump is created if an abend occurs.
- If SYSUDUMP is not coded and an abend occurs, then a system, unformatted dump is generated.

## Report DDNAME statement reference

Report DDNAME statements refer to the data sets that contain messages and formatted results from various processes, including recovery and its auxiliary utilities.

The following report DDNAMES are dynamically allocated by IMS Database Recovery Facility to SYSOUT unless you explicitly override them in the procedure or JCL:

### FRXWTO

This statement contains the write-to-operator messages that the control task captures when RPTTYPE=SEP or RPTTYPE=SAS is specified.

If RPTTYPE=APP is specified, it is written to the REPORT data set. It is dynamically allocated if not provided in the JCL.

### FRXPRINT

This statement is used to distinguish between the IMS Database Recovery Facility and integrated auxiliary utility messages that are written to the SYSPRINT DD statement.

### REPORT

This statement identifies the location for the following items:

- Commands, control statements and parameters that are specified for the IMS Database Recovery Facility execution
- Final status reports for all address spaces
- Utility status reports per database data set
- Input and output (for example, image copies, change accumulations, and logs)

If the RPTTYPE=APP option is specified in the master address space, the REPORT DD statement contains the messages for the following DDNAME statements.

If RPTTYPE=SEP or RPTTYPE=SAS is specified, the reports for the following DDNAME statements are written to each DDNAME statement instead, whether in the MAS Job output, or the SAS Job output, respectively.

#### DDNAME statements:

- FABAMSG
- FABARPRT
- FABASNAP
- FRXWTO
- HPFPRINT
- HFPRPTS
- IIUPRINT
- IIUSNAP
- IIUSOUT
- IIUSTAT
- PRPRINT
- SNAPPIT
- STATIPRT
- VALIDPRT

### SYSPRINT

This statement contains control card usage information and progress messages from the master and subordinate address spaces.

## IMS Index Builder report DDNAME statements

The following report DDNAME statements are associated with the IMS Index Builder auxiliary utility.

If IMS Index Builder is invoked and either RPTTYPE=SEP or RPTTYPE=SAS is specified, these DDNAME statements contain the specified information, whether in the MAS job output, or in the SAS job output, respectively.

Otherwise, if RPTTYPE=APP is specified, messages are written to the REPORT DD.

#### **IIUSNAP**

This statement identifies where Index Builder writes control flow event messages.

#### **IIUSOUT**

This statement identifies where Index Builder writes SORT messages.

#### **IIUSTAT**

This statement identifies where Index Builder writes DBRC notification messages for indexes that are rebuilt.

#### **IIUPRINT**

This statement identifies where the main output from Index Builder is written.

### **Build Index function of FPA report DDNAME statements**

The following report DDNAME statements are associated with the Build Index function of FPA.

#### **HFPRPTS**

This statement identifies where the main output from FPA is written, which is the Audit Report.

#### **HFPPRINT**

This statement identifies where processing messages are written.

### **IMS High Performance Image Copy report DDNAME statements**

The following report DDNAME statements are associated with the IMS High Performance Image Copy auxiliary utility.

If IMS HP Image Copy is invoked and either RPTTYPE=SEP or RPTTYPE=SAS is specified, these DDNAME statements contain the specified information, whether in the MAS job output, or in the SAS job output, respectively.

Otherwise, if RPTTYPE=APP is specified, messages are written to the REPORT DD.

#### **ICEPRINT**

This statement identifies where IMS HP Image Copy writes the ICEIN statements report and the active global option report.

#### **ICERPRT**

This statement identifies where IMS HP Image Copy writes the image copy dump process report.

#### **DFSPRINT**

This statement identifies where the IMS HP Image Copy writes control card usage and image copy statistics and return codes for each image copy.

#### **ICE#DOUT**

This statement identifies where IMS HP Image Copy writes DBRC notification messages for each image copy.

### **DFSPREC0 report DDNAME statements**

The following report DDNAME statements are associated with the DFSPREC0 auxiliary utility.

If DFSPREC0 is invoked and either RPTTYPE=SEP or RPTTYPE=SAS is specified, these DDNAME statements contain the specified information, whether in the MAS job output, or in the SAS job output, respectively.

Otherwise, if RPTTYPE=APP is specified, messages are written to the REPORT DD.

#### **PRPRINT**

This statement identifies where DFSPREC0 writes the number of records that are inserted for each HALDB INDEX or ILE that is rebuilt.

### IMS High Performance Pointer Checker report DDNAME statements

The following report DDNAME statements are associated with the IMS High Performance Pointer Checker auxiliary utility.

If IMS High Performance Pointer Checker is invoked and either RPTTYPE=SEP or RPTTYPE=SAS is specified, these DDNAME statements contain the specified information, whether in the MAS job output, or in the SAS job output, respectively.

Otherwise, if RPTTYPE=APP is specified, messages are written to the REPORT DD.

#### **PRIMAPRT**

This statement identifies where HP Pointer Checker writes the DMB directory report and the PROCCTL statements report.

**Tip:** You can specify DD DUMMY to suppress this report and get only a PC return code.

#### **EVALUPRT**

This statement identifies where HP Pointer Checker writes the evaluation reports.

#### **SUMMARY**

This statement identifies where HP Pointer Checker writes the summary report.

#### **STATIPRT**

This statement identifies where HP Pointer Checker writes statistical reports.

#### **VALIDPRT**

This statement identifies where HP Pointer Checker writes validation reports.

#### **SNAPPIT**

This statement identifies where HP Pointer Checker writes snap reports.

### IMS DEDB Pointer Checker report DDNAME statements

The following report DDNAME statements are associated with the DEDB Pointer Checker auxiliary utility.

If DEDB Pointer Checker is started and either RPTTYPE=SEP or RPTTYPE=SAS is specified, these DDNAME statements contain the specified information, whether in the MAS job output, or in the SAS job output, respectively.

Otherwise, if RPTTYPE=APP is specified, messages are written to the REPORT DD.

#### **FABARPRT**

This statement identifies where DEDB Pointer Checker writes evaluation reports.

#### **FABAMSG**

This statement identifies where DEDB Pointer Checker writes processing and summary messages.

#### **FABASNAP**

This statement identifies where DEDB Pointer Checker writes snap reports.

**Tip:** You can specify DD DUMMY to suppress this report and get only a PC return code.

## Recovery sort subordinate address space JCL

---

The recovery sort subordinate address space is started by the master address space and performs the following tasks:

- Sorts log and change accumulation records
- Reads the input image copy
- Recovers the database
- Creates the new image copy
- Invokes the DEDB Pointer Checker

The procedure that is used to create this address space is specified by using the DRFPROC= environmental control statement. The default procedure name is FRXJCLSB. Samples are located in the

SFRXSAMP(FRXRSS|FRXJCLSB) members. The FRXJCLSB procedure is used to start the recovery sort subordinate address space and must be placed in a valid z/OS system procedure library.

**Tip:** If you have multiple maintenance levels of IMS Tools, you must customize separate FRXRSSxx members, where xx is the maintenance level used. For example, create FRXRSST1 for the IMS Tools load library that is used in your test environments and FRXRSSP1 for the IMS Tools load library that is used in your production environments.

Multiple recovery sort subordinate address spaces can be started. The names assigned to these address spaces are specified by using the ASPREF() keyword. The maximum number allowed is specified by using the SORTPARM(NUM(x)) keyword. For more information, see [“Environmental control statements” on page 40](#).

All DD statements required for running in the subordinate address spaces, except STEPLIB DD and the SYSUDUMP DD, are passed from the master address space to this address space and dynamically allocated.

The required STEPLIB DD must be specified in the subordinate address space JCL and it must contain a single data set that points to the IMS Recovery Solution Pack executable load library, SFRXLOAD.

## **DDNAME statement reference**

### **STEPLIB**

This required DD statement identifies the IMS Database Recovery Facility load library.

The STEPLIB DD statement must include a single load library that names the executable IMS Recovery Solution Pack load library (SFRXLOAD).

All other libraries that are required for execution are passed from the master address space and dynamically allocated for execution.

### **BSYOUT**

This DD statement contains the backup profile information, control cards, and IMS Recovery Expert messages.

### **SYSUDUMP**

This statement identifies the data set definition where dump output is sent after an abnormal termination.

**Recommendation:** For debugging and diagnostic purposes, IBM support prefers an unformatted dump.

- If SYSUDUMP is coded, then a formatted dump is created if an abend occurs.
- If SYSUDUMP is not coded and an abend occurs, then a system, unformatted dump is generated.

### **FRXDEBUG**

This optional DD statement identifies the data set definition where certain debugging and diagnostic messages are written.

The following DD statements can be either coded directly in the procedure JCL, or specified in the master address space JCL and dynamically allocated in the recovery sort subordinate address space (recommended).

Dynamic allocation eliminates the need to manually update subordinate address space JCL procedures when the master address space JCL is updated .

- IMS
- PROCLIB
- DFSRESLB
- IMSDALIB
- SORTLIB
- RECON1
- RECON2

- RECON3
- DFSVSAMP
- DBDLIB
- STEPLIB

## IMS Index Builder, DFSPREC0, and Build Index function of FPA address space JCL

---

The IMS Index Builder address space is started when IMS Index Builder is requested to rebuild indexes. The DFSPREC0 address space is started when DFSPREC0 is requested to rebuild HALDB primary indexes and ILDS by using either HALDB(BOTHF) or HALDB(ILEF). The Build Index function of FPA address space (FS-UAS) is started when the Build Index function of FPA is requested to rebuild secondary indexes for a Fast Path database.

All DD statements required for running in the subordinate address spaces, except STEPLIB DD and the SYSUDUMP DD, are passed from the master address space to this address space and dynamically allocated.

The required STEPLIB DD must be specified in the subordinate address space JCL and it must contain a single data set that points to the IMS Recovery Solution Pack executable load library, SFRXLOAD.

The address spaces for IMS Index Builder, DFSPREC0, IMS High Performance Image Copy, and the Build Index function of FPA are started by the master address space. The procedure that is used to create these address spaces is specified by using the DRFIAX() environmental control statement. The default procedure name is FRXJCLIP. Samples are located in the SFRXSAMP(FRXJCLIP|FRXIAX) member.

**Tip:** Customize the FRXRSS procedure and rename to FRXRSSxx, where xx is the maintenance level or IMS subsystem used (for example, TEST or PROD and IMS V14 or IMS V15). For example:

- If you have different IMS subsystems, you can create FRXIAX14 for your IMS V14 system and FRXIAX15 for your IMS V15 system.
- If you have different environments with different maintenance levels, you can create FRXRSST1 for IMS SSID IMT1 of your test environment and FRXRSP1 for your IMS SSID IMP1 of your production environment.

The name of the address space for the Build Index function of FPA is specified by using the FSPREF() keyword. The name of the IMS Index Builder address space is specified by using the IBPREF() keyword. The name of the DFSPREC0 address space is specified by using the PRPREF() keyword. For more information, see [“Environmental control statements” on page 40](#).

Multiple address spaces for IMS Index Builder, DFSPREC0, and the Build Index function of FPA can be started. One address space is started for each index that is being rebuilt.

### DDNAME statement reference

#### STEPLIB

This required DD statement identifies the IMS Database Recovery Facility load library.

The STEPLIB DD statement must include a single load library that names the executable IMS Recovery Solution Pack load library (SFRXLOAD).

All other libraries that are required for execution are passed from the master address space and dynamically allocated for execution.

#### SYSUDUMP

This statement identifies the data set definition where dump output is sent after an abnormal termination.

**Recommendation:** For debugging and diagnostic purposes, IBM support prefers an unformatted dump.

- If SYSUDUMP is coded, then a formatted dump is created if an abend occurs.

- If SYSUDUMP is not coded and an abend occurs, then a system, unformatted dump is generated.

The following DD statements can be either directly coded in the procedure JCL, or specified in the master address space JCL and dynamically allocated in the recovery sort subordinate address space (recommended).

Dynamic allocation eliminates the need to manually update subordinate address space JCL procedures when the master address space JCL is updated.

- IMS
- PROCLIB
- DFSRESLB
- IMSDALIB
- SORTLIB
- DFSVSAMP
- RECON

## IMS High Performance Pointer Checker JCL

---

The IMS High Performance Pointer Checker address space is started by the master address space when IMS HP Pointer Checker is requested for a full-function database.

The procedure that is used to create this address space is specified by using the PCPROCNM() parameter. The default procedure name is FABPATH0. A sample is located in the SFRXSAMP(FRXPAT0) member.

The name of this address space is specified by using either the PCPREF() or the PCJOBNM() keyword. If either PCPREF() or PCJOBNM() is specified, then the procedure name that is used to build this address space is FABPATHZ. A sample is located in the SFRXSAMP(FRXPATZ) member.

A maximum of one IMS HP Pointer Checker address space is started.

### DDNAME statement reference

The following IMS Database Recovery Facility pointer checker address space DDNAME statements are located in the SFRXSAMP(FABPATH0/FABPATHZ) member:

#### STEPLIB

This statement contains the executable libraries that are required by the IMS HP Pointer Checker utility. When this address space is started, the IMS HP Pointer Checker and IMS RESLIB load libraries are passed to the procedure.

In addition to these load libraries, if you process a HALDB and it uses a partition selection exit, the exit must be either in your IMS RESLIB or specified in the library in STEPLIB.

#### SYSUDUMP

This statement identifies the data set definition where dump output is sent after an abnormal termination.

**Recommendation:** For debugging and diagnostic purposes, IBM support prefers an unformatted dump.

- If SYSUDUMP is coded, then a formatted dump is created if an abend occurs.
- If SYSUDUMP is not coded and an abend occurs, then a system, unformatted dump is generated.





---

## Part 3. Using IMS Database Recovery Facility

The topics in this section show how to use IMS Database Recovery Facility and how to perform a database recovery.

**Topics:**

- [Chapter 7, “Using batch recovery control statements,” on page 105](#)
- [Chapter 8, “Using the integrated auxiliary utilities,” on page 123](#)
- [Chapter 9, “Reference: DBRC commands for database recovery,” on page 147](#)
- [Chapter 10, “Performing a database recovery,” on page 149](#)



---

## Chapter 7. Using batch recovery control statements

This information describes the batch recovery control statements that identify the databases used for recovery, and how these control statements are used to start the recovery process.

There are three types of control statements:

- Environmental control statements.
- Batch recovery control statements.
- Utility control statements.

This section describes the use of the batch recovery control statements.

### Topics:

- [“Batch recovery control statement commands” on page 105](#)
- [“Reference: ADD command” on page 105](#)
- [“Reference: REMOVE command” on page 111](#)
- [“Reference: START command” on page 113](#)
- [“Reference: Timestamp format” on page 119](#)

---

### Batch recovery control statement commands

You can control the recovery performed by IMS Database Recovery Facility by supplying batch recovery control statements. IMS Database Recovery Facility operates on a list of database data sets and areas that you define with a series of batch recovery control statements.

By comparison, the standard IMS recovery utility, DFSURDB0, can recover only one database data set per invocation.

The following three commands are used to specify batch recovery control statements.

#### **ADD**

Use this command to add one or more database data sets to the recovery list.

You can specify an unlimited number of ADD statements that can be specified in a single job step.

#### **REMOVE**

Use this command to remove one or more database data sets from the recovery list.

This command is useful when a database data set was added as part of a group but does not need to be recovered.

You can specify an unlimited number of REMOVE statements in a single job step.

#### **START**

Use this command to specify recovery options and to initiate the recovery of all database data sets in the recovery list.

Only a single START command can be entered for a single job step.

The batch recovery control statements must be entered in the //SYSIN DD input.

They are not allowed in the FRXDRFxx PROCLIB member.

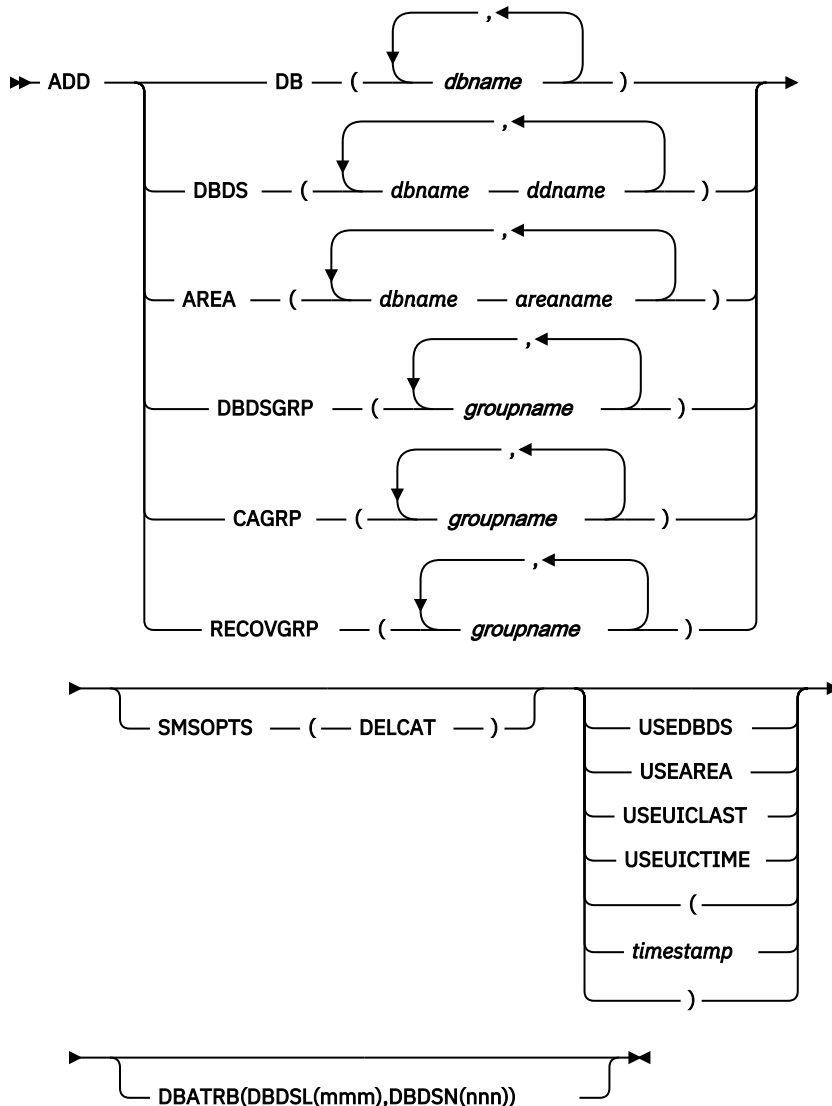
---

### Reference: ADD command

The ADD command is used to add one or more database data sets to the recovery list.

For details on invoking the integrated auxiliary utilities, see [Chapter 8, “Using the integrated auxiliary utilities,” on page 123](#).

## Syntax for ADD command



## Parameters for ADD command

There is no practical limit to the number of **ADD** commands that you can specify except for that which is limited by your environment.

The following parameters are available for your use with the **ADD** command:

### **DB(dbname)**

Specifies that all areas or full-function database data sets or database data sets from a full-function HALDB partition or HALDB master database are to be added to a recovery list.

#### **dbname**

Specifies the database and the associated database data sets or areas that are to be added to a recovery list.

### **DBDS(dbname ddname)**

Specifies that one or more full-function database data sets are to be added to a recovery list.

Specify full-function database data sets with the **ADD DBDS** command as an ordered pair. If you specify more than one full-function database data set, you must specify the complete ordered pair for each database data set. If you specified **DBDS** on the **ADD** command, you must specify *dbname* and *ddname* together, separated by a blank space.

***dbname***

Specifies the database name of the database data set to be added to a recovery list.

***ddname***

Specifies the DD name of the database data set.

**AREA(*dbname areaname*)**

Specifies that one or more Fast Path database areas are to be added to a recovery list. If you specified AREA on the **ADD** command, you must specify *dbname* and *areaname* together (separated by a blank space).

***dbname***

Specifies the database name of the area to be added to a recovery list.

***areaname***

Specifies the area to be added to a recovery list.

**DBDSGRP(*groupname*)**

Specifies that one or more DBDS groups as defined in the RECON data sets will have their database data sets and areas added to a recovery list.

***groupname***

Specifies that the database data sets and areas belonging to the named DBDS group are to be added to the recovery list.

**CAGRP(*groupname*)**

Specifies that one or more change accumulation groups as defined in the RECON data sets are to have their database data sets and areas added to a recovery list.

***groupname***

Specifies that the database data sets and areas belonging to the named CA group are to be added to the recovery list.

**RECOVGRP(*groupname*)**

Specifies that the listed groups are recovery groups. A recovery group is a group of full-function databases, DEDB areas, or both that you define to IMS as related. All DBDSs that make up the full-function databases and all the DEDB areas that make up the recovery groups that you specified in the command are added to a recovery list.

***groupname***

Specifies the name of the group.

**SMSOPTS(*optionname*)**

Specifies the DFSMSdss options that are to be associated with the entries being added to the recovery list. The options are only used when restoring image copies created by the Image Copy 2 utility.

***optionname***

Specifies a unique SMS option.

**DELCAT**

Specifies that the data set is to be restored using the DFSMSdss optional keyword, DELETECATALOGENTRY

**Important:** Use this option with extreme care. DELCAT is required if SMSOPTS is supplied. This option provides you the ability to recover from a scenario in which an entire volume or volumes are lost but the catalog entries remain. When you specify this option, SMS deletes the prior catalog entries for the database data sets and areas being restored as part of recovery. See the *z/OS DFSMSdss Storage Administration Reference* manual before using this option. Read the caution under the DELETECATALOGENTRY option of the **RESTORE** command.

**USED BDS**

Indicates that you restored the image copy for the database and IMS Database Recovery Facility needs only to apply log updates.

IMS Database Recovery Facility does not differentiate between the USED BDS and USE AREA parameters. These parameters exist to be compatible with the DBRC **GENJCL . RECOV** command. Both

parameters have the same effect because IMS Database Recovery Facility does not check whether the database is a full function database or a Fast Path area.

IMS Database Recovery Facility does not verify whether a nonstandard image copy exists in the RECON.

**Restrictions:**

- USEDDBDS and USEAREA work only with batch-type, nonstandard image copies. Concurrent, nonstandard image copies are not supported. If you must use concurrent, nonstandard image copies, see USEUICLAST and USEUICTIME.
- You must restore nonstandard image copies before recovery.

For information about how to recover with the USEDDBDS and USEAREA parameters, see [“Recovering with a nonstandard image copy”](#) on page 160.

**USEAREA**

Indicates that you restored the image copy for the database and IMS Database Recovery Facility needs only to apply log updates.

IMS Database Recovery Facility does not differentiate between the USEDDBDS and USEAREA parameters. These parameters exist to be compatible with the DBRC **GENJCL . RECOV** command. Both parameters have the same effect because IMS Database Recovery Facility does not check whether the database is a full function database or a Fast Path area.

IMS Database Recovery Facility does not verify whether a nonstandard image copy exists in the RECON.

**Restrictions:**

- USEDDBDS and USEAREA work only with batch-type, nonstandard image copies. Concurrent, nonstandard image copies are not supported. If you must use concurrent, nonstandard image copies, see USEUICLAST and USEUICTIME.
- You must restore nonstandard image copies before recovery.

For information about how to recover with the USEDDBDS and USEAREA parameters, see [“Recovering with a nonstandard image copy”](#) on page 160.

**USEUICLAST**

Indicates that you restored the image copy for the database and IMS Database Recovery Facility needs only to apply log updates.

IMS Database Recovery Facility applies logs to the database or area from the latest user image copy that is recorded in the RECON. You must restore the database or area to the last user image copy time stamp before you recover.

IMS Database Recovery Facility verifies that a user image copy was recorded in the RECON.

**Restrictions:**

- The nonstandard image copy must be registered as a user image copy in the RECON. You can register the image copy by using the DBRC **NOTIFY . UIC** command. The user image copy can be batch or concurrent.
- You must restore nonstandard image copies before recovery.

For information about how to recover with the USEUICLAST parameter, see [“Recovering with a nonstandard image copy”](#) on page 160.

**USEUICTIME(timestamp)**

Indicates that you restored the image copy for the database and IMS Database Recovery Facility needs only to apply log updates.

IMS Database Recovery Facility applies logs to the database data set or area data set from the time stamp of the user image copy that is recorded in the RECON. You must restore the database data set or area data set to the time stamp of the user image copy before you recover.

The time stamp that is specified for the USEUICTIME parameter must be a valid user image copy time stamp in the RECON for the database data set. If the user image copy time stamps differ for each database data set, you must specify a separate **ADD** statement for each data set and specify the applicable user image copy time stamp.

IMS Database Recovery Facility verifies that a user image copy is recorded for that time stamp in the RECONS for each database data set.

For information about how to recover with the USEUICTIME parameter, see [“Recovering with a nonstandard image copy”](#) on page 160.

#### ***timestamp***

Timestamp format reference: [“Reference: Timestamp format”](#) on page 119

### **DBATRB**

The DATRB parameter allows you to specify the DBDSL*mmm* and DBDSN*nnn* data set characteristics to be used for allocating new database data sets. The parameters that follow DBATRB; DBDSL and DBDSN identify the DBDSL*mmm* and DBDSN*nnn* parameters in the environmental control statement parameters to use for the ADD command for recover to copy.

#### **DBDSL(*mmm*)**

*mmm* identifies a specific DBDSL*mmm* parameter in the environmental control statement parameters. The DBDSL*mmm* parameter it identifies will have a matching value for the *mmm*. For example, DBDSL234 is referred to by DBDSL(234) on the **ADD** command.

#### ***mmm***

You can specify the value of *mmm* as any alphanumeric three digit value that corresponds with the value specified on a DBDSL*mmm* environmental control statement parameter.

**Note:** If a database data set needs to be automatically deleted and redefined, the ADD command must include the keyword DBATRB(DBDSL(*mmm*)). The DBDSL(*mmm*) specification will reference the environmental control statement DBDSL*mmm* that should have DISP(NEW) as an option.

#### **DBDSN(*nnn*)**

*nnn* identifies a specific DBDSN*nnn* parameter in the environmental control statement parameters. The environmental control statement keyword it identifies will have a matching value for the *nnn*. For example, DBDSN234 is referred to by DBDSN(234) on the **ADD** command.

#### ***nnn***

You can specify the value of *nnn* as any alphanumeric three digit value that corresponds with the value specified on a DBDSN*nnn* environmental control statement parameter.

**Note:** If you specify OUTPUT(PRO), the DBATRB(DBDSN(*nnn*)) parameter is ignored. The name of the existing production database data set cannot be changed.

If you specify OUTPUT(DUP), you are required to specify the DBATRB(DBDSN(*nnn*)) option to identify the name of the duplicated database data set. Failure to specify this option will result in a parsing error.

If you specify OUTPUT(BOTH), the DBATRB(DBDSN(*nnn*)) parameter is ignored for the recovery to production but required for use with the duplicate database data set. Failure to specify this option will result in a parsing error.

## **ADD command examples**

Here are some of examples of using the ADD command with IMS Database Recovery Facility.

### **Delete and redefine example: two production databases prior to recovery**

The following example will delete and redefine two production databases prior to recovery:

```
DBDSL004(DISP(NEW))
ADD DB(DBOVLFPD DIVNTZ02) DBATRB(DBDSL(004))
START
```

### Delete and redefine example: two fast path areas to a different volume

The following example will delete and redefine two fast path areas to a different volume:

```
DBDSL004(DISP(NEW),VOL(SCR003))
OUTPUT(PRO)
ADD AREA(DEDBJN22 DB22AR0 DEDBJN22 DB22AR2) DBATRB(DBDSL(004))
START
```

### Single data set example: recover to copy

The following is an example of naming and locations of a single data set.

Original data set name: PROD.MYDATA.INV.NEW

Duplicate output data set that you want: HIQUAL.MYDATA.INV.NEW

The SYSIN DD specification you need:

```
SYSIN DD *
OUTPUT(DUP)
DBDSN002('HIQUAL')
...
DBDSL002(DISP(NEW),DATAC(myclass),VOL(myvol1))
...
ADD DBDS(MYDATA INV) DBATRB(DBDSL(002),DBDSN(002))
```

The output report will look like this:

| Database | Recovery | Facility              | Duplicate | Data Set | Report |
|----------|----------|-----------------------|-----------|----------|--------|
| DB       | DD       | Dsname                |           |          |        |
| MYDATA   | INV      | HIQUAL.MYDATA.INV.NEW |           |          |        |

### Multiple data set example: recover to copy

The following is an example of naming and locations of multiple data sets.

Original data sets names: PROD.MYDATA.INV.NEW PROD.MYDATA.SHIPMENT.NEW

Duplicate output data set that you want with both data sets having the same attributes:  
HIQUAL.MYDATA.INV.NEW HIQUAL.Mydata.Shipment.new

SYSIN DD specification:

```
SYSIN DD *
OUTPUT(DUP)
DBDSN003('HIQUAL')
...
DBDSL004(DISP(NEW),DATAC(myclass),VOL(myvol1))
...
ADD DBDS(MYDATA INV MYDATA SHIPMENT MYDATA ORDERS)
DBATRB(DBDSL(004),DBDSN(003))
```

### Change accumulation group example: recover to copy

The following is an example of naming and locations of multiple data sets within a change accumulation group.

Original data set names in change accumulation group mycagrp:

```
PROD.MYDATA.INV.NEW
PROD.MYDATA.SHIPMENT.NEW
PROD.MYDATA.ORDERS.NEW
PROD.MYDATA.ACCTRECV.NEW
```

Copy output data sets that you want:

```
HIQUAL.MYDATA.INV.NEW
HIQUAL.MYDATA.SHIPMENT.NEW
```



```
HIQUAL.MYDATA.ORDERS.NEW
HIQUAL.MYDATA.ACCTRECV.NEW
```

You want the data set HIQUAL.MYDATA.INV.NEW to reside on myvol1, the data set HIQUAL.MYDATA.SHIPMENT.NEW to reside on myvol2, and for HIQUAL.MYDATA.ORDERS.NEW and HIQUAL.MYDATA.ACCTRECV.NEW to reside on the default volume, myvol3.

The SYSIN DD specification you need is:

```
SYSIN DD *
OUTPUT(DUP)
DBDSN005('HIQUAL')
...
DBDSL006(DISP(NEW), DATAC(myclass) VOL(myvol3) -
DBN(MYDATA) DDN(INV) DATAC(myclass),VOL(myvol1), -
DBN(MYDATA) DDN(SHIPMENT) DATAC(myclass),VOL(myvol2))
...
ADD CAGRP(MYCAGRP) DBATRB(DBDSL(006),DBDSN(005))
```

The output report will look like this:

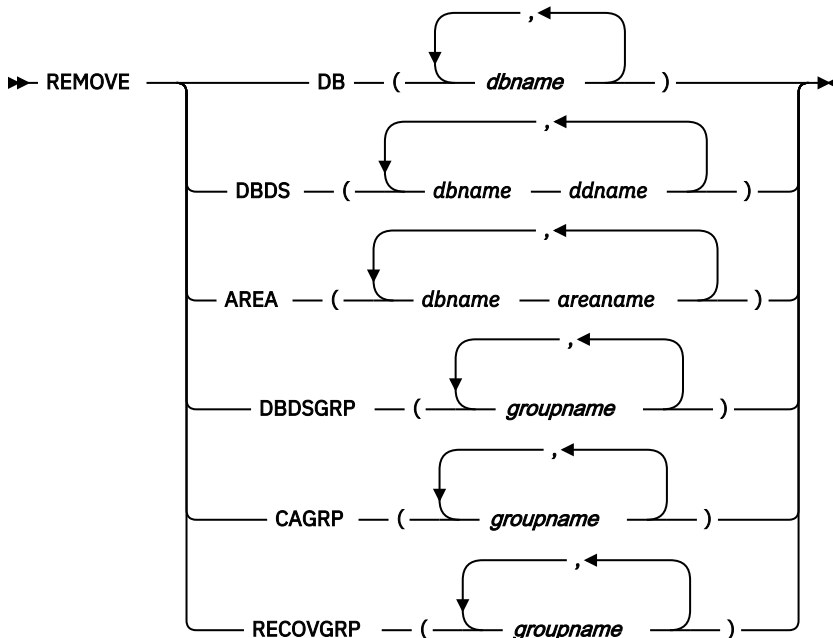
| Database | Recovery | Facility                   | Duplicate | Data Set | Report |
|----------|----------|----------------------------|-----------|----------|--------|
| DB       | DD       | Dsname                     |           |          |        |
| MYDATA   | INV      | HIQUAL.MYDATA.INV.NEW      |           |          |        |
| MYDATA   | SHIPMENT | HIQUAL.MYDATA.SHIPMENT.NEW |           |          |        |
| MYDATA   | ORDERS   | HIQUAL.MYDATA.ORDERS.NEW   |           |          |        |
| MYDATA   | ACCTRECV | HIQUAL.MYDATA.ACCTRECV.NEW |           |          |        |

All data sets are created using the SMS data class myclass and they reside on different volumes.

## Reference: REMOVE command

The **REMOVE** command is used to remove one or more selected database data sets from the recovery list that might have been added as a result of using a group name in a preceding **ADD** command.

### Syntax for REMOVE command



### Parameters for REMOVE command

The **REMOVE** command removes one or more selected database data sets from a recovery list. It can remove an individual DBDS entry, database, or a DBRC group. It reverses **ADD** command actions. The list

must have been built by one or more **ADD** commands and the **REMOVE** command must come before the **START** command.

DBDSs and areas are specified the same way as in the **ADD** command.

Use the following parameters with the **REMOVE** command.

#### **DB**

Specifies that the full-function database data sets or Fast Path areas making up one or more databases are to be removed from the recovery list.

##### ***dbname***

Specifies the database and the associated database data sets or areas that are to be removed from a recovery list.

#### **DBDS**

Specifies that one or more full-function database data sets are to be removed from the recovery list.

##### ***dbname***

Specifies the database name of the database data set to be removed from a recovery list.

##### ***dbname***

Specifies the DD name of the database data set. If you specified DBDS on the **REMOVE** command, you must specify *dbname* and *ddname* together.

#### **AREA**

Specifies that one or more Fast Path database areas are to be removed from a recovery list. If you specified AREA on the **REMOVE** command, you must specify *dbname* and *areaname* together (separated by a blank space).

##### ***dbname***

Specifies the database name of the area to be removed from a recovery list.

##### ***areaname***

Specifies the area to be removed from a recovery list.

#### **DBDSGRP**

Follow this keyword with the name of the group of database data sets defined to DBRC.

##### ***groupname***

Specifies the name of the group.

#### **CAGRP**

Follow this keyword with the name of the group of database data sets in a change accumulation group as defined to DBRC.

##### ***groupname***

Specifies the name of the group.

#### **RECOVGRP**

Specifies that this group is a recovery group. All DBDSs that make up the full-function databases and all the DEDB areas are removed from the recovery list.

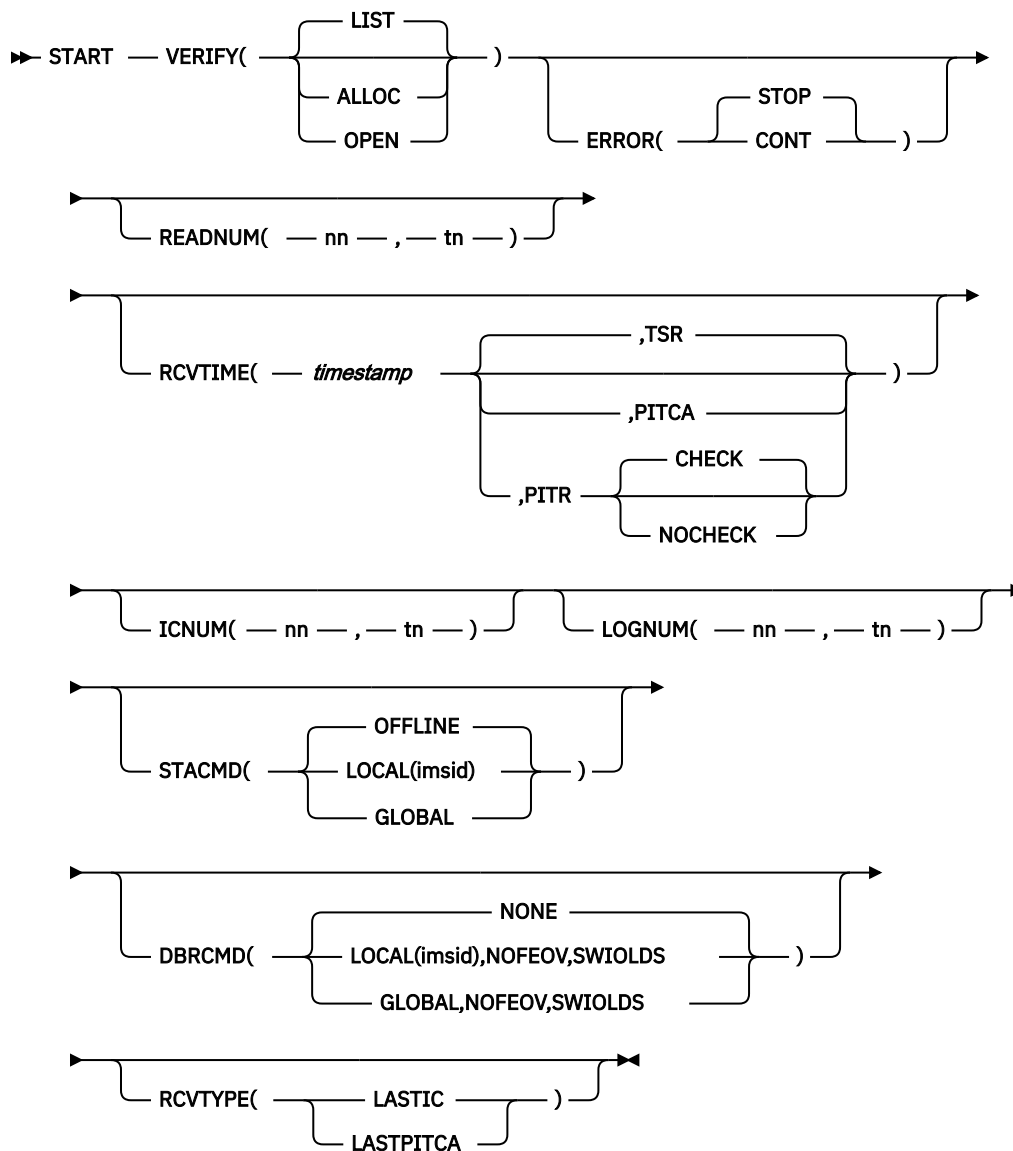
##### ***groupname***

Specifies the name of the group.

## Reference: START command

The START command initiates the IMS Database Recovery Facility processing. The syntax and use of the START command is described here.

### Syntax for START command



### Parameters for START command

You can specify only one **START** command for a single execution, and it is the last command processed in the list. If you have specified any commands following the **START** command, they will result in errors and the process will end.

Use the following parameters with the **START** command.

#### VERIFY(LIST,ALLOC,OPEN)

This parameter specifies the VERIFY level that you want IMS Database Recovery Facility to perform. When VERIFY is specified, no recovery or output processing is performed. VERIFY is used to verify the recovery assets that are needed to perform the requested action.

You do not need to stop the database to use the VERIFY parameter. However, if you do not stop the database when using VERIFY, the results may differ from an actual recovery that is performed when

the database is stopped. For example, the list of log data sets that are needed for a recovery do not include the active log when using VERIFY, and you can do a new change accumulation or an image copy when the database is stopped.

If you specify VERIFY without any options, it defaults to VERIFY(LIST).

**Note:** When verifying logs, if the databases are allocated or updated during the VERIFY, only archived logs can be verified. In this case, the recovery (without VERIFY) will show a different number of logs since the databases will be de-allocated by recovery.

Three VERIFY levels can be specified.

#### **LIST**

If you specify VERIFY(LIST), IMS Database Recovery Facility provides a list of all log, image copy, and change accumulation data sets that is required for recovery. The list contains information that is extracted from DBRC about each data set. LIST is the default value.

#### **ALLOC**

If you specify VERIFY(ALLOC), IMS Database Recovery Facility obtains the list of data sets and then allocates each one, with a deferred mount for tape data sets, to verify the existence and physical accessibility of each data set. The LIST function is also performed.

If an error is encountered in the allocation process, the data set entry on the report is flagged as an error with the ALLOC RC = *nn* message next to the name. *nn* This parameter specifies the return code from the DYNALLOC macro.

If any data set receives a non-zero return code for ALLOC, all members in the recovery list shows a Final Status of "Verify alloc failure" in the Summary section of the report. Check the I/O Section to determine which data sets failed.

**Tip:** Search for FRD4302I or FRD4210I messages in your job log for additional reason codes regarding the cause of image copy, log, or change accumulation processing errors.

#### **OPEN**

If you specify VERIFY(OPEN), each data set is opened to verify its availability and usability. The LIST and ALLOC functions are also performed.

If an error is encountered during the OPEN process, the data set entry on the report is flagged as an error and an OPEN RC=*nn* message is displayed next to the name. *nn* This parameter specifies the return code from the OPEN macro.

If any data set receives a non-zero return code for ALLOC, all members in the recovery list shows a Final Status of "Verify alloc failure" in the Summary section of the report. Check the I/O Section to determine which data sets failed.

#### **ERROR(CONT,STOP)**

Tells IMS Database Recovery Facility whether to stop or continue when an error authorizing or recovering a DBDS named in the recovery list is encountered.

#### **CONT**

Indicates that recovery of DBDS entries in the recovery list is to continue, other than the one for which an error was encountered.

#### **STOP**

Indicates that the IMS Database Recovery Facility is to shutdown all recovery tasks and end processing. This is the default.

#### **READNUM(*nn* [,*tn*])**

You can use this parameter to override the values set in the FRXDRExx member for the execution of this START command.

**Note:** If the values of either *nn* or *tn* are omitted or set to zero, the default values are imposed. The values entered here have precedence; they override the system defaults and any override values set in the FRXDRExx member.

**Note:** For a more detailed description of this parameter, see [“Environmental control statements” on page 40](#).

**RCVTIME(timestamp,TSR,PITCA,PITR(CHECK,NOCHECK))**

You must specify the RCVTIME parameter whenever you want to:

- Perform a timestamp recovery - RCVTIME(timestamp,TSR)
- Perform a point in time recovery - RCVTIME(timestamp,PITR)
- Perform a point in time recovery using a point in time change accumulation data set - RCVTIME(timestamp,PITCA)

The RCVTIME parameter is optional when you want to create an incremental image copy. If OUTPUT(ICR) is specified without RCVTIME, the image copy is created to the current time. If RCVTIME(timestamp) is specified, the image copy is created to the specified time stamp. The state of the database at the specified (or implied) time determines the type of image copy that is created:

- If the database is allocated, a concurrent image copy is created.
- If the database is not allocated, a batch image copy is created.

For more information about the incremental image copy function, see [“Environmental control statements”](#) on page 40, OUTPUT parameter.

**timestamp**

Timestamp format reference: [“Reference: Timestamp format”](#) on page 119

**TSR**

Use time stamp recovery (TSR) to perform a time stamp recovery. When a TSR is performed, the IMS Database Recovery Facility checks the status of the databases being recovered to make sure that they are not allocated or in use at the specified timestamp. You must have issued / DBRecovery commands from all online IMS systems which had them in use, and there must not be any batch jobs updating the databases at the time specified by the RCVTIME parameter. A listing of the RECON can help you determine valid timestamps for performing a TSR. There must be no ALLOC record for any database data set or area being recovered that spans the recovery time.

When performing a TSR recovery, IMS Database Recovery Facility verifies that all related DBDSs are also being recovered to the same time. For each related DBDSs which are not being recovered, message FRD6024A is issued to warn you of this condition. Recovery fails if any related DBDSs are not also being recovered.

Input which is available for use by TSR is:

- A prior image copy, concurrent or batch
- A complete change accumulation which has a stop time earlier than the recovery timestamp specified
- Archived logs (SLDS or RLDS) which have a stop time later than the stop time from the input image copy and earlier than the recovery timestamp specified
- If the input image copy is a concurrent image copy, then log data prior to the image copy stop time might be needed in order to resolve any in-flight work at the time the image copy was taken

**PITR**

Use point-in-time recovery (PITR) to perform a point-in-time recovery. When a PITR is performed, the databases being recovered can be in any state; they may be allocated or unallocated. There is no restriction on database allocation status if point-in-time recovery is selected. All committed updates, up to and including the specified recovery time, are applied to the database data sets and areas in the recovery list. If you specify the PITR parameter, the recovery time can be specified to any time before the current time.

When performing a PITR recovery, IMS Database Recovery Facility verifies that all related DBDSs are also being recovered to the same time. For each related DBDSs which are not being recovered, message FRD6024A is issued to warn you of this condition.

Input which is available for use by PITR is:

- A prior image copy, concurrent or batch

- A change accumulation which has a stop time earlier than the recovery timestamp specified, but only if there is a deallocation point (DBRECOVERY command) between the CA stop time and the PITR recovery time. This is the only way for the IMS Database Recovery Facility to ensure that all database updates on the CA data set are committed
- Archived logs (SLDS or RLDS) which have a stop time later than the stop time from the input image copy and earlier than the recovery timestamp specified.
- If the input image copy is a concurrent image copy, then log data prior to the image copy stop time may be needed in order to resolve any in-flight work at the time the image copy was taken

When you specify PITR, you can also specify the following options.

#### **CHECK**

Use CHECK to indicate that recovery fails when IMS Database Recovery Facility discovers that there are related DBDSs which are not being recovered to the same time. This processing ensures that DBDSs are not recovered to inconsistent times, causing integrity issues. This is the default value.

#### **NOCHECK**

Use NOCHECK to allow recovery to continue when IMS Database Recovery Facility discovers that there are related DBDSs which are not being recovered to the same time. Specifying NOCHECK means that you are responsible for ensuring related DBDSs are recovered consistently.

#### **PITCA**

Use point in time change accumulation (PITCA) to perform a point in time recovery, using a point in time change accumulation data set as input. IMS High Performance Change Accumulation (HPCA) is able to generate a point in time change accumulation file to a point in time where databases are still online. This CA file is called a PIT HPCA and contains only committed updates. This type of CA is marked invalid in the RECON to prevent usage by other utilities. Use PITCA to perform a point in time recovery using only a prior image copy and a PIT HPCA as input. No additional data sets containing log updates are used as input to this recovery.

When HPCA marks the PIT HPCA file invalid in the RECON data sets, DBRC switches the contents of the RUN and START time fields in the record. Therefore, the timestamp you specify on the RCVTIME parameter must be the RUN time of the CA record as recorded by DBRC. In order for the IMS Database Recovery Facility to select the PIT HPCA file, the CA record must be marked ERR and the CA data set name must have the suffix ".HPCAP".

Input which is available for use by PITCA is:

- A prior image copy, concurrent or batch
- A PIT IMS HP Change Accumulation Utility point in time change accumulation data set which has a RUN time matching the recovery time specified, is marked in ERR, and has a suffix of ".HPCAP"

#### **RCVTYPE(LASTIC, LASTPITCA)**

The RCVTYPE parameter is used whenever you want to recover using only:

- A prior image copy, it may be concurrent or batch
- An IMS HP Change Accumulation Utility point in time change accumulation data set which has a RUN time matching the recovery time specified, is marked in ERR, and has a suffix of ".HPCAP"

#### **LASTIC**

Use last image copy (LASTIC) to perform a recovery using only the latest usable batch image copy. No logs or change accumulation data sets are used.

The latest image copy where the database data set (DBDS) area was not allocated for the time the image copy was taken are used for the recovery. There are several types of image copies. Batch is one where the DBDS area has been stopped. A concurrent image copy (CIC) can be taken while the DBDS area has not been stopped. However, a CIC can also be taken while the DBDS area has been stopped. It is still recorded as a CIC. Any type of image copy taken while the DBDS area has been stopped are used for a LASTIC recovery.

LASTIC is another way of specifying a time stamp and performing either a full recovery or a time stamp recovery (TSR). A new parameter, RCVTYPE, allows for either LASTIC or LASTPITCA. A LASTIC recovery uses only the image copy. If there were changes made after the IC was taken, then this is a TSR. Otherwise, it is a full recovery.

Unlike RCVTIME, recoveries with LASTIC can be to different times and can be full recoveries or time stamp recoveries. The RECOV record that is recorded to DBRC reflects this. For time stamp recoveries (including PITR), the CHECK option causes IMS Database Recovery Facility to fail the recovery when there is a DBDS area that should also be recovered to the same time but is not. When NOCHECK is specified, IMS Database Recovery Facility does not make this check. Because LASTIC does not have a constant RCVTIME and the resulting recoveries can be a mixture of full and time stamp recoveries, CHECK is not possible for LASTIC. Although NOCHECK is implied, CHECK and NOCHECK are not allowed with LASTIC. An error message is issued if either is specified.

The use of RCVTYPE(LASTIC) is:

- Mutually exclusive with RCVTIME
- Allowed for OUTPUT(PRO | DUP | BOTH)
- Not allowed for OUTPUT(ICR | ICRCA)

### LASTPITCA

Use LASTPITCA to perform a point in time recovery using an image copy and the last point in time change accumulation (PIT HPCA) as input. No log data sets are used.

Using LASTPITCA is another way of specifying a timestamp and the recovery is always a point in time recovery (PITR). A new parameter, RCVTYPE, allows for either LASTIC or LASTPITCA.

The RECOV record that is recorded to DBRC reflects the type of recovery. For time stamp recoveries (including PITR), the CHECK option causes IMS Database Recovery Facility to fail the recovery when there is a database data set (DBDS) area that should also be recovered to the same time but is not. When NOCHECK is specified, IMS Database Recovery Facility does not make this check. Because LASTPITCA does not have a constant RCVTIME and the resulting recoveries can be a mixture of full and time stamp recoveries, CHECK is not possible for LASTPITCA. Although NOCHECK is implied, CHECK and NOCHECK are not allowed with LASTPITCA. An error message is issued if either is specified.

The use of RCVTYPE(LASTPITCA) is:

- Mutually exclusive with RCVTIME
- Allowed for OUTPUT(PRO | DUP | BOTH)
- Not allowed for OUTPUT(ICR | ICRCA)

### ICNUM(*nn* [,*tn*])

The system defaults and any override values you set in the FRXDRFxx member for *nn* and *tn* are overridden by the values supplied in this **START** command.

**Note:** If you omit or set the values of either *nn* or *tn* to zero, the default values in FRXDRFxx are imposed. The values you enter here take precedence; they override the system defaults and any override values set in the FRXDRFxx member.

**Note:** For a more detailed description of this parameter, refer to [“Environmental control statements” on page 40](#).

### LOGNUM(*nn* [,*tn*])

The system defaults and any override values you set in the FRXDRFxx member for *nn* and *tn* are overridden by the values you supply in this **START** command.

**Note:** If you omit or set the values of either *nn* or *tn* to zero, the default values are imposed. The values entered here have precedence; they override the system defaults and any override values set in the FRXDRFxx member.

For a more detailed description of this parameter, refer to [“Environmental control statements”](#) on page 40.

### **STACMD(OFFLINE,LOCAL((*imsid*)),GLOBAL)**

Use STACMD to tell IMS Database Recovery Facility what the disposition of the DBDS should be after it has been recovered. STACMD is not issued when VERIFY is also specified. The following are the subparameters of STACMD:

#### **OFFLINE**

Indicates that after recovery of DBDS entries in the recovery list, the resource is to be kept offline. This is the default.

#### **LOCAL((*imsid*))**

Indicates that after recovery of DBDS entries in the recovery list, the **START** command is issued against the specified IMS subsystem, *imsid*.

#### **GLOBAL**

Indicates that after recovery of DBDS entries in the recovery list, the **START** command is issued with the GLOBAL keyword and applies to all sharing subsystems.

### **DBRCMD(NONE,LOCAL((*imsid*)),GLOBAL)**

Use DBRCMD to tell IMS Database Recovery Facility to issue the DBRECOVERY command before attempting to recover the DBDSs in the recovery list. DBRCMD is not issued when VERIFY is also specified.

#### **NONE**

Indicates that the DBRECOVERY command should not be issued for any entry in the recovery list. This is the default.

#### **LOCAL((*imsid*))**

Indicates the DBRECOVERY command is issued on the specified IMS subsystem, *imsid*.

#### **NOFEOV**

Indicates that the NOFEOV parameter is specified on all /DBR commands that are issued, including the last one. This means that a log switch is not done.

**Important:** Typically the NOFEOV parameter is specified on all /DBR commands except for the last one. This means that an OLDS switch occurs only once, after the last database is recovered by using the /DBR command.

#### **SWIOLDS**

Indicates that after all databases have been taken offline by using the /DBR command, a /SWI OLDS command is issued on the specified IMS system.

If SWIOLDS is specified, NOFEOV is assumed to prevent multiple OLDS switches.

#### **GLOBAL**

Indicates that the DBRECOVERY command is issued with the GLOBAL keyword and applies to all sharing subsystems which have resources registered to DBRC.

#### **NOFEOV**

Indicates that the NOFEOV parameter is specified on all /DBR commands that are issued, including the last one. This means that a log switch is not done.

**Important:** Typically the NOFEOV parameter is specified on all /DBR commands except for the last one. This means that an OLDS switch occurs only once, after the last database is recovered by using the /DBR command.

#### **SWIOLDS**

Indicates that after all databases have been taken offline by using the /DBR command, a /SWI OLDS command is issued on all active IMS systems.

If SWIOLDS is specified, NOFEOV is assumed to prevent multiple OLDS switches.



## Reference: Timestamp format

---

IMS Database Recovery Facility applies additional syntax rules to the standard DBRC timestamp syntax.

### General format rules for the timestamp and time zone offset

For more complete information about the formats that are recognizable to IMS Database Recovery Control (DBRC), refer to the topics about DBRC timestamps in the *IMS Database Recovery Control (DBRC) Guide and Reference*.

- The format of the timestamp value is expressed as either compressed or punctuated.
- Timestamps should always be enclosed within single quotation marks ( ' ).

**Note:** Compressed timestamps containing no spaces are accepted without quotes.

- In the punctuated timestamp format, the delimiter character ( | ) can be any non-alphanumeric character (including blank) with the exception of the single quotation mark ( ' ).
- Alphabetic characters (A-Z) can not be used as field delimiters in a timestamp.
- The first alphabetic character encountered in a timestamp determines the first character of a **symbolic** time zone offset (for example: EST, PST, ABC, etc.).
- A blank space must precede a **symbolic** time zone offset in both punctuated and compressed timestamp formats.
- The occurrence of the plus (+) or minus (–) character is considered the start of a **numeric** time zone offset.
- A blank space is **not** required to precede a **numeric** time zone offset in both punctuated and compressed timestamp formats.

### Punctuated timestamp format

In the following timestamp format, the use of brackets [ ] indicates that the value contained in the brackets is optional.

Punctuated timestamps follow this format:

```
'[yy]yy|ddd|hh|mm|ss|thmi|ju [offset]'
```

### Compressed timestamp format

In the following timestamp format, the use of brackets [ ] indicates that the value contained in the brackets is optional.

Compressed timestamps follow this format:

```
'yydddhmmsssthmi|ju [offset]'
```

### Timestamp specification reference

#### [yy]yy

The year specification (1978 – 9999).

You can abbreviate the year to yy, specifying only the last two digits of the year. For example, 2020 and 20 are equivalent values.

**Note:** Compressed timestamps only accept the two digit year format.

#### ddd

The day specification (001 – 366).

#### hh

The hour specification (0-23).

**mm**

The minute specification (0-59).

**ss**

The second specification (0-59).

**thmiju**

The specification (known as precision) down to the millionth of a second (000000 – 999999)

t (ten), h (hundred), m (1 thousand), i (10 thousand), j (100 thousand), u (1 million)

**offset**

- Offset is an optional specification and represents a value that, when added to (or subtracted from) UTC (Coordinated Universal Time), provides local time.
- Offset can be specified as a **numeric** value in the form (punctuated and compressed):

```
±h[h[:mm]]
```

```
±h[h[mm]]
```

*h[h]* is a numeric value from 0 to 14, representing UTC time zones east or west of Greenwich, England. (Although the normal time zones east and west of Greenwich are ±12 hours, there are actually a few time zones on the earth that are 13 and 14 hour zones.)

*mm* is a numeric value from the set {00, 15, 30, 45}.

**Note:** For the compressed timestamp format, if you specify *mm*, then you must also specify *h[h]*.

- Offset can also be specified as a predefined **symbolic** string that represents the required offset value.

The **symbolic offset** string and value must be previously defined to the RECON data sets using the following command:

```
CHANGE.RECON TIMEZONE(NNNNN, offset)
```

For example (first the defining RECON command, and then its usage in a punctuated timestamp):

```
CHANGE.RECON TIMEZONE(ABC, +11:45)
```

```
'2020 045 16.24.45.7 ABC'
```

To represent the symbolic offset string, you can use any set of characters that is meaningful for your usage - such as GMT (Greenwich Mean Time), PDT (Pacific Daylight Time), EST (Eastern Standard Time), ABC (see example above), or even XYZ.

The symbolic offset string (ABC in the example above) has a limit of 5 characters.

For a further description of symbolic time zone offsets, see the TIMEZONE keyword under the CHANGE.RECON command in *IMS Commands: IMS Component and Z/OS Commands* user documentation.

**Timestamp examples**

Here are 7 examples of valid **punctuated** timestamps:

```
'20.252/08:24:45.712345'
'20.252 08:24:45.712 -8'
'20/252/08.24.45.71 -8:00'
'20/252/08.24.45.71-8:00'
'2020 252 16.24.45.7 +0'
'2020 252 16.24.45.7'
'2020 045 16.24.45.7 ABC'
```

Here are 4 examples of valid **compressed** timestamps:

```
'20252082445712345'  
'20252082445712-0800'  
'202521624457+0'  
'202521624457 ABC'
```

### Timestamp technical notes:

- The precision level specified on the timestamp is used by DBRC to locate information in the RECON.

The precision level can range from a tenth of a second to a microsecond, or 1 to 6 digits to the right of the decimal point. This is sometimes called precision 1 to 6.

Precision 6 is known as full precision. DBRC uses full precision (precision 6).

- The timestamp you specify is converted into Coordinated Universal Time (UTC), which is used internally by IMS and DBRC to store all timestamps.

The format of the external timestamp that you specify provides a more user-readable format.

- If a timestamp does not contain an offset, it is assumed to be local time and the offset of the local MVS (the one on which the job executes) is used to calculate the UTC timestamps that are used internally.
- If you need to recover to a time prior to a seasonal time change, take care when specifying the timestamp to ensure that the correct results are obtained. In these cases, you must specify the offset that is associated with the time period being recovered to.

For example, if daylight savings time went into effect today and you are in the Pacific Time zone, the offset for calculating UTC time yesterday would have been -8 hours. Today, since daylight savings time is in effect, the offset is now -7 hours.

In this example, in order to recover to the correct point in time (yesterday), you must use -8 hours as the offset value in the timestamp.



---

## Chapter 8. Using the integrated auxiliary utilities

This information describes how to use the IMS Database Recovery Facility and invoke the integrated auxiliary utilities.

This section extends the information discussed in [Chapter 7, “Using batch recovery control statements,”](#) on page 105.

Refer also to the descriptions for the control statement parameters DRFIAX, LIU@GOPT, and REPORT, described in [“Environmental control statements”](#) on page 40.

### Topics:

- [“Utility control statement specification”](#) on page 123
- [“Reference: ADD IB\(\) command”](#) on page 124
- [“Reference: ADD IC\(\) command”](#) on page 131
- [“Reference: ADD PC\(\) command”](#) on page 140
- [“Return code handling”](#) on page 144

---

### Utility control statement specification

The utility control statements are used to direct the execution of the integrated auxiliary utilities.

The utility control statements can be specified by using either of these methods:

- On the global UTILGBL() environmental control statement, either in the FRXDRFxx PROCLIB member or the //SYSIN DD input.
- On the ADD command, by using the appropriate utility keyword, IB(), IC(), or PC().

Specifying the utility control statements on the UTILGBL() statement sets the default values that are used for any invocation of the utility. The values specified on the UTILGBL() statement are used by IMS Database Recovery Facility anytime any of the utilities are executed.

You can also specify most of the utility control statements on the ADD command by using the appropriate IB(), IC(), or PC() keyword. If multiple ADD commands are used, you must specify utility control statements that are specific to each ADD command. All databases that are specified on the ADD command use the same utility options that are specified on that command.

To invoke any of the integrated auxiliary utilities, the IB(), IC(), and PC() keyword must be specified on the ADD command. The utilities cannot be executed by only specifying utility control statements on the UTILGBL() statement, except the IMS Library Integrity Utility which is invoked by using the LIU@GOPT keyword on the UTILGBL command.

If all the necessary parameters are already specified on the UTILGBL() statement, you can simply code the IB(), IC(), or PC() keyword without parameters. However, if you want to specify additional parameters or override a parameter that was specified on the UTILGBL() statement, you must use the appropriate utility keyword and specify the necessary parameters.

For example, to invoke all the utilities by using the default values, code the following ADD command:

```
ADD DB(HIDAM1) IC() IB() PC()
```

However, if you want to specify custom parameters for running IMS High Performance Image Copy, use the ADD IC() command with parameter specifications:

```
ADD DB(HIDAM1) DBATRB(DBDSL(101),DBDSN(101)), -  
    IC (ICHLQ(ichlq),SPACE(CYL,3,1),ICNMRULE(Y),ICCAT(Y), -  
    DSNTYPE(BASIC),COMP(N), -  
    STORCLAS(storcl),MGMTCLAS(mgmtcl),DATACLAS(datacl),UNIT(unit))
```

The syntax and usage of the ADD IB(), ADD IC(), and ADD PC() commands is explained in the sections that follow.

## Reference: ADD IB() command

---

This topic describes the syntax and use of the ADD IB() command for invoking IMS Index Builder and the Build Index function of FPA for building indexes.

Any parameters specified on the ADD IB() command override those parameters that were specified on the UTILGBL() control statement.

There are also parameters for this auxiliary utility that can only be specified on the UTILGBL() control statement.

For a list and description of those parameters, refer to [“Reference: Utility control statements” on page 69](#).

### Topics:

- [“ADD IB\(\) command for IMS Index Builder” on page 124](#)
- [“ADD IB\(\) command for the Build Index function of FPA” on page 129](#)

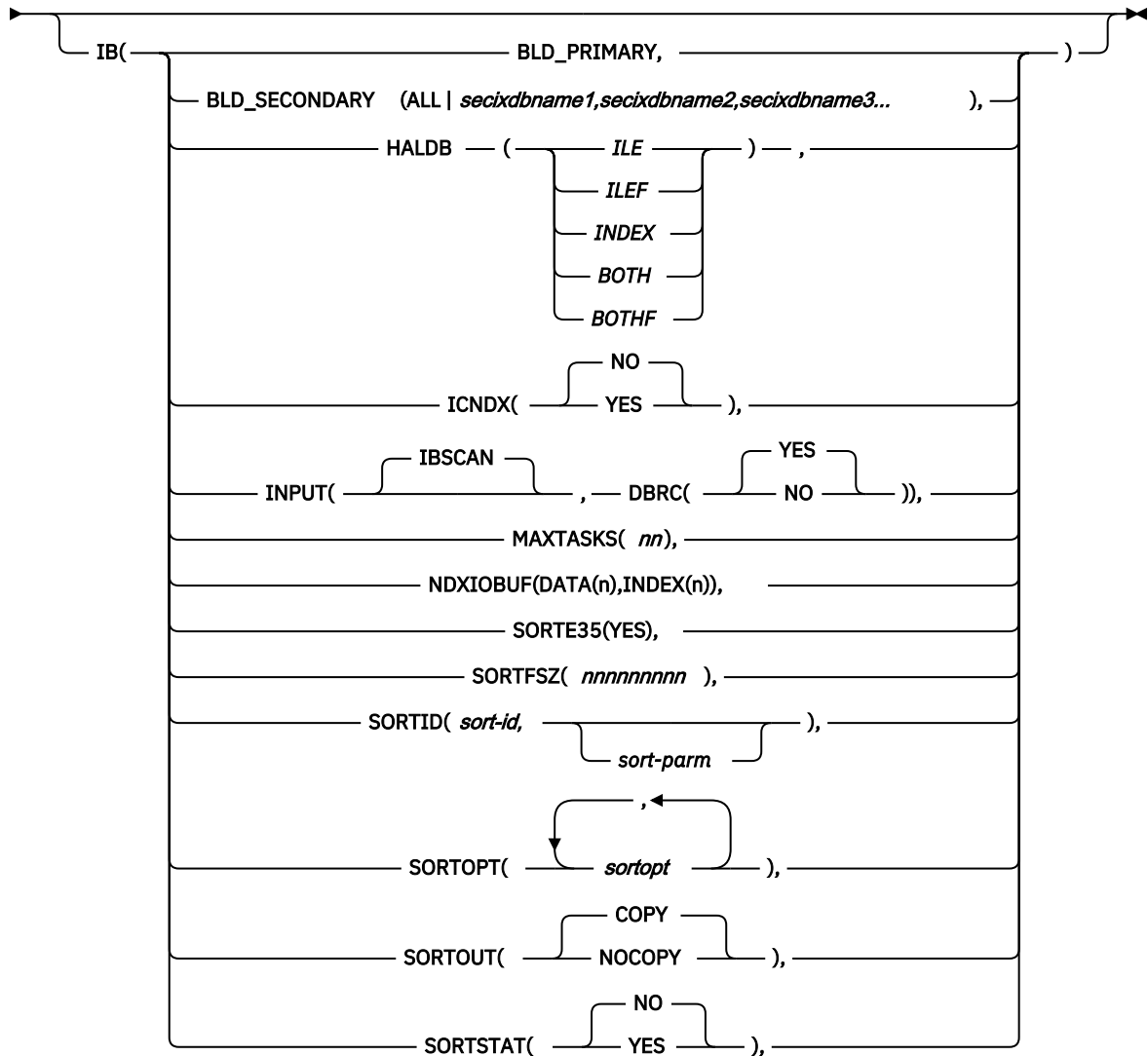
## ADD IB() command for IMS Index Builder

This topic describes the syntax and use of the ADD IB() command for invoking IMS Index Builder.

### Syntax for ADD IB() command for IMS Index Builder

The basic format of the ADD command is described in [“Parameters for ADD command” on page 106](#). Shown here is the syntax for those parameters that are associated with the IMS Index Builder usage.

➡ ADD ➡



### Parameter reference for ADD IB() command for IMS Index Builder

The use of the IB() keyword on the ADD command indicates that indexes associated with recovered database data sets are to be rebuilt after the DBDS is recovered for OUTPUT(PRO) only. The IB statement is ignored for OUTPUT(DUP).

Any combination of the IB statements that are used for rebuilding primary, secondary, or HALDB ILDS data sets can be specified regardless of what type of databases are being recovered. IMS Database Recovery Facility determines the type of database and the indexes present and only invokes IMS Index Builder or DFSPRECO for those that are present.

The following reference shows the IMS Index Builder and DFSPRECO utility options that can be specified on the IB() keyword within the ADD command.

#### BLD\_PRIMARY

Use this parameter to specify that IMS Database Recovery Facility should rebuild the primary index of the *dbname* specified on the ADD DB() command.

HALDB primary indexes are rebuilt using the DFSPRECO utility. In IMS Database Recovery Facility, specifying the BLD\_PRIMARY command does not rebuild HALDB primary indexes. Use the parameters in the HALDB() statement to build HALDB primary indexes.

When the DFSPREC0 routine is needed for HALDB Primary Index and ILE processing, you must modify module DFSDAPL0.

**BLD\_SECONDARY(ALL | *secixdbname1,secixdbname2,secixdbname3...*)**

Use this parameter to rebuild the secondary index databases of the *dbname/areaname* specified on the ADD DB() command.

**Important:** When BLD\_SECONDARY(ALL) is specified, all secondary indexes are rebuilt. It is not necessary to specify BLD\_SECONDARY(*secixdbname1, secixdbname2, secixdbname3*).

**Note:** Only BLD\_SECONDARY(ALL) is supported when this parameter is specified on the UTILGBL() environmental statement.

**HALDB(ILE | ILEF | INDEX | BOTH | BOTHF)**

This parameter applies only to HALDB databases. Use this parameter to specify whether to rebuild only the ILE data set, the Primary Index, or both.

Specify this parameter only on the ADD statements for PHIDAM database types.

**ILE**

The indirect list entry (ILE) is to be rebuilt.

**ILEF**

The ILDS of the specified partition is rebuilt by using the free space option. DFSPREC0 must have five z/OS data spaces available for processing if you select the free space option.

**INDEX**

The HALDB primary index is to be rebuilt.

**BOTH**

The specified ILE and the primary index are to be rebuilt.

**BOTHF**

The primary index and the ILDS of the specified partition are to be rebuilt. The ILDS is rebuilt by using the free space option.

The utility must have five z/OS data spaces available for processing if you select the free space option.

**Important:** If IMS Index Builder Version 3.1 or higher is used, and the free space option is not specified, the ILDS and PRIMARY index are rebuilt by IMS Index Builder.

**ICNDX(NO | YES)**

This parameter specifies that indexes are to be image copied.

This parameter does not apply to HALDB primary index and ILE data sets because they are not recoverable.

This parameter also does not apply to secondary indexes that are not recoverable.

However, if you specify ICNDX(YES), secondary indexes for Full Function, Fast Path and HALDB databases are image copied. For Full Function databases, recoverable primary indexes are image copied.

NO is the default.

**Note:** If ICNDX(YES) is coded, it does not take effect unless the IC keyword is present on the ADD command.

**INPUT(IBSCAN,DBRC(Y | N))**

This parameter specifies the type of input used by IMS Index Builder to build indexes.

IBSCAN specifies that a scan of the physical data base is used as input. In IMS Database Recovery Facility, this option applies to secondary indexes for both HALDB and Full Function databases and to primary indexes for Full Function databases.

IBSCAN does not apply to FPA and is ignored if specified.

DBRC specifies whether DBRC is (DBRC(Y)) or is not (DBRC(N)) used. DBRC(Y) is the default.



## MAXTASKS(nn)

Use this parameter to specify that this is a MAXTASKS statement.

**nn**

Valid values are from 0 to 36, with leading zeros.

**Important:** MAXTASKS(2) has special meaning in that it instructs IMS Index Builder V3.1 to use the hierarchical HPU scan method for building indexes, such as in IMS Index Builder V2.3, instead of the sequential scan method. For more information, see the *IMS Index Builder User's Guide*.

## NDXIOBUF(DATA(n),INDEX(n))

Use this parameter to specify the number of buffers that VSAM uses when IMS Index Builder loads index data sets. Use this parameter when:

- IMS Index Builder loads index data sets without needing a sort.
- You specify the SORT E35 statement to use the E35 exit interface to sort.

## SORTE35(NO | YES)

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces.

Use this parameter to specify whether IMS Index Builder can use the sort E35 exit to load sorted index records into the index data set.

When NO is specified, the sort program loads the sorted records directly into the index data set.

The default is NO unless you have coded IC() and ICNDX(YES) is intended. Then, the default is YES.

## SORTFSZ(nnnnnnnnn)

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces.

Use this parameter to specify the estimated number of records to be sorted. *nnnnnnnnn* is an integer from 1 to 999999999 that specifies the estimated number of records to be sorted.

The value is appended to the SORT option and is overridden if a non-zero index size is specified in the INDEX option for the DBD, or by the reused file size of the index.

If not specified, and not overridden, no FILSZ parameter is appended to the generated SORT option.

## SORTID(sort\_id,{sort\_parm,...})

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces.

Use this parameter to specify the ID of the SORT command used to start the address space for the sort program.

Only one *sort\_id* keyword is allowed. It must be a single alphanumeric character appended to the sort procedure name IIUBSRT. For example, if you specify the character A, IMS Index Builder uses the procedure IIUBSRTA to start address spaces used by the sort program.

The specified procedure must exist in one of the libraries in the SYS1.PROCLIB concatenation of the MVS primary subsystem. If a *sort\_id* is not specified, IMS Database Recovery Facility uses IIUBSRTT.

*sort\_parms* is a string of sort option parameters, separated by commas or blanks. The string is appended, unchanged, to the SORT OPTION command generated by IMS Index Builder.

The maximum length of the string is 34 characters, and the entire string must be on one line.

You can use this string to specify performance options or to override installation settings. For example:

```
MAINSIZE(100) FILSZ(E3000000)
```

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

### **SORTOPT(*sortopt...*)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate address spaces. Use this parameter to specify sort options to be appended to the SORT OPTION command.

Only one SORTOPT parameter is allowed.

**Important:** The FILSZ specified in the SORTOPT parameter takes precedence.

*sortopt...* is a string of sort options, separated by commas or blanks. The string is appended, unchanged, to the SORT OPTION command generated by IMS Index Builder.

The maximum length of the string is 34 characters, and the entire string must be on one line.

You can use this string to specify performance options or to override installation settings. For example:

```
MAINSIZE(100) FILSZ(E30000000)
```

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

### **SORTOUT(COPY | NOCOPY)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate sort address spaces.

Use this parameter to specify whether IMS Index Builder can copy sort messages to the data set defined in the IIUSOUT DD statement or leave them in the data set defined in the SORT DD command, which is in the sort address space.

If you do not specify the SORTOUT parameter, COPY is assumed.

### **SORTSTAT(NO | YES)**

This parameter applies to IMS Index Builder subordinate address spaces that are involved in sorting, not IMS Database Recovery Facility subordinate sort address spaces.

Use this parameter to tell IMS Index Builder to produce sort count statistics in the SYSPRINT data set for each index sorted.

If you specify this statement, the following counts are produced:

#### **SORT RECORD COUNT — E15**

The number of records passed to sort

#### **SORT BUFFER COUNT — E15**

The number of buffers processed

#### **SORT WAIT COUNT — E15**

The number of WAITs issued while waiting for a buffer

#### **SORT OUTPUT COUNT — E35**

The number of records sorted and loaded into the index data set

**Note:** If an index *dbname1* is included in the ADD DB() command and that index is marked as recoverable in DBRC, then IMS Database Recovery Facility recovers the index data set. Also, if an image copy is requested as part of the ADD command, then IMS Database Recovery Facility will image copy the recovered index.

Additionally, if as part of the IB parameter, there is a request to rebuild index *dbname1*, either through the BLD\_PRIMARY or BLD\_SECONDARY parameter, and if IB parameter, ICNDX(YES) is specified, then IMS Index Builder rebuilds the index and makes an image copy of the index data set.

### **Example of using the ADD IB() command**

The following example instructs IMS Database Recovery Facility to invoke the IMS Index Builder or the DFSPREC0 utility to rebuild indexes:

```
SORTPARM(ASPREF(DR3S))  
UTILGBL(IBPREF(DR3I),PRPREF(DR3P),PCPROCNM(FRXPAT0),PCJOBNM(DR3C))
```

```

REPORT(RPTTYPE=APP,DRFUNIT=SYSDA,DRFHLQ=h1q)
DRFIAX(FRXJCLIP)
START ERROR(CONT) DBRCMD(GLOBAL) STACMD(GLOBAL) READNUM(5)
ADD DB(PRIMEDB) -
    IB(BLD_PRIMARY(YES),BLD_SECONDARY(ALL), INPUT(IBSCAN,DBRC(Y)), -
        NDXIOBUF(DATA(10),INDEX(5)),ICNDX(YES))

```

The following events occur:

- The primary index database is rebuilt for the Full Function primary database (PRIMEDB)
- All secondary indexes are rebuilt for the PRIMEDB database
- The PRIMEDB database is scanned for input when rebuilding the indexes and DBRC is used
- VSAM uses 10 buffers for the data portion and 5 buffers when loading the index portion of the data sets. IMS Index Builder loads the indexes without needing a sort.
- All indexes that are rebuilt, and their image copies, are marked as RECOV in DBRC.

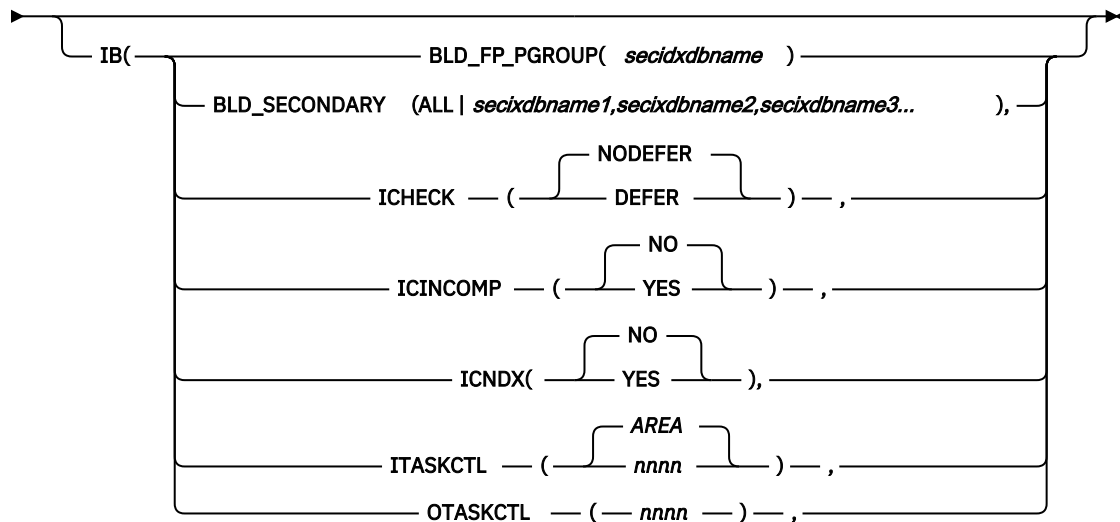
## ADD IB() command for the Build Index function of FPA

This topic describes the syntax and use of the ADD IB() command for invoking the Build Index function of FPA for building indexes.

### Syntax for ADD IB() command for the Build Index function of FPA

The basic format of the ADD command is described in “Parameters for ADD command” on page 106. Shown here is the syntax for those parameters that are associated with the Build Index function of FPA.

➡ ADD ➡



### Parameter reference for ADD IB() command for the Build Index function of FPA

The use of the IB() keyword on the ADD command indicates that indexes associated with recovered database data sets are to be rebuilt after the DBDS is recovered for OUTPUT(PRO) only. The IB statement is ignored for OUTPUT(DUP).

The following reference shows the options for the Build Index function of FPA that can be specified on the IB() keyword within the ADD command.

#### BLD\_FP\_PGROUP(secidxdbname)

##### *secidxdbname*

Use this parameter to identify the first index in a logical partition group of indexes, and to indicate that all indexes in the partition group are to be built.

**BLD\_SECONDARY(ALL | *secixdbname1,secixdbname2,secixdbname3...*)**

Use this parameter to rebuild the secondary index databases of the *dbname/areaname* specified on the ADD DB() command.

**Important:** When BLD\_SECONDARY(ALL) is specified, all secondary indexes are rebuilt. It is not necessary to specify BLD\_SECONDARY(*secixdbname1, secixdbname2, secixdbname3*).

**Note:** Only BLD\_SECONDARY(ALL) is supported when this parameter is specified on the UTILGBL() environmental statement.

**ICHECK(NODEFER | DEFER)**

Use this parameter to control the input devices.

**NODEFER**

Gets access to all devices that are associated with the input data sets at a time.

**DEFER**

Access to devices for obtaining input data sets are limited to the number that is specified by the ITASKCTL keyword.

**ICINCOMP (No | Yes)**

Use this parameter to specify, when an image copy is used as an input data set, whether the image copy records are to be compressed by using the CSRCESTRV MVS macro when they are stored into a data space storage.

IMS Fast Path Advanced Tool IMS Fast Path Advanced Tool stores all image copy records into a data space storage when an image copy is used as an input data set. Because the maximum size of one data space is 2 GB, if the total size of the records exceeds the limit, IMS Fast Path Advanced Tool ends with an error.

Try to reduce the total size of the records to under 2 GB by specifying ICINCOMP=YES. FPA compresses the image copy records by using the CSRCESTRV macro before storing them into a data space and expands them when they are processed.

The data compression and expansion process increases the amount of CPU time.

**No**

The image copy records are stored into a data space storage without compression.

**Yes**

The image copy records are stored into a data space storage after compression.

**ICNDX(NO | YES)**

This parameter specifies that indexes are to be image copied.

This parameter does not apply to HALDB primary index and ILE data sets because they are not recoverable.

This parameter also does not apply to secondary indexes that are not recoverable.

However, if you specify ICNDX(YES), secondary indexes for Full Function, Fast Path and HALDB databases are image copied. For Full Function databases, recoverable primary indexes are image copied.

NO is the default.

**Note:** If ICNDX(YES) is coded, it does not take effect unless the IC() keyword is present on the ADD command.

**ITASKCTL(AREA | *nnnn*)**

Use this parameter to specify the maximum number of input areas to be processed concurrently.

**AREA**

The number of input areas.

***nnnn***

Specify a value from 1 to 2048. Specify a value less than or equal to the number of input areas.

If you specify a value that is greater than the number of input areas, it is replaced by the number of input areas.

In case of the CHANGE command in the REORG mode, the following value is set:  $(ITASKCTL + OTASKCTL)/2$  (the value is rounded off)

### **OTASKCTL(nnnn)**

Use this parameter to specify the maximum number of output areas to be processed concurrently.

#### ***nnnn***

Specify a value from 1 to 2048. Specify a value less than or equal to the number of output areas.

If you specify a value that is greater than the number of output areas, it is replaced by the number of output areas.

In case of the CHANGE command in the REORG mode, the following value is set:  $(ITASKCTL + OTASKCTL)/2$  (the value is rounded off)

## **Reference: ADD IC() command**

---

This topic describes the syntax and use of the ADD IC() command for invoking IMS High Performance Image Copy.

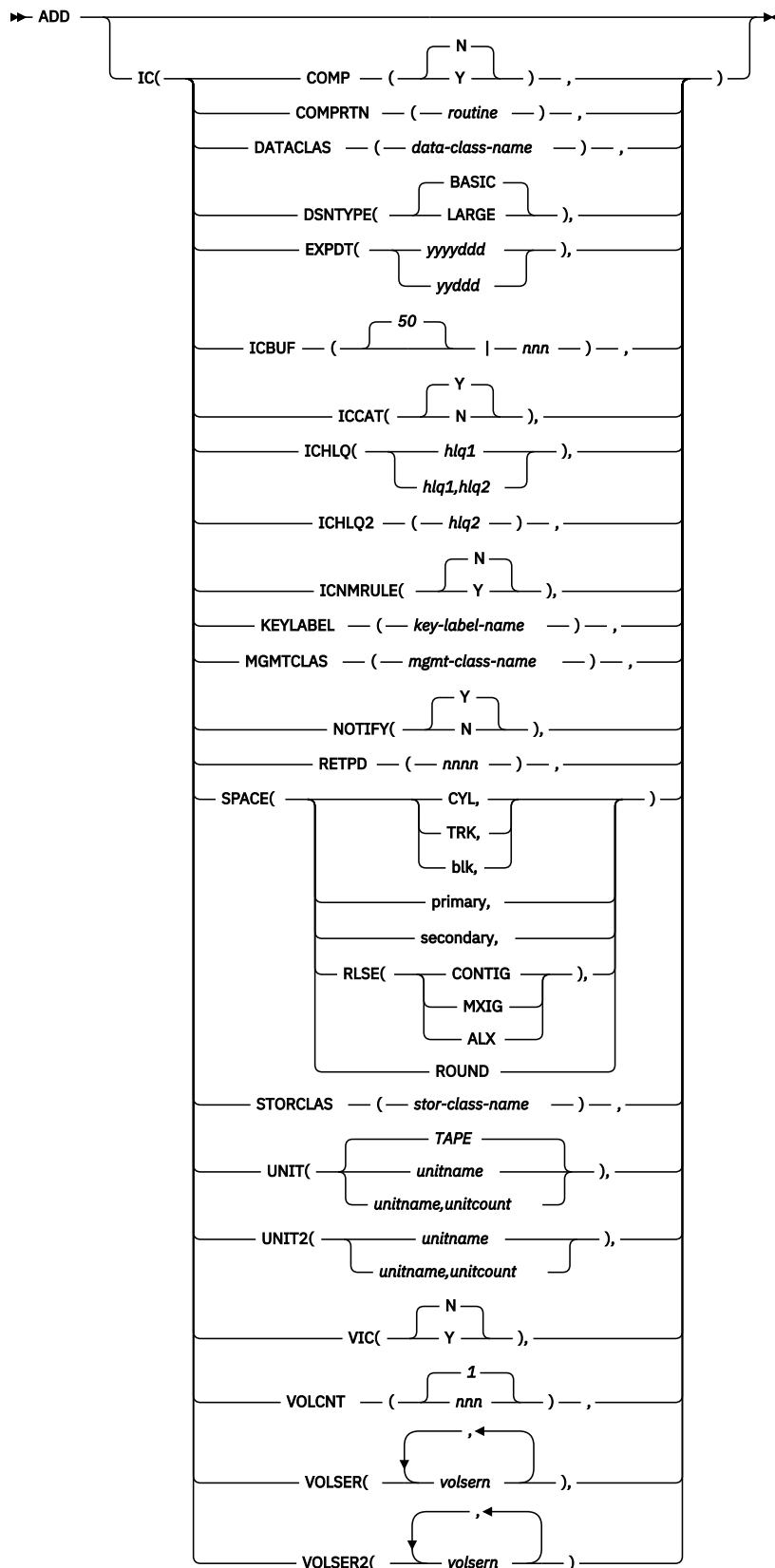
Any parameters specified on the ADD IC() command override those parameters that were specified on the UTILGBL() control statement.

There are also parameters for this auxiliary utility that can only be specified on the UTILGBL() control statement.

For a list and description of those parameters, refer to [“Reference: Utility control statements” on page 69](#).

### **Syntax for ADD IC() command**

The basic format of the ADD command is described in [“Parameters for ADD command” on page 106](#). This section shows the syntax for those parameters that are associated with the IMS High Performance Image Copy usage.



### Parameter reference for ADD IC() command

The use of the IC() keyword on the ADD command indicates that an image copy is to be created for each recovered database.



**Attention:** No image copy of any database or index is created unless the IC() keyword is specified on the ADD command. Specifying only IB(ICNDX(YES)) on the ADD command does not create an image copy.

If you specify OUTPUT(ICR) then an incremental image copy is produced using the prior image copy and archived logs or change accumulation data sets as input. The production database is not accessed and no database data sets are recovered.

If you specify OUTPUT(DUP), any IC statement is ignored and DBRC is unaffected.

Use this keyword with the ADD DB(*dbname*) command to indicate that an image copy is to be run for the recovered database. No other functions of IMS High Performance Image Copy (such as CIC, CRC) are allowed, nor are parallel processes, such as AIC.

When you specify the OUTPUT(ICR) option, the incremental image copy that is created might be batch or concurrent. A concurrent image copy is created when only a subset of the required logs are available. The time stamp that is used to register the image copy to DBRC is the latest stop time of all the log data sets that were used as input. It is not necessary to stop the database for an incremental image copy.

The following reference shows the IMS High Performance Image Copy options that can be specified on the IC keyword and used by the IMS Database Recovery Facility interface to IMS High Performance Image Copy.

#### **COMP(Y | N)**

This parameter specifies whether the Image Copy function is to compress the output image copy data.

##### **Y**

Specifies that the output image copy data is to be compressed by the Image Copy function.

##### **N**

Specifies that the output image copy data is not to be compressed by the Image Copy function.

The value N is the default.

#### **COMPRTN(*routine*)**

This parameter specifies the name of the compression routine that the Image Copy function invokes to compress the output image copy data.

**Note:** When the IDRC feature is installed and is active on the native tape drive, software compression is suppressed for the IC1 and IC2 output data sets when the compression is not FABJCMP2.

The setting for *routine* can have the following meanings:

##### ***routine***

The name of the compression routine. IMS High Performance Image Copy provides four compression exit routines: FABJCMP1, FABJCMP2, FABJCMP3, and FABJCMP4.

Specify one of these compression routines to be called by the Image Copy function.

If the COMP=Y keyword is specified without the COMPRTN= keyword, FABJCMP1 is used as the default.

The same compression routine is automatically called by the Database Recovery function to extract the compressed data.



**Attention:** If you want to create a compressed image copy data set for a DEDB that has an SDEP segment, you must use FABJCMP4. FABJCMP1, FABJCMP2, and FABJCMP3 cannot be used for a DEDB that has an SDEP segment.

IMS High Performance Image Copy provides the following compression routines:

##### **FABJCMP1**

Does repeated character compression.

Free space can also be compressed.

**FABJCMP2**

Does only free space compression.

Segmented data is not compressed.

**FABJCMP3**

Does repeating-characters compression of the z/OS program compression method.

Free space might also be compressed.

**FABJCMP4**

Does repeating-characters compression of the z/OS program compression method.

This routine runs the block compression, which does not distinguish between the data portion, free space, and unused space in the database.

**DATACLAS(*data-class-name*)**

This parameter specifies the name of the data class for the new SMS managed image copy output data sets that are dynamically allocated.

***data-class-name***

The name of the data class to be used for allocating the data set.

The name that you define is one-to eight-characters.

This parameter does not have a default.

**DSNTYPE(**BASIC** | **LARGE**)**

This parameter specifies whether the Image Copy can be allocated as Large Format data sets or Basic Format data sets.

More detail information for Large Format data sets is shown in the z/OS V1R7.0 DFSMS Using Data Sets.

**BASIC**

Specifies that the Image Copy function allocates Image Copy data sets as Basic Format data sets. BASIC is the default.

**LARGE**

Specifies that the Image Copy function allocates Image Copy data sets as Large Format data sets.

**EXPDT(*yyyyddd* | *yyddd*)**

This parameter specifies the expiration date of the output image copy data set.

***yyyyddd* or *yyddd***

The expiration date of the output image copy.

The date format must comply with the DFSMS format (the Julian calendar format).

This parameter does not have a default.

**ICBUF(*nnn*)**

This parameter specifies the number of buffers to be used to access the output image copy data set.

The value specified for ICBUF is used for BUFNO of QSAM DCB.

***nnn***

The left-aligned number of buffers that are used to access the output image copy data set.

Applicable values can range from 0 – 255.

The default value is 50.

**ICCAT(**Y** | **N**)**

This parameter specifies whether the Image Copy function is to catalog the output image copy data set.

**Y**

Catalogs the output image copy data set.

This is the default.



## N

Does not catalog the output image copy data set.

### ICHLQ(*hlq* | *hlq1* , *hlq2*)

This parameter specifies the data set name prefix of the image copy data set.

#### *hlq*

The data set name prefix of the output image copy data set for the Image Copy function or it symbolizes a value in &ICHLQ.

#### *hlq1*

The data set name prefix of the primary output image copy data set for the Image Copy function or it symbolizes a value in &ICHLQ.

#### *hlq2*

The data set name prefix of the secondary output image copy data set for the Image Copy function or it symbolizes a value in &ICHLQ2.

**Note:** The *hlq2* parameter and ICHLQ2= cannot be specified together.

When you use the default ICOUT naming (ICNMRULE=N), specify a data set name prefix containing 33 or fewer characters. This prefix is used as the high-level qualifier of the data set name.

If you use the default STACK naming, specify a data set name prefix containing 7 or fewer characters. MVS naming convention requires that a generated data set name contain 44 or fewer characters.

This prefix must conform to the normal data set naming standards; it can include periods but must not end with a period.

If this parameter is left blank, a data set name prefix is not assigned.

### ICHLQ2(*hlq2*)

This parameter specifies the data set name prefix of the image copy data set.

You can use a data set name prefix when you allocate an output image copy data set dynamically.

#### *hlq2*

The data set name prefix of the secondary output image copy data set for the Image Copy function or sets a symbolic value for &ICHLQ2.

ICHLQ2= parameter cannot be specified with ICHLQ=(*hlq1*,*hlq2*).

When you use the default ICOUT naming (ICNMRULE=N), specify a data set name prefix containing 33 or fewer characters.

This prefix is used as the high-level qualifier of the data set name.

If you use the default STACK naming, specify a data set name prefix containing 7 or fewer characters. MVS naming convention requires that a generated data set name contain 44 or fewer characters.

This prefix must conform to the normal data set naming standards; it can include periods but must not end with a period.

If this parameter is left blank, a data set name prefix is not assigned.

### ICNMRULE(Y | N)

This parameter is used to select a naming convention for the output image copy data set.

#### Y

Selects the naming convention of the output image copy data set as follows:

```
ichlq.ICn.dbdname.ddname.Dyyddd.Thhmmss
```

#### N

Selects the naming convention of the image copy data set as follows:

```
ichlq.ICn.dbdname.ddname
```

Within these naming conventions, the following meanings apply:

***ichlq***

Is specified with the ICHLQ= keyword.

***n***

Is 1 or 2, indicating the primary or secondary copy.

***dbdname***

Is the DBD name of the database to be copied.

***ddname***

Is the DD name of the target data set group or area to be copied.

***yyddd***

Is the date.

***hhmmss***

Is the time stamp.

N is the default value.

**KEYLABEL(*key-label-name*)**

This parameter specifies the name of the key label used to encrypt the image copy output data sets.

***key-label-name***

The name of the key label to be used for allocating the encrypted image copy output data set.

The name that you define, is 1- to 64-characters long.

This parameter does not have a default.

**MGMTCLAS(*mgmt-class-name*)**

This parameter specifies the name of the management class for the new SMS managed image copy output data sets that are dynamically allocated.

***mgmt-class-name***

The name of the management class to be used for allocating the data set.

The name that you define, is one-to eight-characters long.

This parameter does not have a default.

**NOTIFY(*Y | N*)**

This parameter specifies whether the Image Copy function registers the image copy with DBRC.

***N***

Does not register the image copy with DBRC.

***Y***

Registers the image copy with DBRC.

Y is the default value.

**RETPD(*nnnn*)**

This parameter specifies the retention period, in days, for the output image copy data set.

***nnnn***

The retention period for the output image copy.

You can specify a value from 0 - 9999.

This parameter does not have a default value.

**SPACE({*CYL | TRK | blk*},*primary*[,*secondary*][,RLSE][,CONTIG | ,MXIG | ,ALX][,DYN][,ROUND])**

This parameter specifies the space parameter of the output image copy data set for dynamic allocation.

***CYL | TRK | blk***

Specifies how the Image Copy function is to allocate the output image copy data set.

***CYL***

The allocation in cylinders.

**TRK**

The allocation in tracks.

**blk**

The allocation in blocks. For example: SPACE(1024,10,10).

**primary**

The number of tracks, cylinders, or blocks to be allocated.

**secondary**

The additional number of tracks or cylinders to be allocated, if more are needed.

If this option is specified, it must be located in the third position.

**RLSE**

Requests that space allocated to an output data set but not used, is to be released when the data set is closed.

**CONTIG**

Requests that space allocated to the data set must be contiguous.

**MXIG**

Requests that space allocated to the data set must be the largest area of available contiguous space on the volume and equal to, or greater than, the primary quantity.

**ALX**

Requests that space allocated to the data set is the largest 5 areas of available contiguous space on the volume.

**DYN**

Using this option causes IMS Database Recovery Facility to determine the space attributes for the output image copy from the space attributes of the input image copy being used for the recovery.

The dynamic space calculation is determined from the number of records in the input image copy:

- If the input image copy space attributes cannot be determined dynamically, the values specified on the SPACE parameter are used.
- If only SPACE(DYN) is specified and the space attributes cannot be determined, it is treated as if the SPACE parameter were omitted.

This option is only valid when the input image copy is on DASD. Otherwise dynamic space calculation is bypassed.

When DYN is specified, allocation is in units of TRK or CYL, RLSE is always appended and ROUND is disregarded.

If the input image copy has attribute CONTIG, MXIG, or ALX and propagation of this attribute to the output image copy is not required, then code a null argument at the end of the SPACE parameter. For example:

```
SPACE (CYL,10,1,DYN,)
```

**ROUND**

When the first option specifies the average block length, this parameter requests that space allocated to the data set must be equal to an integral number of cylinders.

More detail information for each of these values is shown in the z/OS JCL reference manual.

This parameter does not have a default.

- Output Device under SMS:

The SPACE parameter in the SMS definition is used as the default.

If you specify the SPACE keyword in the ICEIN control statement or the site default table, the SMS definition is overwritten by the specified SPACE parameters.

- Non-SMS Output Device:

The SPACE keyword is a required parameter.

If you omit this keyword, dynamic allocation fails.

If you defined SPACE in the site default table, IMS High Performance Image Copy uses the SPACE parameter in the site default.

### **STORCLAS(*stor-class-name*)**

This parameter specifies the name of the storage class for the new SMS managed image copy output data sets that are dynamically allocated.

#### ***stor-class-name***

The name of a storage class to be used for allocating the data set.

The name that you define, is one-to eight-characters long.

An ACS routine can override the storage class that you specify in the STORCLAS keyword.

This keyword does not have a default.

### **UNIT(TAPE | *unitname* | *unitname,unitcount*)**

This parameter specifies the UNIT of the output image copy data set for allocation.

**Note:** When an image copy is requested, the UNIT parameter must be present either on the IC keyword or on the UTILGBL control statement; otherwise, an error results and no IC is created.

#### **TAPE | *unitname* | *unitname,unitcount***

The option for allocating the output image copy.

For example, UNIT(TAPE) or UNIT(TAPE,2).

The default value is TAPE.

### **UNIT2(*unitname* | *unitname,unitcount*)**

This parameter specifies the UNIT of the secondary output image copy data set, which is used for allocation.

#### ***unitname* or *unitname,unitcount***

The option for allocating the secondary output image copy.

For example, UNIT2(TAPE) or UNIT2(TAPE,2).

This parameter does not have a default.

If both UNIT2 and STORCLAS are specified, the value specified in the STORCLAS keyword is not applied to the secondary output image copy data set.

### **VIC(Y | N)**

This parameter specifies whether the Image Copy function registers the dummy data set name that is specified on the VICDSN parameter with DBRC as a virtual image copy.

#### **Y**

Registers the image copy with DBRC.

#### **N**

Does not register the image copy with DBRC. N is the default.

### **VOLCNT(1 | *nnn*)**

This parameter specifies the number of volumes used for the output image copy data set.

#### ***nnn***

The maximum number of volumes that an output image copy data set requires.

The volume count is a decimal number from 1 through 255 for a tape data set and from 1 through 59 for a DASD data set.

If you omit VOLCNT, 1 is used as the default.

### **VOLSER(*volsern* | *volsern, volsern...*)**

This parameter specifies the volume serial number of the volume used for the primary data set of the output image copy.

**volser or (volser1, volser2,...)**

The volume serial number of the volume allocated to the primary data set of the output image copy.

This parameter does not have a default.

**VOLSER2(volsern | volsern, volsern...)**

This parameter specifies the volume serial number of the volume used for the secondary data set of the output image copy.

**volser or (volser1, volser2,...)**

The volume serial number of the volume allocated to the secondary data set of the output image copy.

This parameter does not have a default.

**Unsupported keywords**

The following IMS High Performance Image Copy keywords are not supported by IMS Database Recovery Facility:

- AREA | DBD=
- CAGRP=
- DBBUF=
- DBDSGRP=
- DSN3-7=
- CRCTYPE=
- DBD=
- DBDALLOC=
- DDN=
- DEDBPC=
- DEDBEFCTL
- FUNC=
- HDPC=
- ICHLQ3-7=
- ICOUT=
- PART=

**Important:** ICOUT=\* is implied if you are requesting one image copy. If you are requesting two image copies, then ICOUT=\*,\* is implied.

**Example of using the ADD IC() command**

```
XCFGROUP(TOIREORG)
SORTPARM(ASPREF(DR3S))
UTILGBL(IBPREF(DR3I),PRPREF(DR3P),PCPROCNM(FRXPAT0),PCJOBNM(DR3C))
REPORT(RPTTYPE=APP,DRFUNIT=SYSDA,DRFHLQ=h1q)
DRFIAX(FRXJCLIP)
LIU@GOPT(XXXX)
DBDSN101(%P(duphlq))
DBDSL101(DISP(NEW),STGC(storclas),VOL(volser))
ADD DB(HIDAM1) DBATRB(DBDSL(101),DBDSN(101)), -
  IC(ICHLQ(ichlq),SPACE(CYL,3,1),ICNMRULE(Y),ICCAT(Y), -
    DSNTYPE(BASIC),COMP(N), -
    STORCLAS(storcl),MGMTCLAS(mgmtcl),DATACLAS(datacl),UNIT(unit))
START ERROR(CONT) DBRCMD(GLOBAL) STACMD(GLOBAL) READNUM(5)
```

## Reference: ADD PC() command

This topic describes the syntax and use of the ADD PC() command for invoking IMS High Performance Pointer Checker and IMS High Performance Fast Path Utilities .

Any parameters specified on the ADD PC() command override those parameters that were specified on the UTILGBL() control statement.

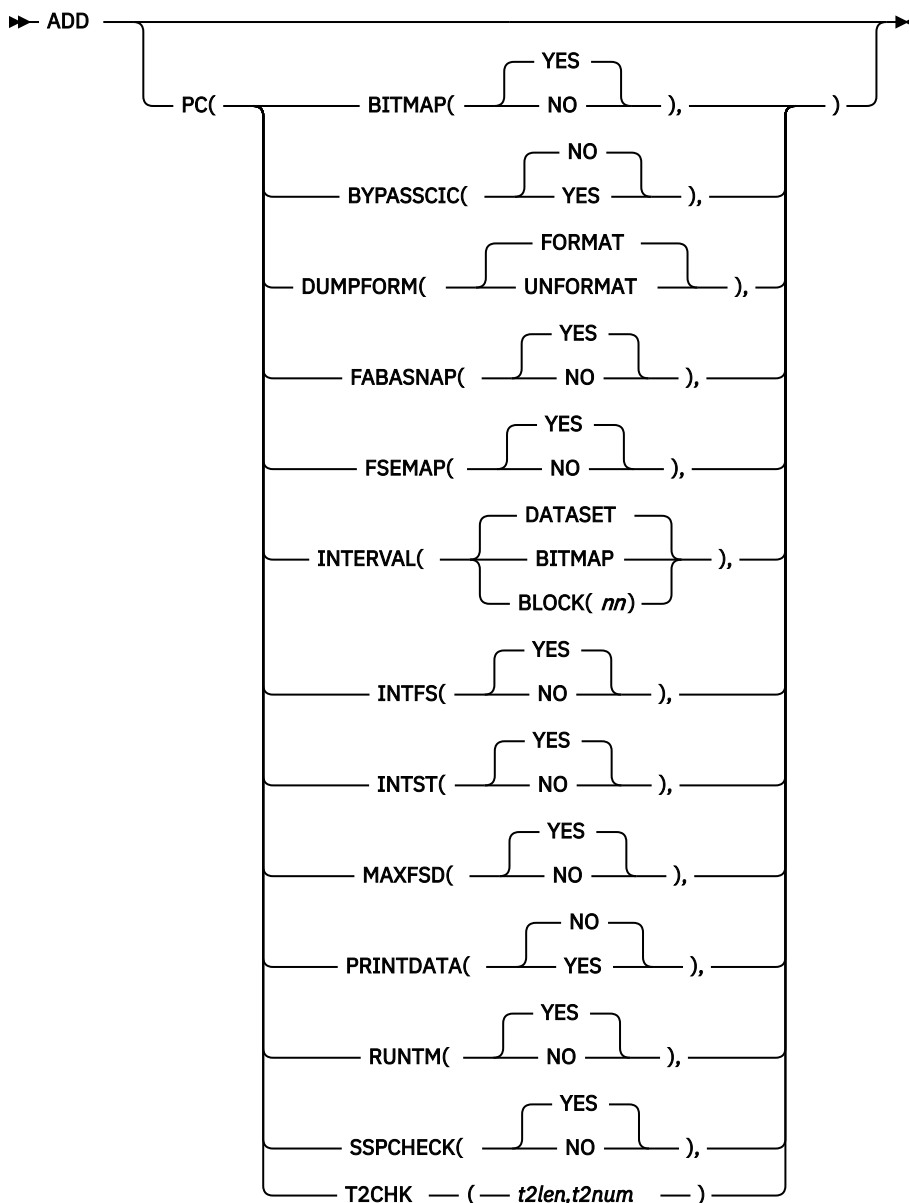
There are also parameters for this auxiliary utility that can only be specified on the UTILGBL() control statement.

For a list and description of those parameters, refer to [“Reference: Utility control statements” on page 69.](#)

### Syntax for ADD PC() command

The basic format of the ADD command is described in [“Parameters for ADD command” on page 106.](#)

This reference shows the syntax for those parameters that are associated with the IMS High Performance Pointer Checker and IMS High Performance Fast Path Utilities usage.



## Parameter reference for the ADD PC() command

The use of the PC() keyword on the ADD command indicates that the pointer checker function of either IMS High Performance Pointer Checker or IMS High Performance Fast Path Utilities can be run for your recovered database data sets.

This processing is limited to the single-step hash-pointer checking feature. Therefore, many of the keyword values for these products are assigned by IMS Database Recovery Facility, as implied by this limited function, and passed to the products.

Note also that because the databases to be pointer checked are identified on the ADD DB control statement, the pointer checker products always use dynamic allocation for their data sets, and thus no DD Names are required in the JCL.

**Note:** The PC options that are listed in this keyword description and listed in the syntax diagram are described in more detail in the *IMS High Performance Pointer Checker for z/OS User's Guide*.

**Note:** For HALDB databases, neither the primary index nor the secondary index is checked by the pointer checker through IMS Index Builder.

The following options are from the pointer checker PROC command in the PROCCTL SYSIN control statements and are fixed from the IMS Database Recovery Facility perspective. They are passed with the following default values:

### **BITMAP(YES | NO)**

This parameter specifies whether you want to generate the **Bit Map Display** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

This option can be specified when TYPE=ALL or SCAN is also specified.

#### **YES**

The report is generated.

This is the default value.

#### **NO**

The report is not generated.

### **BYPASSCIC(YES | NO)**

This parameter specifies whether you want to bypass IMS HP Pointer Checker hash checking, when OUTPUT(ICR) is specified and the image copy that is created is concurrent.

This parameter is ignored if OUTPUT(ICR) is not specified or if IMS HP Pointer Checker is not requested.

#### **YES**

The IMS HP Pointer Checker processing is bypassed.

#### **NO**

The IMS HP Pointer Checker is not bypassed.

This is the default value.

### **DUMPFORM(UNFORMAT | FORMAT)**

This parameter specifies the dump format that you want to print block dumps with.

This parameter can be specified when DIAGDUMP=FIRST100 in the OPTION is also specified.

This parameter can be specified with any TYPE= specification.

#### **FORMAT**

This option specifies that the formatted dumps are to be printed.

This is the default value.

#### **UNFORMAT**

This option specifies that the unformatted dumps are to be printed.

**FABASNAP(YES | NO)**

This parameter indicates whether you request the **FABASNAP** report when using DEDB Pointer Checker.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**FSEMAP(YES | NO)**

This parameter specifies whether you want to generate the **Free Space Map** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

This parameter can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**INTERVAL(DATASET | BITMAP | BLOCK(nn))**

This parameter specifies whether you want to define the interval at which the **Interval Statistics** report and the **Interval Free Space Summary** report are produced.

This parameter can be specified when TYPE=ALL or SCAN is specified on the PROC statement, and when INTFS=YES or INTST=YES is specified on the REPORT statement.

It is effective only for HDAM, HIDAM, PHDAM, or PHIDAM databases.

**DATASET**

The reports are produced for the entire database data set only once.

This is the default value.

**BITMAP**

The reports are produced each time a bitmap block is processed.

**BLOCK(nn)**

The number represented by *nn* times 100, is the number of blocks that is processed between statistics intervals.

To code this field, you must include two integer digits. Use leading zeros, if necessary.

**INTFS(YES | NO)**

This parameter specifies whether you want to generate the **Interval Free Space Summary** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

The report is produced each time an interval is processed and the information in the report is added to the next report. For example, the *N*th report provides the total information of the 1st through *N*th reports.

This parameter can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**INTST(YES | NO)**

This parameter specifies whether you want to generate the **Interval Statistics** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.



The report is produced each time an interval is processed; the information in the report is added to the next report. For example, the *N*th report provides the total information of the 1st through *N*th reports.

This parameter can be specified when TYPE=ALL or SCAN is specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**MAXFSD(YES | NO)**

This parameter specifies whether you want to generate the **Maximum Free Space Distribution** report for the HDAM, HIDAM, PHDAM, or PHIDAM database.

This parameter can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**PRINTDATA(NO | YES)**

This parameter specifies whether you want to print the pointer data that is extracted by the program.

If you specify YES, you might get a large report that is of little use.

This parameter can be specified when TYPE=ALL or SCAN is also specified.

**Note:** This parameter must be used for debugging purpose only.

**NO**

Any extracted pointer data is not printed.

This is the default value.

**YES**

The extracted pointer data is printed.

**RUNTM(YES | NO)**

This parameter specifies whether you want to generate the separator page for DB/DSG reports with Run Time Option.

This parameter can be specified when TYPE=ALL or SCAN is also specified.

**YES**

The report is generated.

This is the default value.

**NO**

The report is not generated.

**SSPCHECK(YES | NO)**

This parameter specifies whether you want to perform subset pointer checking during the hash check process.

**YES**

Subset pointer checking is performed.

**NO**

Subset pointer checking is not performed.

This is the default value.

### **T2CHK(t2len,t2num)**

Use the T2CHK parameter to ignore the short and known T2 errors that are not really errors in HISAM, HDAM, HIDAM, PHDAM, and PHIDAM databases.

This parameter requires the following values:

#### **t2len**

The minimum T2 record length to be reported (for HDAM, HIDAM, PHDAM, and PHIDAM).

The T2 records whose lengths are shorter than the specified minimum value are not to be reported.

This setting allows you to ignore short T2s that might not be actual errors.

#### **t2num**

The maximum number of T2 records not to be reported (for HISAM, HDAM, HIDAM, PHDAM, and PHIDAM).

By specifying a T2 record threshold value for this specification, the known T2 records are not reported.

This setting allows you to ignore known T2s that might not be actual errors, but are reported until the database is reorganized.

### **Example of using the ADD PC() command**

The following example instructs IMS Database Recovery Facility to perform pointer checking by using IMS High Performance Pointer Checker with diagnosis and dump formatting.

```
ADD DB(PAYROLL) -  
  PC(INTERVAL(DATASET),RUNTM(NO),INTST(YES),BITMAP(YES), -  
  FSEMAP(YES),MAXFSD(NO),INTFS(NO),FABASNAP(NO))
```

The following reports are generated:

- Interval Statistics Report
- Bit Map Display Report
- Free Space Element Map Report

## **Return code handling**

---

An optional input data set FRXRETCD allows you to specify the return code from auxiliary utilities run under IMS Database Recovery Facility.

The FRXRETCD data set can be used to override the return code values, nn = 0-16, from the following utilities:

- IMS High Performance Pointer Checker (PCERROR=)
- IMS Index Builder (IBERROR=)
- IMS High Performance Pointer Checker (DEDB) (DPERROR=)
- IMS High Performance Image Copy (ICERROR=)
- DFSPRECO (PRERROR=)

The return codes *nn* must be two-digit number between 0 and 16.

The functionality is similar to the HPSRETCD data set provided in IMS High Performance Image Copy and IMS High Performance Pointer Checker. The keywords used by FRXRETCD are unique to IMS Database Recovery Facility.

The IMS Database Recovery Facility final return code is based on two criteria:

- Any recovery-related errors produce RC equal to or greater than 0.

- If there are no recovery-related errors, the final return code is set to the larger value of either the IMS Database Recovery Facility return code or the integrated auxiliary utility return code.

For example, if IMS Database Recovery Facility returns RC=8 and the IMS Index Builder auxiliary utility returns RC=12, the IMS Database Recovery Facility final return code becomes 12.

### FRXRETCD data set format

The FRXRETCD data set contains your specifications for the return codes that are returned from the integrated auxiliary utilities.

The following conditions apply to the FRXRETCD data set:

- The function is disabled if the FRXRETCD data set is not present in the JCL, or if the data set is specified as DD DUMMY.
- The control data set can be specified as part of the input stream, a sequential data set, or a member of a partitioned data set.
- The data set must contain 80 byte fixed length records. BLKSIZE is optional. If BLKSIZE is used, it must be specified as a multiple of 80.

The FRCRETCD data set contains (DRF) statements using the following format:

```
//FRXRETCD DD *
(DRF)
PCERROR=nn
IBERROR=nn
DPERROR=nn
ICERROR=nn
PRERROR=nn
/*
```

*Figure 3. Control statement format for the FRCRETCD data set*

These statements must be coded as shown in the following example:

```
//FRXRETCD DD *
(DRF)
PCERROR=4,
IBERROR=8
```

*Figure 4. Example control statement format for the FRCRETCD data set*

### Control statement format conventions

Follow these format conventions when you write control statements in the FRXRETCD data set:

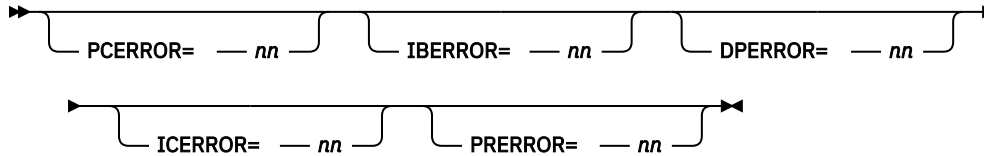
- You must code an (DRF) statement in the first line of the FRXRETCD data set, and you must code keywords and their values on the second or subsequent lines.
- You must code the (DRF) statement, keywords, and their values within columns 1 and 72.
- When you code multiple keywords, they can be separated by commas or a new line.
- Keywords are not positional; you can specify them in any order of sequence. You cannot specify a null value for any keyword.
- Comments can follow the last keyword value on each control statement record, separated by at least one blank.
- A comment statement must begin with an asterisk in column 1.
- Comments can follow the last keyword value on each control statement record, separated by at least one blank.
- Comments can be located right after the value (without any blanks) or on a new line, but they should start with a forward slash and an asterisk, not just with an asterisk.

### (DRF) statement syntax

The (DRF) statement must contain optional parameters that are specified by the keywords shown in the following syntax diagram.

➡ (DRF) ⬅

#### Optional keywords



#### Keyword reference

##### PCERROR=

Specify the return code when the PC (IMS High Performance Pointer Checker) error is detected.

The value nn is from 0 to 16.

##### IBERROR=

Specify the return code when the IB (IMS Index Builder) error is detected.

The value nn is from 0 to 16.

##### DPERROR=

Specify the return code when the DP (IMS High Performance Pointer Checker (DEDB)) error is detected.

The value nn is from 0 to 16.

##### ICERROR=

Specify the return code when the IC (IMS High Performance Image Copy) error is detected.

The value nn is from 0 to 16.

##### PRERROR=

Specify the return code when the PR (DFS PRECO) error is detected.

The value nn is from 0 to 16.

---

## Chapter 9. Reference: DBRC commands for database recovery

This information describes some of the IMS DBRC commands that can be useful when using IMS Database Recovery Facility.

### Topics:

- [“Using the INIT.DBDSGRP command” on page 147](#)
- [“Using the CHANGE.DBDSGRP command” on page 147](#)
- [“Using the LIST.DBDSGRP command” on page 148](#)
- [“Using the NOTIFY.RECOV command” on page 148](#)

---

### Using the INIT.DBDSGRP command

IMS provides the keyword, RECOVGRP for the INIT.DBDSGRP DBRC command. The RECOVGRP indicates that a database group is a recovery group.

A recovery group is a group of full-function databases, HALDB partitions, or DEDB areas that you consider to be related for recovery purposes.

If you use the IMS Database Recovery Facility to perform a time stamp recovery (TSR) on any member of a group, you must recover all the members of the group to an equivalent time.

In the following example, a group of DBDSs is defined:

```
//INITDBGRP      JOB
//S01 EXEC PGM=DSPURX00,...
.
.
.
//SYSIN          DD *
      INIT.DBDSGRP      GRPNAME(DBDSG1)      -
                        RECOVGRP(DB1,DB2,DB3)
/*
```

---

### Using the CHANGE.DBDSGRP command

IMS provides the parameters, ADDRECOV and DELRECOV for the CHANGE.DBDSGRP DBRC command.

Use ADDRECOV to identify one or more full-function databases, HALDB partitions, and DEDB areas to add to a recovery group.

Use DELRECOV to identify one or more full-function databases, HALDB partitions, and DEDB areas to delete from a recovery group.

In the following example, the first CHANGE.DBDSGRP command adds members to the recovery group and the second command deletes members from the recovery group:

```
//CHANGE1 JOB
//S01 EXEC PGM=DSPURX00,...
.
.
.
//SYSIN DD *
      CHANGE.DBDSGRP  GRPNAME(DBDSG1)      -
                        ADDRECOV(DB4,DB5)
      CHANGE.DBDSGRP  GRPNAME(DBDSG1)      -
                        DELRECOV(DB1,DB2)
/*
```

## Using the LIST.DBDSGRP command

---

Use the DBRC LIST.DBDSGRP command to display the recovery group for a database.

In the following example, the members of a specified DBDS group are displayed:

```
//LISTDBGP      JOB
//S01 EXEC PGM=DSPURX00,...
.
.
.
//SYSIN        DD *
LIST.DBDSGRP   GRPNAME(DBDSG1)
/*
```

## Using the NOTIFY.RECOV command

---

Use the DBRC NOTIFY.RECOV command to add information about recovery of a specified DBDS or DEDB area to the RECON data set.

NOTIFY.RECOV provides an optional parameter called point-in-time recovery (PITR) on the NOTIFY.RECOV DBRC command. If you use the PITR parameter, you must also use the RCVTIME parameter.

In the following example, information about recovery of a specified DBDS will be added to the RECON data set.

- The RUNTIME parameter specifies the time stamp of the recovery of the DBDS.
- The RCVTIME parameter specifies the time stamp to which the specified DBDS was recovered.

The PITR parameter specifies a point-in-time recovery.

```
//NFYRECOV     JOB
//S01 EXEC PGM=DSPURX00,...
.
.
.
//SYSIN        DD *
NOTIFY.RECOV   DBD(DB1)   DDN(DDN1) -
                RUNTIME(202671015366) -
                RCVTIME(202670905297) -
                PITR
/*
```

**Note:** The NOTIFY.RECOV command does not need to be used when performing database recoveries using IMS Database Recovery Facility. Recovery information will be updated after successful recovery.

---

# Chapter 10. Performing a database recovery

This information describes the steps required to perform a database recovery using the IMS Database Recovery Facility.

This information does not describe all features but is meant to provide a basic understanding of the IMS Database Recovery Facility recovery process.

## Topics:

- [“1. Defining recovery scope” on page 149](#)
- [“2. Taking databases offline” on page 151](#)
- [“3. Building a recovery list” on page 151](#)
- [“4. Defining physical data sets” on page 152](#)
- [“5. Starting recovery” on page 152](#)
- [“6. Using integrated auxiliary utility processing” on page 152](#)
- [“7. Performing post-recovery activities” on page 154](#)

---

## 1. Defining recovery scope

The first step in the recovery process is to define the scope of recovery.

This step is important because it allows you to clearly define which database data sets are recovered and under what conditions.

The IMS Database Recovery Facility obtains the definition of the following data sets from DBRC and the system catalog. Access to these data sets is obtained through dynamic allocation. These data sets are not specified in the IMS Database Recovery Facility JCL:

- Log data sets
- Change accumulation data sets
- Image copy data sets
- Database data sets
- Area data sets

Using IMS Database Recovery Facility to recover full-function databases, HALDB partitions, and DEDB areas depends upon recovery information that is recorded in the RECON data sets. Specify the database data sets you need to recover by using the **ADD** command and its keyword parameters. On the **ADD** command, you can specify:

- Database data sets or areas
- Databases (you can specify both full-function or DEDB on the same command)
- One of the following groups as defined in RECON:
  - CAGRP (change accumulation groups)
  - DBDSGRP (database data set groups)
  - RECOVGRP (recovery groups)

### Full function databases

For full function databases, most recoveries include all of the related databases. These could be IMS logically-related databases, primary indexes, and secondary indexes.

If the IMS Database Recovery Facility has dependencies between updates to different databases, you can also include these in the recovery process. Many installations use DBDS groups to predefine sets of related databases.

### **Fast Path areas**

You can recover a Fast Path area independently of the other areas. Multiple areas can be recovered concurrently.

You can recover a single area while the rest of the areas remain available for online processing.

### **HALDB**

A HALDB (high availability large database) is a partitioned database. You can recover each partition independent of the others. You can recover the entire database with individual database data sets being recovered concurrently for all partitions.

If you only need to recover one partition, the other partitions can remain available for application processing.

### **Individual database or database data set**

You can use the IMS Database Recovery Facility to recover any database data set, group, or area that can be recovered by the IMS Database Recovery utility.

### **RECON groups**

The IMS Database Recovery Facility can use any of these database groups that are defined in the RECON data set:

- DBDSGRP (database data set group)
- CAGRP (change accumulation group)
- RECOVGRP (recovery group)

If you issue the **RECOVER** command with a group name, every database data set included in the group is recovered. If you include multiple groups in the recovery list, a DBDS is included only once.

### **DBD names**

Using the primary DBD name causes all database data sets within all partitions of the primary database to be included in the recovery. Using the partition name causes only those database data sets in that partition to be included in the recovery.

Neither of these usage scenarios will include a primary index or ILE data set in the recovery, nor will primary indexes or ILE data sets need to be rebuilt until either a media failure has occurred or time stamp recovery is performed.

If a media failure occurs or you perform a time stamp recovery, you must rebuild the primary indexes or ILE data sets from the primary data sets. Use the Index/ILDS Rebuild utility (DFSPREC0) to do so.

### **Secondary indexes**

Using the secondary index DBD name causes the recovery of the entire secondary index database if it is recoverable. If the secondary index database is partitioned, you can use the partition name to recover only that partition.

Unless a secondary index is partitioned to match the partitioning of the target database, the secondary index will have target segments in more than one partition of the main database.

Use time stamp recovery on the target database and the secondary index database with care to ensure the integrity of the indirect pointers.

### **Recovery group**

You can use recovery groups to define a complete set of databases to be recovered together.

The IMS Database Recovery Facility indicates when a recovery is completed for one of the DBDSs within the group, if all of the other DBDSs within the group were also recovered within the same the IMS Database Recovery Facility recovery list.

### **Nonrecoverable databases**

You cannot recover databases that are marked as nonrecoverable in the RECON with the IMS Database Recovery Facility unless an image copy for the DBDs in the database is registered in the RECON data set.

Then you can use the IMS Database Recovery Facility to recover the image copy, including the secondary indexes.



You can use the IMS Database Recovery Facility to ensure that the databases comprising the predefined group of databases, called a recovery group, have been consistently recovered.

## 2. Taking databases offline

---

In order to recover databases with the IMS Database Recovery Facility, you must first take the databases offline.

Ensure that the following conditions are met:

- Ensure that the databases are not authorized to any subsystem.
- Ensure that the databases, area, or partition are in a stopped state because of a **DBR** command or the normal stop of an IMS subsystem.
- Ensure that all of the OLDSs containing data for the recovery are archived.

If the database was online to more than one IMS system, all archive jobs must complete before can recover the database.

By using the DBRCMD keyword on the START command, you indicate that the IMS Database Recovery Facility will issue the /DBR commands necessary to being all databases in the recovery list offline.

You can specify the DBRCMD parameter on the START command in order to allow the IMS Database Recovery Facility to take all databases in the recovery list offline. You can also do this manually.

### Examples: Manually taking databases offline

- Take the database offline by issuing the IMS **DBR DB** command. For example:

```
/DBR DB xxx
```

- Take a DEDB area offline by issuing the IMS **DBR AREA** command. For example:

```
/DBR AREA xxxx
```

- If you use the primary name in the **DBR** command for the database, it must be started.

You must issue a separate **DBR** command for each partition that is to be recovered. For example:

```
/DBR DB partition name
```

- You can specify a database group name in the **DBR DATAGROUP** command to identify the list of databases that must be taken offline before recovery.

If a HALDB needs recovery, you can create a database group containing all the HALDB partitions to allow a single operand on the **DBR** command. For example:

```
/DBR DATAGROUP groupname
```

## 3. Building a recovery list

---

Use the **ADD** and **REMOVE** commands to build and manipulate your list of data sets to be recovered.

Although multiple ADD and REMOVE control statements can be used with SYSIN, only a single recovery list is created.

Defining the scope of the recovery means determining:

- Which databases, areas, partitions, and database groups need to be recovered.
- Which kind of recovery is to be run:
  - Full recovery.
  - Time stamp recovery.
  - Point-in-time recovery.

- The type of checking to be done:
  - Ensure that an entire recovery group is recovered.
  - No checking is done.

## 4. Defining physical data sets

---

You can automatically delete and redefine database data sets as part of the recovery process by using data set definitions from either the original data set characteristics or the environmental control statement data set characteristics specification. You should first try to use the original data set characteristics to do so.

Use DISP(OLD | NEW) on the DBDSL $nnn$  keyword to identify whether your database data sets will be deleted and redefined. Use the DATRB() keyword on the ADD command to specify the attributes for deleting and redefining database data sets.

If you do not want to use automatic **DELETE** and **DEFINE**, but you do not want to reallocate data sets, you must delete and define physical data sets outside of IMS Database Recovery Facility.

When you do not need to change the data set attributes or location for OSAM data sets, those data sets can simply be overwritten without your having to delete and reallocate them.

The IMS Database Recovery Facility uses the VSAM REUSE parameter when you specify the **DEFINE CLUSTER** command. If the database has been allocated with the VSAM REUSE parameter on the **DEFINE CLUSTER** command, the physical data sets need not be empty in order for the recovery to work.

For VSAM data sets, if you want recovery to run without deleting and redefining the VSAM clusters, you must specify the REUSE parameter.

There are many reasons that you might want to reallocate data sets. For example:

- Changing SPACE parameters.
- Changing VOLUME parameters.
- Changing FREESPACE parameters.

You also need to delete and reallocate if you want to change any of the physical characteristics of the data sets.

## 5. Starting recovery

---

References to information on starting the recovery process by using IMS Database Recovery Facility are described here.

Starting recovery with the IMS Database Recovery Facility is described in detail in:

- [Chapter 7, “Using batch recovery control statements,” on page 105](#)

## 6. Using integrated auxiliary utility processing

---

IMS Database Recovery Facility has a one-button recovery solution that allows you to define recovery and post-recovery tasks and to submit one job to perform these tasks.

Integrated auxiliary utility processing allows you to use the DB Control Suite interface or batch JCL and control statements to tell IMS Database Recovery Facility that the following four processes (or a combination of them) are to be performed as part of the database recovery, rather than being performed in separate job steps after the recovery is complete:

- Image Copy generation by using IMS High Performance Image Copy.
- Hash Pointer Checking for a Full Function DBDS using IMS High Performance Pointer Checker or for a Fast Path DEDB by using IMS Fast Path Basic Tools - DEDB Pointer Checker.
- For HALDB, after recovery completes, prime index (PHIDAM) is rebuilt for each partition recovered by IMS Database Recovery Facility using DFSPRECO (which is a part of the IMS base product).

- Full Function primary index rebuild, and HALDB or Full Function secondary index rebuild using IMS Index Builder.
- Fast Path secondary index rebuild using the Build Index function of FPA.

For example, an IC(*options*) keyword is added to the IMS Database Recovery Facility **ADD** control statement indicating that an image copy of the database is to be performed as the database is being recovered. IMS Database Recovery Facility is responsible for communicating to IMS High Performance Image Copy all the data sets and options necessary to create image copies.

Invoke integrated auxiliary processing by supplying control cards in the SYSIN DD statement and by supplying the appropriate DDNAMEs in the IMS Database Recovery Facility primary and subordinate address space procedures.

## Calling IMS HP Image Copy

Use the IMS Database Recovery Facility **ADD** control statement IC keyword to indicate that the image copy function is required which will result in an IC of the recovered DBDS or of each of the DBDSs in a group (CAGROUP or RECOVGRP).

Only IMS HP Image Copy output is generated.

The following output is not generated:

- IMS Standard Image Copy Utility output.
- IMS Online Image Copy Utility output.
- IMS Image Copy 2 Utility output.
- IMS HPIC Flash Copy output.

## Calling IMS HP Pointer Checker

Use the IMS Database Recovery Facility **ADD** control statement PC keyword to indicate that you require the single step hash pointer checking function.

HALDB primary indexes and ILDS' are not hash checked by IMS Database Recovery Facility because IMS HP Change Accumulation Utility does not allow for hash pointer checking on HALDB primary indexes and ILDSs. Therefore, IMS HP Change Accumulation Utility is not called for primary indexes and ILDSs after DFSPRECO rebuilds them.

If you want pointer checking by IMS HP Change Accumulation Utility done, ensure that all logically-related data sets are available to IMS Database Recovery Facility by specifying the PC() keyword on the corresponding **ADD** commands.

If the logically-related data sets are not available, then IMS HP Change Accumulation Utility returns multiple pointer errors to the IMS Database Recovery Facility primary address space. This can be acceptable if, for example, you only want the Free Space Report from IMS HP Change Accumulation Utility.

## Calling IMS Index Builder or DFSPRECO

Use the IMS Database Recovery Facility **ADD** control statement IB keyword to indicate that the Build Index function is required.

For primary and secondary indexes, index build processing is run after the database has been recovered because the entire database is required in order to build the indexes. Index build (IB) processing cannot occur while each individual data set in the database is being recovered.

For HALDB databases, IB processing consists of first calling DFSPRECO to rebuild the prime index and/or the ILDS. For Full Function databases, IB is called to rebuild both the prime and the secondary indexes. For Fast Path DEDB databases, the Build Index function of FPA is called to build secondary indexes.

You need only rebuild indexes if the media containing the index fails or if a database is recovered to a particular time stamp (either PITR or non-PITR). This means that index data sets do not need to be rebuilt for all recoveries.

Index data sets are available to be image copied only if the index data sets are marked recoverable in the RECON. Image copies are not taken of index data sets that are marked non-recoverable.

## Message and report output handling

Write to Operator (WTO) messages generated by IMS Index Builder, DFSPREC0, IMS High Performance Image Copy, IMS High Performance Pointer Checker, DEDB Pointer Checker, and the Build Index function of FPA in the subordinate address spaces are collected from the IMS Database Recovery Facility master address space.

If **REPORT(SEP)** is specified, then all messages from utilities run in IMS Database Recovery Facility subordinate address spaces are written to FRXWTO in the IMS Database Recovery Facility master address space.

Both the REPORT and FRXWTO files in the IMS Database Recovery Facility master address space separate the messages by DBDname and DD areaname and also indicate the job name of the IMS Database Recovery Facility subordinate address space that created them.

IMS Database Recovery Facility gathers all utilities' reports from the IMS Database Recovery Facility subordinate address spaces in the IMS Database Recovery Facility master address space and distributes them as you request.

The reports are written to various output DDs in the master address space so that you can distinguish between IMS Database Recovery Facility and auxiliary utility report output.

The standard utility report DD names are added to the master address space JCL to contain any separate report data.

The IMS Database Recovery Facility REPORT file indicates which auxiliary processes were requested and driven.

**Tip:** You can specify DD DUMMY to suppress the IMS High Performance Pointer Checker reports and get only a return code only if there are pointer errors. For example:

```
//*   HPPC   reports
//PRIMAPRT DD DUMMY
//EVALUPRT DD DUMMY
//SUMMARY  DD DUMMY
//STATIPRT DD DUMMY
//VALIDPRT DD DUMMY
//SNAPPIT  DD DUMMY
//*
//*   DEDBPC reports
//FABARPRT DD DUMMY
//FABAMSG  DD DUMMY
//FABASNAP DD DUMMY
```

## 7. Performing post-recovery activities

You might not want IMS Database Recovery Facility to call the tools and utilities required for post-recovery.

After you run recovery with the IMS Database Recovery Facility, you might need to perform the activities described in the following list.

- Rebuild HALDB data sets.

After a time stamp recovery, you must rebuild the ILDS data sets and a PHIDAM primary index data set. Use the HALDB Index/ILDS Rebuild Utility (DFSPREC0) to do so.

If you have already run a full recovery on the ILDS and index data sets, you need not perform the rebuild. This can be done as part of recovery by using the IB keyword on the ADD statement.

- Rebuild index data sets.

Index data sets can be either recoverable or nonrecoverable. You can recover the recoverable index data sets as part of the IMS Database Recovery Facility recovery process by adding them to the recovery list. Alternatively, you can rebuild them after recovery is complete with an index rebuild utility.

You might need to rebuild nonrecoverable index data sets. If you have performed a time stamp recovery, you must rebuild the index data sets.

If you have not performed a time stamp recovery and if the media that the index data sets are on did not experience a failure, then you might not need to rebuild the index data sets. Rebuilding index data set can be done by using the IB keyword on the ADD statement.

- Run pointer checker.

Optionally, you can ensure that the recovery has completed its specific task by running a pointer checker to validate the internal IMS pointers.

If you normally run a pointer checker in the image copy process, then you might be able to omit this step.

You can run the pointer checker function by specifying the PC keyword on the ADD statement.

- Run image copies.

Running an image copy for those databases which have just been recovered resets the recovery point and averts the need for a more complicated recovery, if one is needed in the near future.

If you used PITR on a time stamp recovery, the IMAGE COPY NEEDED flag has been set on. The databases will not be available to authorize for update processing until the image copy is done.

**Note:** Resetting the IMAGE COPY NEEDED flag after a PITR recovery is not recommended because the PITR recovery processed only some records on a log file and subsequent recoveries will not manage those log files correctly. You can run the image copy function by using IMS High Performance Image Copy by specifying the IC keyword on the ADD statement.



---

## Part 4. Administration and management

These topics provide information you need in order to administer IMS Database Recovery Facility, and to manage performance and time stamp recovery.

**Topics:**

- [Chapter 11, “Administering IMS Database Recovery Facility,” on page 159](#)
- [Chapter 12, “Managing performance,” on page 165](#)
- [Chapter 13, “Managing time stamp recovery,” on page 169](#)





---

# Chapter 11. Administering IMS Database Recovery Facility

This information describes the tasks required to administer IMS Database Recovery Facility.

Administration tasks include managing inputs, outputs, environment, performance, and recovery points.

## Topics:

- [“Database administration checklist” on page 159](#)
- [“Managing sources of input” on page 159](#)
- [“Managing output” on page 162](#)
- [“Managing the environment” on page 162](#)

---

## Database administration checklist

This topic describe the tasks that are required to administer IMS Database Recovery Facility databases.

As the person responsible for administering databases, you must regulate the following tasks:

### Managing inputs:

Ensure that log, change accumulation, and image copy data sets are available.

### Managing outputs:

Ensure that database recovery tools are capable of:

- Performing an image copy of a database while it is being recovered.
- Performing an ICR (incremental image copy), in which an updated image copy data set is generated from a previous image copy data set.

### Managing the environment:

- Provide new JCL and PROCLIB members to run the IMS Database Recovery Facility.
- **Optional:** Delete and redefine data sets before you run the IMS Database Recovery Facility.

Specifying `DBDSLmmm (DISP(NEW))` causes database data sets and area data sets to be deleted and redefined as part of the recovery process.

The person that administers databases typically establishes procedures to control deleting and redefining data sets.

### Managing performance:

Performance of each recovery depends on your execution environment at the time of the recovery.

---

## Managing sources of input

The topics in this section discuss how you can manage IMS Database Recovery Facility input data and control data.

## Topics:

- [“Input data sets from earlier product releases” on page 160](#)
- [“DBRC controls” on page 160](#)
- [“Image copy input types” on page 160](#)
- [“Recovering with a nonstandard image copy” on page 160](#)
- [“Change accumulation input data sets” on page 161](#)
- [“Log data sets usage for recovery” on page 161](#)
- [“Image copy frequency” on page 162](#)

- [“Recovery from secondary sources” on page 162](#)

## Input data sets from earlier product releases

IMS Database Recovery Facility processes image copies, logs, and change accumulation input that is generated under any currently-supported IMS release.

It is also possible to have IMS Database Recovery Facility restore from image copies that are generated under prior IMS releases. This might be useful in cases where you maintain an archive of old image copies that have not yet been updated to the latest IMS release type.

In order to be able to restore from these image copies, the following conditions must be met:

- The database must be registered with DBRC.
- The image copy must be registered with DBRC.
- No logs or change accumulations can be supplied to recovery.
- No ALLOC entries can be registered under the same database that is being restored from the earlier IMS release image copy.

## DBRC controls

IMS Database Recovery Facility uses recovery information that is recorded in the RECON. Therefore, only resources that are registered to DBRC can be used for database recovery by IMS Database Recovery Facility.

For resources that were migrated to different volumes than those registered in the RECON, DBRC determines the correct locations of the resources by doing a search of the system catalog.

DBRC restricts access to database data sets and areas that are undergoing recovery by granting exclusive authorization to IMS Database Recovery Facility.

DBRC performs the following functions for IMS Database Recovery Facility:

- Determines the resources that are required for recovery
- Maintains the status of database data sets and areas that are being processed by IMS Database Recovery Facility

## Image copy input types

You can use standard image copy types as input to IMS Database Recovery Facility.

These types include:

- Standard IMS image copy (both batch and concurrent image copy)
- Image copy 2 generated image copies
- Image copies that are generated by the IMS High Performance Image Copy tool
- OLIC (online image copy) data sets
- Image copies that you supply (User Image Copy)
- Restored image copies

You can stack image copies by using the SAMEDS (same data set) option of image copy 2.

## Recovering with a nonstandard image copy

You can recover database data sets and areas even if a nonstandard image copy, such as a user image copy, is restored. However, the nonstandard image copies are not used as input for IMS Database Recovery Facility.

### Before you begin

Concurrent nonstandard image copies must be registered to DBRC with the correct USID. See the DBRC **NOTIFY.UIC** command. If the USID is incorrect, the recovery will be faulty. The correct USID is located in

the RECON ALLOC record that corresponds to the time that the concurrent nonstandard image copy was taken.

### About this task

You can recover with nonstandard image copies by using the USEDDBDS, USEAREA, USEUICLAST, or USEUICTIME parameters for the **ADD** command. You must restore the data set with the nonstandard image copy before you recover. You can automatically restore any standard or nonstandard image copy that is registered to DBRC as an IMS image copy.

### Procedure

1. Restore the database data set or area from the nonstandard image copy.
2. Depending on the parameter that you use for the **ADD** command, you might need to use the DBRC **NOTIFY.RECOV** command:
  - If you are recovering by using the USEDDBDS or USEAREA parameters, record the restoration by using a DBRC **NOTIFY.RECOV** command. Specify the image copy run time for the RCVTIME parameter.
  - If you are recovering by using the USEUICLAST or USEUICTIME parameters, do not use the **NOTIFY.RECOV** command. Proceed to the next step.
3. Add the database data set or area to the recovery list by using the **ADD** command with the USEDDBDS, USEAREA, USEUICLAST, or USEUICTIME parameter.
4. Proceed with recovery by using the **RECOVER START** command or the **ADD** command in batch.
5. If you use FASTIC=DUMP in your IMS High Performance Image Copy job, you must also specify RECFM=V for IMS Database Recovery Facility to restore the image copy.

### Change accumulation input data sets

You can use change accumulation data sets as input to IMS Database Recovery Facility.

However, there are cases where change accumulation data sets cannot be used for recovery. For example, if there is no deallocation point between the change accumulation stop time and the recovery time, IMS Database Recovery Facility cannot use the change accumulation data set.

However, it is not necessary to first run the IMS High Performance Change Accumulation Utility before database recovery. If no acceptable change accumulation data set is found, IMS Database Recovery Facility will read and process all of the logs needed to ensure updates from all systems are included in the recovery.

### Log data sets usage for recovery

IMS Database Recovery Facility uses RLDS (recovery log data sets) as input when they are available.

IMS Database Recovery Facility uses SLDS (system log data sets) as input in the following situations:

- When no RLDS is available
- In case of an error reading RLDSs

This log usage scheme minimizes the impact of recovery on other IMS activity.

Ensure that every OLDS (online log data set) that contains log records that are required for recovery is archived before you start the recovery process. If an OLDS is required for recovery but it has not been archived, recovery is not started and an error message is issued.

If an error occurs while reading the primary copy of a log data set (either RLDS or SLDS), IMS Database Recovery Facility automatically starts reading from the secondary copy.

If errors occur reading from the copies of an RLDS, IMS Database Recovery Facility automatically starts reading from the SLDS for the interval covering the RLDS.

## Image copy frequency

The frequency with which you take database image copies affects the time needed for database recovery.

Periodically you should consider whether to increase this frequency.

Your actual time savings depends upon the amount of log data to be read.

## Recovery from secondary sources

For some situations, you might want to recover a database by using only secondary image copies and log data sets.

This situation might be one where copies of your input data sets are shipped to a remote site where the databases can be reconstructed in the event of a disaster at the production site. You might also use this option if an unrecoverable I/O error occurs on the primary data set.

You can use secondary sources for the default processing mode at a location by using the SOURCE keyword with the SEC parameter in the FRXDRFxx member or in the SYSIN DD statements. If you choose the secondary source option, none of the primary image copy or log data sets are used.

In the event that you issue this form of recovery without the availability of secondary copies of the primary data sets, an FRD6022A message is issued to describe the error.

## Managing output

---

These topics show how you can manage your IMS Database Recovery Facility output data sets.

### Topics:

- [“Stacked image copy output” on page 162](#)

## Stacked image copy output

Stacking can place two or more output image copy data sets on the same tape volume or set of tape volumes.

You can indicate to IMS Database Recovery Facility that stacking of output image copies is to be done by specifying the STACK option on the UTILGBL statement. IMS Database Recovery Facility then passes this indication on to IMS High Performance Image Copy during image copy generation.

Stacking can increase the efficiency of tape media use and decrease the number of tape volumes that are needed by allocation. However, use of the STACK option with IMS Database Recovery Facility will serialize the recovery process and might decrease the overall recovery performance.

When STACK is specified, IMS Database Recovery Facility processes all DBDSs on the stacked image copy under the same task.

## Managing the environment

---

Requirements for managing your recovery environment include ensuring that data sets and areas that are to be recovered are defined to DBRC and are available on the system running IMS Database Recovery Facility.

All required resources must be in the RECON, and DBDLIBs and PSBLIBs must be available.

### Topics:

- [“Data integrity” on page 163](#)
- [“How security is assigned” on page 163](#)
- [“Database availability during recovery” on page 163](#)
- [“Starting databases and areas” on page 163](#)
- [“DEDBs with MADS” on page 163](#)
- [“Restart considerations” on page 164](#)

- [“Auditability considerations” on page 164](#)
- [“Spill data space management” on page 164](#)

## Data integrity

Before a recovery can be performed, all database data sets in the recovery list must be brought offline. This can be done by either issuing the DBRECOVERY command, or specifying DBRCMD option on the START command.

The database data sets that are being recovered are not made available for processing by other IMS subsystems or batch jobs until the successful completion of recovery is recorded in the RECON by DBRC. It is essential that all jobs accessing database data sets and areas that are being recovered use DBRC to control access to the data sets during recovery.

If you run a time stamp recovery to any prior point in time (PITR option), the database data sets and areas are not made available for processing by other IMS subsystems or batch jobs until they are successfully image copied and the image copies are registered with DBRC.

**Note:** If you use time stamp forward recovery, it is strongly recommended that all related database data sets and areas be recovered to the same point in time. This recommendation includes database data sets and areas, secondary indexes, and logically related databases. Failure to comply with this recommendation can result in databases that contain inconsistent data.

## How security is assigned

IMS Database Recovery Facility requires specific RACF and security settings.

For more information, see the topics on security in the *IMS Recovery Solution Pack: Overview and Customization*.

## Database availability during recovery

The databases and areas that are being recovered are authorized exclusively to IMS Database Recovery Facility during the recovery process, and are unavailable for any non-recovery-related IMS or batch processing until the successful completion of the recovery process.

If recovery is unsuccessful, the database data sets (and their associated databases) and areas that fail recovery are left in a recovery needed state.

## Starting databases and areas

The full-function databases, HALDB partitions, and DEDB areas that have data sets on the recovery list can be started automatically on all IMS systems to which they are defined by using the STACMD parameter on the START command.

After recovery completes and if you did not select time stamp recovery with PITR, a **START DB** command is issued automatically if you specified STACMD(GLOBAL) parameter.

However, if a point in time recovery (PITR) was performed, database data sets must first be image copied before they can be started.

## DEDBs with MADS

IMS Database Recovery Facility attempts to recover the first unavailable area data set in the area data set list.

If the area is restored from an SMS image copy that was generated by the image copy 2 utility, the area data set that is restored is the one that was dumped to create the image copy.

**Note:** IMS Database Recovery Facility's recovery of MADS (multiple area data sets) differs from IMS Database Recovery Utility (DFSURDB0) processing.

IMS Database Recovery Facility requires at least one area data set in the area data set list to be unavailable.

When using DFSURDB0, the RECON recovery-needed flag must be set for recovery of MADS.

When using IMS Database Recovery Facility, this flag does not need to be set.

## **Restart considerations**

If recovery by IMS Database Recovery Facility is in progress when IMS abnormally ends, IMS Database Recovery Facility does not automatically start during IMS emergency restart.

You can start recovery by using one of the following methods:

- You can initiate IMS Database Recovery Facility in batch mode.
- You can initiate IMS Database Recovery Facility on a different IMS.

You must enter the DBDSs and areas again.

The old recovery list cannot be transferred from one IMS to another.

- You can initiate IMS Database Recovery Facility on the same IMS after restart completes.

You must enter the DBDSs and areas.

The old recovery list is not retained across restarts.

- You can use database recovery utilities other than IMS Database Recovery Facility to perform recovery.

## **Auditability considerations**

Log records are written as part of IMS Database Recovery Facility processing in online mode and are ignored during IMS restart because they are intended as diagnostic aids only.

Trace entries are made to the internal trace tables of the address spaces that are participating in IMS Database Recovery Facility.

The trace tables are dumped in the event of IMS Database Recovery Facility abnormal end.

Database data set and area recovery status is maintained in the RECON by DBRC.

This RECON status determines the availability of database data sets and areas to IMS subsystems.

## **Spill data space management**

IMS Database Recovery Facility reads change accumulation records and holds them in the RSS until the data is needed for the data set restore process.

If there is more data to be held than can fit in the address space, it is spilled into one or more data spaces.

You can control the size of the data spaces that are allocated by setting SPSIZE in the FRXDRFxx member.

Choosing a smaller size can result in additional data spaces being allocated if the data will not fit into the initial data space.

Smaller data space sizes can result in more efficient paging device utilization if the amount of spill-data fits into the initial data space because data space storage requires backing up data on paging devices.

---

## Chapter 12. Managing performance

IMS Database Recovery Facility provides improved performance over other IMS database recovery solutions.

Database recovery performance improvements result both from the design of IMS Database Recovery Facility and from your management of the database recovery operational environment.

### Topics:

- [“Performance improvement” on page 165](#)
- [“Control of input parallelism” on page 166](#)
- [“Virtual tape system management” on page 166](#)
- [“Tape device availability management” on page 166](#)

---

### Performance improvement

IMS Database Recovery Facility should provide an improvement over IMS database recovery procedures for the time that is required to recover database data sets.

This improvement is due to the following reasons:

- Log data sets that must be read to recover the set of database data sets are read in parallel.
- Multiple database data sets and areas are recovered simultaneously.
- Shared database data sets and areas do not require a change accumulation step before running the IMS Database Recovery Facility.

Not running change accumulation before recovering shared databases can be beneficial to database recovery performance.

- Multiple processing steps are not required when using auxiliary utilities.

Each database is read only once for processing by IMS Database Recovery Facility, along with IMS High Performance Pointer Checker and IMS High Performance Image Copy (if IMS High Performance Pointer Checker or IMS High Performance Image Copy was requested in batch).

Processing by IMS Index Builder, the DFSPRECO utility, or the Build Index function of FPA is not included.

After the database is processed by IMS Database Recovery Facility, it is passed to two separate tasks, one for IMS High Performance Image Copy and one for IMS High Performance Pointer Checker, which run in parallel. IMS Database Recovery Facility continues processing as a third task.

Before IMS Database Recovery Facility reads the next database, it waits for completion of the other two tasks.

Using this method improves performance because databases need to be read only once, not three times, and parallel processing occurs.

Performance of each recovery depends on your processing environment at the time of the recovery.

The improvement in performance depends on the following:

- The number of log data sets that are required for recovery
- The number of tape drives and physical data paths that are available to IMS Database Recovery Facility for reading data sets
- The number of database data sets and areas that are being recovered

## Control of input parallelism

---

The ability for IMS Database Recovery Facility to read multiple input devices simultaneously is called input parallelism.

You control the amount of parallelism by providing an appropriate number of input devices. You can control the number of both mountable (tape) and non-mountable (DASD) devices separately through the values specified with the READNUM parameter, or LOGNUM and ICNUM parameters, in the FRXDRExx PROCLIB member or in the **START** command.

The number of input devices has a direct impact on the performance of IMS Database Recovery Facility. Use the READNUM parameter to specify the number of image copy, change accumulation, and log data sets that can be read in parallel. The time that is required for recovery of databases and areas is directly proportional to this value.

If you set the value too low for the READNUM parameter, or LOGNUM and ICNUM parameters, reading image copies and log data sets will impede recovery.

If the image copies and log data sets reside on tape, specifying a value that is too high for the READNUM parameter, or LOGNUM and ICNUM parameters, impacts the availability of tape drives for other operations during the recovery process.

You can specify the READNUM parameter as a startup parameter or as a parameter on the **START** command. If you do not specify the READNUM maximum number of log read tasks that are started in parallel for tape only (*nn*) subparameter by one of the following methods, the system default value of 3 is used:

- As a startup parameter
- As a **START** statement for batch invocation

SORTPARM(NUM(*xx*)) also aids input parallelism by controlling the number of recovery sort subordinate address spaces that are generated. Each address space processes one IC at a time.

**Note:** If multiple image copies are stacked on tape, they are separated into separate groups by tape volume. Each tape volume is processed by a separate address space.

## Virtual tape system management

---

If you manage your image copy data sets through a VTS (virtual tape server), the system automatically moves the contents of the tape data set to a temporary cache data set when the data set is allocated.

As part of its initialization IMS Database Recovery Facility can help you take advantage of the VTS by pre-allocating the image copy data sets that are required for recovery.

IMS Database Recovery Facility facilitates efficient use of the VTS by overlapping the time that is required to stage image copy data from tape to cache with the operation of reading the log data sets.

Specifying CACHE in the FRXDRExx member initiates the preallocation of image copy data sets. You must have VTS available to take advantage of this option.

## Tape device availability management

---

Determine how many tape devices you have available before you run your recovery job. If you are restoring from image copies that reside on separate tapes, ideally you would have enough tape devices to handle each parallel recovery task.

For example, if you have five image copies and you set up your recovery environment to start five recovery RSSs to run in parallel, you would want to have at least five tape devices available for recovery.

If you have fewer tape devices than are necessary, MVS displays an IEF238D message prompting you to reply with either the device name when it becomes available, or a cancel command for the recovery job.

If you cannot increase the number of available tape devices to match your parallel recoveries, here are two other methods to help manage your recoveries:



- You can decrease the number of RSSs that run in parallel.

For example, if you have five image copies and only one available tape device, you might consider overriding your recovery SORT parameters with SORTPARM(NUM(1)).

This forces IMS Database Recovery Facility to serialize your recovery, but because you only have one available tape device it is already serialized.

With this method you will not have to reply with the tape device name for the IEF238D message each time.

- You can specify TAPECHK(Y).

With a TAPECHK(Y) control statement setting, IMS Database Recovery Facility uses the value specified in READNUM or ICNUM as a way to control tape device allocations in each RSS..

If you have five image copies, five RSSs, and only one tape device, you would specify TAPECHK(Y) and ICNUM(1,13). The number 1 in the ICNUM specification informs IMS Database Recovery Facility that only one tape device is available. That tape device is allocated to only one address space and places the other address spaces in a wait state until the device becomes available.

However, if you specify ICNUM(2,13) erroneously when there is only one tape device available, IMS Database Recovery Facility attempts to allocate two tape devices but MVS issues message IEF238D on the second allocation attempt.

Also consider that when you restore from image copies residing on tape and expect IMS Database Recovery Facility to generate IMS HP Image Copy output on tape, you are using twice as many tape devices.

In this situation, IMS Database Recovery Facility requires one tape device to read the input image copy from and another tape device to generate the output image copy.



---

## Chapter 13. Managing time stamp recovery

Time stamp recovery is the recovery of full-function databases, HALDB partitions, and DEDB areas to a specified time stamp or recovery time.

IMS Database Recovery Facility supports time stamp recovery to database allocation boundaries or to any prior PITR (point-in-time recovery).

### Topics:

- [“Time stamp recovery without PITR” on page 169](#)
- [“Time stamp recovery with PITR” on page 169](#)
- [“RECOVGRP” on page 170](#)
- [“Time stamp recovery with PITR and UOR” on page 170](#)
- [“Time stamp recovery with PITR and database updates on multiple UOR” on page 171](#)

---

### Time stamp recovery without PITR

Time stamp recovery without PITR (point-in-time recovery) is described here.

At your specified recovery time, all database data sets and areas that are being recovered cannot be in use for update by any IMS job. You must issue a **DBRECOVERY** command from all of the online IMS systems that are using the data sets and areas, and there must have been no batch jobs updating the databases and areas. A listing of the RECON can help you to determine valid time stamps.

There must be no ALLOC record for any database data set or area that is being recovered that spans the recovery time. That is, all ALLOC records with an ALLOC time that is earlier than the recovery time must have either:

- A DEALLOC time that is also earlier than the recovery time.
- The associated PRILOG record must have a STOP time earlier than the recovery time.

The associated PRILOG record is the one with a START time that is equal to the START time specified in the ALLOC record.

---

### Time stamp recovery with PITR

This topic describes time stamp recovery with PITR (point-in-time recovery).

When you specified PITR, the database data sets and areas can have been at any allocation status in the RECON. Recovery resources must still be available only for the time stamp you specified. There is no restriction on database allocation status if you specified time stamp recovery with PITR.

In the recovery process only committed updates, up to and including the specified recovery time, are applied to the database data sets and areas in the recovery list.

Change accumulation input is used under these two conditions:

1. The change accumulation data set must have a stop time before the PITR recovery time.
2. You must also ensure that the stop time does not fall within the same ALLOC / DEALLOC range that spans the recovery time.

If IMS Database Recovery Facility cannot guarantee that the updates on a change accumulation data set are committed, it is not used and log data sets are used instead.

**Restriction:** If you selected time stamp recovery or time stamp recovery with PITR, and there is a REORG record for any database data set or area being recovered that is after the time specified on the RCVTIME parameter on the DBRC **NOTIFY . RECOV** command, warning message FRD6022A is issued and recovery is not performed for this data set. If you did not specify the ERRORCONT parameter on the **START** command, recovery is not started. This prevents recovery from using the wrong database structure.

If you selected time stamp recovery, you can also specify that you want to be notified of database data sets and areas that might be inconsistent with recovery of database data sets and areas on the recovery list.

## RECOVGRP

A RECOVGRP (recovery group) is a group type in the RECON. It is a kind of database group that can be used with the IMS Database Recovery Facility and DBRC commands anywhere that a database group can be used.

RECOVGRP (recovery group) differs from a regular database group in the following ways:

- All members must be registered with DBRC.
- A member can belong to only one recovery group.

If a member belongs to a recovery group, the group name is stored in the DB record for full-function or in the AAUTH record for an area.

- A member's presence in a recovery group can affect the IMS Database Recovery Facility process.

When the IMS Database Recovery Facility is about to perform a time stamp recovery, either to an allocation boundary or to any point in time for a recovery group member, it ensures that all members of the recovery group are recovered in the same recovery operation unless you specified otherwise with the NOCHECK parameter.

## Time stamp recovery with PITR and UOR

Time stamp recovery with PITR (point-in-time recovery) and UOR (units of recovery) is described here.

The following diagram shows time stamp recovery with PITR (point-in-time recovery) and UOR (units of recovery).

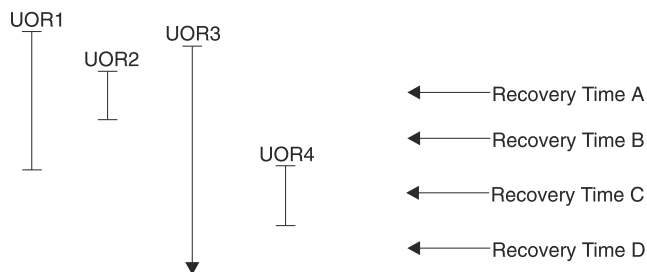


Figure 5. Time stamp recovery with PITR (point-in-time recovery) and UOR (units of recovery)

The vertical lines in this diagram represent the scope of a UOR. The top of each line is the beginning of a UOR. The bottom of each line is the end of the sync point for a UOR. For purposes of this figure, it is assumed that no other UORs have updated any of the database data set or areas on the recovery list. Updates from UORs shown are applied based on the recovery time:

- If time stamp recovery with PITR is performed with recovery time A, no updates apply to any database data set or area on the recovery list.

Recovery consists of image copies being restored to the database data sets and areas on the recovery list.

- If time stamp recovery with PITR is performed with recovery time B, only updates from UOR2 apply to the database data sets and areas on the recovery list.

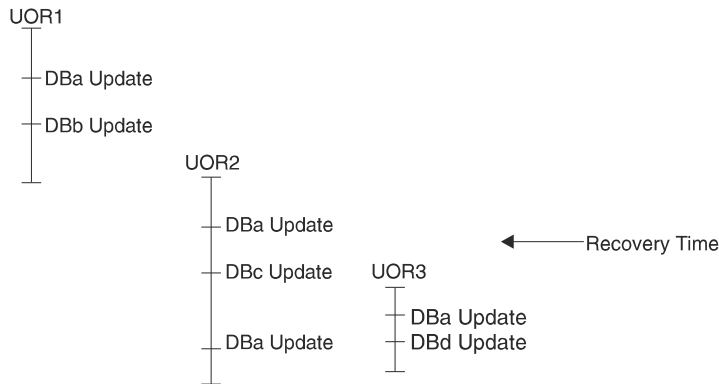
It is the only UOR that has committed data.

- If time stamp recovery with PITR is performed with recovery time C, updates apply to the database data sets and areas on the recovery list for UOR1 and UOR2 only.
- If time stamp recovery with PITR is performed with recovery time D, updates apply to the database data sets and areas on the recovery list for UOR1, UOR2 and UOR4.

## Time stamp recovery with PITR and database updates on multiple UOR

Time stamp recovery with PITR (point-in-time recovery) and database updates on multiple UOR (units of recovery) is described here.

The following diagram shows time stamp recovery with PITR (point-in-time recovery) and database updates on multiple UOR (units of recovery).



*Figure 6. Time stamp recovery with PITR (point-in-time recovery) and database updates on multiple UORs (units of recovery)*

The vertical lines in this diagram represent UOR scopes for three UORs that update database data sets in common.

- The first entry of each line is the beginning of a UOR.
- The last entry of each line is the end of the sync point for that UOR.

For purposes of this figure, it is assumed that no other UORs have updated any of the database data sets or areas on the recovery list. A time stamp recovery with PITR is run for databases DBa and DBb.

If you did not specify the NOCHECK parameter, a message is issued that indicates that databases DBc and DBd might need to be recovered by using time stamp recovery with PITR with the original recovery time.

After a time stamp recovery with PITR completes, the databases that are marked as recoverable in the RECON and areas cannot be started (the **START** command fails) nor can they be used by IMS batch jobs until an image copy is taken for each database data set and area (the IC NEEDED flag is set ON in the RECON).

If you take a nonstandard image copy, you must set the IC NEEDED flag to OFF in the RECON by issuing the DBRC **CHANGE . DBDS** command with the ICOFF parameter. Therefore, the STAGLOBAL and STALocal parameters that you specified are ignored for time stamp recovery with PITR.



---

## Part 5. Usage scenarios for optimization

This document addresses some of the more common and useful ways to recover IMS databases by using IMS Database Recovery Facility.

IMS Database Recovery Facility supports many different types of recoveries.

Instructions for using IMS Database Recovery Facility to perform the following types of recoveries are described in the following topics:

**Topics:**

- [Chapter 14, “Setting up a basic IMS Database Recovery Facility environment,” on page 175](#)
- [Chapter 15, “Recover the database to the current time,” on page 179](#)
- [Chapter 16, “Validating recovery by using IMS HP Pointer Checker ,” on page 181](#)
- [Chapter 17, “Taking an incremental image copy of offline databases,” on page 185](#)
- [Chapter 18, “Taking an incremental image copy of an online database to the current time or point-in-time,” on page 191](#)
- [Chapter 19, “Using IMS Index Builder to rebuild indexes during recovery,” on page 197](#)
- [Chapter 20, “Recovering HALDB databases that use online reorganization,” on page 201](#)





# Chapter 14. Setting up a basic IMS Database Recovery Facility environment

Complete these step-by-step instructions to set up and run basic IMS Database Recovery Facility for the usage scenarios.

In this topic:

- [“Setting up the IMS Recovery Solution Pack” on page 175](#)
- [“Installing IMS Recovery Solution Pack libraries” on page 175](#)
- [“Reviewing the IMS Recovery Solution Pack installation checklist” on page 176](#)
- [“Customizing the procedures and JCL members” on page 176](#)
- [“Tips and best practices” on page 177](#)

## Setting up the IMS Recovery Solution Pack

This section outlines the steps to install and customize the IMS Recovery Solution Pack according to your environment and then, begin running the tools and products that are included in this solution pack.

The IMS Recovery Solution Pack includes these products:

- IMS Database Recovery Facility
- IMS Database Recovery Facility Extended Functions
- IMS High Performance Change Accumulation Utility
- IMS High Performance Image Copy
- IMS Index Builder

You must also install IBM IMS Tools Base so that you can save output reports from IMS Tools products in the IMS Tools Knowledge Base (ITKB) repository.

**Recommendation:** Install and configure IMS High Performance Pointer Checker (HPPC) and High Performance DEDB Pointer Checker (DEDB PC) (a component of the IMS Fast Path Solution Pack for z/OS) on your system to take advantage of the additional integrated auxiliary utilities that are supported by IMS Database Recovery Facility during and after recovery.

## Installing IMS Recovery Solution Pack libraries

You must install the IMS Recovery Solution Pack load libraries by completing the instructions that are in the *Program Directory for IMS Recovery Solution Pack for z/OS*.

You must install the following IMS Recovery Solution Pack load libraries:

| Load library              | Description                 |
|---------------------------|-----------------------------|
| <i>your.hlq</i> .SFRXLOAD | DRF and DRF-XF load library |
| <i>your.hlq</i> .SHPSLMDO | HPIC load library           |
| <i>your.hlq</i> .SHPCLMDO | HPCA load library           |
| <i>your.hlq</i> .SIIULMOD | IB load library             |
| <i>your.hlq</i> .SFRXSAMP | Sample PROCs and JCLs       |

The SMP/E installation process for the IMS Recovery Solution Pack is documented in the *Program Directory for IMS Recovery Solution Pack for z/OS*.

## Reviewing the IMS Recovery Solution Pack installation checklist

---

Review the IMS Recovery Solution Pack installation checklist member, which is located in *your.hlq.SFRXSAMP(FRXCHECK)*, to verify that your environment is configured correctly for IMS Recovery Solution Pack.

## Customizing the procedures and JCL members

---

Depending on which IMS Database Recovery Facility tasks you are doing, you must customize the applicable procedures and JCLs.

Review the IMS Recovery Solution Pack sample library, which is named *your.hlq.SFRXSAMP*. This sample library contains all of the relevant procedures and JCL that is required to set up and run IMS Database Recovery Facility with utilities on your system, including DEDB PC, HPPC, IB, and HPIC.

To customize the members, follow the instructions that are provided in each sample member.

The following lists the sample members that are used in this scenarios document.

### **FRXDRFZZ**

This configuration file is required for IMS Database Recovery Facility.

Make a copy of *your.hlq.SFRXSAMP(FRXDRFZZ)* and customize it for your environment.

### **FRXBPECF**

This configuration file is required for BPE.

Make a copy of *your.hlq.SFRXSAMP(FRXBPECF)* and customize it for your environment.

### **FRXFOI**

This configuration file is required for TOSI.

Make a copy of *your.hlq.SFRXSAMP(FRXFOI)* and customize it for your environment.

### **FRXITKB**

This configuration file is required for copying compatible IMS Database Recovery Facility and utility reports to ITKB.

Make a copy of *your.hlq.SFRXSAMP(FRXITKB)* and customize it for your environment.

### **DRFMAS procedure**

This generic procedure is used to allocate the IMS Database Recovery Facility MAS.

Make a copy of *your.hlq.SFRXSAMP(FRXMAS)*.

All of the variables in this procedure are controlled by the SET statements in the execution JCL that are called by *your.hlq.SFRXSAMP(FRXDRF)*.

If you are using a consolidated IMS Tools load library, copy *your.hlq.SFRXSAMP(FRXMASC)* and customize the SET statements in *your.hlq.SFRXSAMP(FRXDRFC)*.

### **DRFIAX procedure**

This IMS Database Recovery Facility UAS/IAX procedure is used to allocate the IB, IC, and DFSPRECO address spaces.

Make a copy of *your.hlq.SFRXSAMP(FRXIAX)* and customize it by adding the name of your IMS Database Recovery Facility load library or consolidated IMS Tools load library.

**Tip:** If you have multiple levels of maintenance for the IMS Tools, customize the FRXIAX procedure and rename to FRXIAXxx, where xx is the maintenance level used for each load library. For example, you can create FRXIAXT1 for IMS SSID IMT1 of your test environment and FRXIAXP1 for your IMS SSID IMP1 of your production environment.

### **DRFPROC procedure**

This procedure is used to allocate the IMS Database Recovery Facility subordinate address spaces.

Make a copy of *your.hlq.SFRXSAMP(FRXRSS)* and customize it by adding the name of your IMS Database Recovery Facility load library or consolidated IMS Tools load library.

**Tip:** If you have multiple levels of maintenance for the IMS Tools, customize the FRXRSS procedure and rename to FRXRSSxx, where xx is the maintenance level used for each load library. For example, you can create FRXRSS1 for IMS SSID IMT1 of your test environment and FRXRSSP1 for your IMS SSID IMP1 of your production environment.

#### **FRXPAT0 and FRXPATZ procedures**

These procedures are required to run HPPC for a full function database or DEDB PC for a fast path database.

Make a copy of *your.hlq.SFRXSAMP(FRXPAT0)* and *your.hlq.SFRXSAMP(FRXPATZ)* in a system PROCLIB that is concatenated in the JES system PROCLIB, and customize them for your environment.

#### **IMS Database Recovery Facility batch job**

This sample batch job runs IMS Database Recovery Facility.

Make a copy of *your.hlq.SFRXSAMP(FRXDRF)* and customize it for your environment.

## **Tips and best practices**

---

This section provides tips to run the IMS Recovery Solution Pack smoothly.

#### **Dynamic allocation of RECONS**

To ensure that IMS Database Recovery Facility recovers the correct set of data, use dynamic allocation of RECONS as opposed to explicitly specifying the RECON data sets in the IMS Database Recovery Facility JCL and procedures.

#### **RACF authority**

If insufficient authority was granted to the user ID that submits the IMS Database Recovery Facility job, RACF permission failures can occur when data sets are created and accessed.

These data sets are required by IMS Database Recovery Facility to perform recovery in the master address space and all subordinate address spaces, including the utility address space for IMS Index Builder (IB UAS). The IB UAS can initiate one or more sort subordinate address spaces (IBSS).

In addition, when specifying the DRFHLQ= option in the REPORT() parameter, confirm with your system administrator that the 1- to 8-character high-level qualifier has the proper authority to allocate and catalog IMS Database Recovery Facility report data sets on your system.

#### **Hard coding JCL statements for databases that are being recovered**

The IMS Database Recovery Facility master address space JCL cannot contain any hard-coded database data sets. This restriction applies to any step in the JCL, regardless of the usage.

Because IMS Database Recovery Facility dynamically allocates the database data sets for recovery in the subordinate address space, dynamic allocation will fail if any of the subsequent steps have these same data sets allocated by DD statements in the MAS.

This restriction is a z/OS restriction.



---

## Chapter 15. Recover the database to the current time

This scenario documents how to recover a database to the current time by restoring the database to the last valid image copy and processing any available change accumulations (CAs) and logs up to the point that IMS Database Recovery Facility is run.

To recover a database to the current time, the database is deleted and then reallocated with the image copy and any updates from the CAs and logs.

In this topic:

- [“Creating and running an IMS Database Recovery Facility job” on page 179](#)
- [“Reviewing the output” on page 180](#)

---

### Creating and running an IMS Database Recovery Facility job

Perform the following steps to set up and create an IMS Database Recovery Facility job to recover a database to the current time.

#### Before you begin

- To perform an IMS Database Recovery Facility recovery, you must have IMS Database Recovery Facility installed.
- You must take your databases offline by issuing the /DBR command for IMS Database Recovery Facility to delete the production databases and redefine them.
- Complete the tasks in sample member [FRXITKB](#).

#### Procedure

1. Install and configure the required IMS Database Recovery Facility load libraries and utilities.

For more information, see [“Installing IMS Recovery Solution Pack libraries” on page 175](#)

2. Customize the procedures and JCL found in the *your.hlq.SFRXSAMP* sample library.

For more information, refer to:

- [“DRFMAS procedure” on page 176](#)
- [“DRFPROC procedure” on page 176](#)
- [“DRFIAX procedure” on page 176](#)

3. Customize the execution JCL located in the *your.hlq.SFRXSAMP* sample library.

For more information, see [“DRFMAS procedure” on page 176](#).

The following example shows modified IMS Database Recovery Facility JCL to run a recovery:

```
@//DRF      EXEC FRXMAS,DRFMBR=ZZ
//SYSIN DD *
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
SORTPARM(ASPREF(AS))
DBDSL101(DISP(NEW))
@OUTPUT(PRO)
@ADD DB(DHVNTZ02) DBATRB(DBDSL(101))
START ERROR(CONT)
//
```

#### Explanation:

- a. When you run IMS Database Recovery Facility, the FRXMAS procedure is invoked.  
DRFMBR=ZZ references the FRXDRFZZ configuration file for IMS Database Recovery Facility.

- b. A combination of OUTPUT(PRO) and DISP(NEW) causes IMS Database Recovery Facility to delete and redefine the production database data sets during recovery to the current time.
- c. The database name is DHVNTZ02 (HIDAM/VSAM).

You can specify one or more databases to recover.

#### 4. Submit the IMS Database Recovery Facility job.

### What to do next

After you submit your IMS Database Recovery Facility batch job, wait until the job finishes and review the output.

## Reviewing the output

This section shows an example of the output from a standard IMS Database Recovery Facility recovery to the current time recovery job. Review the output to help you understand the results of running IMS Database Recovery Facility.

### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
FRD7201I SORTPARM(ASPREF(AS))
FRD7201I DBDSL101(DISP(NEW))
FRD7201I OUTPUT(PRO)
FRD7201I ADD DB(DHVNTZ02) DBATRB(DBDSL(101))
FRD7201I START ERROR(CONT)
```

### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | IC | Records | Read CA | LOG | Records Written | Subord. Reg Name | STC # | Final Status           |
|---------------|--------------|------|----|---------|---------|-----|-----------------|------------------|-------|------------------------|
| DHVNTZ02      | HIDAM        | 1    | 49 | 0       | 1501    |     | 49              | AS068801         | 689   | Delete/define complete |
| DHVNTZ02      | HIDAM2       | 2    | 49 | 0       | 0       |     | 49              | AS068802         | 690   | Delete/define complete |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/12/2020 Time: 23:15 Page: 2

### DATABASE RECOVERY FACILITY UTILITY REPORT

| Database | DDN    | Database Data Set Name     | IC  | PC/DP | IB  | PR  | Utility Final Status |
|----------|--------|----------------------------|-----|-------|-----|-----|----------------------|
| DHVNTZ02 | HIDAM  | IMSTESTL.DHVNTZ02.FKXXI01E | N/A | N/A   | N/A | N/A |                      |
| DHVNTZ02 | HIDAM2 | IMSTESTL.DHVNTZ02.FKXXI02E | N/A | N/A   | N/A | N/A |                      |

Final Return (RC) and Reason (RSN) Codes

| ---IC--- |     | ---PC--- |     | ---DP--- |     | ---IB--- |     | ---PR--- |     | ---LIU--- |     |
|----------|-----|----------|-----|----------|-----|----------|-----|----------|-----|-----------|-----|
| RC       | RSN | RC       | RSN | RC       | RSN | RC       | RSN | RC       | RSN | RC        | RSN |
| N/A      | N/A | N/A      | N/A | N/A      | N/A | N/A      | N/A | N/A      | N/A | N/A       | N/A |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/12/2020 Time: 23:15 Page: 3

### DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: Not Specified

| Image Copy Data Set Name          | Volume Serial | IC DS Read Count | IC Type | Time Stamp Range       |
|-----------------------------------|---------------|------------------|---------|------------------------|
|                                   |               |                  |         | 1st Record Last Record |
| IMSVS.DHVNTZ02.HIDAM.IC.IC225745  | 222222        | 49               | STD     |                        |
| IMSVS.DHVNTZ02.HIDAM2.IC.IC225745 | 222222        | 49               | STD     |                        |

| Change Accum Data Set Name               | Volume Serial | CA DS Read Count | Time Stamp Range       |
|------------------------------------------|---------------|------------------|------------------------|
|                                          |               |                  | 1st Record Last Record |
| No data available for this type data set |               |                  |                        |

No data available for this type data set

| Log Data Set Name                    | Volume Serial | Log DS Read Count | IMS SYSID | Time Stamp Range                                    |
|--------------------------------------|---------------|-------------------|-----------|-----------------------------------------------------|
|                                      |               |                   |           | 1st Record Last Record                              |
| IMSVS.RLDSP.IMS1.D20043.T2305207.V00 | 000000        | 0                 | IMS1      | 2020.043 23:05:20.752699 -02020.043 23:05:55.264546 |
| Prilog: 2020.043 23:02:53.600000     |               |                   |           |                                                     |
| IMSVS.RLDSP.IMS1.D20043.T2302536.V00 | 000000        | 1489              | IMS1      | 2020.043 23:02:53.600000 -02020.043 23:03:37.030997 |
| Prilog: 2020.043 23:02:53.600000     |               |                   |           |                                                     |
| IMSVS.RLDSP.IMS1.D20043.T2304155.V00 | 000000        | 0                 | IMS1      | 2020.043 23:04:15.544239 -02020.043 23:05:20.752699 |
| Prilog: 2020.043 23:02:53.600000     |               |                   |           |                                                     |
| IMSVS.RLDSP.IMS1.D20043.T2303370.V00 | 000000        | 12                | IMS1      | 2020.043 23:03:37.030997 -02020.043 23:04:15.544239 |
| Prilog: 2020.043 23:02:53.600000     |               |                   |           |                                                     |

---

## Chapter 16. Validating recovery by using IMS HP Pointer Checker

The IMS Database Recovery Facility recovery of the database to the current time job is validated for any pointer errors by using IMS High Performance Pointer Checker (HPPC).

You can use IMS HP Pointer Checker to validate the recovery for any pointer errors.

In this topic:

- [“Creating and running an IMS Database Recovery Facility job” on page 181](#)
- [“Reviewing the output” on page 182](#)

---

### Creating and running an IMS Database Recovery Facility job

Perform the following steps to create and run an IMS Database Recovery Facility job to validate the recovery by using HPPC.

#### Before you begin

- To validate the recovery, you must have IMS Database Recovery Facility and IMS HP Pointer Checker (HPPC) installed.
- Complete the tasks in sample member [FRXITKB](#).

#### Procedure

1. Install and configure the required IMS Database Recovery Facility load libraries and utilities.

For more information, see [“Installing IMS Recovery Solution Pack libraries” on page 175](#)

2. Customize the procedures found in the *your.hlq.SFRXSAMP* sample library.

For more information, refer to:

- [“DRFMAS procedure” on page 176](#)
- [“DRFPROC procedure” on page 176](#)

3. Add the PC() keyword to the IMS Database Recovery Facility ADD statement.

The PC() keyword causes IMS Database Recovery Facility to run HPPC (for full-function databases) and DEDB PC (for Fast Path databases).

For more information about the PC() keyword, see [“Reference: ADD PC\(\) command” on page 140](#).

4. Customize the FRXDRF or FRXDRFC execution JCL found in the *your.hlq.SFRXSAMP* sample library.

For more information see [“DRFMAS procedure” on page 176](#).

The following example shows modified IMS Database Recovery Facility JCL to run an IMS Database Recovery Facility/PC recovery:

```
a//DRF      EXEC FRXMAS,DRFMBR=ZZ
//SYSIN DD *
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
SORTPARM(ASPREF(AS))
DBDSL101(DISP(NEW))
bOUTPUT(PRO)
cADD DB(DHVNTZ02,DEDBJN23) PC() -
  DBATRB(DBDSL(101))
START ERROR(CONT)
//
```

#### Explanation:

- a. When you run IMS Database Recovery Facility, the FRXMAS procedure is invoked.

DRFMBR=ZZ references the FRXDRFZZ configuration file for IMS Database Recovery Facility.

- b. A combination of OUTPUT(PRO) and DISP(NEW) causes IMS Database Recovery Facility to delete and redefine the production database data sets during recovery to the current time.
- c. DHVNTZ02 (HIDAM/VSAM) is a full-function database and DEDBJN23 is a Fast Path database with eight areas.

You can specify one or more databases to recover.

5. Submit the IMS Database Recovery Facility job.

## Reviewing the output

This section describes the important parts of the completed IMS Database Recovery Facility output job to help you understand what exactly happened.

The IMS Database Recovery Facility Utility Report is generated when any auxiliary utility is started by using IMS Database Recovery Facility, for example PC(). It lists the final return code from the operation performed on any of the databases recovered by IMS Database Recovery Facility including the final reason codes from each auxiliary utility.

The following example shows the output from an IMS Database Recovery Facility/PC recovery job.

### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
FRD7201I SORTPARM(ASPREF(AS))
FRD7201I DBDSL101(DISP(NEW))
FRD7201I OUTPUT(PRO)
FRD7201I ADD DB(DHVNTZ02,DEDBJN23) PC() -
FRD7201I DBATRB(DBDSL(101))
FRD7201I START ERROR(CONT)
```

### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | IC  | Records Read | CA   | LOG | Records Written | Subord. Reg Name | STC #         | Final Status |
|---------------|--------------|------|-----|--------------|------|-----|-----------------|------------------|---------------|--------------|
| DEDBJN23      | DB23AR0      | 1    | 180 | 0            | 104  | 180 | AS069201        | 694              | Delete/define | complete     |
| DEDBJN23      | DB23AR1      | 2    | 21  | 0            | 104  | 21  | AS069202        | 695              | Delete/define | complete     |
| DEDBJN23      | DB23AR2      | 3    | 147 | 0            | 0    | 147 | AS069203        | 696              | Delete/define | complete     |
| DEDBJN23      | DB23AR3      | 4    | 147 | 0            | 0    | 147 | AS069201        | 694              | Delete/define | complete     |
| DEDBJN23      | DB23AR4      | 5    | 231 | 0            | 0    | 231 | AS069202        | 695              | Delete/define | complete     |
| DEDBJN23      | DB23AR5      | 6    | 99  | 0            | 0    | 99  | AS069203        | 696              | Delete/define | complete     |
| DEDBJN23      | DB23AR6      | 7    | 165 | 0            | 0    | 165 | AS069201        | 694              | Delete/define | complete     |
| DEDBJN23      | DB23AR7      | 8    | 21  | 0            | 0    | 21  | AS069202        | 695              | Delete/define | complete     |
| DHVNTZ02      | HIDAM        | 1    | 49  | 0            | 1501 | 49  | AS069203        | 696              | Delete/define | complete     |
| DHVNTZ02      | HIDAM2       | 2    | 49  | 0            | 0    | 49  | AS069201        | 694              | Delete/define | complete     |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 00:04 Page: 2

### DATABASE RECOVERY FACILITY UTILITY REPORT

| Database | DDN     | Database Data Set Name     | IC  | PC/DP | IB  | PR  | Utility Final Status |
|----------|---------|----------------------------|-----|-------|-----|-----|----------------------|
| DEDBJN23 | DB23AR0 | IMSTESTL.DB23AR0           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR1 | IMSTESTL.DB23AR1           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR2 | IMSTESTL.DB23AR2           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR3 | IMSTESTL.DB23AR3           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR4 | IMSTESTL.DB23AR4           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR5 | IMSTESTL.DB23AR5           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR6 | IMSTESTL.DB23AR6           | N/A | 00    | N/A | N/A |                      |
| DEDBJN23 | DB23AR7 | IMSTESTL.DB23AR7           | N/A | 00    | N/A | N/A |                      |
| DHVNTZ02 | HIDAM   | IMSTESTL.DHVNTZ02.FKXXI01E | N/A | 00    | N/A | N/A |                      |
| DHVNTZ02 | HIDAM2  | IMSTESTL.DHVNTZ02.FKXXI02E | N/A | 00    | N/A | N/A |                      |

### Final Return (RC) and Reason (RSN) Codes

| ---IC--- | ---PC--- | ---DP--- | ---IB--- | ---PR--- | ---LIU--- |
|----------|----------|----------|----------|----------|-----------|
| RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN    |
| N/A N/A  | 00 00    | 00 00    | N/A N/A  | N/A N/A  | N/A N/A   |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 00:04 Page: 3

### DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: Not Specified

| Image Copy Data Set Name           | Volume Serial | IC DS Read Count | IC Type | Time Stamp Range       |
|------------------------------------|---------------|------------------|---------|------------------------|
|                                    |               |                  |         | 1st Record Last Record |
| IMSVS.DEDBJN23.DB23AR1.IC.IC225831 | 222222        | 21               | STD     |                        |
| IMSVS.DEDBJN23.DB23AR2.IC.IC225832 | 222222        | 147              | STD     |                        |
| IMSVS.DEDBJN23.DB23AR0.IC.IC225831 | 222222        | 180              | STD     |                        |
| IMSVS.DEDBJN23.DB23AR5.IC.IC225832 | 222222        | 99               | STD     |                        |
| IMSVS.DEDBJN23.DB23AR3.IC.IC225832 | 222222        | 147              | STD     |                        |
| IMSVS.DEDBJN23.DB23AR4.IC.IC225832 | 222222        | 231              | STD     |                        |
| IMSVS.DEDBJN23.DB23AR7.IC.IC225832 | 222222        | 21               | STD     |                        |
| IMSVS.DEDBJN23.DB23AR6.IC.IC225832 | 222222        | 165              | STD     |                        |
| IMSVS.DHVNTZ02.HIDAM.IC.IC225745   | 222222        | 49               | STD     |                        |
| IMSVS.DHVNTZ02.HIDAM2.IC.IC225745  | 222222        | 49               | STD     |                        |



| Change Accum Data Set Name<br>-----                                       | Volume<br>Serial | CA DS<br>Read Count  |              | ----- Time Stamp Range -----<br>1st Record                      Last Record |
|---------------------------------------------------------------------------|------------------|----------------------|--------------|-----------------------------------------------------------------------------|
| No data available for this type data set                                  |                  |                      |              |                                                                             |
| Log Data Set Name<br>-----                                                | Volume<br>Serial | Log DS<br>Read Count | IMS<br>SYSID | ----- Time Stamp Range -----<br>1st Record                      Last Record |
| IMSVS.RLDSP. IMS1.D20043.T2305207.V00<br>Prilog: 2020.043 23:02:53.600000 | 000000           | 10                   | IMS1         | 2020.043 23:05:20.752699 -02020.043 23:05:55.264546                         |
| IMSVS.RLDSP. IMS1.D20043.T2302536.V00<br>Prilog: 2020.043 23:02:53.600000 | 000000           | 1489                 | IMS1         | 2020.043 23:02:53.600000 -02020.043 23:03:37.030997                         |
| IMSVS.RLDSP. IMS1.D20043.T2304155.V00<br>Prilog: 2020.043 23:02:53.600000 | 000000           | 204                  | IMS1         | 2020.043 23:04:15.544239 -02020.043 23:05:20.752699                         |
| IMSVS.RLDSP. IMS1.D20043.T2303370.V00<br>Prilog: 2020.043 23:02:53.600000 | 000000           | 12                   | IMS1         | 2020.043 23:03:37.030997 -02020.043 23:04:15.544239                         |



---

## Chapter 17. Taking an incremental image copy of offline databases

This scenario describes how to recover offline databases by taking an incremental image copy (ICR).

Taking an offline ICR is useful in situations when your databases are offline and you want to produce a new image copy that contains only the information that was updated since the last image copy was taken without performing a full recovery.

Taking an offline ICR is recommended for first-time IMS Database Recovery Facility users.

In this topic:

- [“Identifying the databases to perform incremental image copy” on page 185](#)
- [“Creating and running an IMS Database Recovery Facility job” on page 186](#)
- [“Reviewing the output” on page 187](#)
- [“Common errors and problems” on page 188](#)
- [“Optional: Verifying the ICR in DBRC” on page 188](#)
- [“Next steps” on page 189](#)

---

### Identifying the databases to perform incremental image copy

Before taking an incremental image copy (ICR), you must first identify which databases to perform an ICR.

#### Before you begin

If this is your first time running IMS Database Recovery Facility, take your databases offline by issuing a /DBR command to ensure that all updates are committed and all logs are archived before running ICR.

#### About this task

If you have attempted ICR with IMS Database Recovery Facility and would like to perform a point-in-time ICR where your databases are online, see [Chapter 18, “Taking an incremental image copy of an online database to the current time or point-in-time,” on page 191](#).

#### Procedure

To list and identify which databases to take an ICR of, run either a LIST.RECON or LIST.HISTORY job.

A LIST.RECON job displays the RECON's current status and a formatted display of all records it contains.

A LIST.HISTORY job provides a history-of-activity listing for DBDSs or DEDB areas.

For more information, see the *IMS Database Recovery Control (DBRC) Guide and Reference*.

In the following examples, an ICR is being performed on a HIDAM/VSAM database named DHVNTZ02 with two database data sets: DHVNTZ02.FKXXI01E and DHVNTZ02.FKXXI02E.

#### Example: LIST.RECON job

In this example, a LIST.RECON job is being run to get a detailed DBRC report of DHVNTZ02.

```
//LISTRCON EXEC PGM=DSPURX00,COND=EVEN
//SYSPRINT DD SYSOUT=*
//RECON1 DD DSN=IMSTESTL. IMS.RECON1,DISP=SHR
//RECON2 DD DSN=IMSTESTL. IMS.RECON2,DISP=SHR
//RECON3 DD DSN=IMSTESTL. IMS.RECON3,DISP=SHR
//SYSIN DD *
LIST.RECON
/*
```

### Example: LIST.HISTORY job

In this example, a LIST.HISTORY job is being run to get a detailed DBRC report of DHVNTZ02.

```
//LISTRCON EXEC PGM=DSPURX00,COND=EVEN
//SYSPRINT DD SYSOUT=*
//RECON1 DD DSN=IMSTESTL.IMS.RECON1,DISP=SHR
//RECON2 DD DSN=IMSTESTL.IMS.RECON2,DISP=SHR
//RECON3 DD DSN=IMSTESTL.IMS.RECON3,DISP=SHR
//SYSIN DD *
LIST.HISTORY DBD(DHVNTZ02)
/*
```

### Example: Output

The following is an example of the output from the LIST.HISTORY or LIST.RECON job:

```
-----
DB
DBD=DHVNTZ02                IRLMID=*NULL          DMB#=2          TYPE=IMS
-----
DBDS
DSN=DHVNTZ02.FKXXI01E                                TYPE=IMS
DBD=DHVNTZ02 DDN=HIDAM          DSID=001 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=2          IC AVAIL=0          IC USED=1          DSSN=00000001
NOREUSE          RECOVERPD=0
DEFLTJCL=**NULL** ICJCL=ICJCL          OICJCL=OICJCL          RECOVERJCL=RECOVERJCL
RECVJCL=ICRCVJCL
FLAGS:                                COUNTERS:
  IC NEEDED          =OFF
  RECOVER NEEDED     =OFF
  RECEIVE NEEDED     =OFF                                EEQE COUNT          =0
-----
IMAGE
RUN      = 20.043 22:57:46.7          * RECORD COUNT =49
STOP     = 00.000 00:00:00.0          BATCH          USID=0000000001

IC1
DSN=IMSVS.DHVNTZ02.HIDAM.IC.IC225745          FILE SEQ=0001
UNIT=SYSDA          VOLS DEF=0001 VOLS USED=0001
                   VOLSER=222222
-----
DBDS
DSN=DHVNTZ02.FKXXI02E                                TYPE=IMS
DBD=DHVNTZ02 DDN=HIDAM2          DSID=002 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=2          IC AVAIL=0          IC USED=1          DSSN=00000000
NOREUSE          RECOVERPD=0
DEFLTJCL=**NULL** ICJCL=ICJCL          OICJCL=OICJCL          RECOVERJCL=RECOVERJCL
RECVJCL=ICRCVJCL
FLAGS:                                COUNTERS:
  IC NEEDED          =OFF
  RECOVER NEEDED     =OFF
  RECEIVE NEEDED     =OFF                                EEQE COUNT          =0
-----
IMAGE
RUN      = 20.043 22:57:47.6          * RECORD COUNT =49
STOP     = 00.000 00:00:00.0          BATCH          USID=0000000001

IC1
DSN=IMSVS.DHVNTZ02.HIDAM2.IC.IC225745          FILE SEQ=0001
UNIT=SYSDA          VOLS DEF=0001 VOLS USED=0001
                   VOLSER=222222
```

## Creating and running an IMS Database Recovery Facility job

Perform the following steps to create and run an IMS Database Recovery Facility job that produces an incremental image copy (ICR).

### Before you begin

- To perform an IMS Database Recovery Facility ICR, you must have IMS Database Recovery Facility and IMS High Performance Image Copy installed.
- Complete the tasks in sample member [FRXITKB](#).

## Procedure

1. Install and configure the required IMS Database Recovery Facility load libraries and utilities.

For more information, see [“Installing IMS Recovery Solution Pack libraries” on page 175](#).

2. Customize the procedures found in the *your.hlq.SFRXSAMP* sample library.

For more information, refer to:

- [“DRFMAS procedure” on page 176](#)
- [“DRFPROC procedure” on page 176](#)

3. Customize the execution JCL found in the *your.hlq.SFRXSAMP* sample library.

For more information see [“DRFMAS procedure” on page 176](#).

The following example shows modified JCL to run an IMS Database Recovery Facility ICR:

```
a//DRF      EXEC FRXMAS,DRFMBR=ZZ
//SYSIN DD *
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
SORTPARM(ASPREF(AS))
DBDSL101(DISP(NEW))
bOUTPUT(ICR)
cADD DB(DHVNTZ02) DBATRB(DBDSL(101)) -
dIC (COMP(Y),-
COMPRTN(FABJCOMP3),-
DSNTYPE(BASIC),-
ICHLQ(TEMP.IC),-
ICNMRULE(Y),-
UNIT(SYSDA),-
SPACE(CYL,1,1,RLSE))
START ERROR(CONT)
//
```

### Explanation:

- a. When you run IMS Database Recovery Facility, the FRXMAS procedure is invoked.

DRFMBR=ZZ references the FRXDRFZZ configuration file for IMS Database Recovery Facility.

- b. Instruct IMS Database Recovery Facility to generate an ICR.

- c. The database name DHVNTZ02 (HIDAM/VSAM) is used by the ICR process in the ADD DB() parameter.

You can specify one or more databases to take an ICR.

- d. You can customize the IC keywords to your specifications.

In this case, the output ICR is compressed by using the FABJCOMP3 routine, and the output IC data set name follows the ICNMRULE naming convention (*ichlq.ICn.dbdname.ddname.Dyyddd.Thhmmss*).

4. Submit the IMS Database Recovery Facility job.

## Reviewing the output

This section describes the important parts of the IMS Database Recovery Facility ICR output. Review the output to help you understand the results of running this job.

The IMS Database Recovery Facility IC/ICR output report lists the output copies that were that were generated by HPIC through IMS Database Recovery Facility.

Notice that the ICRs generated by IMS Database Recovery Facility are registered to DBRC as batch image copies because the specified database was not allocated and offline.

The following example shows the output from a sample IMS Database Recovery Facility ICR job.

```
DATABASE RECOVERY FACILITY COMMANDS/CONTROL STATEMENTS
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
FRD7201I SORTPARM(ASPREF(AS))
FRD7201I DBDSL101(DISP(NEW))
FRD7201I OUTPUT(ICR)
```

```

FRD7201I ADD DB(DHVNTZ02) DBATRB(DBDSL(101)) -
FRD7201I IC (COMP(Y),-
FRD7201I COMPTN(FABJCMP3),-
FRD7201I DSNTYPE(BASIC),-
FRD7201I ICHLQ(TEMP.IC),-
FRD7201I ICNMRULE(Y),-
FRD7201I UNIT(SYSDA),-
FRD7201I SPACE(CYL,1,1,RLSE))
FRD7201I START ERROR(CONT)

```

#### D A T A B A S E R E C O V E R Y F A C I L I T Y S U M M A R Y R E P O R T

| Database Name | DD/Area Name | DSID | IC | Records Read CA | LOG  | Records Written | Subord. Reg Name | STC # | Final Status             |
|---------------|--------------|------|----|-----------------|------|-----------------|------------------|-------|--------------------------|
| DHVNTZ02      | HIDAM        | 1    | 49 | 0               | 1501 | 49              | AS069901         | 700   | Delete / define complete |
| DHVNTZ02      | HIDAM2       | 2    | 49 | 0               | 0    | 49              | AS069902         | 701   | Delete / define complete |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 00:50 Page: 2

#### D A T A B A S E R E C O V E R Y F A C I L I T Y U T I L I T Y R E P O R T

| Database Name | DDN    | Database Data Set Name     | IC | PC/DP | IB  | PR  | Utility Final Status |
|---------------|--------|----------------------------|----|-------|-----|-----|----------------------|
| DHVNTZ02      | HIDAM  | IMSTESTL.DHVNTZ02.FKXXI01E | 00 | N/A   | N/A | N/A |                      |
| DHVNTZ02      | HIDAM2 | IMSTESTL.DHVNTZ02.FKXXI02E | 00 | N/A   | N/A | N/A |                      |

Final Return (RC) and Reason (RSN) Codes

| ---IC--- | ---PC--- | ---DP--- | ---IB--- | ---PR--- | ---LIU--- |
|----------|----------|----------|----------|----------|-----------|
| RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN    |
| 00 00    | N/A N/A  | N/A N/A  | N/A N/A  | N/A N/A  | N/A N/A   |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 00:50 Page: 2

#### D A T A B A S E R E C O V E R Y F A C I L I T Y I C / I C R O U T P U T R E P O R T

| Database Name | DD/Area Name | IC / ICR Data Set Name                     | Compr Count | Volume Serial Seq. | File Seq. | IC Type | IC Time           | Compr Ratio | Compr Routine |
|---------------|--------------|--------------------------------------------|-------------|--------------------|-----------|---------|-------------------|-------------|---------------|
| DHVNTZ02      | HIDAM        | TEMP.IC.IC1.DHVNTZ02.HIDAM.D20044.T005017  | 48          | 333333             | 00001     | Bat.    | 20.044 00:50:21.6 | -0800       | 10.1 FABJCMP3 |
| DHVNTZ02      | HIDAM2       | TEMP.IC.IC1.DHVNTZ02.HIDAM2.D20044.T005017 | 48          | 000000             | 00001     | Bat.    | 20.044 00:50:21.6 | -0800       | 7.6 FABJCMP3  |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 00:50 Page: 4

```

NOTIFY.IC DBD(DHVNTZ02) DDN(HIDAM )
RUNTIME('2020.044 00:50:21.6 -08:00')
BATCH
ICDSN(TEMP.IC.IC1.DHVNTZ02.HIDAM.D20044.T005017)
FILESEQ(00001) UNIT(3390)
VOLLIST(333333 )
RECDCT(0000000049)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
NOTIFY.IC DBD(DHVNTZ02) DDN(HIDAM2 )
RUNTIME('2020.044 00:50:21.6 -08:00')
BATCH
ICDSN(TEMP.IC.IC1.DHVNTZ02.HIDAM2.D20044.T005017)
FILESEQ(00001) UNIT(3390)
VOLLIST(000000 )
RECDCT(0000000049)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00

```

## Common errors and problems

You might encounter error messages B37 or E37 due to the output IC running out of space.

To resolve this problem, increase the space allocation size for the output IC by using the SPACE= keyword.

## Optional: Verifying the ICR in DBRC

You can verify that the ICR created by IMS Database Recovery Facility was registered in DBRC as a batch image copy.

### Procedure

To verify the ICR in DBRC:

1. Run the same LIST.RECON or LIST.HISTORY job setup that you ran in [“Identifying the databases to perform incremental image copy”](#) on page 185.
2. Locate the database you chose to ICR in your RECON output job: DHVNTZ02.

For example:

```

-----
DB
DBD=DHVNTZ02                IRLMID=*NULL          DMB#=2          TYPE=IMS
-----
DBDS
DSN=DHVNTZ02.FKXXI01E                                TYPE=IMS

```

```

DBD=DHVNTZ02 DDN=HIDAM DSID=001 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=1 DSSN=00000001
NOREUSE RECOVPD=0
DEFLTJCL=**NULL** ICJCL=ICJCL OICJCL=OICJCL RECOVJCL=RECOVJCL
RECVJCL=ICRCVJCL
FLAGS: COUNTERS:
IC NEEDED =OFF
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF EEQE COUNT =0
-----
IMAGE
RUN = 20.043 22:57:46.7 * RECORD COUNT =49
STOP = 00.000 00:00:00.0 BATCH USID=0000000001

IC1
DSN=IMSVS.DHVNTZ02.HIDAM.IC.IC225745 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

IMAGE
RUN = 20.044 00:50:21.6 * RECORD COUNT =49
STOP = 00.000 00:00:00.0 BATCH USID=0000000000

IC1
DSN=TEMP.IC.IC1.DHVNTZ02.HIDAM.D20044.T005017 FILE SEQ=0001
UNIT=3390 VOLS DEF=0001 VOLS USED=0001
VOLSER=333333
-----
DBDS
DSN=DHVNTZ02.FKXXI02E TYPE=IMS
DBD=DHVNTZ02 DDN=HIDAM2 DSID=002 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=2 IC AVAIL=0 IC USED=1 DSSN=00000000
NOREUSE RECOVPD=0
DEFLTJCL=**NULL** ICJCL=ICJCL OICJCL=OICJCL RECOVJCL=RECOVJCL
RECVJCL=ICRCVJCL
FLAGS: COUNTERS:
IC NEEDED =OFF
RECOV NEEDED =OFF
RECEIVE NEEDED =OFF EEQE COUNT =0
-----
IMAGE
RUN = 20.043 22:57:47.6 * RECORD COUNT =49
STOP = 00.000 00:00:00.0 BATCH USID=0000000001

IC1
DSN=IMSVS.DHVNTZ02.HIDAM2.IC.IC225745 FILE SEQ=0001
UNIT=SYSDA VOLS DEF=0001 VOLS USED=0001
VOLSER=222222

IMAGE
RUN = 20.044 00:50:21.6 * RECORD COUNT =49
STOP = 00.000 00:00:00.0 BATCH USID=0000000002

IC1
DSN=TEMP.IC.IC1.DHVNTZ02.HIDAM2.D20044.T005017 FILE SEQ=0001
UNIT=3390 VOLS DEF=0001 VOLS USED=0001
VOLSER=000000

```

## Next steps

If the ICR was successfully created by IMS Database Recovery Facility, it is now registered to DBRC as a BATCH image copy, and you are now able to bring your databases online and continue running updates.

All future updates (logs and CAs) start from the newly created ICR.

If you perform another ICR or standard recovery, IMS Database Recovery Facility uses the previous image copy (in this scenario, it is the ICR) and uses any archived logs or change accumulation data sets as input.





---

## Chapter 18. Taking an incremental image copy of an online database to the current time or point-in-time

This scenario describes how to recover an online database to the current time or to a point in time by taking an incremental image copy (ICR).

An ICR can be generated to the current or a point-in-time where the updates are being applied.

Taking an online ICR is useful in situations when you do not want to bring your databases offline, but still want to create an ICR.

The ICR process is flexible, providing the ability to generate ICRs to the current time or the point-in-time where updates are being applied.

In this topic:

- [“Identify the databases to perform incremental image copy” on page 191](#)
- [“Creating and running your IMS Database Recovery Facility job” on page 192](#)
- [“Reviewing the output” on page 193](#)
- [“Common errors and problems” on page 194](#)
- [“Optional: Verifying the ICR in DBRC” on page 195](#)
- [“Next steps” on page 196](#)

---

### Identify the databases to perform incremental image copy

Before taking an incremental image copy (ICR), you must first identify which databases to perform an ICR.

#### About this task

- If you would like to perform an ICR where your databases are offline, see [Chapter 17, “Taking an incremental image copy of offline databases,” on page 185](#).
- IMS V10 or later provides greater time precision for the ICR output than earlier versions.

#### Procedure

To list and identify which databases to take an ICR of, run either a LIST.RECON or LIST.HISTORY job.

Running these jobs is helpful when you want to select and define a specific time to perform ICR in your IMS Database Recovery Facility job.

A LIST.RECON job displays the RECON's current status and a formatted display of all records it contains. A LIST.HISTORY job provides a history-of-activity listing for DBDSs or DEDB areas.

For more information, see the *IMS Database Recovery Control (DBRC) Guide and Reference*.

In the following examples, an ICR is being performed on a HIDAM/VSAM database named DHVNTZ02.

#### Example: LIST.RECON job

In this example, a LIST.RECON job is being run to get a detailed DBRC report of DHVNTZ02.

```
//LISTRCON EXEC PGM=DSPURX00,COND=EVEN
//SYSPRINT DD SYSOUT=*
//RECON1 DD DSN=IMSTESTL. IMS.RECON1,DISP=SHR
//RECON2 DD DSN=IMSTESTL. IMS.RECON2,DISP=SHR
//RECON3 DD DSN=IMSTESTL. IMS.RECON3,DISP=SHR
//SYSIN DD *
LIST.RECON
/*
```

### Example: LIST.HISTORY job

In this example, a LIST.HISTORY job is being run to get a detailed DBRC report of DHVNTZ02.

```
//LISTRCON EXEC PGM=DSPURX00,COND=EVEN
//SYSPRINT DD SYSOUT=*
//RECON1 DD DSN=IMSTESTL. IMS.RECON1,DISP=SHR
//RECON2 DD DSN=IMSTESTL. IMS.RECON2,DISP=SHR
//RECON3 DD DSN=IMSTESTL. IMS.RECON3,DISP=SHR
//SYSIN DD *
LIST.HISTORY DBD(DHVNTZ02)
/*
```

### Example: Output

The following is an example of the output from the LIST.HISTORY or LIST.RECON job:

```
+-----+
|Timeline for DBDS: DHVNTZ02 HIDAM
|      USID=00000003   AUTHORIZED=00000003
|      RECEIVE=00000000   HARD=00000003
+-----+
|Time-----+Events-----+-----+
|      |IC|      |US|Subsystem
|      |REORG|      |CA|ID|Logs and Allocs
|      |RECOV|      |ID|
+-----+
20.044 14:03:25.0 B      1
20.044 14:09:14.4      |   IMS1
20.044 14:10:00.4      |   2 A
20.044 14:10:01.5      |   s
20.044 14:10:59.5      |   s
20.044 14:12:15.0      |   s
20.044 14:12:56.2      |   2 D
20.044 14:12:56.7      |   s
20.044 18:24:41.8      |   3 A
20.044 18:24:42.8      |   s
20.044 18:25:37.5      |   s
20.044 18:27:19.3      |   s
+-----+
+-----+
|Timeline for DBDS: DHVNTZ02 HIDAM2
|      USID=00000003   AUTHORIZED=00000003
|      RECEIVE=00000000   HARD=00000003
+-----+
|Time-----+Events-----+-----+
|      |IC|      |US|Subsystem
|      |REORG|      |CA|ID|Logs and Allocs
|      |RECOV|      |ID|
+-----+
20.044 14:03:26.8 B      1
+-----+
```

## Creating and running your IMS Database Recovery Facility job

Perform the following steps to create and run an IMS Database Recovery Facility job that produces an incremental image copy (ICR) of an online database.

### Before you begin

**Important:** Because the database is allocated, the result of the ICR might be a concurrent image copy or fuzzy image copy that can contain committed and uncommitted log updates.

- To perform an IMS Database Recovery Facility ICR, you must have IMS Database Recovery Facility and IMS High Performance Image Copy installed.
- Complete the tasks in sample member [FRXITKB](#).

### About this task

This scenario uses the time “20.044 18:25:37.5,” which was extracted from the RECON that has updates being processed.

### Procedure

1. Install and configure the required IMS Database Recovery Facility load libraries and utilities.

For more information, see [“Installing IMS Recovery Solution Pack libraries” on page 175](#)

2. Customize the procedures found in the *your.hlq.SFRXSAMP* sample library.

For more information, refer to:

- [“DRFMAS procedure” on page 176](#)
- [“DRFPROC procedure” on page 176](#)

3. Customize the execution JCL found in the *your.hlq.SFRXSAMP* sample library.

For more information, see [“DRFMAS procedure” on page 176](#).

The following example shows modified IMS Database Recovery Facility ICR JCL in which an ICR is being taken with the database allocated:

```
a//DRF      EXEC FRXMAS,DRFMBR=ZZ
//SYSIN DD *
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
SORTPARM(ASPREF(AS))
DBDSL101(DISP(NEW))
bOUTPUT(ICR)
  cADD DB(DHVNTZ02) DBATRB(DBDSL(101)) -
  dIC (COMP(Y),-
    COMPRTN(FABJCOMP3),-
      DSNTYPE(BASIC),-
      ICHLQ(TEMP.IC),-
      ICNMRULE(Y),-
      UNIT(SYSDA),-
      SPACE(CYL,1,1,RLSE))
  eSTART ERROR(CONT) RCVTIME('20.044 18:25:37.5')
//
```

#### Explanation:

- a. When you run IMS Database Recovery Facility, the FRXMAS procedure is invoked.

DRFMBR=ZZ references the FRXDRFZZ configuration file for IMS Database Recovery Facility.

- b. Instruct IMS Database Recovery Facility to generate an ICR.

- c. The database name DHVNTZ02 (HIDAM/VSAM), which is identified in step [“1” on page 192](#), is used to take an ICR by using the ADD DB() parameter.

You can specify one or more databases for taking an ICR.

- d. You can customize the IC keywords to your specifications.

In this case, the output ICR is compressed by using the FABJCOMP3 routine, and the output IC data set name follows the ICNMRULE naming convention (*ichlq.ICn.dbdname.ddname.Dyyddd.Thhmmss*).

- e. Specifying RCVTIME() is optional.

If you want to run ICR to current time, you do not need to include this parameter because IMS Database Recovery Facility automatically calls HPIC to generate concurrent and fuzzy ICs for the allocated databases that are specified in the ADD DB() statement.

However, if you want to generate a point-in-time ICR that is not the current time, you must input the specific time by using RCVTIME().

In this scenario, the chosen time was inserted.

4. Submit the IMS Database Recovery Facility job.

## Reviewing the output

This section describes the important parts of the IMS Database Recovery Facility ICR output. Review the output to help you understand the results of running this job.

The following example shows the output from a sample IMS Database Recovery Facility ICR job.

One of the ICRs that was generated was a CIC or concurrent image copy and was successfully registered to DBRC as a CIC image copy because the database was allocated and had updates.

#### D A T A B A S E   R E C O V E R Y   F A C I L I T Y   C O M M A N D S / C O N T R O L   S T A T E M E N T S

```
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
FRD7201I SORTPARM(ASPREF(AS))
FRD7201I DBDSL101(DISP(NEW))
FRD7201I OUTPUT(ICR)
FRD7201I ADD DB(DHVNTZ02) DBATRB(DBDSL(101)) -
FRD7201I IC (COMP(Y),-
FRD7201I COMPTN(FABJCOMP3),-
FRD7201I DSNTYPE(BASIC),-
FRD7201I ICHLQ(TEMP.IC),-
FRD7201I ICNMRULE(Y),-
FRD7201I UNIT(SYSDA),-
FRD7201I SPACE(CYL,1,1,RLSE))
FRD7201I START ERROR(CONT) RCVTIME('20.044 18:25:37.5')
```

#### D A T A B A S E   R E C O V E R Y   F A C I L I T Y   S U M M A R Y   R E P O R T

| Database Name | DD/Area Name | DSID | ----- IC | Records Read CA | ----- LOG | Records Written | Subord. Reg Name | STC # | Final Status          |
|---------------|--------------|------|----------|-----------------|-----------|-----------------|------------------|-------|-----------------------|
| DHVNTZ02      | HIDAM        | 1    | 49       | 0               | 3001      | 0               | AS013901         | 140   | No errors encountered |
| DHVNTZ02      | HIDAM2       | 2    | 49       | 0               | 0         | 0               | AS013902         | 141   | No errors encountered |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 18:39 Page: 2

#### D A T A B A S E   R E C O V E R Y   F A C I L I T Y   U T I L I T Y   R E P O R T

| Database | DDN    | Database Data Set Name     | IC | PC/DP | IB  | PR  | Utility Final Status |
|----------|--------|----------------------------|----|-------|-----|-----|----------------------|
| DHVNTZ02 | HIDAM  | IMSTESTL.DHVNTZ02.FKXXI01E | 00 | N/A   | N/A | N/A |                      |
| DHVNTZ02 | HIDAM2 | IMSTESTL.DHVNTZ02.FKXXI02E | 00 | N/A   | N/A | N/A |                      |

Final Return (RC) and Reason (RSN) Codes

| ---IC--- | ---PC--- | ---DP--- | ---IB--- | ---LIU--- |
|----------|----------|----------|----------|-----------|
| RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN    |
| 00 00    | N/A N/A  | N/A N/A  | N/A N/A  | N/A N/A   |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 18:39 Page: 2

#### D A T A B A S E   R E C O V E R Y   F A C I L I T Y   I C / I C R   O U T P U T   R E P O R T

| Database Name | DD/Area Name | IC / ICR Data Set Name                     | Compr Count | Volume Serial | File Seq. | IC Type | IC Time           | Compr Ratio | Compr Routine  |
|---------------|--------------|--------------------------------------------|-------------|---------------|-----------|---------|-------------------|-------------|----------------|
| DHVNTZ02      | HIDAM        | TEMP.IC.IC1.DHVNTZ02.HIDAM.D20044.T183934  | 48          | 222222        | 00001     | Con.    | 20.044 18:25:37.5 | -0800       | 10.1 FABJCOMP3 |
| DHVNTZ02      | HIDAM2       | TEMP.IC.IC1.DHVNTZ02.HIDAM2.D20044.T183934 | 48          | 333333        | 00001     | Bat.    | 20.044 18:25:37.5 | -0800       | 7.6 FABJCOMP3  |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 18:39 Page: 4

#### D A T A B A S E   R E C O V E R Y   F A C I L I T Y   D A T A   S E T   I / O   R E P O R T

Recover to point: 2020.044 18:25:37.500000

| Image Copy Data Set Name          | Volume Serial | IC DS Read Count | IC Type | ----- Time Stamp Range ----- |
|-----------------------------------|---------------|------------------|---------|------------------------------|
|                                   |               |                  |         | 1st Record Last Record       |
| IMSVS.DHVNTZ02.HIDAM2.IC.IC140323 | 222222        | 49               | STD     |                              |
| IMSVS.DHVNTZ02.HIDAM2.IC.IC140323 | 222222        | 49               | STD     |                              |

```
NOTIFY.IC DBD(DHVNTZ02) DDN(HIDAM )
RUNTIME('2020.044 18:25:37.5 -08:00')
STOPTIME('2020.044 18:25:37.5 -08:00')
CIC
ICDSN(TEMP.IC.IC1.DHVNTZ02.HIDAM.D20044.T183934)
FILESEQ(0001) UNIT(3390)
VOLLIST(222222 )
RECDCT(0000000049)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
NOTIFY.IC DBD(DHVNTZ02) DDN(HIDAM2 )
RUNTIME('2020.044 18:25:37.5 -08:00')
BATCH
ICDSN(TEMP.IC.IC1.DHVNTZ02.HIDAM2.D20044.T183934)
FILESEQ(0001) UNIT(3390)
VOLLIST(333333 )
RECDCT(0000000049)
DSP0203I COMMAND COMPLETED WITH CONDITION CODE 00
```

## Common errors and problems

You might encounter error messages B37 or E37 due to the output IC running out of space.

To resolve this problem, increase the space allocation size for the output IC by using the SPACE= keyword.

## Optional: Verifying the ICR in DBRC

You can verify that the ICR created by IMS Database Recovery Facility was registered in DBRC as a concurrent image copy.

### Procedure

To verify the ICR in DBRC:

1. Run the same LIST.RECON or LIST.HISTORY job setup that you ran in [“Identify the databases to perform incremental image copy”](#) on page 191.
2. Locate the database you chose to ICR in your RECON output job: DHVNTZ02.

For example:

```
-----
DB
DBD=DHVNTZ02                IRLMID=*NULL        DMB#=2        TYPE=IMS
-----
DBDS
DSN=DHVNTZ02.FKXXI01E                TYPE=IMS
DBD=DHVNTZ02 DDN=HIDAM        DSID=001 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=2        IC AVAIL=0        IC USED=1        DSSN=000000001
NOREUSE        RECOVPD=0
DEFLTJCL=**NULL** ICJCL=ICJCL        OICJCL=OICJCL        RECOVJCL=RECOVJCL
RECVJCL=ICRCVJCL
FLAGS:                COUNTERS:
IC NEEDED        =OFF
RECOV NEEDED     =OFF
RECEIVE NEEDED   =OFF                EEQE COUNT                =0
-----
IMAGE
RUN      = 20.044 14:03:25.0                * RECORD COUNT =49
STOP     = 00.000 00:00:00.0                BATCH        USID=00000000001
-----
IC1
DSN=IMSVS.DHVNTZ02.HIDAM.IC.IC140323        FILE SEQ=0001
UNIT=SYSDA                VOLS DEF=0001 VOLS USED=0001
VOLSER=222222
-----
IMAGE
RUN      = 20.044 18:25:37.5                * RECORD COUNT =49
STOP     = 20.044 18:25:37.5                CONCUR        USID=00000000003
-----
IC1
DSN=TEMP.IC.IC1.DHVNTZ02.HIDAM.D20044.T183934        FILE SEQ=0001
UNIT=3390                VOLS DEF=0001 VOLS USED=0001
VOLSER=222222
-----
DBDS
DSN=DHVNTZ02.FKXXI02E                TYPE=IMS
DBD=DHVNTZ02 DDN=HIDAM2        DSID=002 DBORG=HIDAM DSORG=VSAM
CAGRP=**NULL** GENMAX=2        IC AVAIL=0        IC USED=1        DSSN=000000000
NOREUSE        RECOVPD=0
DEFLTJCL=**NULL** ICJCL=ICJCL        OICJCL=OICJCL        RECOVJCL=RECOVJCL
RECVJCL=ICRCVJCL
FLAGS:                COUNTERS:
IC NEEDED        =OFF
RECOV NEEDED     =OFF
RECEIVE NEEDED   =OFF                EEQE COUNT                =0
-----
IMAGE
RUN      = 20.044 14:03:26.8                * RECORD COUNT =49
STOP     = 00.000 00:00:00.0                BATCH        USID=00000000001
-----
IC1
DSN=IMSVS.DHVNTZ02.HIDAM2.IC.IC140323        FILE SEQ=0001
UNIT=SYSDA                VOLS DEF=0001 VOLS USED=0001
VOLSER=222222
-----
IMAGE
RUN      = 20.044 18:25:37.5                * RECORD COUNT =49
STOP     = 00.000 00:00:00.0                BATCH        USID=00000000003
-----
IC1
DSN=TEMP.IC.IC1.DHVNTZ02.HIDAM2.D20044.T183934        FILE SEQ=0001
UNIT=3390                VOLS DEF=0001 VOLS USED=0001
VOLSER=333333
```

## Next steps

---

If the ICR was successfully created by IMS Database Recovery Facility, it is now registered to DBRC as a CONCUR image copy, and you are now able to continue running updates on your online or allocated databases.

All future updates (logs and CAs) start from the newly created ICR.

If you perform another ICR or standard recovery, IMS Database Recovery Facility uses the previous image copy (in this scenario, it is the ICR) and uses any archived logs or change accumulation data sets as input.

---

## Chapter 19. Using IMS Index Builder to rebuild indexes during recovery

In this scenario, IMS Database Recovery Facility is used to run IMS Index Builder against HALDBs to rebuild the indexes.

The database that is used in this scenario is a PHIDAM or OSAM database with four partitions and one secondary index with four partitions.

When you include the IB() keyword in the IMS Database Recovery Facility ADD statement, you are instructing IMS Database Recovery Facility to run IMS Index Builder for HALDBs to rebuild your indexes.

You can include many optional keywords in the IB() statement, however, for the purposes of this scenario the standard defaults are used.

In this topic:

- [“Creating and running an IMS Database Recovery Facility job” on page 197](#)
- [“Reviewing the output” on page 198](#)
- [“Common errors and problems” on page 199](#)

---

### Creating and running an IMS Database Recovery Facility job

Perform the following steps to create and run an IMS Database Recovery Facility job that runs IMS Index Builder (IB) against HALDBs to rebuild the indexes.

#### Before you begin

- To rebuild the indexes, you must have IMS Database Recovery Facility and IMS Index Builder (IB) installed.
- Complete the tasks in sample member [FRXITKB](#).

#### Procedure

1. Install and configure the required IMS Database Recovery Facility load libraries and utilities.

For more information, see [“Installing IMS Recovery Solution Pack libraries” on page 175](#)

2. Customize the procedures found in the *your.hlq.SFRXSAMP* sample library.

For more information, refer to:

- [“DRFMAS procedure” on page 176](#)
- [“DRFPROC procedure” on page 176](#)
- [“DRFIAX procedure” on page 176](#)

3. Customize the execution JCL found in the *your.hlq.SFRXSAMP* sample library.

For more information, see [“DRFMAS procedure” on page 176](#).

The following example shows modified IMS Database Recovery Facility JCL to run a recovery and use IB to rebuild the indexes:

```
//IMS Database Recovery Facility      EXEC FRXMAS,DRFMBR=ZZ
//SYSIN DD *
@DRFIAX(FRXJCLIP)
REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
SORTPARM(ASPREF(AS))
DBDSL101(DISP(NEW))
  OUTPUT(PRO)
  @ADD DB(DBOHIDK5) IB(HALDB=BOTH,BLD_SECONDARY(ALL)) -
    DBATRB(DBDSL(101))
```

```
START ERROR(CONT)
//
```

### Explanation:

- DRFIAX specifies the name of the procedure that is used to initiate the utility address space for IB.
- IMS Database Recovery Facility recovers the DBOHIDK5 (PHIDAM/OSAM) database and rebuilds the primary index and ILE, including rebuilding all of the secondary indexes by using the IB() keyword.

You can customize the IB keywords to your specifications.

- Submit the IMS Database Recovery Facility job.

## Reviewing the output

This section describes the important parts of the IMS Database Recovery Facility output. Review the output to help you understand the results of running this job.

The IMS Database Recovery Facility utility report lists the return codes for the primary indexes and ILEs that were rebuilt by IMS Index Builder, including the rebuilt secondary indexes.

The following example shows the output from a sample IMS Database Recovery Facility/IMS Index Builder job.

### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I DRFIAX(FRXJCLIP)
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=MYDRF1)
FRD7201I SORTPARM(ASPREF(AS))
FRD7201I DBDSL101(DISP(NEW))
FRD7201I OUTPUT(PRO)
FRD7201I ADD DB(DBOHIDK5) IB(HALDB=BOTH,BLD_SECONDARY(ALL)) -
FRD7201I DBATRB(DBDSL(101))
FRD7201I START ERROR(CONT)
```

### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | ----- IC | Records Read CA | ----- LOG | Records Subord. Written Reg Name | STC # | Final Status           |
|---------------|--------------|------|----------|-----------------|-----------|----------------------------------|-------|------------------------|
| POHIDKA       | POHIDKAA     | 1    | 36       | 0               | 0         | 36 AS029501                      | 296   | Delete/define complete |
| POHIDKB       | POHIDKBA     | 1    | 2        | 0               | 0         | 2 AS029502                       | 297   | Delete/define complete |
| POHIDKC       | POHIDKCA     | 1    | 2        | 0               | 0         | 2 AS029503                       | 298   | Delete/define complete |
| POHIDKD       | POHIDKDA     | 1    | 36       | 0               | 0         | 36 AS029501                      | 296   | Delete/define complete |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 23:27 Page: 2

### DATABASE RECOVERY FACILITY UTILITY REPORT

| Database | DDN      | Database Data Set Name     | IC  | PC/DP | IB  | PR  | Utility Final Status |
|----------|----------|----------------------------|-----|-------|-----|-----|----------------------|
| POHIDKA  | POHIDKAA | IMSTESTS.DBOHIDK5.A00001   | N/A | N/A   | N/A | N/A |                      |
| POHIDKB  | POHIDKBA | IMSTESTS.DBOHIDK5.A00002   | N/A | N/A   | N/A | N/A |                      |
| POHIDKC  | POHIDKCA | IMSTESTS.DBOHIDK5.A00003   | N/A | N/A   | N/A | N/A |                      |
| POHIDKD  | POHIDKDA | IMSTESTS.DBOHIDK5.A00004   | N/A | N/A   | N/A | N/A |                      |
| POHIDKA  | PRIMINDX | IMSTESTS.DBOHIDK5.X00001   | N/A | N/A   | 00  | N/A |                      |
| POHIDKB  | PRIMINDX | IMSTESTS.DBOHIDK5.X00002   | N/A | N/A   | 00  | N/A |                      |
| POHIDKC  | PRIMINDX | IMSTESTS.DBOHIDK5.X00003   | N/A | N/A   | 00  | N/A |                      |
| POHIDKD  | PRIMINDX | IMSTESTS.DBOHIDK5.X00004   | N/A | N/A   | 00  | N/A |                      |
| POHIDKA  | ILDS/ILE | IMSTESTS.DBOHIDK5.L00001   | N/A | N/A   | 00  | N/A |                      |
| POHIDKB  | ILDS/ILE | IMSTESTS.DBOHIDK5.L00002   | N/A | N/A   | 00  | N/A |                      |
| POHIDKC  | ILDS/ILE | IMSTESTS.DBOHIDK5.L00003   | N/A | N/A   | 00  | N/A |                      |
| POHIDKD  | ILDS/ILE | IMSTESTS.DBOHIDK5.L00004   | N/A | N/A   | 00  | N/A |                      |
| P2XIDKD  | P2XIDKDA | (Partition Index Database) | N/A | N/A   | 00  | N/A |                      |

### Final Return (RC) and Reason (RSN) Codes

| ---IC--- | ---PC--- | ---DP--- | ---IB--- | ---PR--- | ---LIU--- |
|----------|----------|----------|----------|----------|-----------|
| RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN    |
| N/A N/A  | N/A N/A  | N/A N/A  | 00 N/A   | N/A N/A  | N/A N/A   |

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 02/13/2020 Time: 23:27 Page: 3



## Common errors and problems

---

To avoid possible IMS Index Builder errors when you attempt to rebuild your indexes, you should define your indexes as REUSE.



---

## Chapter 20. Recovering HALDB databases that use online reorganization

Follow the instructions in the following sections to use IMS Database Recovery Facility to recover HALDB databases that use online reorganization.

### About this task

IMS Database Recovery Facility supports HALDB recoveries across an OLR boundary or "into" an OLR (between OLR start and stop times). IMS Database Recovery Facility also fully supports IMS Reorg verification if your RECONS are registered with REORGV.

#### Notes:

- Whenever a HALDB is recovered to a previous time (regardless of whether or not it crossed an OLR boundary or was "into an OLR"), all indexes/ILDSs must be rebuilt either with IMS Index Builder or DFSPRECO (load mode).
- It is recommended that you register your RECONS with REORGV so the reorganization number verification is enabled.

In this topic:

- [“Recovering HALDB databases "into an OLR"” on page 201](#)
- [“Recovering HALDB databases "prior" to an OLR” on page 202](#)
- [“Recovering HALDB databases "after" an OLR ” on page 203](#)

---

## Recovering HALDB databases "into an OLR"

This section describes using IMS Database Recovery Facility for HALDB recoveries that are within an OLR (that is, between an OLR start and stop times).

### About this task

By specifying an RCVTIME that is greater than the start time and less than the stop time of an OLR recorded in DBRC, you can recover to a point-in-time within the OLR. In this case, IMS Database Recovery Facility will recover both the active and inactive data sets to that point-in-time.

After the recovery, the ACTIVE DBDS will be set to BOTH and OLREORG CURSOR ACTIVE will be set to YES.

The indexes/ILDSs must be rebuilt prior to manually resuming the OLR.

#### Notes:

- A recovery into an OLR must be a Point-In-Time recovery (RCVTIME('timestamp',PITR)).
- You cannot image copy or pointer check the databases during this type of recovery and/or before the OLR is resumed/completed.

### Procedure

1. **Note:** It must be a PITR type recovery (RCVTIME(timestamp,PITR)).

Specify an IMS Database Recovery Facility RCVTIME that is greater than the OLR start time and less than the OLR stop time that is recorded in DBRC.

2. **Notes:**

- The databases must either be /DBRD prior to the recovery or specify DBRCMD(GLOBAL) and STACMD(GLOBAL) on the **START** command.

The indexes/ILDSs must be rebuilt for both active and inactive data sets.

It is recommended to specify HALDB(BOTH) and ICNDX(N) as well as IB(BLD\_PRIMARY,BLD\_SECONDARY(ALL)) and IC(...) statements to perform these tasks under IMS Database Recovery Facility.

- You cannot run IC and PC when performing a recovery "into" an OLR. Remove any IC or PC statement.

Run IMS Database Recovery Facility.

3. Verify that the ACTIVE DBDS flag was set to BOTH.
4. Verify that the OLREORG CURSOR ACTIVE flag was set to YES.
5. Verify that the IMAGE COPY NEEDED COUNT is 0 and/or IC NEEDED flag was set to OFF for all database components (partition data sets, ILDSs, primary indexes if PHIDAM, and secondary indexes if applicable).

**Note:** IC RECOMMENDED flag may be set to ON.

6. Verify that the RECOVERY NEEDED COUNT is 0 and/or RECOV NEEDED flag was set to OFF for all database components.
7. Verify that the databases are started.
8. Resume the OLR.
9. Verify that the OLR was successfully resumed by looking at the SYSTEM logs.
10. Verify that the databases can be accessed online.

## Recovering HALDB databases "prior" to an OLR

---

This section describes using IMS Database Recovery Facility for HALDB recoveries that are prior to any OLR recorded in DBRC (that is, before an OLR start time).

### About this task

By specifying an RCVTIME that is equal to or less than the OLR start time recorded in DBRC, you can recover to a state prior to when the OLR began. After the recovery, the ACTIVE DBDS is set to whichever data sets were active at that recov-to-time and OLREORG CURSOR ACTIVE is set to NO.

If the RCVTIME exactly matches the OLR start time recorded in DBRC, it is not considered to be "into" an OLR. The data set is recovered to a time just prior to the OLR starting. The ACTIVE DBDS is set to whichever data set was active before the OLR started and OLREORG CURSOR ACTIVE is set to NO.

### Notes:

- You are not restricted to recovering back to the most recent OLR.
- Any recovery of a HALDB partition which crosses an OLR boundary causes the Partition REORG# to be incremented in the RECON.

### Procedure

1. Specify an IMS Database Recovery Facility RCVTIME that is equal to or less than the OLR start time.
2. **Note:** The databases must either be /DBRD prior to the recovery or specify DBRCMD(GLOBAL) and STACMD(GLOBAL) on the IMS Database Recovery Facility **START** command. The indexes/ILDSs must be rebuilt and all recoverable data sets must be image copied. It is recommended to specify HALDB(BOTH) and ICNDX(Y) as well as IB(BLD\_PRIMARY,BLD\_SECONDARY(ALL)) and IC(...) statements to perform these tasks under IMS Database Recovery Facility. The PC statement is recommended if you have the IMS pointer checker tool(s).

Run IMS Database Recovery Facility.

3. Verify that the ACTIVE DBDS flag was set to whichever data set was previously active for that time stamp.
4. Verify that the OLREORG CURSOR ACTIVE flag was set to NO.

5. Verify that the ICNEEDED COUNT is 0 and/or ICNEEDED was set to OFF for all database components (partition data sets, ILDSs, primary indexes if PHIDAM, and secondary indexes if applicable).
6. Verify that the RECOVERY NEEDED COUNT is 0 and/or RECOV NEEDED flag was set to OFF for all database components.
7. Verify that the databases are started and can be accessed online.

## Recovering HALDB databases "after" an OLR

---

This section describes using IMS Database Recovery Facility for HALDB recoveries that are after any OLR recorded in DBRC (that is, after an OLR stop time).

### About this task

By specifying an RCVTIME that is equal to or greater than an OLR stop time recorded in DBRC, you can recover to a state after the OLR completed. After the recovery, the ACTIVE DBDS is set to whichever data sets were active at that recov-to-time and OLREORG CURSOR ACTIVE is set to NO.

If the RCVTIME exactly matches the OLR stop time recorded in DBRC, it is not considered to be "into the OLR". The data set is recovered to a time just after the OLR completed. The ACTIVE DBDS is set to whichever data sets were active after the OLR completed and OLREORG CURSOR ACTIVE is set to NO.

### Notes:

- You are not restricted to recovering back to the most recent OLR.
- Any recovery of a HALDB partition which crosses an OLR boundary causes the Partition REORG# to be incremented in the RECON.

### Procedure

1. Specify an IMS Database Recovery Facility RCVTIME that is equal to or greater than the OLR stop time.
2. **Note:** The databases must either be /DBRD prior to the recovery or specify DBRCMD(GLOBAL) on the **START** command. The indexes/ILDSs must be rebuilt and all recoverable data sets must be image copied if it is not a full recovery. It is recommended to specify HALDB(BOTH) and ICNDX(Y) as well as IB(BLD\_PRIMARY,BLD\_SECONDARY(ALL)) and IC(...) statements to perform these tasks under IMS Database Recovery Facility. The PC statement is recommended if you have the IMS pointer checker tool(s).

Run IMS Database Recovery Facility.

3. Verify that the ACTIVE DBDS flag was set to whichever data set was previously active for that time stamp.
4. Verify that the OLREORG CURSOR ACTIVE flag is set to NO.
5. Verify that the IMAGE COPY NEEDED COUNT is 0 and/or IC NEEDED flag was set to OFF for all database components (partition data sets, ILDSs, primary indexes if PHIDAM, and secondary indexes if applicable).
6. Verify that the RECOVERY NEEDED COUNT is 0 and/or RECOV NEEDED flag was set to OFF for all database components.
7. Verify that the databases are started and can be accessed online.



---

## Part 6. Troubleshooting

IMS Database Recovery Facility issues messages and codes that can help you to diagnose and correct problems that you experience with the product.

**Topics:**

- [Chapter 21, “Runtime messages \(FRD\),” on page 207](#)
- [Chapter 22, “Abend codes,” on page 263](#)
- [Chapter 23, “Batch condition codes,” on page 273](#)
- [Chapter 24, “Gathering diagnostic information,” on page 275](#)





---

## Chapter 21. Runtime messages (FRD)

This topic describes the runtime messages that are issued by IMS Database Recovery Facility.

### Message format

The explanations and user responses provided in this reference can help you diagnose, troubleshoot, and solve IMS Database Recovery Facility problems.

IMS Database Recovery Facility messages adhere to the following format:

```
FRDnnnnx
```

where:

#### **FRD**

Indicates that the message was issued by IMS Database Recovery Facility

#### **nnnn**

Indicates the message identification number

#### **x**

Indicates the severity of the message:

##### **A**

Indicates that operator intervention is required before processing can continue.

##### **E**

Indicates that an error occurred, which might or might not require operator intervention.

##### **I**

Indicates that the message is informational only.

##### **W**

Indicates that the message is a warning to alert you to a possible error condition.

Each message also includes the following information:

#### **Explanation:**

The Explanation section explains what the message text means, why it occurred, and what its variables represent.

#### **System action:**

The System action section explains what the system will do in response to the event that triggered this message.

#### **User response:**

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

#### **Module**

The Module section indicates which module or modules are affected.

#### **Message Variables**

In the message text, there can be lowercase variables (for example, xxx...). The variables represent values when the message appears such as:

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

---

**FRD0000I**      **IMS RECOVERY SOLUTION PACK**  
**VxRy: IMS DATABASE RECOVERY**  
**FACILITY Date: mm/dd/yyyy Time:**  
**hh:mm Page: nnn**

#### Explanation

This message displays the product, version, date, time, and page number on the first line of each new page of the IMS Database Recovery Facility reports.

#### System action

Processing continues.

#### User response

None. This message is informational.

#### Module

FRXGRPT0

---

**FRD0001I**      *asname msgnum msgtext*

#### Explanation

An IMS Database Recovery Facility subordinate address space, indicated by *asname*, issued a message indicated by *msgnum*. The message is sent to the destination identified by the IMS Database Recovery Facility master address space SYSPRINT DD statement.

#### *asname*

The IMS Database Recovery Facility subordinate address space jobname.

#### *msgnum*

The message number issued by the IMS Database Recovery Facility subordinate address space.

#### *msgtext*

The message text of the message issued by the IMS Database Recovery Facility subordinate address space.

#### System action

Processing continues.

#### Module

FRXFDSR0

---

**FRD1000I**      **IRP/DRF VxRy: IMS DATABASE**  
**RECOVERY FACILITY**  
**INITIALIZATION COMPLETE**

#### Explanation

The message indicates the version of IMS Recovery Solution Pack: IMS Database Recovery Facility in use and that initialization is complete.

The message destination is the z/OS system console.

#### System action

Processing continues.

#### Module

FRXMSTR0

---

**FRD1002I**      **IRP/DRF VxRy: IMS DATABASE**  
**RECOVERY FACILITY SAS**  
**INITIALIZATION COMPLETE**

#### Explanation

The message indicates the version of IMS Database Recovery Facility subordinate address space. The message is sent to the z/OS system console.

The message destination is the z/OS system console.

#### System action

Processing continues.

#### Module

FRXMSTR1

---

**FRD1004E**      **DBRC INITIALIZATION FAILED**  
**reason, RC=return code**

#### Explanation

Initialization of the DBRC interface detected an error.

#### *reason*

Identifies the problem and is one of the following occurrences:

- UNABLE TO OBTAIN STORAGE
- INVALID FEEDBACK FROM DBRC

#### *return code*

Represents internal diagnostic information.

#### System action

The address space shuts down and recovery processing is halted.

#### User response

Determine if the IMS Database Recovery Facility address space REGION is too small; if so, increase the size.

**Module**

FRXDTINO

---

**FRD1005E      DRF SAS STARTUP FAILED *reason*,  
RC=*return code***


---

**Explanation**

Startup processing for the IMS Database Recovery Facility subordinate address space failed while attempting to initialize the environment.

***reason***

Identifies the problem and is one of the following occurrences:

- ATTACH PIPE RECEIVE TCB
- ATTACH PIPE SEND TCB
- CREATE PIPE RECEIVE THREAD
- CREATE PIPE SEND THREAD
- CREATE DATA SPACE MANAGER THREAD
- CREATE IMAGE COPY THREAD
- CREATE DATA MANAGER THREAD
- CREATE SORT MANAGER THREAD

***return code***

Represents internal diagnostic information.

**System action**

The address space shuts down and the recovery attempt stops.

**User response**

Determine if the IMS Database Recovery Facility subordinate address space REGION is too small; if so, increase the size.

**Module**

FRXMSTR1

---

**FRD1006A      LOAD FAILED FOR *modulename***


---

**Explanation**

Startup processing for the IMS Database Recovery Facility failed while attempting to load the initialization or termination modules during initial startup processing.

***modulename***

Identifies the module that could not be loaded.

**System action**

The address space shuts down and the recovery attempt stops.

**User response**

Determine if the load module named in the message is included in a partitioned data set which is part of the STEPLIB concatenation.

**Module**

FRXSDR00

---

**FRD1007I      UNABLE TO OPEN *dd* DATA SET**


---

**Explanation**

Unable to open data set of sysprint|report|frxdebug.  
*dd* may be one of the following:

- SYSPRINT
- REPORT
- FRXDEBUG

**System action**

Processing continues.

**User response**

Ensure that the *dd* is specified in the JCL procedure.

**Module**

FRXSDR00

---

**FRD1008A      PROGRAM IS NOT AUTHORIZED**


---

**Explanation**

The IMS Database Recovery Facility program is not authorized to z/OS.

**System action**

The IMS Database Recovery Facility ends with a return code of 12.

**User response**

First, determine that each data set in the STEPLIB and JOBLIB concatenation is APF authorized. Then, ensure that the IMS Database Recovery Facility is an authorized program. After performing these two tasks you might submit the recovery job.

**Module**

FRXSDR00

---

**FRD1009I      USERID *userid* PROPAGATED  
FROM DRF MASTER JOB**


---

**Explanation**

The userid propagation processing is in effect and this message indicates that the userid that was used to submit the IMS Database Recovery Facility master job was propagated to this subordinate address space. This means that this address space will execute with the same level of authority as the IMS Database Recovery Facility master job. This process is controlled by the OPTION(USERPROP()) keyword.

**userid**

Identifies the userid that was associated with the IMS Database Recovery Facility master job and has been propagated to this subordinate address space.

**System action**

Processing continues.

**User response**

None. This message is informational.

**Module**

FRXSDR20, FRXICTL0

---

**FRD1011I      DATABASE RECOVERY  
TERMINATING, reason**

**Explanation**

The IMS Database Recovery Facility is starting the ending process. The *reason* that is displayed indicates the reason that the process is ending and is one of the following:

- INITIALIZATION FAILURE
- NORMAL END OF RECOVERY
- EARLY END OF RECOVERY
- END OF RECOVERY WITH ERRORS
- REQUESTED BY IMS

**System action**

The IMS Database Recovery Facility address space ends. END OF RECOVERY WITH ERRORS means that one or more database data sets failed recovery; however, other database data sets in the recovery list were allowed to complete successfully.

**User response**

If the reason is INITIALIZATION FAILURE, or EARLY END OF RECOVERY, or END OF RECOVERY WITH ERRORS, examine accompanying messages and take the action indicated by those messages.

**Module**

FRXMTRM0

---

**FRD1012E      IMS AND DBDLIB SPECIFIED**

**Explanation**

If Integrated Auxiliary Processing is called, this message is issued when both IMS and DBDLIB DDs are specified for an IMS Database Recovery Facility subordinate address space.

**System action**

The address shuts down, and recovery processing is halted.

**User response**

Ensure that you specify either IMS DDs or DBDLIB DDs, but not both.

**Module**

FRXSDR00

---

**FRD1013E      vartxt ERROR RC=rc**

**Explanation**

IMS Database Recovery Facility has failed in an attempt to process a load library data set. *vartxt* identifies one of the following reasons for the failure:

- OPEN - Load library data set failed on open.
- LOAD - Load library data set failed on load.

**System action**

IMS Database Recovery Facility might continue processing.

**User response**

Identify the load library data set that failed and determine its availability.

**Module**

FRXMDDL0

---

**FRD1014I      UNABLE TO OPEN FRXPRINT  
DATA SET**

**Explanation**

If Integrated Auxiliary Processing is called and FRXPRINT cannot be opened, this message is issued for an IMS Database Recovery Facility subordinate address space.

**User response**

Processing continues.

**Module**

FRXSDR00

---

**FRD2807E      LOG READ PHASE *action***

**Explanation**

The log data set read phase of recovery has completed either normally or encountered an error and ends early. In the message text the following values apply:

***action***

Indicates the success or failure of the log read phase.

**COMPLETED**

The log read phase completed normally.

**EARLY END**

The log read phase encountered an error and ended early.

**System action**

If *action* is COMPLETE, processing continues. If *action* is EARLY END, recovery ends early.

**User response**

If *action* is EARLY END, examine accompanying messages to determine the type of error. Correct the problem and re-initiate recovery.

**Module**

FRXRCTL0

---

**FRD2885I      WRITE TERMINATION COMPLETE**

**Explanation**

Ending of the connection sending data from the IMS Database Recovery Facility address space and the IMS control region is complete.

This message is only issued when the IMS Database Recovery Facility is started by the IMS control region. The message destination is the z/OS system console.

**System action**

None.

**Module**

FRXPDIS0

---

**FRD2890I      DATA SET READ ERROR FOR *dsname***

**Explanation**

A read error is detected for the change accumulation or log data set with the name *dsname* during database data set or fast path area recovery by the IMS Database Recovery Facility. If the IMS Database Recovery Facility is unable to read from a copy of the data set or switch to the SLDS if the error was on an RLDS, the recovery attempt stops.

**User response**

See the accompanying FRD4208I message to determine what caused the error. Correct the error and start the recovery again.

**Module**

FRXRBUF0

---

**FRD2891      READ ERROR SYNAD STRING  
*dbname, sequence number***

**Explanation**

A read error is detected for the change accumulation or log data set.

***dbname***

Identifies the database data set or area that had read error.

***sequence number***

Log record sequence number.

**System action**

FRD2890I has already been issued regarding the read error. This message is then issued to provide the error string returned from the SYNAD ERROR routine. Processing continues.

**User response**

If the problem cannot be solved on inspection, provide this message to IBM support.

**Module**

FRXRBUF0

---

**FRD2892I      *reason* IN LOG RECORD *seqnum*  
DETECTED IN *dsname***

**Explanation**

An invalid record content is detected for the log data set of *dsname* during database data set recovery by the IMS Database Recovery Facility. This message is followed by an ABEND 385-00A.

The meanings of the message variables are as follow:

**reason**

Identifies the problem as invalid time stamp or an invalid rec length.

**seqnum**

Is the sequence number that identifies the log record in the log data set. The sequence number can be used to determine which record is invalid.

**dsname**

The data set from which the log record was read.

**System action**

The IMS Database Recovery Facility address space terminates.

**User response**

**Invalid time stamp:** Examine the log record that is identified in the message within the log data set that is listed in the message. Use the IMS DFSLOG06 macro mapping of the log record to determine the offset to the ACPRILOG field. If this time stamp is zero, determine if anything in your environment interacts with the IMS Logger component initialization or termination processing. If not, report this problem to IBM. In any case, use the appropriate tool or procedure to place the Prilog time for the subsystem or batch job that created the log in the ACPRILOG field of the log record. Refer to the appropriate IMS documentation for the format of the Prilog time stamp for the 06 log record. Ensure that the 06 log records at the beginning and end of the log data set have time stamps provided.

**Invalid record length:** An invalid record length is detected when processing a record while reading from a log data set. The record length is zero. This can occur when the log data set is created with unformatted data set characteristic. IMS Database Recovery Facility terminates early without completing the recovery.

Examine the log data set identified in the message and repair it to the log data set characteristics which the IMS product supports.

**Module**

FRXRBUF0

---

**FRD3001I UNEXPECTED RETURN CODE  
FROM SORT, RC=return code**

---

**Explanation**

The SORT program ended with an unexpected RC. See the prior SORT messages for more information. In the message text the following value applies:

**return code**

Represents the return code received from the SORT program. See the appropriate SORT program documentation for an explanation of the return code.

**System action**

Recovery processing by this subordinate address space is stopped. If ERRORCONT was specified, recovery processing continues for the recovery list entries not being processed by this subordinate address space.

**User response**

Determine the cause of the failure and rerun recovery for the entries that did not complete recovery.

**Module**

FRXLSRT1

---

**FRD3300E RECOVERY ABORTED: reason**

---

**Explanation**

While running, the IMS Database Recovery Facility job encountered an unrecoverable condition. *reason* is one of the following:

**CURRENT HPIC MAINTENCE LEVEL DOES NOT  
SUPPORT GROUP SYMBOLIC IN IC DSN TEMPLATES**

"&GROUP" was specified in the IC DSN or DSN2 template to include the CA Group in the output image copy data set name, but the IMS HP Image Copy maintenance level is not sufficient to support the feature.

**CURRENT xxxx MAINTENCE LEVEL DOES NOT  
SUPPORT ICNOTIFY(DRF)**

The ICNOTIFY(DRF) parameter was specified, but the IMS HP Image Copy or DBRC maintenance level is not sufficient to support this IMS Database Recovery Facility feature.

xxxx identifies either HPIC or DBRC.

**System action**

The IMS Database Recovery Facility job abnormally terminates producing a U0384-00000035 SVC dump.

**User response**

Apply the latest IMS HP Image Copy or DBRC maintenance, or remove the "&GROUP" and/or ICNOTIFY(DRF) from the DRF job.

**Module**

FRXVSTA3

**FRD3301W      DYNAMIC SPACE CALCULATION  
BYPASSED: DBD=*dbname*  
DDN=*ddname***

**Explanation**

The user specified the keyword sub-parameter DYN in the SPACE parameter of the IMS Database Recovery Facility job, but the calculation of the required space could not be completed.

Message FRD3302I is issued with additional information, and message FRD3303I is issued with the input image copy data set name.

*dbname* and *ddname* identify the DBDS being recovered.

**System action**

The IMS Database Recovery Facility job completes with return code 4 unless other errors are encountered.

The SPACE of the output image copy created by the job is the value specified on the SPACE parameter.

**User response**

Use message FRD3303I to identify the input image copy. Then verify the volume, device type, and IC record count.

The image copy record count can be verified using the following DBRC command:

```
LIST.DBDS DBD(dbname)
```

If these values are correct, save all job output and report the condition to IBM Software Support.

**Module**

FRXVSTA3

**FRD3302I      *reason text* : RC=*rc* RSN=xxxxxxx**

**Explanation**

This is an informational message issued with message FRD3301W that explains the reason the dynamic space calculation was bypassed.

*reason text* indicates the reason for the aborted calculation and is one of the following:

**UCB LOOKUP FAILED: RC=*rc* RSN=xxxxxxx**

The UCBLOOK macro was issued to return the UCB of the device containing the input image copy being used for the recovery, but failed with return code *rc* and reason code xxxxxxx.

**DSCB OBTAIN FAILED : RC=*rc* RSN=xxxxxxx**

The OBTAIN macro was issued to return the DSCB of the input image copy being used for the recovery, but failed with return code *rc* and reason code xxxxxxx.

**INPUT IMAGE COPY RECORD COUNT INVALID :  
RC=04 RSN=xxxxxxx**

The record count returned by DBRC for the input image copy being used for the recovery is less than or equal to zero.

xxxxxxx is the value of record count returned by DBRC.

**System action**

The IMS Database Recovery Facility job completes with condition code 4 unless other errors are encountered.

The SPACE of the output image copy created by the job is the value specified on the SPACE parameter.

**User response**

Use message FRD3303I to identify the input image copy. Then verify the volume, device type, and IC record count.

The image copy record count can be verified using the following DBRC command:

```
LIST.DBDS DBD(dbname)
```

If these values are correct, save all job output and report the condition to IBM Software Support.

**Module**

FRXVSTA3

**FRD3303I      Input Image Copy: *data set name***

**Explanation**

The keyword sub-parameter DYN was specified in the SPACE parameter of the IMS Database Recovery Facility job, but the calculation of the required space could not be completed.

Message FRD3303I is issued after messages FRD3301W and FRD3302I and identifies the input image copy data set name.

*data set name* is the data set name of the input image copy being used for recovery.

**System action**

The IMS Database Recovery Facility job completes with return code 4 unless other errors are encountered.

The SPACE of the output image copy created by the job is the value specified on the SPACE parameter.

#### User response

Use message FRD3303I to identify the input image copy. Then verify the volume, device type, and IC record count.

The image copy record count can be verified using the following DBRC command:

```
LIST.DBDS DBD(dbname)
```

If these values are correct, save all job output and report the condition to IBM Software Support.

#### Module

FRXVSTA3

---

**FRD3304E RECOVERY ABORTED : *reason***

#### Explanation

While running, the IMS Database Recovery Facility job encountered an unrecoverable condition.

*reason* is one of the following:

- DBDS ATTRIBUTES COULD NOT BE DETERMINED
- MVS NAME/TOKEN CREATE FAILED FOR DRF

#### System action

The IMS Database Recovery Facility job abnormally terminates with a U0385 abnormal end.

A U0385 SVC dump is produced, unless suppressed by DAE or SLIP.

#### User response

These are internal errors.

Collect all job output and the SVC dump. Then report the problem to IBM Software Support and submit the collected documentation.

#### Module

FRXVSTA3

---

**FRD4100I DSNNAME=*data set name***

#### Explanation

Generic message that displays either an image copy data set or a database data set. This message is usually a follow on to another message. In the message text the following value applies:

#### *data set name*

The full data set name.

If the data set name is of the form SLB . *Ixxxx* . *Dxxxx* . *dbname* . *ddname*, and it was preceded by message FRD4302I (allocation failure), then it can mean one of two things:

- If USESLBIC(Y) is set, then it means that relative to the RCVTIME specified (or current), the SLB does not exist.

It is recommended that the corresponding IC record be removed from the RECON.

- If USESLBIC(N) is set, then it means, the DSN could not be allocated. This will always be the case.

To find a non-SLB recovery point, you can choose a different RCVTIME. However, it is not recommended that these IC records be removed, unless it is known that the SLB does not exist.

#### System action

Processing continues.

#### User response

None. This message is informational.

#### Module

FRXIDYN0, FRXIOM0, FRXIOCNO, FRXIOSMO, FRXISMS0, FRXIVCNO, FRXIVSMO

---

**FRD4101W PREVIOUS DSNNAME WAS NOT DELETED. DELETE IF RPTRET=Y IS USED.**

#### Explanation

IMS Database Recovery Facility was attempting to dynamically allocate DSNNAME with DISP=(OLD, DELETE, DELETE) so that it could be deleted. However, the attempt failed for the reason that is indicated in message FRD9003A. For example, if the data set is in use by another JOB, RC=0004 and RSN=0210 are issued on FRD9003A. DSNNAME is a temporary data set that contains messages that are related to the recovery. It does not contain recovery assets such as a DBDS, a log, an image copy, or a change accumulation file.

This message always follows messages FRD9003A, which indicates a dynamic allocation error and FRD4100I, which identifies the DSNNAME that failed dynamic allocation.



**System action**

Processing continues and the IMS Database Recovery Facility MAS return code is set to 4.

**User response**

If the RPTRET=Y parameter on the REPORT control card is used, manually delete the DSNNAME because it is possible that IMS Database Recovery Facility attempts to use the DSNNAME again on a subsequent recovery.

If the RPTRET=N parameter on the REPORT control card is used, IMS Database Recovery Facility automatically deletes this DSNNAME and then allocates DSNNAME as a new data set. No user response is required.

**FRD4202I**

**DATABASE RECOVERY DATA  
MANAGER TERMINATION  
COMPLETE**

**Explanation**

The IMS Database Recovery Facility address space completed normally.

**System action**

The IMS Database Recovery Facility address space ends.

**Module**

FRXMTRMO

**FRD4203I**

**STARTUP FAILED *reason* ERROR,  
RC=*return code***

**Explanation**

Startup processing for the IMS Database Recovery Facility address space failed while attempting to create a persistent thread. In the message text the following values apply:

***reason***

Identifies the problem and is one of the following:

- CREATE CHANGE ACCUMULATION THREAD
- CREATE DATA SPACE MANAGER THREAD
- CREATE PIPE RECEIVE THREAD
- CREATE PIPE SEND THREAD
- CREATE COMMAND SERVICES THREAD
- CREATE SORT MANAGER THREAD
- CREATE READ SERVICES THREAD
- ATTACH SORT TASK
- ATTACH PIPE RECEIVE TASK

- ATTACH PIPE SEND TASK
- ATTACH CHANGE ACCUMULATION TASK
- ATTACH SORT MANAGER TASK
- ATTACH DBRC TASK

***return code***

Represents internal diagnostic information.

**System action**

The address space shuts down and recovery is aborted.

**User response**

Determine if the IMS Database Recovery Facility address space REGION is too small; if so, increase the size.

**Module**

FRXMSTRO

**FRD4204I**

**READER INITIALIZATION  
COMPLETE**

**Explanation**

The log-data set-read subcomponent of the IMS Database Recovery Facility address space in the IMS Database Recovery Facility completed initialization successfully. The message destination is the z/OS system console.

**System action**

Processing continues.

**Module**

FRXRDTHO

**FRD4205I**

**READER INITIALIZATION FAILED,  
*reason*, RC=*return code***

**Explanation**

Initialization for the log-data set-read subcomponent of the IMS Database Recovery Facility failed. In the message text the following values apply:

***reason***

Indicates the reason for the error, and is one of the following:

- ENQUEUE AWE
- GET AWE
- GET STORAGE
- GET THREAD

- TCB ATTACH

#### **return code**

Represents internal diagnostic information.

The message destination is in the z/OS system console.

#### **System action**

The database recovery control system shuts down.

#### **User response**

If *reason* is GET STORAGE or GET AWE, increase the region size in the startup JCL.

All other reasons indicate an internal IMS error. Record the error information and call IBM Software Support for assistance.

#### **Module**

FRXCAMG0, FRXRCTL0

---

**FRD4206I      UNABLE TO START READING,**  
**reason, RC=return code**

#### **Explanation**

An individual log-data set-read thread of the IMS Database Recovery Facility address space in the IMS Database Recovery Facility was unable to read a log data set. This problem results from an internal IMS error. The message destination is the MVS system console.

In the message text the following values apply:

#### **reason**

If *reason* is OBTAIN DATA SET, the IMS Database Recovery Facility experienced a failure when allocating either a log or change accumulation data set during data set read processing. All of the other reasons that follow indicate an internal IMS error:

- ENQUEUE ALLOCATE AWE
- ENQUEUE AWE FOR READER TERMINATION
- ENQUEUE DEALLOCATE AWE
- GET AWE FOR DEALLOCATE
- GET AWE FOR READER TERMINATION
- POST ECB
- WAIT FOR ALLOCATE

#### **return code**

If *reason* is OBTAIN DATA SET, *return code* is DYNALLOC (SVC 99). For additional information on this return code, see the appropriate z/OS manuals. The message destination is the z/OS system console.

All other values represent internal diagnostic information.

#### **System action**

The recovery instance and the IMS Database Recovery Facility address space ended early.

#### **User response**

If *reason* is GET AWE FOR DEALLOCATE or GET AWE FOR READER TERMINATION, increase the region size of your the IMS Database Recovery Facility startup JCL.

All other reasons indicate an internal IMS errors. Record the error information and call the IBM Software Support for assistance.

#### **Module**

FRXRDTH0

---

**FRD4207I      number RECORDS READ,**  
**DSN=dsname**

#### **Explanation**

The log data set read subcomponent of the IMS Database Recovery Facility completed reading the log data. In the message text the following values apply:

#### **number**

The total number of records read from the data set *dsname*.

#### **dsname**

The data set from which records are read.

#### **System action**

Processing continues.

#### **Module**

FRXRDTH0

---

**FRD4208I      UNABLE TO CONTINUE READING,**  
**reason, RC=return code**

#### **Explanation**

An individual log data set read thread of the IMS Database Recovery Facility is unable to continue reading a log data set. This problem is as a result of an internal IMS error. In the message text the following values apply:

#### **reason**

Indicates the reason for the error, and is one of the following:

- POSTING, DONE WITH BUFFER

- POSTING NEXT READ
- READ ERROR DETECTED
- WAITING FOR BUFFER
- WAITING TO READ

#### return code

Represents internal diagnostic information.

#### System action

The IMS Database Recovery Facility ends.

#### User response

This message indicates an internal IMS error. Record the error information and call IBM Software Support for assistance.

#### Module

FRXRBUF0

---

|                 |                                                 |
|-----------------|-------------------------------------------------|
| <b>FRD4209I</b> | <b>DATA SET ALLOCATED,</b><br><b>DSN=dsname</b> |
|-----------------|-------------------------------------------------|

---

#### Explanation

The log or change accumulation data set indicated by *dsname*, is successfully allocated for input to the IMS Database Recovery Facility address.

#### System action

Processing continues.

#### Module

FRXYALLO

---

|                 |                                                                     |
|-----------------|---------------------------------------------------------------------|
| <b>FRD4210I</b> | <b>ALLOCATE FAILED, reason,</b><br><b>RC=return code FOR dsname</b> |
|-----------------|---------------------------------------------------------------------|

---

#### Explanation

The IMS Database Recovery Facility failed to allocate a log or change accumulation data set indicated in the message text by *dsname*. In the message text the following values apply:

#### reason

Indicates the reason for the error, and is one of the following:

##### GET STORAGE

The IMS Database Recovery Facility component of IMS Database Recovery Facility experienced an error while attempting to obtain storage related to data set allocation.

##### DYNALLOC

An invalid return code was detected from z/OS.

#### OPEN

An invalid return code was detected from the BSAM OPEN macro.

#### return code

If reason is DYNALLOC or OPEN, a return code is issued. The return code consists of two parts. The first four characters are the contents of register 15 returned from the DYNALLOC or OPEN macro. The last four characters represent the DYNALLOC or OPEN error reason code from the S99ERROR field. For example: RC=04841156. For additional information on the return code and reason code, see the *z/OS MVS Programming: Authorized Assembler Services Guide*.

#### System action

The IMS Database Recovery Facility ends early.

#### User response

If reason is GET STORAGE, increase the region size for the IMS Database Recovery Facility address space in the startup JCL and start recovery processing again. If reason is DYNALLOC or OPEN, see the *z/OS MVS Programming: Authorized Assembler Services Guide* for the appropriate action.

#### Module

FRXYALLO

---

|                 |                                                 |
|-----------------|-------------------------------------------------|
| <b>FRD4211I</b> | <b>DATA SET UNALLOCATED,</b><br><b>DSN=name</b> |
|-----------------|-------------------------------------------------|

---

#### Explanation

The log data set or change accumulation data set indicated by *name*, was successfully deallocated by the IMS Database Recovery Facility address space in the IMS Database Recovery Facility.

#### System action

Processing continues.

#### Module

FRXYUNAO

---

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>FRD4212I</b> | <b>DATA SET DOES NOT EXIST,,</b><br><b>RC=xxxxxxx DSN=DSname</b> |
|-----------------|------------------------------------------------------------------|

---

#### Explanation

Before a data set is allocated, a verification call is performed. This message is issued if the data set does not exist.

xxxxxxx: The return code returned by verification call.

DS name: Data set to be allocated.

**System action**

Open will fail for this data set. Recover continues for the rest of the data sets.

**Module**

FRXYALLO, FRXIDYNO

---

|                 |                                                                                |
|-----------------|--------------------------------------------------------------------------------|
| <b>FRD4214I</b> | <b>RECORD PIPE FAILURE<br/>DETECTED: <i>reason</i>, RC= <i>return code</i></b> |
|-----------------|--------------------------------------------------------------------------------|

---

**Explanation**

Internal processing detected a failure while processing the communication pipe between the master address space and one of the sort address spaces. In the message text the following values apply:

***reason***

Indicates the reason for the error, and is one of the following:

- GET ANCHOR BLOCK
- GET OPEN PARMS
- OPEN
- WRITE

***return code***

Represents internal diagnostic information.

**System action**

The IMS Database Recovery Facility system shuts down.

**User response**

Record the error information and call IBM Software Support for assistance.

**Module**

FRXPDIS0, FRXPDSS0, FRXPDSO

---

|                 |                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------|
| <b>FRD4215I</b> | <b>LEAP SECONDS: CVTL<i>SO</i> <i>iso leap</i>,<br/>LOG <i>log leap</i>, USER <i>drf leap</i></b> |
|-----------------|---------------------------------------------------------------------------------------------------|

---

**Explanation**

Informational message displaying Leap seconds offset values.

***iso leap***

CVTL*SO* value defined in MVS.

***log leap***

Leap second offset calculated from first applicable log record.

***drf leap***

Leap second offset specified by user via LEAPSECS= keyword.

**System action**

None.

**Module**

FRXRBUFO

---

|                 |                                                                                 |
|-----------------|---------------------------------------------------------------------------------|
| <b>FRD4216W</b> | <b>LEAP SECONDS APPLIED <i>drf leap</i><br/>DIFFER FROM LOG <i>log leap</i></b> |
|-----------------|---------------------------------------------------------------------------------|

---

**Explanation**

The leap second offset specified by the user differs from the expected value calculated using the first applicable log record.

***drf leap***

Leap second offset specified by user through the LEAPSECS= keyword.

***log leap***

Leap second offset calculated from first applicable log record.

**System action**

Job ends with RC=4.

**User response**

Correct the LEAPSECS=keyword value to correspond with the LOG value that is displayed in the message, and rerun the job.

**Module**

FRXRBUFO

---

|                 |                                                  |
|-----------------|--------------------------------------------------|
| <b>FRD4219I</b> | <b>xxxxxxx RECORDS PROCESSED<br/><i>type</i></b> |
|-----------------|--------------------------------------------------|

---

**Explanation**

Indicates how many records were sent across the pipe. In message FRD4219I the following message variables have these meanings:

**xxxxxxx**

The number of records processed for the type of record indicated by *type*. If *type* is present in the message, it indicates one of the following types:

**FOR CHANGE ACCUMULATION**

The number indicated in the message of change accumulation records related to the current recovery has been processed.

## FOR LOG DATA SETS

The number indicated in the message of log records related to the current recovery has been processed.

### System action

Processing continues.

### User response

None. This message is informational.

### Module

FRXPDSS0, FRXPSDS0

---

**FRD4220I**      **GENERATING type COPY:**  
**DBD=dbd, DDN/AREA=ddn**

### Explanation

The IMS Database Recovery Facility is starting to generate a copy of a database data set or area using an image copy, change accumulation, and/or logs of the production database. This message is accompanied by FRD4100I to identify the generated data set name.

#### type

Identifies the type of copy being produced. The value supplied in the message can be one of the following types:

#### DATA SET

A duplicate of the production database data set is being produced

#### IMAGE

An incremental image copy is being produced.

#### dbd

Identifies the production database name.

#### ddn

Identifies the production ddname.

### System action

Processing continues.

### User response

None. This message is informational.

### Module

FRXICTLO

---

**FRD4221I**      **DATABASE COPY GENERATION**  
**COMPLETE | FAILED: DBD=dbd,**  
**DDN/AREA= ddn - xxxxxx OF**  
**yyyyyy PROCESSED**

## Explanation

The database copy generation for the identified production database data set has completed either successfully (COMPLETE) or unsuccessfully (FAILED). The recovery process might fail due to errors or end early by operator command. This message is followed by FRD4100I specifying the data set name.

#### dbd

Identifies the production database name.

#### ddn

Identifies the production ddname.

#### xxxxxx

xxxxxx is the number of database data sets that have been restored by the RSS.

#### yyyyyy

yyyyyy is the number of database data sets that will be restored by the RSS, unless an error occurs.

### System action

Processing continues.

### User response

This is an informational message to indicate the progress of the recovery operation. If a failure was indicated, look for additional error messages.

### Module

FRXIRTH0

---

**FRD4226I**      **CATALOG SEARCH INTERFACE**  
**ERROR, ID=aa RC=xx, RSN=yy**

### Explanation

During the data set restore process for the IMS Database Recovery Facility, an error occurred while using the Catalog Search Interface to obtain information from the catalog for the database data set being restored.

In the message text the following values apply:

#### aa

The two-character module identification used by IBM service personnel.

#### xx

The return code.

#### yy

The reason code.

### System action

Recovery for this DBD and DDN stops. If the ERRORCONT parameter was not specified on the START command, the entire recovery process stops.

### User response

This is an internal logic error. Contact IBM Software Support.

### Module

FRXIVCNO, FRXUAPIO

---

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| <b>FRD4227I</b> | <b>Data set catalog information is not verified</b> |
|-----------------|-----------------------------------------------------|

---

### Explanation

This message is issued if CATDS(Y) is specified, but neither VERIFY(ALLOC) nor VERIFY(OPEN) is requested during the VERIFY process.

### System action

This is an informational message. Processing continues.

### User response

None. This message is informational.

### Module

FRXEPSS0

---

|                 |                                                                                |
|-----------------|--------------------------------------------------------------------------------|
| <b>FRD4230I</b> | <b>VSAM SHOWCB GENCB MODCB<br/>MACRO ERROR, BLOCK=block,<br/>RC=xx, RSN=yy</b> |
|-----------------|--------------------------------------------------------------------------------|

---

### Explanation

During an data set restore process, a VSAM macro error occurred for either a SHOWCB, GENCB, or MODCB. In the message text the following values apply:

#### **block**

Identifies the VSAM control block being generated or modified.

#### **xx**

The return code returned by VSAM.

#### **yy**

The reason code returned by VSAM.

### System action

Recovery for this DBD/DDN (as identified by message FRD4301I that follows this message) stops. If ERRORCONT was not specified on the START command, the entire recovery process stops.

### User response

For an explanation of the return and reason code, see the interpreting DYNALLOC return codes information in

the *z/OS MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALE-DYN)*. Take the appropriate action indicated by the return and reason codes.

### Module

FRXIVCNO, FRXIVSMO

---

|                 |                                             |
|-----------------|---------------------------------------------|
| <b>FRD4231I</b> | <b>UNABLE TO OBTAIN STORAGE<br/>FOR xxx</b> |
|-----------------|---------------------------------------------|

---

### Explanation

During an data set restore process, a request to obtain storage could not be satisfied. In the message, xxx identifies what the storage request was for and can be one of the following values:

- VSAM PLIST
- VSAM ACB AND RPL
- IC RESTORE DYNALLOC
- CSI BLOCK
- FIRST CI

### System action

Recovery for this DBD/DDN (as identified by message FRD4301I that follows this message) stops. If the ERRORCONT parameter was not specified on the START command, the entire recovery process stops.

### User response

Increase the region size in the IMS Database Recovery Facility procedure. For additional information, see the FRXJCLDF Procedure.

### Module

FRXIVCNO, FRXIVSMO

---

|                 |                                                                                          |
|-----------------|------------------------------------------------------------------------------------------|
| <b>FRD4236I</b> | <b>HEADER RECORD IN IC DATA SET<br/>DOES NOT MATCH REQUEST<br/>DBD=dbd, DDN/AREA=ddn</b> |
|-----------------|------------------------------------------------------------------------------------------|

---

### Explanation

The IMS Database Recovery Facility has detected a problem with the image copy data set selected to restore the DBDS or AREA. The information in the image copy data set indicates that it is for a different DBDS or AREA. In the message the following values apply:

#### **dbd**

The database name being restored.

#### **ddn**

The ddname or area name that is being restored.

## System action

The data set restore process fails.

## User response

Since the image copy is selected by DBRC, this message indicates that the image copy has been incorrectly registered in RECON. Use either a DELETE .IC DBRC command, or a CHANGE .IC DBRC command to correct the information in the RECON. Then, rerun the recovery.

## Module

FRXIIOM0

---

|                 |                                                              |
|-----------------|--------------------------------------------------------------|
| <b>FRD4284E</b> | <b>DATA SET RESTORE I/O ERROR,</b><br><b>dbd ddn: reason</b> |
|-----------------|--------------------------------------------------------------|

---

## Explanation

A data set restore process failed during I/O processing for the DBD and DDN identified by *dbd* and *ddn*, respectively. In the message text the following values apply:

### **dbd**

Identifies the database name in error.

### **ddn**

Identifies the ddname in error.

### **reason**

Indicates the cause of the data set restore I/O error and can be one of the following:

#### **AUTOMATIC DELETE REDEFINE FAILED**

An attempt to automatically delete and redefine a database data resulted in error. Verify the parameters provided to the DBDSN and DBDSL keywords.

#### **DATABASE COPY DEFINE FAILED**

A failure occurred while trying to allocate an alternate data set name. Check the data set naming convention rules that you specified in your JCL. If you specified DISP (NEW) to have IMS Database Recovery Facility generate the name for you, ensure that all of the following conditions exist:

- the rules used in your DBDSN() option are valid
- the name being generated does not conflict with an existing data set name
- that you have enough free space on your volume

#### **IC DECOMPRESS ROUTINE UNKNOWN**

The image copy used for recovery was compressed by a compression tool that is not

supported in this release of IMS Database Recovery Facility.

#### **IC DECOMPRESS OPERATION FAILED**

Data set restored failed to decompress an image copy. Verify if the image copy used was generated by a compression tool supported by this release of IMS Database Recovery Facility.

#### **INPUT IC I/O ERROR**

An I/O error occurred during an attempt to read an input image copy data set on a tape volume. The IMS Database Recovery Facility job output includes an IEC\* or IOS\* series system message that identifies the cause of the I/O error. These messages are documented in *z/OS MVS System Messages Vols 1-10*.

#### **INVALID IC TYPE FOR OUTPUT TO COPY**

The recover-to-copy function OUTPUT (DUP) does not support user-ICs or other SMS-type ICs as input.

#### **INVALID IC TYPE FOR OUTPUT TO IC**

The Incremental image copy function does not support user-ICs or other SMS-type ICs as input.

#### **INVALID VOLUME FOR OUTPUT IC**

One or more of the tape volumes that you provided with either the VOLSER or VOLSER2 parameters, cannot be used. This condition can occur because your input image copy already resides on that tape volume.

#### **OLR DEF REC RETRIEVE**

Data set restore is performing recovery on an OLR capable data set to a point where OLR definition log records (X'2930') are required to provide the block image attributes. These records were not found. Verify in the RECON listing and IMS Database Recovery Facility reports that all required log or change accumulation data sets are available.

#### **OLR DATA SET CREATE**

Data set restore failed during data set create and allocation of an OLR target data set. Check for additional messages describing the cause of this error.

#### **TIME CONVERSION ROUTINE LOAD ERROR**

The IMS Database Recovery Facility subordinate address space failure occurred while attempting to load an IMS time conversion routine. Check your IMS Database Recovery Facility subordinate address space JCL and include the IMS Load library the IMS Database Recovery Facility STEPLIB concatenation.

#### **DELETE/DEFINE FROM PDS FAILED**

The recover-to-copy function as specified with OUTPUT (DUP) or OUTPUT (BOTH) does not

support the allocation of duplicate database data sets that are based on definitions supplied in a PDS member through DISP(PDS).

### System action

Recovery for this DBD and DDN stops. If ERRORABORT was specified, none of the entries in the recovery list were recovered.

### User response

Verify the conditions described above and contact IBM Software Support for further assistance.

### Module

FRXIRTHO

---

**FRD4285I      ERROR ENCOUNTERED DOING  
                 process, RC=ccrrzz,  
                 RBN=nnnnnnnn**

---

### Explanation

During an image copy restore process, a service request failure was detected. This message might be followed by the U0385-0D abend. In this message, the following values apply:

**process has one of the following meanings:**

#### CA GETMAIN

Requested storage was not obtained. In the message the following values apply:

**cc**  
Location code.

**rr**  
Return code from BPEDETM macro.

#### APPLY CA UPDATES

The version of the CA record is not supported. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
CA record version.

#### EXPAND CA SERVICE

The call to expand the CA record was unsuccessful. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
Return code from CSRCEsrv.

### QUERY CSRCEsrv FOR CA

The call to query the expand service was unsuccessful. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
Return code from CSRCEsrv.

### GET CA RECORD

The call to obtain the next CA record failed. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
Zero (0).

### CA RECORD TYPE MATCH

The type of the CA record is invalid. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
Type code of the CA record.

### INPUT/CA RECORD SYNC

CA record ID is less than the IC record ID. This is an unrecoverable error. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
Not used.

**nnnnnnnnnn**  
First four bytes of the record ID.

### CA DBDNAME MATCH

CA header record DBDname does not match the DBDname of the Image Copy. In the message text, the variable values have the following meanings:

**cc**  
Location code.

**rr**  
CA record type.

### LOG GETMAIN

Requested storage was not obtained. In the message text, the variable values have the following meanings:

**cc**  
Location code.



**rr**

Return code from BPEDETM macro.

#### EXPAND LOG SERVICE

The call to expand the log record was unsuccessful. In the message text, the variable values have the following meanings:

**cc**

Location code.

**rr**

Return code from CSRCESRV.

#### QUERY CSRCESRV FOR LOG

The call to query the expand service was unsuccessful. In the message text, the variable values have the following meanings:

**cc**

Location code.

**rr**

Return code from CSRCESRV.

#### GET LOG RECORD

The call to obtain the next log record failed. This message is informational only if issued as a result of a job cancellation or other early ending of the recovery process. In the message text, the variable values have the following meanings:

**cc**

Location code.

**rr**

Zero (0).

#### INPUT/LOG RECORD SYNC

Log record ID is less than the IC record ID. This is an unrecoverable error. In the message text, the variable values have the following meanings:

**cc**

Location code.

**rr**

Not used.

**nnnnnnnn**

Log record RBN.

#### crrzz means that:

The return code from the service request. In every case the value for zz corresponds to the process.

#### nnnnnnnn means that:

If the value of **nnnnnnnn** is significant it is cited in the process entry. In some cases this is the RBN of the database record in process; otherwise it is zero.

#### System action

Recovery for this DBD/DDN (as identified by message FRD4301I that follows this message) stops. If the Error Continue option was not specified on the START command or the START statement, the entire recovery process stops. If this message is followed by the U0385-0D abend, then save the job documentation and call IBM Software Support for assistance.

#### Module

FRXIOCN0, FRXIVCN0, FRXIVSM0, FRXIOSM0

**FRD4301I**

**DATA SET RESTORE COMPLETE |  
FAILED *dbdname ddname* - *xxxxxx*  
OF *yyyyyy* PROCESSED**

#### Explanation

The data set restore for the identified *dbdname ddname* has completed either successfully (COMPLETE) or unsuccessfully (FAILED). The data set restore might fail due to errors or end early by operator command.

*xxxxxx* is the number of database data sets that have been restored by the RSS.

*yyyyyy* is the number of database data sets that will be restored by the RSS, unless an error occurs.

#### System action

Processing continues.

#### User response

None. This is an informational message only to indicate the progress of the recovery operation. If the data set restore fails due to errors, address those errors, which are indicated by other messages.

#### Module

FRXIRTH0

**FRD4302I**

**IC | DBDS ALLOC ERROR:  
DBD=*dbd* DDN=*ddn* RC=*xxxxxxxx*  
RSN=*yyyyzzzz* WITH**

#### Explanation

During the data set restore process for the IMS Database Recovery Facility, an error occurred while allocating either the image copy (IC) data set or the database data set (DBDS). Possible causes might include, but are not limited to the following list:

- Data sets need to be defined
- Data sets are not available

- Data sets are allocated with I/O errors
- Data sets are allocated to another task

This message is accompanied with FRD4100I to identify the data set being allocated.

The message variable meanings are described as follows:

**dbd**  
Identifies the DBD being restored

**ddn**  
Identifies the DDN being restored

**xxxxxxxx**  
The return code set by Dynamic Allocation (SVC 99)

**yyyyzzzz**  
The reason code returned where the following values apply:

**yyyy**  
The error reason code (S99ERROR)

**zzzz**  
The information reason code set (S99INFO)

### System action

Recovery for this DBD and DDN is stops. If the ERRORCONT parameter was not specified on the START command, the entire recovery process stops.

### User response

For an explanation of the return and reason code, see the interpreting DYNALLOC return codes information in the *z/OS MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALE-DYN)*. Take the appropriate action indicated by the return and reason codes.

### Module

FRXIDYNO

|                 |                                                                                                       |
|-----------------|-------------------------------------------------------------------------------------------------------|
| <b>FRD4303I</b> | <b>IC   DBDS UNALLOC ERROR:<br/>DBD=dbd DDN=ddn RC=xxxxxxxx<br/>RSN=yyyyzzzz DDNAME=nnnnnnnn WITH</b> |
|-----------------|-------------------------------------------------------------------------------------------------------|

### Explanation

During the data set restore process for the IMS Database Recovery Facility, an error occurred while unallocating either the image copy data set (IC) or the database data set (DBDS). This message is accompanied with FRD4100I to identify the data set being unallocated. In the message text the following values apply:

**dbd**  
Identifies the DBD being restored

**ddn**  
Identifies the DDN being restored

**xxxxxxxx**  
The return code set by Dynamic Allocation (SVC 99)

**yyyyzzzz**  
The reason code returned

**nnnnnnnn**  
The DDNAME that the data set was allocated under

### System action

Recovery for this DBD and DDN stops. If the ERRORCONT parameter was not specified on the START command, the entire recovery process stops.

### User response

For an explanation of the return and reason code, see the interpreting DYNALLOC return codes information in the *z/OS MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALE-DYN)*. Take the appropriate action indicated by the return and reason codes.

### Module

FRXIDYNO

|                 |                                                                                           |
|-----------------|-------------------------------------------------------------------------------------------|
| <b>FRD4304I</b> | <b>IC DATA SET DOES NOT MATCH<br/>MADS REGISTERED TO DBRC:<br/>DBD= dbd DDN= ddn WITH</b> |
|-----------------|-------------------------------------------------------------------------------------------|

### Explanation

During a database recovery, the image copy selected was created by the Database Image Copy 2 Utility, but it no longer matches any area data set defined to DBRC for the area being restored. This message is accompanied with FRD4100I to identify the data set that was being restored. In the message text the following values apply:

**dbd**  
The DBD being restored

**ddn**  
The DDN being restored

### System action

Recovery of the specified area stops. If the ERRORABORT parameter was specified on the recovery operation, recovery of all members in the recovery list stops.

User response

Add the area data set back to the area data set list by issuing the DBRC INIT .ADS command; or mark the image copy as INVALID by issuing the DBRC CHANGE .IC command, which causes recovery to select a different image copy if one exists. Then, restart the recovery.

Module

FRXISMSO

**FRD4305I DATA SET NOT FOUND IN CATALOG: DDNAME=ddname WITH**

Explanation

During the data set restore process for the IMS Database Recovery Facility, the VSAM database data set being recovered could not be found in the catalog. This probably happened because a necessary catalog is not available to the IMS Database Recovery Facility job. For additional information, see the FRXJCLDF Procedure. This message is accompanied with FRD4100I to identify the data set name of the VSAM database data set. In the message text the following values apply:

ddname

Identifies the DDNAME of the VSAM database data set.

dsname

Identifies the DSNAME of the VSAM database data set.

System action

Recovery for this DBD and DDN stops. If the ERRORCONT parameter was not specified on the START command, the entire recovery process stops.

User response

Ensure that all necessary catalogs are available to the IMS Database Recovery Facility job.

Module

FRXIVCNO

**FRD4306I VSAM READ | WRITE LOGICAL | PHYSICAL ERROR: RC=xx RSN=yy DDNAME=nnnnnn WITH**

Explanation

During an data set restore process, an I/O error was encountered. This message is accompanied with

FRD4100I to identify the data set that experienced the error. In the message text the following values apply:

xx

The return code returned by VSAM

yy

The reason code returned by VSAM

nnnnnn

The DDN the data set was allocated under

System action

Recovery for this DBD and DDN stops. If the ERRORCONT parameter was not specified on the START command, the entire recovery process stops.

User response

For an explanation of the return and reason code, see the *z/OS DFSMS Macro Instructions for Data Sets* manual. Take the appropriate action indicated by the return and reason codes.

Module

FRXIVCNO, FRXIVSMO

**FRD4307I vartxt ERROR: RC=return code RSN=rsn DDNAME=dddddddd WITH**

Explanation

While restoring an image copy, an OPEN or CLOSE error occurred on either the image copy data set or the data set being restored. The operating system and the access method, or both, might have issued messages prior to this message that pertain to the error. Sequential access method services are used to process the image copy data set.

The data set being restored might be either VSAM or Non-VSAM (sequential). The data set name has been determined by DBRC from information in the RECON.

An OPEN error most likely occurs because the data set to be processed does not exist or has been defined incorrectly for the type of database data set organization. This message is accompanied with FRD4100I to identify the data set that experienced the error.

In the message text the following values apply:

vartext

can be one of the following errors:

- DYNALLOC STORAGE
- IC OPEN
- IC CLOSE

- NON-VSAM OPEN
- NON-VSAM CLOSE
- VSAM OPEN
- VSAM CLOSE

#### **return code**

The VSAM return code. For additional information, see the documentation on VSAM Macro Return and Reason Codes in the *z/OS DFSMS Macro Instructions for Data Sets* manual. For NON-VSAM data sets, the return code has no meaning and is always set to 0.

#### **rsn**

The VSAM reason code. For additional information, see the documentation on VSAM Macro Return and Reason Codes in the *z/OS DFSMS Macro Instructions for Data Sets* manual. For NON-VSAM data sets, the reason code has no meaning and is always set to 0.

#### **dddddddd**

The DDN the data set was allocated under.

#### **System action**

If ERRORABORT was specified, recovery stops for all entries in the recovery list.

#### **User response**

Correct the error indicated by the return code, the reason codes, and other possible messages, then rerun the recovery.

#### **Module**

FRXIDYN0, FRXIIOM0, FRXIOSM0, FRXIVSM0, FRXIOCN0, FRXIVCN0, FRXISMS0

---

**FRD4308I**      **IC RESTORE STOPPED. vartxt:**  
**DBD=dbd DDN=ddn WITH**

#### **Explanation**

An data set restore process was stopped for the DBD and DDN identified by *dbd* and *ddn*, respectively. This message is accompanied with FRD4100I to identify the data set that was restored.

In the message text the following values apply:

#### **dbd**

identifies the DBD for which the data set restore was stopped

#### **ddn**

identifies the DDN for which the data set restore was stopped

#### **vartxt**

indicates the reason that the process could not continue, and is one of the following:

#### **KSDS KEYLENGTH ERROR**

The key length defined for the KSDS is not the same as the key length when the IC was taken.

#### **OUTPUT DATA SET MUST BE EMPTY**

The data set being restored contained data. It should be deleted and redefined prior to recovery.

#### **OUTPUT RECORD LENGTH ERROR**

The record length defined for the output data set is not the same as the record length when the IC was taken.

#### **VSAM CLUSTER MUST BE AN ESDS**

The data set being restored is defined to DBRC as non-indexed, yet it has been defined as a KSDS.

#### **VSAM CLUSTER MUST BE A KSDS**

The data set being restored is defined to DBRC as indexed, yet it has been defined as an ESDS.

#### **System action**

If ERRORABORT was specified, none of the entries in the recovery list were recovered.

#### **User response**

Correct the problem as identified by *vartxt* and rerun the recovery.

#### **Module**

FRXIVCN0, FRXIVSM0

---

**FRD4309I**      **ERROR ON vartxt: DBD=bbbbbbbb**  
**DDN=dddddddd I/O ERROR=aaa,**  
**tt, ddname, op, error, block,**  
**accessmethod WITH**

#### **Explanation**

An I/O error occurred while restoring an image copy for the DBD and DDN specified by *bbbbbbbb* and *dddddddd*, respectively, in the data set identified by message FRD4100I.

In the message text the following values apply:

#### **vartxt**

Represents is either INPUT (the error occurred while reading the IC data set) or OUTPUT (the error occurred while writing to the output data set).

#### **bbbbbbbb**

The DBD which experienced the I/O error.

**dddddddd**

The DDN which experienced the I/O error.

**aaa**

The device number that the error occurred on.

**tt**

The type of device.

**ddname**

The name of the DD statement that the data set was allocated to.

**op**

The I/O operation that resulted in the error.

**error**

The type of error as determined by the device. For example, if it is device dependent.

**block**

The block number in error.

**accessmethod**

The access method being used.

**System action**

Recovery stops for the identified DBD and DDN. If ERRORABORT was specified, recovery stops for all members in the recovery list.

**User response**

Correct the indicated problem and rerun the recovery.

**Module**

FRXIOM0, FRXIOCNO, FRXIVCNO, FRXIOSMO

---

**FRD4310I      RESTORING DBD=*dbd*, DDN/  
                         AREA=*ddn* WITH**

---

**Explanation**

The IMS Database Recovery Facility is starting to recover a database data set or area using an image copy. This message is accompanied with FRD4100I to identify the image copy data set name.

In the message the following values apply:

**dbd**

Defines the DBD name that is being recovered.

**ddn**

Define the DDNAME or AREA name that is being recovered.

**System action**

Processing continues.

**User response**

None. This message is informational.

**Module**

FRXIOM0, FRXISMS0

---

**FRD4312E      THE DATA SET OR AREA CANNOT  
                         BE RECOVERED FROM SLB IC  
                         BECAUSE *rsn* RC= *nnnnnnnn* RSN=  
                         *nnnnnnnn***

---

**Explanation**

A call to the IMS Recovery Expert API for System Level Backups failed.

In the message text:

**rsn**

Indicates the reason for which the data is invalid and can be one of the following:

**API NOT LOADED IN MAS ADDRESS SPACE**

System level backup library is not installed in DRF master address space.

**API INIT CALL FAILED**

The Initial call to SLB API failed.

**API PREPARE\_RECOV CALL FAILED**

The Prepare call to SLB API failed

**API RESTORE IC CALL FAILED**

The Restore call to SLB API failed.

**API CONNECT CALL FAILED**

The Reconnect call to SLB API failed.

**API QUERY CALL FAILED**

The Query Call to SLB API failed

**API QUERY(ALLOC) CALL FAILED**

The Query call for Allocating the data set failed.

**API QUERY(OPEN) CALL FAILED**

The Query call for Opening the data set failed.

**API DISCONN CALL FAILED**

The Disconnect call to SLB API failed

**API NOT LOADED IN SUB ADDRESS SPACE**

System level backup library is not installed in DRF Subordinate Address Space.

**System action**

Recovery continues as if USESLBIC(N) is specified.

**User response**

Save the job log, provide the return code information, and contact the IBM Software Support for further assistance.

**Module**

FRXVSTA0, FRXICTL0, FRXIRTH0, FRXVVER0

---

**FRD4314W      SLB OFFLOAD DATA SET LOOKUP  
FAILED IN MODULE=*mod-name*  
API CALL=*call-type***

**Explanation**

The database has been successfully restored from the System Level Backup (SLB), but if it was restored from an offloaded SLB, the SLB data set name could not be determined. This is an internal error.

**System action**

Recovery continues normally, but the final report will be incomplete.

Check the RSS job log to determine the offload data set name.

**User response**

Save the job log and contact IBM Software Support for further assistance.

**Module**

FRXPSDS0

API CALL *call-type* is either CONNECT or QUERY.

---

**FRD4316I      DATABASE MIGHT HAVE BEEN  
ALLOCATED DURING RECOVERY -  
CHECK RSS JOB LOG**

**Explanation**

This message is preceded by FRD4312E with the reason "API RESTORE IC CALL FAILED", which indicates that the database or area could not be recovered.

One reason the recovery might have failed for OUTPUT(PRO) processing is that the production database data set was already allocated to another process.

The RSS JOB log indicates if this was the case.

**System action**

Processing continues.

**User response**

If the RSS indicates that the production database data set was already allocated to another process, resolve this issue and run the recovery again.

**Module**

FRXIRTH0

---

**FRD4318I      DBD=*dbdname* DDNAME=*ddname***

**Explanation**

This message is issued together with the immediately prior message to indicate the *dbdname* and *ddname* that is referred to in that previous message.

**System action**

None.

**User response**

Refer to the immediately prior message.

**Module**

This message is issued from any number of modules. Refer to the immediately prior message.

---

**FRD4320I      DATABASE NOT CONTAINED IN  
SLB OR SLB DOES NOT EXIST.  
CHECK RECON, SLB BACKUP AND  
OBJECT REPORTS.**

**Explanation**

The database listed in prior messages FRD4301I, FRD4212I, and FRD4302I is not contained in the SLB. There are several reasons why the database might not be contained in the SLB. Among them are:

- The Enable DB Restore option in the Backup Profile as set to N
- The Recovery Needed flag in the RECON was on at the time of the backup
- The database data set did not exist at the time of the backup.

**System action**

Recovery continues.

**User response**

List the database in the RECON and check the Recovery Needed flag.

Check the SLB Backup and Object reports for the status of the database.

**Module**

FRXIRTH0

---

**FRD4322I      DATABASE NOT CONTAINED IN  
ANY SLB. DRF WILL LOOK FOR A  
LATER NON-SLB IC:  
DBD=*dbdname* DD/AREA=*ddname***

## Explanation

IMS Database Recovery Facility has determined that there is no System Level Backup (SLB), either at - or earlier than the specified recovery time, that contains *dbdname ddname*.

IMS Database Recovery Facility looks for a later non-SLB image copy and uses it for recovery, provided that it is later than the most recent SLB, and earlier than the specified recovery time.

If a non-SLB image copy does not exist in that range, messages FRD4212I and/or FRD4301I, FRD4302I, FRD4100I, and (under certain conditions) FRD4320I are written in the job log.

Note that under these circumstances, IMS Database Recovery Facility cannot use - for recovery - a non-SLB image copy that is earlier than the time of the most recent SLB. However, an earlier recovery time can be specified that then can be successfully used for recovery.

## System action

Recovery continues.

## User response

Check the recovery report to ensure that database recovery was successful with the use of a non-SLB image copy.

## Module

FRXVSTA0

**FRD4410E**      **macro MACRO,**  
                 **RETURN=X'returncode',**  
                 **[ REASON=X'reasoncode', ],**  
                 **PART=partname DSN=dsname**

## Explanation

IMS Database Recovery Facility internally invoked a macro instruction to gather information about the data set *dsname* for the HALDB partition *partname* as part of either of these following tasks:

- Starting a HALDB OLR for the HALDB partition *partname*.
- Tracking these changes at the RSR tracking site.

This macro instruction completed with a register 15 return code value of *returncode* and a register 0 value of *reasoncode*. These values were not expected by IMS Database Recovery Facility.

## System action

If this error occurred while starting the HALDB OLR, then the HALDB OLR is not started, but the partition remains accessible with the original active set of data sets--either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

In either of these cases, one or more of the output data sets might have already been created automatically.

## User response

Based on the macro indicated in the message, see the appropriate publication in the following table to determine the cause of the error, and follow any recommended actions therein:

Table 4. Publications That Contain Macro Information

| Macro   | Publication                                                                              |
|---------|------------------------------------------------------------------------------------------|
| DEVTYPE | <i>z/OS DFSMSdfp Advanced Services</i>                                                   |
| GETDSAB | <i>z/OS MVS Programming: Authorized Assembler Services Reference, Volume 2 (EDX-IXG)</i> |
| LOCATE  | <i>z/OS DFSMSdfp Advanced Services</i>                                                   |
| OBTAIN  | <i>z/OS DFSMSdfp Advanced Services</i>                                                   |
| SWAREQ  | <i>z/OS MVS Programming: Authorized Assembler Services Reference, Volume 4 (SET-WTO)</i> |
| TRKCALC | <i>z/OS DFSMSdfp Advanced Services</i>                                                   |

If the error was detected while processing the INITIATE OLREORG command, reissue the command for partition *partname*.

If the error was detected at the RSR tracking site, issue a START DB command for the partition in order for Online Forward Recovery (OFR) to continue updating the partition.

## Module

FRXIOCR0, FRXIOCC0, FRXDDCR0, FRXDDCC0

**FRD4411E**      **UNACCEPTABLE DATA SET,**  
                 **REASON CODE reasoncode.**  
                 **PART=partname DSN=dsname**

## Explanation

In determining whether to create the data set *dsname* for the HALDB partition *partname*, IMS Database Recovery Facility found one of the following situations:

- A data set of that same name exists; however, because of certain characteristics of this pre-existing data set, that data set could not be used for your intended purpose.
- No data set by that name was found when one should have existed previously.

#### Situation 1:

In starting a HALDB online reorganization for the HALDB partition *partname*, IMS Database Recovery Facility found that the data set that was to be used as the output of the reorganization had one of the following limitations:

- It was not of a type that could used for this purpose.
- It had certain characteristics that did not match those of the corresponding input data set.
- It did not exist.

#### Situation 2:

On the RSR tracking site, IMS Database Recovery Facility found that in tracking database changes for a HALDB online reorganization for the HALDB partition *partname*, the data set that was to be used as the shadow of an output data set had one of the following limitations:

- It was not of a type that could used for this purpose.
- It had certain characteristics that did not match those of the corresponding input data set.
- It did not exist.

In any case, *reasoncode* is a reason code the specific problem. These reason codes, environments, and explanations are shown in the table below. The Environment column indicates whether the error is detected at the beginning of the reorganization itself, the tracking of the reorganization at the RSR tracking site, or both.

Table 5. Reason Codes, Environments, and Errors Detected for Message FRD4411I

| Rsn Code | Environ | Errors Detected for Pre-Existing Data Set                                 |
|----------|---------|---------------------------------------------------------------------------|
| 1        | Both    | The data set is not a DASD data set.                                      |
| 2        | Both    | The data set is a VSAM data set, but the database is defined to use OSAM. |
| 3        | Both    | The data set is a PDS, PDSE, or a member of a PDS or PDSE.                |
| 4        | Both    | The data set is not a VSAM ESDS, but the database is defined to use VSAM. |

Table 5. Reason Codes, Environments, and Errors Detected for Message FRD4411I (continued)

| Rsn Code | Environ  | Errors Detected for Pre-Existing Data Set                                                                                                                                                                                                                                                                             |
|----------|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 5        | Both     | The VSAM data set does not have the REUSE attribute.                                                                                                                                                                                                                                                                  |
| 6        | Reorg.   | The VSAM data set has a different record length than that of the corresponding input data set.                                                                                                                                                                                                                        |
| 7        | Reorg.   | The VSAM data set has a different control interval size than that of the corresponding input data set.                                                                                                                                                                                                                |
| 8        | Reorg.   | The VSAM ESDS has a different control interval size than that of the corresponding input data set.                                                                                                                                                                                                                    |
| 10       | Reorg.   | The primary index data set of a PHIDAM database (the KSDS), is defined with a different key offset or key length than that of the corresponding input data set.                                                                                                                                                       |
| 11       | Tracking | The VSAM data set has a different record length than that of the tracked output data set at the active site.                                                                                                                                                                                                          |
| 12       | Tracking | The VSAM data set has a different control interval size than that of the tracked output data set at the active site.                                                                                                                                                                                                  |
| 13       | Tracking | The primary index data set of a PHIDAM database (the KSDS), is defined with a different key offset or key length than that of the tracked output data set at the active site.                                                                                                                                         |
| 18       | Both     | One of these situations exist: <ul style="list-style-type: none"> <li>• The data set is cataloged but it is not on the volume to which the catalog entry refers.</li> <li>• The data set was allocated with a DD statement that had a VOLUME parameter, but the data set was not on the referenced volume.</li> </ul> |
| 19       | Both     | The data set was allocated with a DD statement, but the data set name in the DSNAMES parameter does not follow the data set naming convention defined for this partition's data sets.                                                                                                                                 |



Table 5. Reason Codes, Environments, and Errors Detected for Message FRD4411I (continued)

| Rsn Code | Environ | Errors Detected for Pre-Existing Data Set                                                                                                |
|----------|---------|------------------------------------------------------------------------------------------------------------------------------------------|
| 20       | Both    | The output data set does not exist, but the corresponding input data set resides on multiple volumes and is not an SMS-managed data set. |

### System action

If this error occurred while starting the reorganization, then the reorganization is not started, but the partition remains accessible with the original active set of data sets--either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

In both cases, one or more of the output data sets might have already been created automatically.

### User response

#### Situation 1:

Before restarting the partition at the RSR tracking site, perform one of the following tasks:

- Delete the pre-existing output data set to allow IMS Database Recovery Facility to create it automatically. When IMS Database Recovery Facility creates the data set, the size and placement of the data set is determined automatically by IMS Database Recovery Facility, MVS, DFSMS/MVS, and also by parameters defined by the storage administrator. Therefore, do not let IMS Database Recovery Facility create the data set if you need precise control over the size and placement of the data set.
- Create a data set that is acceptable for tracking the output of the reorganization. Ensure that for an OSAM-defined database the output data set for the database segments (one of the A-through-J or the M-through-V data sets) meets the following criteria:
  - A DASD data set.
  - Not a VSAM data set.
  - Neither a PDS, PDSE, or a member of a PDS or PDSE.

Ensure that for a VSAM-defined database the output data set for the database segments (one of the A-through-J or the M-through-V data sets) has the following characteristics:

- A VSAM entry-sequenced data set (ESDS).
- The REUSE attribute.

- The same fixed-length record length as the tracked output data set at the active site (as shown in the X'2930' log record).
- The same control interval size as the tracked output data set at the active site (as shown in the X'2930' log record) or that for a PHIDAM database the output data set for the primary index (the X or Y data set) at the RSR tracking site has the following characteristics:
  - A VSAM key-sequenced data set (KSDS).
  - The REUSE attribute.
  - The same fixed-length record length as the tracked output data set at the active site (as shown in the X'2930' log record).
  - The same key offset and key length as the tracked output data set at the active site (as shown in the X'2930' log record).

Then issue a `INITIATE OLREORG` command to start the HALDB online reorganization for partition *partname*.

#### Situation 2:

Before restarting the partition at the RSR tracking site, perform one of the following tasks:

- Delete the pre-existing output data set at the tracking site to allow IMS Database Recovery Facility to create it automatically. When IMS Database Recovery Facility creates the data set, the size and placement of the data set is determined automatically by IMS Database Recovery Facility, MVS, DFSMS/MVS, and also by parameters defined by the storage administrator. Therefore, do not let IMS Database Recovery Facility create the data set if you need precise control over the size and placement of the data set.
- Create a data set that is acceptable for tracking the output of the reorganization. Ensure that for an OSAM-defined database, the output data set for the database segments (one of the A-through-J or the M-through-V data sets) at the RSR tracking site meets the following criteria:
  - It is a DASD data set.
  - It is not a VSAM data set.
  - It is neither a PDS, PDSE, nor a member of a PDS or PDSE.

Ensure that for a VSAM-defined database the output data set for the database segments (one of the A-through-J or the M-through-V data sets) at the RSR tracking site has the following characteristics:

- A VSAM entry-sequenced data set (ESDS).
- The REUSE attribute.

- The same fixed-length record length as the tracked output data set at the active site (as shown in the X'2930' log record).
- The same control interval size as the tracked output data set at the active site (as shown in the X'2930' log record) or that for a PHIDAM database the output data set for the primary index (the X or Y data set) at the RSR tracking site has the following characteristics:
  - A VSAM key-sequenced data set (KSDS).
  - The REUSE attribute.
  - The same fixed-length record length as the tracked output data set at the active site (as shown in the X'2930' log record).
  - The same key offset and key length as the tracked output data set at the active site (as shown in the X'2930' log record).

Then issue a /START DB command for partition partname in order for Online Forward Recovery to continue updating the partition.

#### Module

FRXIOCR0, FRXDDCR0

---

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>FRD4413E</b> | <b>NON-DASD DATA SET. DEVICE CLASS=X'devclass' PART=partname DSN=dsname</b> |
|-----------------|-----------------------------------------------------------------------------|

---

#### Explanation

The data set *dsname*, which is an input data set for a HALDB online reorganization of the HALDB partition *partname*, is not a DASD data set. The device class reported by the internally executed DEVTYPE macro instruction is *devclass* (hexadecimal) rather than X'20'. This is inconsistent with its use as part of a database.

#### System action

If this error occurred while starting the reorganization, then the reorganization is not started. The partition remains accessible, but successful use of it is unlikely.

If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

#### Module

FRXIOCR0, FRXDDCR0

---

|                 |                                                                                                                |
|-----------------|----------------------------------------------------------------------------------------------------------------|
| <b>FRD4414E</b> | <b>IDCAMS RETURN CODE <i>returncode</i> CREATING AN ONLINE REORG OUTPUT DATA SET. PART=partname DSN=dsname</b> |
|-----------------|----------------------------------------------------------------------------------------------------------------|

---

#### Explanation

IMS Database Recovery Facility tried to create the VSAM output data set *dsname* for the HALDB partition *partname* as part of either of the following tasks:

- Starting a HALDB online reorganization for the partition.
- Tracking these changes at the RSR tracking site.

Access Method Services (IDCAMS) provided a return code of *returncode*.

#### System action

The message, if one was produced by IDCAMS was displayed just before this message.

For return code 4, the data set is assumed to have been created. In this case, none of the following actions apply and processing continues.

If this error occurred while starting the reorganization, then the reorganization is not started, but the partition remains accessible with the original active set of data sets--either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

In either case, one or more of the output data sets might have already been created automatically.

#### User response

See the *z/OS MVS System Messages* to determine the cause of the error and follow any recommended actions.

Then if the error was detected while processing the INITIATE OLREORG command, reissue the command for partition *partname*.

If the error was detected at the RSR tracking site, issue a /START DB command for the partition in order for Online Forward Recovery to continue updating the partition.

#### Module

FRXIOCD0, FRXDCCD0

---

|                 |                                                                                                               |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| <b>FRD4415E</b> | <b>IDCAMS RETURN CODE <i>returncode</i> DELETING AN ONLINE REORG INPUT DATA SET. PART=partname DSN=dsname</b> |
|-----------------|---------------------------------------------------------------------------------------------------------------|

---

#### Explanation

IMS Database Recovery Facility tried to delete the VSAM input data set *dsname* for the HALDB partition *partname* as part of one of these tasks:

- Completing a HALDB online reorganization for the partition.
- Completing the tracking of these changes at the RSR tracking site.

Access Method Services (IDCAMS) provided a return code of *returncode*.

### System action

The message, if one was produced by IDCAMS was displayed just before this message.

The reorganization or the tracking of the reorganization at the RSR tracking site is still considered to have completed successfully, and there is now a single new active set of data sets for partition *partname*--either the A-through-J and X or the M-through-V and Y data sets.

### User response

See the *z/OS MVS System Messages* to determine the cause of the error and follow any recommended actions, then delete the data set if it has not already been deleted.

### Module

FRXIOCD0, FRXDDCD0

---

|                 |                                                                                                                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FRD4416E</b> | <b>NOT ENOUGH DASD SPACE.</b><br><b>ERROR REASON=X's99errorcode'</b><br><b>[ SMS REASON=s99ersn ]</b><br><b>BLOCKS=blocks BLKSIZE=blksize</b><br><b>VOLUMES=volcount</b><br><b>{ UNIT=SYSALLDA STORAGE</b><br><b>CLASS=storageclassname }</b><br><b>PART=partname DSN=dsname</b> |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

### Explanation

IMS Database Recovery Facility tried to create the OSAM output data set *dsname* for the HALDB partition *partname* as part of either of the following tasks:

- Starting a HALDB OLR for the partition.
- Tracking these changes at the RSR tracking site.

The internally executed DYNALLOC macro instruction (SVC 99) that was used for this purpose reported that the data set could not be created because of DASD space or DASD volume constraints. The requested space was *blocks* blocks, each with a block size of *blksize*. There were *volcount* volumes requested.

If the text showing the group name (SYSALLDA) appears in the message text, IMS Database Recovery Facility attempted to create a non-SMS-managed data set because the corresponding input data set was not SMS-managed. If the text showing the storage class

appears, then an attempt was made to create an SMS-managed data set using storage class *storageclassname* because the corresponding input data set was SMS-managed and was created using that storage class.

If the reorganization was being started, these space allocation values were derived from those of the corresponding input data set. On the RSR tracking site, these space allocation values were derived from those of the tracked output data set at the active site where the reorganization occurred.

For further information of this error, see the *z/OS MVS Programming: Authorized Assembler Services Guide*. The value *s99errorcode* is the hexadecimal value of the error reason code (field S99ERROR in the SVC 99 request block) that is provided when the DYNALLOC macro instruction (SVC 99) fails with return code 4. In the SMS case, *s99ersn* is the SMS reason code (field S99ERSN in the SVC 99 request block extension). The SMS reason code corresponds to the message number of an IGD message describing the error. For further information on the IGD message, see the *z/OS MVS System Messages*, Vol 8.

### System action

If this error occurred while starting the reorganization, then the reorganization is not started, but the partition remains accessible with the original active set of data sets--either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

In either case, one or more of the output data sets might have already been created automatically.

### User response

Determine whether the requested space (*blocks* blocks of size *blksize*) is really required for the output data set. In some cases, this space allocation request could include space acquired in a secondary allocation for the input data set and might not be required for the reorganized database. Then perform one of the following tasks:

- Ensure that there is sufficient DASD space on the volumes referenced by the group name SYSALLDA (for non-SMS-managed) or referenced by the storage class *storageclassname* (for SMS-managed), possibly by scratching unneeded data sets. Note that the primary allocation amount (*blocks* blocks of size *blksize*) must be available in a total of no more than five extents on a single volume. For SMS-managed data sets, if the storage class definition includes the GUARANTEED SPACE attribute, then the primary allocation amount must be available and will be

allocated on each of the volumes (rather than on just the first volume).

- Create and catalog this output data set yourself. In this case, you can reserve whatever amount of space you choose, and you can control the number of volumes. See the *IMS Administration Guide: Database Manager* for information on allocating OSAM data sets and for a technique for pre-allocating multi-volume data sets.

Then if the error was detected in processing the INITIATE OLREORG command, reissue the command for partition *partname*.

If the error was detected at the RSR tracking site, issue a START DB command for the partition in order for Online Forward Recovery to continue updating the partition.

## Module

FRXIOCD0, FRXDCCD0

---

|                 |                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------|
| <b>FRD4417E</b> | <b>DATA SET ALLOCATED TO ANOTHER JOB AND CANNOT BE CREATED. PART=<i>partname</i> DSN=<i>dsname</i></b> |
|-----------------|--------------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility tried to create the OSAM output data set *dsname* for the HALDB partition *partname* as part of one of the following tasks:

- Starting a HALDB OLR for the partition.
- Tracking these changes at the RSR tracking site.

Creating the data set required that the data set be allocated as though DISP=NEW had been specified on a JCL DD statement. However, a data set of the same name was already in use by another job, causing the allocation to fail.

## System action

If this error occurred while starting the reorganization, then the reorganization is not started, but the partition remains accessible with the original active set of data sets--either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

In either case, one or more of the output data sets might have already been created automatically.

## User response

Ensure that no other job is using a data set of the name *dsname* even if that data set is not cataloged

and even if that data set is not intended to be the same data set as the output of the reorganization.

Then if the error was detected while processing the INITIATE OLREORG command, reissue the command for partition *partname*.

If the error was detected at the RSR tracking site, issue a START DB command for the partition in order for Online Forward Recovery to continue updating the partition.

## Module

FRXIOCD0, FRXDCCD0

---

|                 |                                                                                                                                                                                                                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FRD4418E</b> | <b>[ DATA SET CREATION ERROR. UNALLOCATION FAILURE. ] RETURN=<i>returncode</i> [ { INFO. REASON=X's99infocode' ERROR REASON=X's99errorcode' } ] [ SMS REASON=<i>s99ersn</i> ] BLOCKS=<i>blocks</i> BLKSIZE=<i>blksize</i> VOLUMES=<i>volcount</i> { UNIT=SYSALLDA STORAGE CLASS=<i>storageclassname</i> } PART=<i>partname</i> DSN=<i>dsname</i></b> |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility tried to create the OSAM output data set *dsname* for the HALDB partition *partname* as part of one of the following tasks:

- Starting a HALDB OLR for the partition.
- Tracking these changes at the RSR tracking site.

As indicated by the message text, the internally executed DYNALLOC macro instruction (SVC 99) reported one of these following problems:

- The data set could not be created.
- The data set that was just created successfully could not be unallocated from the IMS Database Recovery Facility job.

The requested space was *blocks* blocks, each with a block size of *blksize*. There were *volcount* volumes requested.

When the text showing the group name (SYSALLDA) appears in the message text, the attempt was to create a non-SMS-managed data set because the corresponding input data set was not SMS-managed. If the text showing the storage class appears, then the attempt was to create an SMS-managed data set using storage class *storageclassname* because the corresponding input data set was SMS-managed and was created using that storage class.

If the reorganization was being started, these space allocation values were derived from those of the

corresponding input data set. On the RSR tracking site, these space allocation values were derived from those of the tracked output data set at the active site where the reorganization occurred.

For the meaning of the various error codes shown in this message, see the *z/OS MVS Programming: Authorized Assembler Services Guide*.

#### **returncode**

The decimal value of the register 15 return code provided by the DYNALLOC macro instruction (SVC 99).

#### **s99infocode**

The hexadecimal value of the information reason code (field S99INFO in the SVC 99 request block).

#### **s99errorcode**

The hexadecimal value of the error reason code (field S99ERROR in the SVC 99 request block).

#### **s99ersn**

For SMS; the SMS reason code (field S99ERSN in the SVC 99 request block extension). The SMS reason code corresponds to the message number of an IGD message describing the error. For more information on the IGD message, see the *z/OS MVS System Messages*, Vol 8.

#### **System action**

If this error occurred while starting the reorganization, then the reorganization is not started, but the partition remains accessible with the original active set of data sets--either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline.

In either case, one or more of the output data sets might have already been created automatically.

#### **User response**

Using the information you can find in the *z/OS MVS Programming: Authorized Assembler Services Guide*, determine the meaning of *returncode*, *s99info*, *s99errorcode*, and *s99ersn* as applicable. Follow any recommended actions to correct the error.

Then if the error was detected while processing the INITIATE OLREORG command, reissue the command for partition *partname*.

If the error was detected at the RSR tracking site, issue a START DB command for the partition in order for Online Forward Recovery to continue updating the partition.

#### **Module**

FRXIOCD0, FRXDDCD0

---

### **FRD4419E**

#### **Explanation**

The message indicates internal error with these fields:

- RVGB\_PTR
- Module ID || module subcode
- ORA Data Set Communication Area
- Reason code indicating which internal error

#### **System action**

System terminates.

#### **System programmer response**

Provide information to IBM Support.

#### **Module**

FRXDDDF0, FRXIOCR0, FRXIODF0, FRXIOMG0.

---

|                 |                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FRD4420E</b> | <b>CSI RETURN CODE=4, [ { CATALOG ERROR, DATA SET ERROR, } ]<br/>CATALOG MANAGEMENT RETURN CODE=<i>cmreturncode</i> REASON CODE=<i>cmreasoncode</i><br/>MODULE=<i>cmmoduleid</i><br/>PART=<i>partname</i> DSN=<i>dsname</i></b> |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

#### **Explanation**

When the CSI return code is 4, a catalog management error was detected. In this case, *cmreasoncode*, *cmreturncode*, and *cmmoduleid* are the catalog management reason code, return code, and two-character module identification, in that order. For more information about catalog management errors, see the *z/OS MVS System Messages*.

#### **System action**

See FRD4421I.

#### **User response**

See FRD4421I.

#### **Module**

See FRD4421I.

---

|                 |                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------|
| <b>FRD4421I</b> | <b>CSI RETURN CODE=<i>csireturncode</i><br/>[ REASON CODE=<i>csireasoncode</i> ]<br/>PART=<i>partname</i> DSN=<i>dsname</i></b> |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------|

---

### Explanation

IMS Database Recovery Facility internally invoked the Catalog Search Interface (CSI) to gather information about the data set *dsname* for the HALDB partition *partname* as part of either:

- Starting a HALDB online reorganization for the partition.
- Tracking these changes at the RSR tracking site.

The CSI completed with a CSI return code value as shown in the message text. For more information on CSI return codes, see the *z/OS DFSMS Managing Catalogs*.

A CSI error was detected. For CSI return code 8, *csireasoncode* is the CSI reason code. Information on this CSI error information, see the *z/OS DFSMS Managing Catalogs*. These values were not expected by IMS Database Recovery Facility.

### System action

If this error occurred while starting the reorganization, then the reorganization is not started, but the partition remains accessible with the original active set of data sets- either the A-through-J and X or the M-through-V and Y data sets. If this error occurred at the RSR tracking site, then the shadow partition is taken offline. In either case, one or more of the output data sets might have already been created automatically.

### User response

Based on value of the CSI return code, see the *z/OS DFSMS Managing Catalogs* to determine the cause of the error and follow the recommended actions. If the error was detected in processing the INITIATE OLREORG command, reissue the command for partition *partname*.

If the error was detected at the RSR tracking site, issue a START DB command for the partition in order for Online Forward Recovery to continue updating the partition.

### Module

FRXIOMG0

---

**FRD6002E DBRC request-type REQUEST FAILURE, RC= return code**

### Explanation

A request of type request-type was made to DBRC that did not successfully complete. The return code from DBRC is returned in the message. See the applicable table under the heading "DBRC Request Return Codes" in the topic titled "DSP Messages" in *IMS*

*Messages and Codes* for a description of the DBRC request return codes. In the message text:

### request-type

The type of DBRC request that failed. request-type is one of the following:

- SIGNON
- SIGNOFF
- INIT
- RECOV START
- RECOV COMPLETE
- RECOV UPDATE
- BUILD RWCR
- AUTHORIZATION
- UNAUTHORIZATION

### System action

The action taken will depend on the request being made:

### SIGNON

Recovery processing will not be started. If this is a batch type recovery, the IMS Database Recovery Facility address space ends.

### SIGNOFF

If no other failure has occurred, then recovery processing has completed. The subsystem record for this recovery job will remain recorded in RECON.

### INIT

The IMS Database Recovery Facility was unable to initialize DBRC. The IMS Database Recovery Facility address space ends.

### RECOV START

The request to obtain recovery information to start recovery processing failed. NO recovery processing will be done.

### RECOV COMPLETE

The request to notify DBRC of the completion of recovery failed. No information regarding the recovery will be recorded in RECON.

### RECOV UPDATE

The request to obtain new recovery information after a REC STOP command failed. The RECOVER STOP request will be ignored.

### BUILD RWCR

The request to resolve the entry name on a RECOVER command to a list of data entries failed. The command processing fails. For a batch recovery, the recovery process stops.

## AUTHORIZATION

The request to authorize full-function databases or Fast Path Areas failed. No recovery processing will be done.

## UNAUTHORIZATION

The request to unauthorize full-function databases or Fast Path Areas failed. Processing continues..

### User response

See the topic titled "DSP Messages" in *IMS Messages and Codes* for an explanation of the DBRC return and reason codes and take appropriate action before attempting the recovery process again. If the explanation indicates an internal error, call IBM Software Support for assistance.

### Module

FRXVADD0, FRXVSTA0, FRXVREM0, FRXVSTO0,  
FRXMSTRO, DFSMSTEO, FRXVAUTO

---

**FRD6003I**      *name* <AUTHORIZED BY *ssid*>

---

### Explanation

This message lists the database data set or area for which a command was processed by the IMS Database Recovery Facility. Either FRD6011I, FRD6016I, FRD6021I or FRD6032I precedes this message. This message is sent to the issuer of the command.

In the message text:

#### **name**

Indicates the database data set name (dbname ddname) or area name for recovery.

#### **ssid**

In response to a RECOVER ADD or ADD command, if the database data set or area is authorized by an IMS, *ssid* identifies the authorizing ssid. This message is issued for each IMS that authorized a the database data set or area in the recovery list.

This message is issued for each IMS that authorized a database data set or area in the recovery list.

### System action

RECOVER command processing continues.

### User response

None. This message is informational.

### Module

FRXVSTA0, FRXVSTO0, FRXVADD0, FRXVREM0

---

## FRD6010W

**UNABLE TO ADD TO *rcvlist*,  
*entrytype*, *entryname*: *reason***

---

### Explanation

The entry name specified in the ADD command could not be added to the specified recovery list. If a duplicate entry was detected, the duplicate is ignored. In the message text the following information is shown:

#### ***rcvlist***

The specified recovery list.

#### ***entrytype***

If a member of a group is not defined in the RECON, "entrytype" is not included in the message.

#### ***entrytype***

One of the following:

- DBDS
- AREA
- DATABASE
- GROUP (if DBDSGRP, CAGRP, or RECOVGRP was specified in the RECOVER command)

#### ***entryname***

A keyword specified on the RECOVER ADD or ADD command and is one of the following:

- DB
- DBDS
- DBDSGRP
- CAGRP

#### ***reason***

The reason can be one of the following:

#### **ALREADY STARTED**

The recovery list specified by *rcvlist* already started recovery processing.

#### **DATABASE HAS NO DATA SETS**

The database does not have any database data sets defined to it in RECON.

#### **DATABASE IS DEFINED AS GSAM**

The IMS Database Recovery Facility cannot recover GSAM database types.

#### **DATABASE IS OLR CAPABLE**

The object specified by *entryname* is OLR capable and the ADD or ADD command was issued to a release of IMS Database Recovery Facility that does not support recoveries with OLR capable databases.

#### **FAST PATH NOT INSTALLED**

The RECOVER ADD command attempted to add a Fast Path AREA to the recovery list on a system that does not have FP installed.

### INDEX OR ILE DATA SET

The DBDS has not been added to the recovery list because it is either a Prime Index or ILDS data set and cannot be recovered. These data sets are recovered by the Index/ILDS Rebuild utility.

### NOT DEFINED IN RECON

The object specified by *entryname* is not defined for RECON.

### NOT FOUND IN ACBGEN

The *AREA* specified on the RECOVER ADD *AREA* command was not defined to the IMS receiving the command.

### NOT IN CAGRP FOR ICRCA

One or more entries in the recovery list do not belong to a CAGRP. The OUTPUT(ICRCA) option requires that all entries belong to a CAGRP.

### UNKNOWN REASON FROM DBRC

DBRC returned an unknown reason code to the IMS Database Recovery Facility.

### DATAGROUP NOT SUPPORTED

A RECOVER ADD DATAGROUP *groupname* command was entered. The IMS Database Recovery Facility does not support the DATAGROUP option.

### INVALID SMSOPT PARAMETER

The only SMSOPT parameter currently supported is DELCAT.

### IT IS A DATAGROUP

The group that is specified for an ADD DBDSGRP or ADD RECOVGRP named a DATAGROUP.

### IT IS A RECOVGRP

The group that is specified for an ADD DBDSGRP named a RECOVGRP.

### IT IS A DBDSGRP

The group that is specified for an ADD RECOVGRP named a DBDSGRP.

### System action

The database data sets that make up the entry name are not added to the recovery list. The databases, areas, or both are not recovered by the IMS Database Recovery Facility.

### User response

Examine the RECOVER ADD or ADD command and compare it with the corresponding ACBGEN or RECON entries. Reissue the command with the correct entry names. If reason is UNKNOWN REASON FROM DBRC, call the IBM Support Center for assistance.

### Module

FRXVADD0

### FRD6011I

---

**THE FOLLOWING ENTRIES ARE  
ADDED TO THE RECOVERY LIST  
*token*:**

### Explanation

This message is followed by a series of FRD6003I messages. It indicates that one or more entries were added to the recovery list as a result of the RECOVER ADD command. *token* is supplied by the user on the RECOVER ADD command or generated by IMS.

This message is sent to the issuer of the command.

### System action

The database data set or area name listed in the subsequent FRD6003I messages are added to the recovery list.

### User response

None. This message is informational.

### Module

FRXVADD0

### FRD6012I

---

**RECOVER OPTIONS IGNORED:  
*SMSOPT***

### Explanation

When recovery list is overflowed, any SMS option will be ignored except DELCAT. The only *SMSOPT* parameter currently supported is DELCAT.

### System action

Processing continues, ignoring listed options.

### Module

FRXVADD0

### FRD6013I

---

**NOTHING ADDED TO RECOVERY  
LIST: *reason***

### Explanation

The processing of the ADD command did not add anything to a recovery list. Either no entry information was found by DBRC for any item listed in the command or all the entries returned by DBRC based on the items in the command were already in the recovery list. In the message text:



**reason**

The *reason* can be one of the following:

**ALREADY ON RECOVERY LIST**

All entries to be added by this command are already in the recovery list.

**NO ENTRIES TO ADD**

No entry information was found by DBRC for any item listed in the command.

**Note:**

See prior FRD6010I message for more details.

**System action**

The recovery list remains unchanged. If this is a batch recovery attempt, recovery processing will not occur.

**User response**

Examine the RECOVER ADD or ADD command and compare it with the corresponding ACBGEN or RECON entries and previous ADD commands to determine if the correct names were being specified. If *reason* is NOENTRIES TO ADD and this is a batch recovery, the command can be removed from the command stream if the correct names were being specified.

**Module**

FRXVADD0

---

|                 |                                                                     |
|-----------------|---------------------------------------------------------------------|
| <b>FRD6014E</b> | <b>DUPLICATE DSN TOO LONG FOR<br/><i>dbname, ddname, reason</i></b> |
|-----------------|---------------------------------------------------------------------|

**Explanation**

IMS Database Recovery Facility constructed a DSN that was too long.

***dbname***

Identifies the database.

***ddname***

Identifies the DD.

***reason***

Identifies the reason, for example date, prefix, suffix, or time.

**System action**

Processing continues.

**User response**

Attempt to adjust the variables used in duplicate DSN construction.

**Module**

FRXVADD0

---

|                 |                                                                           |
|-----------------|---------------------------------------------------------------------------|
| <b>FRD6016I</b> | <b>THE FOLLOWING ENTRIES WERE<br/>REMOVED FROM THE RECOVERY<br/>LIST:</b> |
|-----------------|---------------------------------------------------------------------------|

**Explanation**

This message is followed by a series of FRD6003I messages. It indicates that one or more entries were removed from the recovery list as a result of RECOVER REMOVE command.

**System action**

Then names listed in the subsequent DFS4265I messages are removed from the recovery list.

**User response**

None. This message is informational.

**Module**

FRXVREM0

---

|                 |                                   |
|-----------------|-----------------------------------|
| <b>FRD6017I</b> | <b>RECOVERY LIST IS NOW EMPTY</b> |
|-----------------|-----------------------------------|

**Explanation**

One of the following instances occurred:

- The RECOVER REMOVE command processed and removed the last database data set or area from the recovery list.
- The RECOVER REMOVE ALLENT command processed.
- The RECOVER STOP command processed and stopped the last database data set or area in the recovery list.
- The RECOVER STOP ALLENT command processed.

This message is sent to the issuer of the command.

**System action**

The recovery instance is deleted.

**User response**

None. This message is informational.

**Module**

FRXVREM0

---

|                 |                                                                                         |
|-----------------|-----------------------------------------------------------------------------------------|
| <b>FRD6018W</b> | <b>UNABLE TO REMOVE<br/><i>resource_type</i><i>resource_name</i>:<br/><i>reason</i></b> |
|-----------------|-----------------------------------------------------------------------------------------|

**Explanation**

An IMS RECOVER REMOVE command was issued but could not process. This message is sent to the issuer of the command.

In the message text:

**resource\_type**

Identifies the type of resource the RECOVER REMOVE command was issued against, and is one of the following:

- LIST
- DBDS
- AREA
- DB

**resource\_name**

Identifies the name of the resource specified in the RECOVER REMOVE command.

**reason**

Indicates one of the following causes for the failures:

**RECOVERY IN PROGRESS**

A RECOVER REMOVE command was issued when recovery was in progress for the recovery list.

**NOT IN RECOVERY LIST**

The database data set or area specified in the command was not part of the recovery list.

**NOT DEFINED IN RECON**

The DB, CAGRP, or DBDSGRP, resource types could not be found in RECON.

**NOTHING REMOVED**

No database data set or area was removed as a result of the RECOVER REMOVE command.

**RECOVERY LIST DOES NOT EXIST**

A RECOVER REMOVE command was issued, but the identified recovery list does not exist.

**THIS IS A FAST PATH AREA**

A RECOVER REMOVE DBDS was specified in the command, but the entry was a Fast Path area.

**THIS IS NOT A FAST PATH AREA**

A REC REMOVE AREA`areaname` command was entered and the `areaname` identified an FF DBDS in the recovery list. Determine the appropriate `areaname`, or enter RECOVER REMOVE DBDS`dbd dbds`.

**DATAGROUP NOT SUPPORTED**

A RECOVER ADD DATAGROUP`groupname` command was entered. The IMS Database Recovery Facility does not support the DATAGROUP option.

**System action**

If the `resource_type` is LIST, the RECOVER REMOVE command is aborted. Otherwise, processing continues for those resources not identified by the error.

**User response**

Determine why the command failed and if necessary, reissue the command.

**Module**

FRXVREMO

|                 |                                            |
|-----------------|--------------------------------------------|
| <b>FRD6021I</b> | <b>RECOVERY STARTED FOR<br/>(options):</b> |
|-----------------|--------------------------------------------|

**Explanation**

The RECOVER START command was issued and no errors were detected. The IMS Database Recovery Facility processing started for the database data set and areas listed in the subsequent FRD6003I messages. In the message text:

**options**

Specifies parameters on the RECOVER START command. If no parameters were specified, the following default values for the options are listed:

- ERRORCONT
- ERRORSTOP
- STAGLOBAL
- STALOCAL
- NOSTA

This message is sent to the issuer of the command.

**System action**

RECOVER START command continues.

**User response**

None. This message is informational.

**Module**

FRXVSTAO

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| <b>FRD6022A</b> | <b>UNABLE TO RECOVER: <code>dbname</code>,<br/>reason</b> |
|-----------------|-----------------------------------------------------------|

**Explanation**

A RECOVER START command was issued, but the database data set or area cannot be recovered. This message is sent to the issuer of the command.

In the message text:

**dbname**

Identifies the database data set or area that could not be recovered.

**reason**

Indicates the cause of the error and is one of the following:

**AUTHORIZATION FAILED, RSN = XX**

Authorization processing failed to properly authorize one or more of the database data sets or areas in the recovery list. XX identifies the DBRC reason code. The DBRC reason codes are documented in the *IMS Messages and Codes* manual, under message DFS047A.

**IMAGE COPY ALREADY TAKEN**

The OUTPUT(ICR) option was specified, however there were no changes since the last incremental image copy was taken.

**IMAGE COPY NOT AVAILABLE**

DBRC was unable to locate a usable image copy for the database data set.

**INCOMPLETE CA NOT ALLOWED**

An incomplete change accumulation was detected during an attempt to generate a batch Incremental Image Copy using the ICRCA option. A complete change accumulation is required.

**INVALID RCVTIME**

DBRC determined that the RCVTIME specified falls in the middle of an ALLOC for this entry, and the PITR option was not specified.

**LOG NOT ARCHIVED**

Logs that are selected for recovery are not archived. Verify that the logs are in the RECON, then run an archive job for the unarchived OLDS.

**LOGS MARKED IN ERROR**

DBRC determined that none of the log data sets are usable. They might be marked in error. Or DBRC was unable to locate log data sets for recovery from only secondary sources.

**NEEDS HIGHER RELEASE LOG**

The IMS Database Recovery Facility detected logs generated by a higher release IMS. The database cannot be selected if it was updated by an IMS operating at a higher release.

**NO BATCH ICR POINT FOUND**

IMS Database Recovery Facility attempted to generate a batch Incremental Image Copy and was unable to. A usable batch window does not exist.

**NO IC FOUND FOR LASTIC**

RCVTYPE(LASTIC) was specified, however no IC exists where the DSDS/Area has been stopped.

**NO CA FOUND FOR PITCA or LASTPITCA**

A recovery was attempted using a PITCA, but no PIT CA was found. A PITCA must have a RUN time which matches the RCVTIME timestamp for a PITCA recovery. A PITCA must be marked ERR, and the CA data set name must have the suffix .HPCAP.

**NO IC TAKEN AFTER A PITR**

The IMS Database Recovery Facility requires input for recovery (in the form of an image copy, logs, and change accumulation). The input that was received is not allowed. Because the later PITR recovery might have removed Backout information from the RECON, the integrity of the database might be compromised by this recovery. After a PITR recovery, an image copy must be taken.

Recovery of the identified DBDS failed.

**NO PRILOG FOUND**

DBRC was unable to locate a PRILOG record. This condition should not occur. Call IBM Software Support for assistance.

**NO RECOV RECORD FOUND**

The USEDDBDS option was specified on the RECOVER ADD command, but the latest recovery recorded in RECON was not for a time stamp recovery, or the DBDS or AREA is marked in RECON as needing recovery.

**NOT FOUND**

One or more of the database data sets or areas in the recovery list is not listed in the RECON data set.

**NOT OFFLINE**

One or more of the database data sets or areas in the recovery list was not taken offline with an IMS DBR command. When the database is in this state, proper operation of the IMS Database Recovery Facility cannot be guaranteed.

**RECOV RECORD UNUSABLE**

The USEDDBDS option was specified on the RECOVER ADD command, but the DBDS or AREA has been updated or reorganized since the latest recovery recorded in the RECON.

**REORG LATER THAN IMAGE COPY**

A recovery to a time that includes an ALLOC range after a REORG, cannot be allowed unless an image copy was taken after the REORG job. If this situation is allowed, recovery applies data produced under one version of the

database (before REORG) and the data from the reorganized version of the database; this will undoubtedly produce a broken database. Take an image copy after your REORG job.

### **System action**

If ERRORCONT was specified for the database data set or area, processing continues for the remaining members in the IMS Database Recovery Facility recovery list. If ERRORCONT was not specified, recovery stops.

### **User response**

Take the appropriate action based on the specified reason:

#### **AUTHORIZATION FAILED**

The DBRC reason codes are documented in the *IMS Messages and Codes* manual under message DFS047A. If the reason indicates that a database or area is in use, determine which IMS did not stop the appropriate database or area, then stop the database or area with the IMS DBR command. Otherwise, take the action indicated in the DFS047A message.

#### **IMAGE COPY ALREADY TAKEN**

This is an informational message. No action is required.

#### **IMAGE COPY NOT AVAILABLE**

If this failure occurred while recovering from only secondary sources (SOURCE(SEC) was specified), verify that secondary image copies were created prior to running recovery.

#### **INCOMPLETE CA NOT ALLOWED**

In order to be able to successfully run with the ICRCA option, modify your process to ensure that complete change accumulations are generated. Ensure that the complete change accumulations are generated to a time where no transactions occur against the database.

#### **INVALID RCVTIME**

Ensure that the RCVTIME specified falls in the middle of an ALLOC for this entry or specify the Pitr option if a point-in-time recovery is desired.

#### **LOGS MARKED IN ERROR**

If this failure occurred while recovering from only secondary sources (SOURCE(SEC) was specified), verify that SECLOGs and/or SECSLDs were generated for this database data set prior to running recovery.

#### **NEEDS HIGHER RELEASE LOGS**

Logs generated on a higher release IMS cannot be used for recovery on a lower release IMS. This situation can occur when running multiple IMS

systems in coexistence mode. Rerun the recovery on the level of IMS in which the logs were created.

#### **NO BATCH ICR POINT FOUND**

A batch image copy cannot be generated. Ensure that a batch window exists.

#### **NO IC FOUND FOR LASTIC**

A RCVTYPE(LASTIC) recovery is not possible.

#### **NO CA FOUND FOR PITCA or LASTPITCA**

For a PITCA recovery, verify that the RCVTIME timestamp matches the RUN time listed for the CA record in the RECON created by the HPCA PIT. Also, the CA must be marked ERR and the CA data set name must have the suffix .HPCAP.

#### **NO IC TAKEN AFTER A Pitr**

After a point-in-time recovery, take an image copy so that subsequent recoveries have a valid starting point. If you know the integrity of the database will not be compromised, delete the RECOV record by using the DBRC DELETE .RECOV command, then rerun the recovery.

#### **NO PRILOG FOUND**

Call IBM Software Support for assistance.

#### **NOT FOUND**

The database data set or area was deleted from the RECON data set after the RECOVER ADD command was issued. If the database data set or area is legitimate and needs to be recovered, define it in RECON.

#### **NOT OFFLINE**

The database data set or area was not stopped on the IMS performing recovery. Stop the database or area with the IMSDBR command.

#### **NO RECOV RECORD FOUND**

Recover the DBDS or AREA using a nonstandard image copy. Then inform DBRC of the recovery with a NOTIFY .RECOV command specifying the time that the recovery was run (current), and the time to which the DBDS or AREA was recovered. Then the IMS Database Recovery Facility can be used to recover the DBDS or AREA with the USEDDBDS parameter.

#### **RECOV RECORD UNUSABLE**

Changes have been made to the DBDS or AREA since the last recovery; as a result, you must run the recovery again. Recover the DBDS or AREA using a nonstandard image copy. Then, inform DBRC of the recovery with a NOTIFY .RECOV command specifying the time that the recovery was run (current) and the time to which the DBDS or AREA was recovered. The IMS Database Recovery Facility can then be used to recover the DBDS or AREA with the USEDDBDS parameter.

### REORG LATER THAN IMAGE COPY

Inspect the database and ensure that it is in the same workable state as it was before the REORG ran. To issue an IMS Database Recovery Facility time stamp recovery to before the REORG runtime you must first delete the REORG by issuing the DELETE . REORG command.

### Module

FRXVSTAO

---

|                 |                                                     |
|-----------------|-----------------------------------------------------|
| <b>FRD6023A</b> | <b>START COMMAND FOR process<br/>FAILED, reason</b> |
|-----------------|-----------------------------------------------------|

---

### Explanation

The RECOVER START command was issued but the recovery or verify process cannot continue, as indicated by reason:

#### ERRORABORT SPECIFIED

An error was detected with one database data set or area while the IMS Database Recovery Facility attempted to start recovery, and ERRORCONT was specified on the RECOVER START command.

#### INCOMPLETE RECOVERY GROUP

Recovery failed because one or more members of a recovery group were not included in the recovery list. This message is followed by one or more FRD6024I messages listing the members of the recovery group that are not in the recovery list.

#### NO RECOVERABLE ENTRIES

None of the entries in the recovery list can be recovered due to one or more errors.

#### RECOVERY LIST NOT FOUND

No recovery list was found with a token matching the one provided with the command.

#### ALREADY IN PROGRESS

The recovery was not started because it is already processing.

#### OTHER RECOVERY IN PROGRESS

The recovery was not started because another recovery is currently in progress.

#### RECOVERY TIME GREATER THAN RUN TIME

The input recovery time on a RECOVER START command is greater than the current time.

#### DBRC RECOVER START FAILURE

The log data sets returned by DBRC for recovery are not ordered correctly. This condition should not occur.

#### INVALID RECOVERY TIME

The RCVTIME specified is not in a valid time stamp format.

This message is sent to the issuer of the command.

### System action

The current the IMS Database Recovery Facility recovery instance is aborted.

### User response

Take the appropriate action based on the reason received:

#### ERRORABORT SPECIFIED

Correct the error and restart recovery. See message DFS4266I in *IMS Messages and Codes* for the proper action.

#### INCOMPLETE RECOVERY GROUP

Specify the whole group, redefine the group, don't specify PITR, or specify NOCHECK. See FRD6024I for additional information.

#### NO RECOVERABLE ENTRIES

Prior messages should have been issued regarding failures for each entry. See the previous messages for the appropriate action.

#### RECOVERY LIST NOT FOUND

Create a recovery list with the name you specified or enter the name of an existing list.

#### ALREADY IN PROGRESS

The recovery is running. No action is necessary.

#### OTHER RECOVERY IN PROGRESS

Stop the other recovery or wait until it finishes.

#### RECOVERY TIME GREATER THAN RUN TIME

Enter the command RECOVER START with the correct time to start recovery.

#### DBRC RECOVER START FAILURE

Call IBM Software Support for assistance.

#### INVALID RECOVERY TIME

Ensure the RCVTIME specified is in an acceptable time stamp format.

### Module

FRXVSTAO

---

|                 |                                                                                        |
|-----------------|----------------------------------------------------------------------------------------|
| <b>FRD6024A</b> | <b>RECOVERY GROUP <i>grpname</i><br/>MEMBER <i>member</i> NOT IN<br/>RECOVERY LIST</b> |
|-----------------|----------------------------------------------------------------------------------------|

---

### Explanation

The RECOVER START command was issued with members of the recovery group indicated in the message by *grpname*. However, not all members of the recovery groups are in the recovery list. FRD6024I is issued for each member of the recovery list. This message follows message FRD6023I.

In the message text:

**grpname**

The name of the recovery groups

**member**

The database or area name in the recovery group.

**System action**

The command is not executed.

**User response**

Add all the members of the recovery groups to the recovery list using the ADD command, then issue START, or issue RECOVER START with the NOCHECK parameter specified.

**Module**

FRXVSTAO

---

**FRD6031E**      **UNABLE TO STOP**  
*resource\_type**resource\_name:*  
*reason*

**Explanation**

An IMS RECOVER STOP command was issued but could not process. This message is sent to the issuer of the command.

In the message text:

**resource\_type**

Identifies the type of resource the RECOVER STOP command was issued against, and is one of the following:

- LIST
- DBDS
- AREA
- DB

**resource\_name**

Identifies the name of the resource specified in the RECOVER STOP command.

**reason**

Indicates the cause of the error, and is one of the following:

**RECOVERY NOT IN PROGRESS**

A RECOVER STOP command was issued when recovery was not in progress.

**ALLENT REQUIRED**

A RECOVER STOP command was issued but ALLENT was not specified. Only RECOVERY STOP ALLENT<save> is supported by the IMS Database Recovery Facility.

**EARLY END OF RECOVERY**

A RECOVER STOP command was entered while early end of recovery was in process. The command is ignored.

**System action**

If the *resource\_type* is LIST, then the RECOVER STOP command aborts. Otherwise, processing continues for other resources.

**User response**

Determine why the command failed and if necessary, reissue the command.

**Module**

FRXVSTO0

---

**FRD6032I**      **THE FOLLOWING ENTRIES WILL**  
**HAVE RECOVERY STOPPED:**

**Explanation**

This message is followed by a series of FRD6003I messages. It indicates that one or more entries will have recovery stopped as a result of the RECOVER STOP command. This message is sent to the issuer of the command.

**System action**

The name listed in the subsequent FRD6003I messages will have their recovery stopped.

**User response**

None. This message is informational.

**Module**

FRXVSTO0

---

**FRD6033I**      **ALL ENTRIES IN RECOVERY LIST**  
**token, ARE BEING STOPPED**

**Explanation**

As the result of a REC STOP ALLENT command, this message is issued indicating nothing is left to recover, and the recovery for list token stops.

This message destination is the z/OS system console.

**System action**

Processing continues.

**Module**

FRXVST00

**FRD6034****Explanation**

Message is used in tracing buffers used for various processes and is issued only when tracing is active.

**System action**

Processing continues.

**Module**

FRXMSTR0, FRXMSTR1, FRXUORM0

**FRD6101E      UNABLE TO GET | FREE STORAGE  
FOR *component*****Explanation**

During a parse operation, a request to get or free storage could not be satisfied. In the message, *xxx* identifies what the storage request was for, and can be one of the following:

***component***

Indicates the target of the storage request.

**System action**

Recovery stops.

**User response**

Increase the region size in the IMS Database Recovery Facility procedure. For additional information, see the FRXJCLDF Procedure.

**Module**

FRXEPSS0, FRXEDRF0, FRXEPCCO

**FRD6102A      UNABLE TO OPEN FILE *fileid*.  
REASON: *vartxt*****Explanation**

During a parse operation, an error occurred while attempting the open of the indicated component.

In the message text:

***fileid***

Identifies the DDNAME for which the open failed.

***vartxt***

Identifies the reason for the open failure.

**System action**

Recovery stops.

**User response**

Correct the reason for the open failure and start the recovery again.

**Module**

FRXEPSS0, FRXEDRF0, FRXEPCCO

**FRD6103A      UNABLE TO ACCESS MEMBER  
*mbrname* FROM PROCLIB FILE.  
KEYWORD: *keyword*****Explanation**

During a parse operation, an error occurred while attempting to read the indicated member from the data set identified by the PROCLIB DD statement. In the message text:

***mbrname***

Identifies the member for which the read failed.

***keyword***

Identifies the keyword which specified the member to be read. *keyword* is the following:

**DRFMBR=**

Indicates that the error occurred while attempting to read the FRXDRFxx proclib member.

**System action**

Recovery stops.

**User response**

Determine the reason for the read failure and start the recovery again.

**Module**

FRXEDRF0

**FRD6105A      ERROR ON FILE *filename* I/O  
ERROR=*aaa*, *tt*, *ddname*, *op*, *error*,  
*block*, *accessmethod*****Explanation**

An I/O error occurred while reading the indicated file. In the message text:

***filename***

Identifies the DD statement that the data set was allocated to.

***aaa***

The device number that the error occurred on.

**tt**

The type of device.

**ddname**

The name of the DD statement that the data set was allocated to.

**op**

The I/O operation that resulted in the error.

**error**

The type of error as determined by the device. For example, if it is device dependent.

**block**

The block number in error.

**accessmethod**

The access method being used.

### System action

Recovery stops.

### User response

Correct the indicated problem and rerun the recovery.

### Module

FRXEPSS0

---

**FRD6110A**      **action HAS TOO MANY TARGET KEYWORDS.**

### Explanation

During a parse operation, two or more target keywords were supplied for the indicated action. Only one target keyword is allowed. In the message text:

#### action

Identifies the operation to be performed, and is one of the following:

- **ADD/REMOVE**

The target keywords associated with ADD/REMOVE are the following:

- DB
- DBDS
- AREA
- DBDSGRP
- CAGRP
- RECOVGRP

- **START**

The target keywords associated with START are the following:

- RCVDBDS
- VERIFY

### System action

Recovery stops.

### User response

Correct the problem with the target keyword and start the recovery again.

### Module

FRXEPSS0

---

**FRD6111A**      **action MISSING TARGET KEYWORDS.**

### Explanation

During a SYSIN parse operation, no target keywords were supplied for the indicated action. A target keyword is required. In the message text:

#### action

Identifies the operation to be performed, and is one of the following:

- **ADD/REMOVE**

The target keywords associated with ADD/REMOVE are the following:

- DB
- DBDS
- AREA
- DBDSGRP
- CAGRP
- RECOVGRP

- **START**

The target keywords associated with START are the following:

- RCVDBDS
- VERIFY

### System action

Recovery stops.

### User response

Correct the problem with the target keyword and start the recovery again.

### Module

FRXEPSS0

---

**FRD6112A**      **DBDNAME ENTRIES MUST BE PAIRED WITH *nametype* ENTRIES ON THE *action entity* STATEMENT**



**Explanation**

One of the following error situations was encountered during SYSIN parsing:

- On an ADD or REMOVE DBDS command, an occurrence of a dbdname parameter was not associated with a corresponding ddname.
- On an ADD or REMOVE AREA command, an occurrence of a dbdname parameter was not associated with a corresponding areaname.

In the message text:

**nametype**

DDNAME or AREANAME depending on full-function or fast path dbd type.

**action**

Identifies the operation to be performed, and is one of the following:

- ADD
- REMOVE

**entity**

One of the following:

- DBDS
- AREA

**System action**

Recovery stops.

**User response**

Correct the problem with the parameter association and start the recovery again.

**Module**

FRXEPSS0

---

**FRD6113A      ONLY ONE SET OF DATASET NAME DESCRIPTIONS IS ALLOWED IN PROCLIB MEMBER**

---

**Explanation**

More than one set of data set names are found in the proclib member.

**System action**

Processing continues.

**Module**

FRXEPCC0

---

**FRD6114A      EXCESSIVE NUMBER OF CONCATENATED STATEMENTS ENCOUNTERED IN *filename***

---

**Explanation**

The indicated file contained control statements which were concatenated. The number of concatenated statements was in excess of what is generally expected for this type of control statement. In the message text:

***filename***

The indicated file.

**System action**

Recovery stops with Condition Code 8.

**User response**

Statements are probably erroneously concatenated. Remove the concatenation character from those statements which do not require it, and start the recovery again.

**Module**

FRXEPSS0

---

**FRD6122A      THE FOLLOWING MUTUALLY EXCLUSIVE KEYWORDS WERE ENCOUNTERED IN FILE *fileid* KEYWORD: *keyword1* KEYWORD: *keyword2***

---

**Explanation**

During a parsing operation, mutually exclusive keywords were encountered. In the message text:

***keyword1***

Identifies one of the keywords in conflict.

***keyword2***

Identifies the other keyword or types of keywords in conflict.

The EXPDT and RETPD parameters might have been specified in the control statement, but they are mutually exclusive.

**System action**

Recovery stops.

**User response**

Correct the problem with the keywords, and start the recovery again.

## Module

FRXEPSS0, FRXEDRF0, FRXEPCC0

---

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <b>FRD6123E</b> | <b>LASTIC/LASTPITCA NOT<br/>ALLOWED WITH OUTPUT(ICR) OR<br/>OUTPUT(ICRCA)</b> |
|-----------------|-------------------------------------------------------------------------------|

---

### Explanation

RCVTYPE(LASTIC) or RCVTYPE(LASTPITCA) was specified along with OUTPUT(ICR) or OUTPUT(ICRCA). LASTIC and LASTPITCA recoveries are allowed only for OUTPUT(PRO | DUP | BOTH).

### System action

Recovery stops.

### User response

Correct the problem with the keywords, and start the recovery again.

## Module

FRXEPSS0

---

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>FRD6124A</b> | <b>INVALID DATA ENCOUNTERED.<br/>KEYWORD: <i>keyword</i> RSN=<i>rsn</i></b> |
|-----------------|-----------------------------------------------------------------------------|

---

### Explanation

The data associated with the indicated keyword was not valid.

In the message text:

#### ***keyword***

Identifies the keyword associated with the invalid data.

#### ***rsn***

Indicates the reason for which the data is invalid and can be one of the following:

#### **ALPHANUMERIC DATA REQUIRED**

The data contained a non-alphanumeric character.

#### **DATA IS INCORRECT LENGTH**

The data did not match the length specifications.

#### **EXCESSIVE NUMERIC VALUE**

The supplied value exceeded the acceptable limit for the keyword.

#### **FIRST PARAMETER IS INVALID**

The keyword required two parameters; the first of which was invalid.

## **IC KEYWORD MISSING WHILE OUTPUT=ICR SPECIFIED**

If Incremental Image Copy is specified as the OUTPUT option, then an IC keyword must be included on every ADD statement.

## **ICNOTIFY(DRF) AND ICNDX(Y) ARE MUTUALLY EXCLUSIVE**

The ICNOTIFY(DRF) and ICNDX(Y) parameters are mutually exclusive and cannot be specified at the same time.

## **INVALID FIRST CHARACTER FOR PDS MEMBER NAME**

The first character in the character string must be alphabetic.

## **INVALID POSITIONAL KEYWORD DETECTED**

An invalid keyword is detected in an invalid position.

## **INVALID TARGET VALUE**

The target name associated with the DB, DBDS, AREA, DBDSGRP, CAGRP or RECOVGRP is invalid. These target names must contain only alphanumeric or special characters allowable for PDS member names.

## **INVALID USE OF KEYWORD**

The indicated keyword is a valid keyword, but was not used in the proper context, or it was expected to be specified along with another keyword and that other keyword was not present. For example, WAITTIME and WAITMSG must be specified together.

## **IS ONLY SUPPORTED WITH OUTPUT OPTION PRO | DUP | BOTH**

The PITCA keyword was specified with an OUTPUT option other than PRO, DUP or BOTH. Only PRO, DUP or BOTH are allowed.

## **LAST PARAMETER IS INVALID**

The keyword required two parameters; the second of which was invalid.

## **NUMERIC VALUE TOO SMALL**

The supplied value was below the acceptable limit for the keyword.

## **REPORT STATEMENT MISSING WHILE IC, IB, OR PC SPECIFIED**

A report statement is required for auxiliary utilities invocation support.

## **NOT SUPPORTED WITH INCREMENTAL IC TYPE**

The OUTPUT(ICR) option will terminate if specified with PITER on the RECOVER START input control statement. This combination is not supported. An incremental image copy cannot be taken to any point-in-time. An incremental image copy can, however, be taken to a specified timestamp if a database

was previously brought offline or if an archive of OLDS occurred.

The OUTPUT(ICRCA) process will terminate if specified with RCVTIME. This combination is not supported. The ICRCA process determines the latest available timestamp that can be used to ensure that a batch image copy is generated.

#### **TWO CHARACTER VALUE IS REQUIRED**

The data supplied for the keyword must be a two character value.

#### **UNIQUE VOLUMES REQUIRED ON VOLSER AND VOLSER2**

If you are generating dual IC output on tape, you can optionally specify VOLSER and VOLSER2 to provide a list of tape volumes for each set. You cannot provide the same volumes in both lists, otherwise the image copies that are generated might overlay one another.

#### **UNIT IS NEEDED IF IC KEYWORD IS SPECIFIED**

If you specified the IC keyword to generate an output image copy with IMS Database Recovery Facility, UNIT is a required parameter.

#### **UNIT2 IS NEEDED IF VOLSER2 IS SPECIFIED**

If you specified VOLSER2 to create a secondary output image copy data set, UNIT2 is a required parameter.

#### **USESLBIC(Y) IS IGNORED WHEN USEDDBDS OR USEAREA IS SPECIFIED**

If USEDDBDS or USEAREA is specified, USESLBIC(Y) is overridden with USESLBIC(N).

#### **USESLBIC(Y) IS IGNORED WHEN USEUICLAST OR USEUICTIME IS SPECIFIED**

If USEUICLAST or USEUICTIME is specified, USESLBIC(Y) is overridden with USESLBIC(N).

#### **VOLSER IS REJECTED FROM IC KEYWORD**

If you are generating STACK image copies and you specified VOLSER or VOLSER2 on the individual ADD statements with the IC() keyword, you must move them to the UTILGBL statement. Specifying VOLSER and VOLSER2 on the UTILGBL statement ensures that the image copy stack is generated with a consistent tape volume list.

#### **VOLSER2 IS REJECTED IF VOLSER IS NOT SPECIFIED**

In UTILGBL or ADD statements, if VOLSER is not specified, VOLSER2 is rejected.

#### **XCFGROUP MISSING WHILE DBRCMD/STACMD SPECIFIED**

XCFGROUP name must be provided if you specified DBRCMD or STACMD.

#### **System action**

Recovery stops.

#### **User response**

Correct the format of the data and resubmit the recovery.

#### **Module**

FRXEPSS0, FRXEDRF0, FRXEPCC0

---

|                 |                                                           |
|-----------------|-----------------------------------------------------------|
| <b>FRD6125A</b> | <b>LAST RECORD FROM FILE <i>fileid</i> WAS INCOMPLETE</b> |
|-----------------|-----------------------------------------------------------|

---

#### **Explanation**

During a parsing operation, the last record in the file did not contain sufficient information such that it could be processed. In the message text:

#### ***fileid***

Identifies the DDNAME or member of a PDS which was being parsed.

#### **System action**

Recovery stops.

#### **User response**

Correct the problem with the last record and start the recovery again.

#### **Module**

FRXEPSS0, FRXEDRF0, FRXEPCC0

---

|                 |                                                                       |
|-----------------|-----------------------------------------------------------------------|
| <b>FRD6126A</b> | <b>NO ADDITIONAL STATEMENTS ARE ALLOWED AFTER THE START STATEMENT</b> |
|-----------------|-----------------------------------------------------------------------|

---

#### **Explanation**

A control statement was encountered after the START command. The START command must be the last control statement contained in the SYSIN file.

#### **System action**

Recovery stops with Condition Code 8.

#### **User response**

Ensure that the START command is the last control statement in the SYSIN file, and start the recovery again.

#### **Module**

FRXEPSS0

---

**FRD6127A      INVALID RCVTIME TIMESTAMP  
ENCOUNTERED. RC= return code  
RSN= reason****Explanation**

The time stamp associated with the RCVTIME keyword was invalid. In the message text:

**reason**

Indicates the cause of the error:

- Reasons associated with RC = 8:

**UNKNOWN FORMAT**

The time stamp was unrecognizable.

**NO END QUOTE**

The time stamp was not completely bracketed by quote marks.

- Reasons associated with RC = 12:

**CLEN**

The compressed input was too long.

**LENZN**

The length of the time stamp was zero or negative.

**YYL**

The length of (YY)YY was not 2 or 4.

**DDL**

The length of DDD was not 3.

**DDD**

The value for ddd was not between 001 and 366 inclusive.

**HHL**

The length of HH was not 2.

**HH**

The value for hh was greater than 23.

**MML**

The length of MM was not 2.

**MM**

The value for *mm* was greater than 59.

**SSL**

The length of SS was not 2.

**SS**

The value for ss was greater than 59.

**MSEPS**

There were multiple consecutive separators between values.

**PREL**

The length of THMIJU was greater than 6.

**XTRA**

There was excess input after the THMIJU value.

**OFFSET**

The time zone offset value was invalid.

**System action**

Recovery stops with Condition Code 8.

**User response**

Correct the problem with the time stamp and start the recovery again.

**Module**

FRXEPSS0

---

**FRD6128A      DUPLICATE KEYWORDS  
ENCOUNTERED****Explanation**

During a SYSIN parse operation, duplicate action keywords were specified on a single control statement. Only one action keyword is allowed on a single control statement. Action keywords include the following:

- ADD
- START
- REMOVE

**System action**

Recovery stops.

**User response**

Ensure that valid control statements are supplied for the indicated action and start the recovery again.

**Module**

FRXEPSS0

---

**FRD6129W      INVALID VALUE FOR KEYWORD:  
keyword VALUE USED: value****Explanation**

During a parse operation, a value was specified which was outside of the range of allowable values for the indicated keyword. A default value was substituted for the specified value. In the message text:

**keyword**

Identifies the keyword associated with the invalid data.

**value**

Indicate the value which was substituted for the invalid data.

**System action**

Processing continues.

**User response**

None required, however, it is suggested that the value be changed to one which is within the allowable range.

**Module**

FRXEDRF0

---

**FRD6130A NO RECORDS WERE INCLUDED IN FILE *fileid* FOR action**

**Explanation**

During a parse operation, no syntactically valid control statements were supplied for the indicated action. Such control statements are required.

In the message text:

***fileid***

Identifies the FILE which contains the records being parsed.

***action***

Identifies the operation to be performed, and is one of the following:

- ADD
- START

**System action**

Recovery stops.

**User response**

Ensure that valid control statements are supplied for the indicated action and start the recovery again.

**Module**

Ensure that valid control statements are supplied for the indicated action and start the recovery again.

---

**FRD6131I DEFAULT RCVDBDS PROCESSING INVOKED.**

**Explanation**

A control keyword indicating the type of recovery was not included in the SYSIN file. When no such keyword is included, the recovery type defaults to RCVDBDS (recover database data set).

**System action**

This message is informational. Recovery continues.

**User response**

None. This message is informational.

**Module**

FRXEPSS0

---

**FRD6133I DRF STARTUP PARAMETER MEMBER *parmmb* SELECTED**

**Explanation**

The indicated parameter member was selected from the file identified by the PROCLIB DD statement. The suffix contained in the parameter member name was specified in the DRFMBR parameter in the execution JCL for the IMS Database Recovery Facility.

**System action**

This message is informational. Recovery continues.

**User response**

None. This message is informational.

**Module**

FRXEDRF0

---

**FRD6134I NO DRF STARTUP PARAMETER MEMBER WAS REQUESTED**

**Explanation**

The DRFMBR parameter was not specified in the execution JCL for the IMS Database Recovery Facility. Consequently, no the IMS Database Recovery Facility Startup Parameter Member was identified nor evaluated. The IMS Database Recovery Facility continues using default values for the startup parameters.

**System action**

This message is informational. Recovery continues.

**User response**

None. This message is informational.

**Module**

FRXEDRF0

---

**FRD6135E ERROR PROCESSING DBDLIB. RSN=*reason***

### Explanation

An error occurred while attempting to read a dbd member from the data set identified by the DBDLIB DD statement. In the message text:

#### *reason*

Indicates the cause of the error:

#### **OPEN ERROR ON DBDLIB**

The IMS Database Recovery Facility was unable to open the data set associated with the DBDLIB DD statement.

#### **UNABLE TO OBTAIN DBD *dbdname***

The IMS Database Recovery Facility was unable to get the named dbd from the DBDLIB.

#### **CLOSE ERROR ON DBDLIB**

The IMS Database Recovery Facility was unable to close the data set associated with the DBDLIB DD statement

### System action

This message is informational. Recovery continues.

### User response

None required, however, for performance reasons, it is recommended that the problem with the DBDLIB resolved.

### Module

FRXMDBDO

---

**FRD6136I      DEFAULT KEYLENGTH OF 255  
WILL BE USED FOR DBD *dbdname***

---

### Explanation

The default value of 255 will be used for the key length of the root segment for the indicated database. This message is associated with message FRD6135I.

### System action

This message is informational. Recovery continues.

### User response

None required, however, the use of the default key length value might have performance implications. It is suggested that the reason for the inability to obtain the DBD be resolved so that an optimal key length value might be used during subsequent executions of the IMS Database Recovery Facility.

### Module

FRXMDBDO

---

**FRD6137I****ERROR PROCESSING****PSBLIB.RSN=*reason***

---

### Explanation

An error occurred while attempting to read a psb member from the data set identified by the PSBLIB DD statement. In the message text, *reason* indicates the cause of the error, which can be one of the following:

- OPEN ERROR ON PSBLIB The IMS Database Recovery Facility was unable to open the data set associated with the PSBLIB DD statement.
- UNABLE TO OBTAIN PSB *psbname* The IMS Database Recovery Facility was unable to get the named psb from the PSBLIB.
- CLOSE ERROR ON PSBLIB The IMS Database Recovery Facility was unable to close the data set associated with the PSBLIB DD statement.

### System action

This message is informational. Recovery continues.

### User response

None required, however, for performance reasons, it is recommended that the problem be resolved regarding the PSBLIB.

### Module

FRXMPSBO

---

**FRD6138E      ERROR PROCESSING DDEFPS.  
RSN=*reason***

---

### Explanation

In the message text, the reason for the failure can be one of the following:

- OPEN ERROR The IMS Database Recovery Facility was unable to open the data set associated with the DDEFPS DD statement.
- UNABLE TO OBTAIN MEMBER *name* The IMS Database Recovery Facility was unable to get the named member for DELETE/DEFINE.
- MEMBER *name* OVER 16K LIMIT The member exceeded the 16K file size limit which is enforced by IMS Database Recovery Facility.
- INVALID FORMAT The data set was not defined with LRECL=80 and fixed/blocked format.
- CLOSE ERROR The IMS Database Recovery Facility was unable to close the data set successfully.

### System action

Recovery stops for the database.

### User response

Correct the problem, and rerun the recovery.

### Module

FRXMPSB0

---

|                 |                                                                  |
|-----------------|------------------------------------------------------------------|
| <b>FRD6140A</b> | <b>PARSING OF RECORDS FROM<br/><i>fileid</i> ENDED IN ERROR.</b> |
|-----------------|------------------------------------------------------------------|

---

### Explanation

One or more control statements in the indicated file were found in error. This message is accompanied by additional messages which describe the error condition.

### System action

Recovery stops with Condition Code 8.

### User response

Correct the problem which is described by the additional messages, and start the recovery again.

### Module

FRXEPSS0, FRXEDRF0, FRXEPCC0

---

|                 |                                                                 |
|-----------------|-----------------------------------------------------------------|
| <b>FRD6141I</b> | <b>PARSING OF RECORDS FROM<br/><i>fileid</i> WAS SUCCESSFUL</b> |
|-----------------|-----------------------------------------------------------------|

---

### Explanation

All of the control statements in the indicated file were syntactically correct and successfully parsed.

### System action

This message is informational. Recovery continues.

### User response

None. This message is informational.

### Module

FRXEPSS0, FRXEDRF0

---

|                 |                                                                                                    |
|-----------------|----------------------------------------------------------------------------------------------------|
| <b>FRD6142E</b> | <b>PARSE INTERNAL ERROR. <i>vartxt</i><br/>GRAMMAR: <i>blockname</i> RC=<i>return<br/>code</i></b> |
|-----------------|----------------------------------------------------------------------------------------------------|

---

### Explanation

During a parsing operation, an error occurred processing the indicated internal grammar control block. In the message text:

#### *vartxt*

Indicates the type of parse internal error:

**GRAMMAR: *blockname* RC=*return code***

#### *blockame*

Indicates an internal component.

#### **RC**

Indicates the *return code* associated with the failure.

**FAILED TIME CONERSION ROUTINE. RC=*return code***

An internal time conversion routine failed with the indicated *return code*.

**INVALID ADD OR REPLACE TARGET KEYWORD**

An invalid keyword associated with DB, DBDS, AREA, DBDSGRP, CAGRP or RECOVGRP was returned after parsing of the ADD or REPLACE commands.

### System action

Recovery stops.

### User response

These errors are all internal. If the *vartxt* associated with the message is GRAMMAR, see the BPEPARSE topic for the meaning of the return code. Contact IBM Software Support and provide the block name and return code information, if applicable.

### Module

FRXEPSS0, FRXEDRF0

---

|                 |                                                                                                                              |
|-----------------|------------------------------------------------------------------------------------------------------------------------------|
| <b>FRD6143E</b> | <b>NAME CONSTRUCT NOT FOUND,<br/>DBDSN(<i>nnn</i>)/DBDSL(<i>mmm</i>)<br/>SPECIFIED BUT NO MATCHING<br/>DBDSN/DBDSL FOUND</b> |
|-----------------|------------------------------------------------------------------------------------------------------------------------------|

---

### Explanation

The data set name keyword, DBDSN/DBDSL, is specified on an ADD command but the corresponding DBDSN environmental control statement is not supplied. *nnn* is the string that associates the ADD command to the DBDSN environmental control statement.

### System action

The ADD command is rejected.

### User response

Provide a matching DBDSN environmental control statement.

### Module

FRXEDSG0

---

**FRD6144E      ERROR PROCESSING IMS reason =  
                 reasons**

---

### Explanation

An error is found during IMS process.

#### reasons

The reason can be one of the following:

- OPEN ERROR ON IMS
- UNABLE TO OBTAIN DBD
- CLOSE ERROR ON IMS

### System programmer response

Check that the IMS DD is correct in your JCL PROC.

### Module

FRXMPSB0

---

**FRD6145E      SYMBOLIC OFFSET *symbolic* IS  
                 INVALID FOR RCVTIME, reason**

---

### Explanation

Symbolic time zone is not valid in the RCVTIME statement.

#### *symbolic*

The symbolic offset value.

#### reason

The reason can be one of the following:

- The symbolic time zone is not defined to DBRC
- The DBRC symbolic time zone table is not available

### System action

Recovery stops.

### Module

FRXEPSS0

---

**FRD6146E      DD DOES NOT MATCH ALLOC. IN  
                 MASTER AS, DD=*ddname***

---

### Explanation

Integrated Auxiliary Utility processing was requested, but the data sets allocated to the same DDs in the master and subordinate address spaces do not match.

*ddname* may be one of the following:

- IMS
- STEPLIB
- JOBLIB

---

**FRD7103A      REPORT PROCESSING FAILED:  
                 vartxt**

---

### Explanation

Generation of the IMS Database Recovery Facility reports failed for the indicated internal reason. In the message text:

#### *vartxt*

Identifies the reason for the report generation failure:

- INVALID REPORT INPUT CONTROL BLOCK
- REPORT INVOKED FOR UNKNOWN REASON
- NULL DATA SET INPUT CONTROL BLOCK
- NULL UOW INPUT CONTROL BLOCK
- NULL SUMMARY REPORT INPUT CONTROL BLOCK
- FAILED TIME CONVERSION ROUTINE RC=xx
- FAILED LOAD OF DATE CONVERSION MODULE
- FAILURE CREATING SUMMARY REPORT
- FAILURE CREATING DATA SET USAGE REPORT
- FAILURE CREATING UOW REPORT

### System action

Processing continues, however, the generated report might be incomplete.

### User response

All of the above error situations are not user correctable. Contact the IBM Service Center and provide the return code information, if applicable.

### Module

FRXGRPT0, FRXGRDS0, FRXGRDM0, FRXGUOW0

---

**FRD7201I      <control statement text>**

---

### Explanation

This message displays the image of a statement processed from the file associated with the SYSIN statement.



### System action

This message is informational. Recovery continues.

### User response

None. This message is informational.

### Module

FRXGRPT0

---

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <b>FRD7202A</b> | <b>PARSING ERRORS FOUND IN<br/>PRIOR STATEMENT</b> |
|-----------------|----------------------------------------------------|

---

### Explanation

The immediately preceding FRD7201E message shows a SYSIN statement in which some invalid syntax was detected. This message is accompanied by additional messages which are contained in the file identified by the SYSPRINT DD statement.

### System action

Recovery has already stopped.

### User response

See the SYSPRINT DD statement for additional messages describing the problem.

### Module

FRXGRPT0

---

|                 |                                                                 |
|-----------------|-----------------------------------------------------------------|
| <b>FRD9000I</b> | <b><i>utiltype</i> IS NOW PROCESSING<br/><i>name1 name2</i></b> |
|-----------------|-----------------------------------------------------------------|

---

### Explanation

IMS Database Recovery Facility has invoked the utility *utiltype*. The value for *utiltype* can be one of the following utilities:

- IB - Index Builder
- PR - DFSPRECO
- IC - High Performance Image Copy
- PC - High Performance Pointer Checker
- DP - DEDB Pointer Checker
- ITKB - IMS Tools Knowledge Base

If the utility is IB, IC, PC, or DP, then *name1* identifies the database name and *name2* identifies the database DD name. If the utility is ITKB, then *name1* will identify ITKBSRVR and *name2* will be the ITKB server name.

### System action

Processing continues.

### Module

FRXIOCN0, FRXIVCN0, FRXIOSM0, FRXIVSM0, FRXVSTA0

---

|                 |                                                                         |
|-----------------|-------------------------------------------------------------------------|
| <b>FRD9001I</b> | <b><i>utiltype</i> SUCCESSFULLY<br/>PROCESSED <i>dbdname ddname</i></b> |
|-----------------|-------------------------------------------------------------------------|

---

### Explanation

IMS Database Recovery Facility invoked the utility *utiltype* and the utility has successfully completed processing the database *dbdname*. The value for *utiltype* values can be one of the following utilities:

- IB - Index Builder
- PR - DFSPRECO
- IC - High Performance Image Copy
- PC - High Performance Pointer Checker
- DP - DEDB Pointer Checker

If the utility processes individual database data sets, *ddname* identifies the database data set that was successfully processed.

### System action

Processing continues.

### Module

FRXIOCN0, FRXIVCN0, FRXIOSM0, FRXIVSM0

---

|                 |                                                                                |
|-----------------|--------------------------------------------------------------------------------|
| <b>FRD9002A</b> | <b><i>utiltype</i> ENCOUNTERED RC=<i>rtn</i><br/>RSN=<i>rsn</i> PROCESSING</b> |
|-----------------|--------------------------------------------------------------------------------|

---

### Explanation

A utility *utiltype* invoked by IMS Database Recovery Facility produced a return code *rtn* and reason code *rsn* while processing the database. The value for *utiltype* values can be one of the following utilities:

- IB - Index Builder
- PR - DFSPRECO
- IC - High Performance Image Copy
- PC - High Performance Pointer Checker
- DP - DEDB Pointer Checker
- FS - The Build Index function of FPA

If the utility processes individual database data sets, *ddname* identifies the database data set that is being processed.

### System action

Processing stops if you have selected ERROR(STOP). Processing continues if you selected ERROR(CONT). If you specified ERROR(CONT) and the error is for a

database data set and subsequent processing requires access to that database data set, the subsequent processing is skipped.

**User response**

Determine the cause of the error. See the documentation related to the utility identified by *utiltype* for an explanation of the return and reason codes. When you have corrected the error, rerun the recovery job.

**Module**

FRXIOCN0, FRXIVCN0, FRXIOSM0, FRXIVSM0

---

|                 |                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------|
| <b>FRD9003A</b> | <b><i>utiltype</i> DYNAMIC ALLOCATION<br/><i>rtn</i> REASON CODE <i>rsn</i> FOR <i>dsname</i></b> |
|-----------------|---------------------------------------------------------------------------------------------------|

---

**Explanation**

IMS Database Recovery Facility attempted to dynamically allocate a data set identified by *dsname* and detected a non-zero return code *rtn* and reason code *rsn*. The data set was invoked on behalf of *utiltype*. The value for *utiltype* values can be one of the following utilities:

- IB - Index Builder
- PR - DFSPRECO
- DP - DEDB Pointer Checker
- IC - High Performance Image Copy
- PC - High Performance Pointer Checker
- WT - WTO capture services
- FS - Build Index function of FPA

**System action**

Processing stops if you have selected ERROR(STOP). Processing continues if you selected ERROR(CONT). If the DRFUNIT=genericunit keyword of the REPORT control statement was set to an invalid generic unit, then messages FRD9003A and FRD4100I will be written and followed by abend U384-02C. The return/reason code fields in message FRD9003A will be 4/021Cx.

If IMS Database Recovery Facility attempted to allocate a duplicate DSN on behalf of Integrated Auxiliary Utility processing, then messages FRD9003A and FRD4100I will be written and followed by ABENDU0384-02C. The return/reason code fields in message FRD9003A will be will be /970Cx if the duplicate DSN is not allocated to another IMS Database Recovery Facility job or 4/210x if the duplicate DSN and is allocated to another IMS Database Recovery Facility job. It is possible to avoid

this error by setting RPTRET=N on the REPORT control card.

**User response**

Determine the cause of the error. For an explanation of the dynamic allocation return and reason codes, see the *z/OS MVS Programming: Authorized Assembler Services Guide*. When you have corrected the error, rerun the recovery job.

**Module**

FRXIOCN0, FRXIVCN0, FRXIOSM0, FRXIVSM0, FRXVSTA0

---

|                 |                                                              |
|-----------------|--------------------------------------------------------------|
| <b>FRD9004I</b> | <b>SECONDARY INDEX NAME IS<br/>INVALID, <i>indexname</i></b> |
|-----------------|--------------------------------------------------------------|

---

**Explanation**

The BLD\_SECONDARY(selected indexes) parameter contains an invalid index name. The name specified as the selected index does not belong to the database.

***indexname***  
The name of the invalid index.

**User response**

None. This is an informational message and the invalid index is ignored.

**Module**

FRXVSTA2

---

|                 |                                  |
|-----------------|----------------------------------|
| <b>FRD9005E</b> | <b><i>utiltype</i> NOT FOUND</b> |
|-----------------|----------------------------------|

---

**Explanation**

The utility *utiltype* was not found when trying to load the utility.

**System action**

The recovery job stops.

**User response**

Determine the cause of the error. Once you have corrected the error, rerun the recovery job.

**Module**

FRXMINI0, FRXMINI1, FRXMINI2

---

|                 |                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------|
| <b>FRD9006W</b> | <b>INDEX DATA SETS WILL NOT BE<br/>IMAGE COPIED AFTER A REBUILD<br/>IF STACK SPECIFIED</b> |
|-----------------|--------------------------------------------------------------------------------------------|

---

### Explanation

IMS Database Recovery Facility is invoked to generate image copies of the recovered database that is stacked on tape. It is also invoked to perform an index rebuild. Following the index rebuild, no image copy will be generated for the index data set if stacking is being performed.

### User response

To generate an image copy of your index data sets, run a separate image copy utility after recovery. Alternately, IMS Database Recovery Facility allows Index Builder to have image copies generated if STACK is not specified.

### Module

FRXVASD0

---

**FRD9007E**      *vartxt* **ERROR RC=rc RSN=rsn**

### Explanation

A failure was encountered while processing ITKB.

*vartxt* can be any of the following:

- PROGRAM LOAD - The ITKB client code failed to load.
- PROGRAM UNLOAD - The ITKB client code failed to unload.
- INIT - ITKB failed on initialization.
- TERM - ITKB failed on termination.

### System action

If ITKB failed to load or to be initialized, then processing stops. During termination, IMS Database Recovery Facility processing is allowed to continue.

### User response

Determine the cause of the error. Check to ensure that the correct ITKB load library is accessible to your job. After you correct the error, rerun the recovery job.

### Module

FRXMSTRO, FRXUAPI0, FRXVSTA0

---

**FRD9009E**      **Library Integrity Utility Module Found In STEPLIB Version Too Low.**

### Explanation

LIU@GOPT(xxxx) was coded and the Library Integrity Utility module was loaded, but its version was lower than the minimum version required.

### System action

### User response

Change the Library Integrity Utility load library the STEPLIB DD of all JCL and procedures to refer to the correct version.

### Module

---

**FRD9010I**      **HPIC ERROR ON THE PRIMARY OR SECONDARY IMAGE COPY: SEE THE FABJ MESSAGES AND THE ICEPRINT REPORT.**

### Explanation

IMS HP Image Copy has issued the final RC=04 RSN=00. IMS HP Image Copy was able to create one of either the primary or secondary image copy, but not both.

### System action

Processing continues.

### User response

Search for the messages prefixed by FABJ in the SYSOUT data set, examine the ICEPRINT REPORT, and correct the problem as indicated.

### Module

FRXVRCV0

---

**FRD9011E**      **HPIC ERROR ON THE DBRC NOTIFY.IC COMMAND. SEE THE FABJ MESSAGES.**

### Explanation

IMS HP Image Copy has issued the final RC=08 RSN=03. IMS HP Image Copy was unable to register the image copy that it created to DBRC.

### System action

Processing continues.

### User response

Search for the messages prefixed by FABJ in the SYSOUT data set and correct the problem as indicated.

### Module

FRXVRCV0

---

**FRD9020I      HALDB(BOTH) RECOMMENDED TO  
ENSURE DATABASE INTEGRITY  
AFTER TIMESTAMP RECOVERY**

---

**Explanation**

HALDB(BOTH) is suggested to ensure the ILE and primary index are both rebuilt so that all sets being recovered are rebuilt to a consistent point for Pitr.TSR.LASTIC or LASTPITCA recovery.

**System action**

Processing continues.

**User response**

None. This message is informational.

---

**FRD9012I      DRF ITKB REPORT INTERFACE  
INITIALIZATION COMPLETE**

---

**Explanation**

The initialization of IMS Tools KB is complete for the entire recovery process, including all of the reports that are being created.

**System action**

Processing continues.

**User response**

None. This message is informational.

**Module**

FRXITKB0

---

**FRD9013E      ITKB REPORT PROCESSING  
ERROR OCCURRED, REASON=  
reason**

---

**Explanation**

IMS Tools KB failed due to one of the following reasons:

**reason**

- INIT NOT COMPLETED
- REPORT ALREADY OPEN
- RECONID NOT DEFINED
- REPORT NOT DEFINED
- REPORT NOT OPEN

**System action**

The IMS Tools KB report is not generated.

**User response**

Correct the error listed in the reason code and continue.

**Module**

FRXITKB0

---

**FRD9014E      ITKB REPORT PROCESSING  
ERROR OCCURRED, TYPE= type**

---

**Explanation**

The IMS Tools KB process cannot be continued because of an error, indicated by *type*.

**type**

- DRF report
- WTO
- SYSPRINT

**System action**

The IMS Tools KB is not generated.

**User response**

Review the reason code listed in message FRD9013E, correct the error, and continue.

**Module**

FRXITKB0

---

**FRD9021I      PRIMARY | SECONDARY INDEXES  
WERE NOT REQUESTED TO BE  
REBUILT FOR A TIME STAMP  
RECOVERY. DBD = dbdname**

---

**Explanation**

For the HIDAM database in a time stamp recovery the IB(BLD\_PRIMARY) keyword is required.

For HALDB or full function databases in a time stamp recovery, the IB(BLD\_SECONDARY) keyword is required.

**System action**

None.

**User response**

Specify BLD\_PRIMARY or BLD\_SECONDARY and rebuild the indexes.

**Module**

FRXVSTA0

---

**FRD9400I      DATABASE *dbname* START  
COMMAND ISSUED [ON *imsid*]**

---

**Explanation**

A START DB command has been issued for *dbname* after recovery completes. If the LOCAL(*imsid*) was specified on the STACMD command, the *imsid* indicates the name of the IMS which is processing the START DB command.

**System action**

None.

**User response**

See the IMS console for the status of the database.

**Module**

FRXVRCV0

---

**FRD9401I      AREA *areaname* START COMMAND  
ISSUED [ON *imsid*]**

---

**Explanation**

A START AREA command has been issued for *areaname* after recovery completes. If the LOCAL(*imsid*) was specified on the STACMD statement, the *imsid* indicates the name of the IMS which is processing the START AREA command.

**System action**

None.

**User response**

See the IMS console for the status of the area.

**Module**

FRXVRCV0

---

**FRD9402I      DATABASE *dbname* DBR  
COMMAND ISSUED [ON *imsid*]**

---

**Explanation**

A DBR DB command has been issued for *dbname* at the beginning of recovery. If the LOCAL(*imsid*) was specified on the DBRCMD statement, the *imsid* indicates the name of the IMS which is processing the DBR DB command.

**System action**

None.

**User response**

See the IMS console for the status of the database.

**Module**

FRXVSTA0

---

**FRD9403I      AREA *areaname* DBR COMMAND  
ISSUED [ON *imsid*]**

---

**Explanation**

A DBR AREA command has been issued for *areaname* at the beginning of recovery. If the LOCAL(*imsid*) was specified on the DBRCMD statement, the *imsid* indicates the name of the IMS that is processing the DBR AREA command.

**System action**

None.

**User response**

See the IMS console for the status of the area.

**Module**

FRXVSTA0

---

**FRD9404E      IMS COMMAND ISSUED BUT NO  
CONNECTION EXISTS: *command*[,  
*imsid*]**

---

**Explanation**

A DBR AREA | DB or START AREA | DB command *command* has been issued but no TOI/XCF connection exists. If the LOCAL(*imsid*) was specified on the DBRCMD | STACMD statement, the *imsid* indicates the name of the IMS which was to process the IMS command.

**System action**

The IMS command is ignored.

**User response**

Determine why no TOI/XCF connection exists, and correct the condition. If recovery failed because the command was not processed by IMS, start recovery after correcting the problem.

**Module**

FRXVSTA0, FRXVRCV0

---

**FRD9405I      DFSOL*cnn* ARCHIVE STATUS IS  
*status* FOR *imsid* ON *sysid***

---

### Explanation

Status for an online log data set (OLDS) to be used in recovery processing is displayed for informational purposes. The *DFSOLcnn* represents the OLDS used by the IMS online system. The *cnn* represents the suffix of the OLDS' where *c*=P (primary) or S (secondary) and *nn*=00-99. For the subject OLDS, *status* indicates the current archive status, *imsid* indicates the name of the IMS and *sysid* indicates the system id.

### System action

None.

### User response

Confirm that archive processing is scheduled to run as quickly as possible for the OLDS data set.

### Module

FRXVSTA0

---

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| <b>FRD9406E</b> | <b>DFSOLcnn ARCHIVE IS NOT COMPLETE FOR <i>imsid</i> ON <i>sysid</i></b> |
|-----------------|--------------------------------------------------------------------------|

---

### Explanation

Archive processing has not been completed for an online log data set (OLDS) that has been selected for use in recovery processing. The *DFSOLcnn* represents the OLDS used by the IMS online system. The *cnn* represents the suffix of the OLDS' where *c*=P (primary) or S (secondary) and *nn*=00-99. For the subject OLDS, *imsid* indicates the name of the IMS, and *sysid* indicates the system id.

### System action

If an OLDS is required for recovery but has not yet been archived, recovery is not started and an error is issued.

### User response

If an OLDS contains log records that are required for recovery, ensure that the OLDS has been archived.

### Module

FRXVSTA0

---

|                 |                                                                   |
|-----------------|-------------------------------------------------------------------|
| <b>FRD9407E</b> | <b>NO ACTIVE MEMBER FOUND IN <i>xcfgroup</i> FOR <i>imsid</i></b> |
|-----------------|-------------------------------------------------------------------|

---

### Explanation

A LOCAL keyword specified an IMS on the DBRCMD or STACMD command, but no active member entry was found in the specified XCF group. The *xcfgroup*

indicates the XCF group name passed in the DRF XCFGROUP() statement. The *imsid* indicates the IMS passed in the DRF LOCAL() keyword for the DBRCMD or STACMD command.

### System action

The IMS command is not processed.

### User response

Determine whether the Tools Online System Interface server program is running on the IMS system and whether the XCF group is correctly specified.

### Module

FRXVSTA0, FRXVRCV0

---

|                 |                                                                       |
|-----------------|-----------------------------------------------------------------------|
| <b>FRD9408I</b> | <b>COMMAND NOT PROCESSED FOR LOCAL IMS SPECIFIED: DBRCMD   STACMD</b> |
|-----------------|-----------------------------------------------------------------------|

---

### Explanation

This is an informational message indicating that a DBRCMD AREA | DB or STACMD AREA | DB command was not processed.

### System action

The IMS command is not processed.

### User response

The identification of DBRCMD or STACMD reveals which LOCAL IMS keyword received the error. Determine the cause of the failure, and resubmit the command.

### Module

FRXVSTA0, FRXVRCV0

---

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>FRD9409I</b> | <b>ERROR DETECTING OLDS ARCHIVE STATUS FOR <i>imsid</i> ON <i>sysid</i></b> |
|-----------------|-----------------------------------------------------------------------------|

---

### Explanation

An error was encountered detecting OLDS archive status for subject IMS on subject system. The *imsid* indicates the name of the IMS and *sysid* indicates the system id.

### System action

Archive status is not collected, and processing continues.

### User response

Determine the status of OLDS for the subject IMS system, and ensure that archive processing is able to proceed.

### Module

FRXVSTA0

---

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <b>FRD9410I</b> | <b>ERROR VALIDATING ARCHIVE<br/>COMPLETE FOR <i>imsid</i> ON <i>sysid</i></b> |
|-----------------|-------------------------------------------------------------------------------|

---

### Explanation

An error was encountered validating that the OLDS archive was completed for the subject IMS on the subject system. The *imsid* indicates the name of the IMS and *sysid* indicates the system id.

### System action

Archive processing is not validated, and processing continues.

### User response

Ensure that every OLDS has been archived for the subject IMS on the subject system.

### Module

FRXVSTA0

---

|                 |                                                                             |
|-----------------|-----------------------------------------------------------------------------|
| <b>FRD9411I</b> | <b>COMMAND SERVICES<br/>COMPONENT MISSING: <i>cmd</i> NOT<br/>PROCESSED</b> |
|-----------------|-----------------------------------------------------------------------------|

---

### Explanation

The DBRCMD | STACMD statement has been specified but not processed due to missing components for command services. The variable *cmd* indicates the DBR or START command was not processed.

### System action

Command processing is bypassed.

### User response

Ensure that the FMID: H2B7110 has been SMP/E installed into the CSI and the proper load library is specified in the concatenation for STEPLIB DD. If the FMID is installed into a separate CSI, then ensure that the load library is included in the concatenation for STEPLIB DD.

### Module

FRXVSTA0, FRCVRCV0

---

|                 |                                                                          |
|-----------------|--------------------------------------------------------------------------|
| <b>FRD9412I</b> | <b>IMS COMMAND PROCESSING<br/>FAILED <i>command</i>, <i>resource</i></b> |
|-----------------|--------------------------------------------------------------------------|

---

### Explanation

An error was encountered while processing a command for resource specified. Where *command* is the failed command, and *resource* is the database or area requested for command processing.

### System action

Command processing continues.

### User response

Review the job log for IMS Database Recovery Facility and IMS to determine the cause of the error. Correct the error, and resubmit the recovery job.

### Module

FRXVSTA0, FRCVRCV0





---

## Chapter 22. Abend codes

This reference section provides detailed reason code information for the service and logic abend codes issued by IMS Database Recovery Facility. The explanations provided in this reference can help you diagnose, troubleshoot, and solve IMS Database Recovery Facility problems.

IMS Database Recovery Facility abnormally ends when it encounters conditions which prevent the continuation of normal operations. There are two types of abends issued by IMS Database Recovery Facility:

- ABENDU0384 - a service invoked by IMS Database Recovery Facility returned an unexpected result
- ABENDU0385 - IMS Database Recovery Facility detected a logic error

### Topics:

- [“Reason codes for service error abend ABENDU0384” on page 263](#)
- [“Reason codes for logic error abend ABENDU0385” on page 269](#)

---

### Reason codes for service error abend ABENDU0384

IMS Database Recovery Facility detected a logic error. Reason codes further identify the type of error:

|                    |                                             |                                                                                                                                                                                                                                                                                                                                           |
|--------------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>001</b>         | <b>Startup failure</b>                      | <b>Explanation</b>                                                                                                                                                                                                                                                                                                                        |
| <b>Explanation</b> |                                             | The IMS Database Recovery Facility is unable to load modules required for recovery. Register 14 contains the return code from the load service.                                                                                                                                                                                           |
| <b>Module</b>      |                                             | FRXIVCNO, FRXMINIO, FRXMINI1, FRXPPIBO, FRXPPRBO, FRXPPSB0, FRXSINIO, FRXSINI1                                                                                                                                                                                                                                                            |
| <b>002</b>         | <b>Obtain variable length storage error</b> | <b>Explanation</b>                                                                                                                                                                                                                                                                                                                        |
| <b>Explanation</b> |                                             | IMS Database Recovery Facility is unable to obtain fixed length storage. Register 14 contains the return code from the load service. For information about how to set a new region size in the JCL for this address space, refer to the section "Region size considerations". After a new region size is set, run the recovery job again. |
| <b>Module</b>      |                                             | FRXCAMG0, FRXCBDM0, FRXICAI0, FRXICLIO, FRXICTLO, FRXLBLD0, FRXPDIRO, FRXPDSR0, FRXIDYNO, FRXIIOMO, FRXIOCN0, FRXIOSM0, FRXISMS0, FRXIVCNO, FRXIVSM0, FRXLMRG0, FRXPPIBO, FRXPPRBO, FRXPPSB0, FRXPSDR0, FRXQBUFO, FRXRCTLO, FRXSDR00, FRXTAU00, FRXUORM0, FRXVADD0, FRXVAUTO, FRXVDISO, FRXVRCV0, FRXVSTAO, FRXVVER0, FRXWSPL0            |
| <b>003</b>         | <b>Load error</b>                           | <b>Explanation</b>                                                                                                                                                                                                                                                                                                                        |
|                    |                                             | IMS Database Recovery Facility is unable to create an data set restore dispatchable thread if the module is FRXICTLO. IMS Database Recovery Facility is unable to gain access to data set restore dispatchable thread blocks if the module is FRXIRTH0. Register 14 contains the return code from the thread service.                     |
| <b>004</b>         | <b>Obtain fixed length storage</b>          | <b>Explanation</b>                                                                                                                                                                                                                                                                                                                        |
|                    |                                             | IMS Database Recovery Facility is unable to obtain fixed length storage. Register 14 contains the return code from the load service. For information about how to set a new region size in the JCL for this address space, refer to the section "Region size considerations". After a new region size is set, run the recovery job again. |
|                    |                                             | <b>Module</b>                                                                                                                                                                                                                                                                                                                             |
|                    |                                             | FRXICTLO, FRXUORM0                                                                                                                                                                                                                                                                                                                        |
| <b>005</b>         | <b>Create thread error</b>                  | <b>Explanation</b>                                                                                                                                                                                                                                                                                                                        |
|                    |                                             | IMS Database Recovery Facility is unable to create an data set restore dispatchable thread if the module is FRXICTLO. IMS Database Recovery Facility is unable to gain access to data set restore dispatchable thread blocks if the module is FRXIRTH0. Register 14 contains the return code from the thread service.                     |

**Module**

FRXICTLO, FRXIRTHO

**006 Address space create error****Explanation**

IMS Database Recovery Facility experienced an error during address space create. Register 14 contains the return code from the address space create service.

**Module**

FRXMINIO, FRXMINI1

**007 Enqueue AWE error****Explanation**

IMS Database Recovery Facility is unable to enqueue an asynchronous work element to a dispatchable thread. Register 14 contains the return code from the enqueue service.

**Module**

FRXCAMGO, FRXCARD0, FRXDRQIO, FRXEDRFO, FRXEDSG0, FRXEPSS0, FRXGRPT0, FRXHBUFO, FRXICAI0, FRXICTLO, FRXIDYNO, FRXIOCNO, FRXIVCNO, FRXIOSMO, FRXIVSMO, FRXIRTHO, FRXLACR1, FRXLSE35, FRXLMRG0, FRXLUTRO, FRXMSTP0, FRXMSTRO, FRXMSTR1, FRXPDIRO, FRXPDISO, FRXPDSR0, FRXPDSS0, FRXPSDR0, FRXPSS0, FRXQBUFO, FRXRBUFO, FRXRCTLO, FRXRDTHO, FRXUORM0, FRXVADD0, FRXVCMD0, FRXVDISO, FRXVRCV0, FRXVREM0, FRXVSTAO, FRXVSTO0, FRXVVER0, FRXYALLO

**008 Attach TCB error****Explanation**

The IMS Database Recovery Facility data set restore controller is unable to attach a TCB (thread control block) for a data set restore instance. Register 14 contains the return code from the attach TCB service.

**Module**

FRXICTLO

**009 Message error****Explanation**

IMS Database Recovery Facility is unable to issue a message. Register 14 contains the return code from the message service.

**Module**

FRXCAMGO, FRXIDYNO, FRXIIOMO, FRXIOCNO, FRXIOSMO, FRXIRTHO, FRXISMSO, FRXIVCNO, FRXIVSMO, FRXMSTRO, FRXMSTR1, FRXPDISO, FRXPDSS0, FRXPSS0, FRXRBUFO, FRXRCTLO, FRXRDTHO, FRXUORM0, FRXYALLO, FRXYUNAO

**00A Buffer error****Explanation**

IMS Database Recovery Facility is unable to obtain a data buffer. Register 14 contains the return code from the obtain buffer service.

**Module**

FRXPDIRO, FRXPDSR0, FRXPSS0

**00B Set storage length****Explanation**

IMS Database Recovery Facility is unable to set values for obtaining fixed storage. Register 14 contains the return code from the set storage service.

**Module**

FRXEPQS0, FRXICLI0, FRXMINIO, FRXMINI1

**00C Post error****Explanation**

IMS Database Recovery Facility is unable to post a dispatchable thread. Register 14 contains the return code from the post service.

**Module**

FRXCAMGO, FRXDAQS0

**00D AWE wait error****Explanation**

IMS Database Recovery Facility failed while waiting for recovery I/O to complete. Register 14 contains the return code from the wait service.

**Module**

FRXIOCNO, FRXIOSMO, FRXIVCNO, FRXIVSMO

**00F Write to IMS pipe error****Explanation**

IMS Database Recovery Facility detected an error sending data to the Database Recovery Manager using

an IMS pipe service. Register 14 contains the return code from the wait service.

#### Module

FRXPDIS0, FRXPDSO, FRXPDSO

---

### 011 Release storage error

#### Explanation

IMS Database Recovery Facility is unable to release storage. Register 14 contains the return code from the wait service.

#### Module

FRXPDIRO, FRXPDSO, FRXPDSO

---

### 012 Terminate thread

#### Explanation

IMS Database Recovery Facility is unable to end a thread. Register 14 contains the return code from the wait service.

#### Module

FRXIRTHO

---

### 013 Create IMS pipe error

#### Explanation

IMS Database Recovery Facility is unable to create an IMS pipe for inter-address space communication. Register 14 contains the return code from the wait service.

#### Module

FRXPDIRO, FRXPDSO, FRXPDSO

---

### 014 Address space parameter error

#### Explanation

IMS Database Recovery Facility experienced an error using the ASEX macro to attempt to access and interpret parameters passed by the Database Recovery Manager as part of creating the IMS Database Recovery Facility address space. Register 14 contains the return code from the create address space service.

#### Module

FRXSDR00

---

### 015 IMS pipe create error - DSPSERV

#### Explanation

The IMS pipe create service detected an error from the DSPSERV macro. Register 14 contains the return code from the DSPSERV macro.

#### Module

FRXPPIB0, FRXWSPL0

---

### 016 IMS pipe create error - ALESERV

#### Explanation

The IMS pipe create service detected an error from the ALESERV macro. Register 14 contains the return code from the ALESERV macro.

#### Module

FRXPPIB0, FRXRAUA0, FRXWSPL0

---

### 017 Establish ESTAE error - ESTAEX

#### Explanation

The IMS pipe service detected an error using the ESTAEX macro.

#### Module

FRXPPIB0, FRXPPRB0, FRXPPSB0

---

### 018 Buffer pool create error

#### Explanation

IMS Database Recovery Facility is unable to create a buffer pool storage for use during recovery. The IMS Database Recovery Facility address space abnormally ends with Register 14 containing the return code from the buffer pool create service.

#### Module

FRXMINI0, FRXMINI1, FRXMSTR0

---

### 019 Obtain buffer pool storage error

#### Explanation

The IMS Database Recovery Facility subordinate address space abnormally ends because it is unable to obtain buffer storage.

#### Module

FRXMSTR1

---

### 01A Startup parameter length error

### Explanation

The IMS Database Recovery Facility address space abnormally ends because an invalid length is detected for the startup parameters provided by IMS. Register 14 containing the invalid parameter length.

### Module

FRXSDR00

|            |                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------|
| <b>01B</b> | <b>Internal error indicating a problem in initializing the IMS Database Recovery Facility RAUX service</b> |
|------------|------------------------------------------------------------------------------------------------------------|

### Explanation

The IMS Database Recovery Facility address space ends abnormally. Acquire an SVC dump of the IMS Database Recovery Facility master address space, and contact the IBM Support Center for assistance.

### Module

FRXLBLD0, FRXMSTR1, FRXUPRE0

|            |                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------|
| <b>01C</b> | <b>Internal error indicating a problem in ending the IMS Database Recovery Facility RAUX service</b> |
|------------|------------------------------------------------------------------------------------------------------|

### Explanation

IMS Database Recovery Facility ends abnormally.

### Module

FRXMTRM0, FRXUPRE0, FRXVRCV0

|            |                                                                                                         |
|------------|---------------------------------------------------------------------------------------------------------|
| <b>01D</b> | <b>Internal error indicating a problem in processing an IMS Database Recovery Facility RAUX service</b> |
|------------|---------------------------------------------------------------------------------------------------------|

### Explanation

IMS Database Recovery Facility ends abnormally. Acquire an SVC dump of the IMS Database Recovery Facility master address space, and contact the IBM Support Center for assistance.

### Module

FRXGAPI0, FRXIOCN0, FRXIVCN0, FRXIOSM0, FRXIVSM0, FRXRAUX0, FRXUAPI0, FRXUPRE0

|            |                                                                                                                   |
|------------|-------------------------------------------------------------------------------------------------------------------|
| <b>01E</b> | <b>Internal error indicating a problem in initializing the IMS Database Recovery Facility WTO capture service</b> |
|------------|-------------------------------------------------------------------------------------------------------------------|

### Explanation

IMS Database Recovery Facility ends abnormally. Acquire an SVC dump of the IMS Database Recovery Facility master address space, and contact IBM Software Support for assistance.

### Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0, FRXMSTR1, FRXUPRE0

|            |                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------|
| <b>01F</b> | <b>Internal error indicating a problem in ending the IMS Database Recovery Facility WTO capture service</b> |
|------------|-------------------------------------------------------------------------------------------------------------|

### Explanation

IMS Database Recovery Facility ends abnormally. Acquire an SVC dump of the IMS Database Recovery Facility master address space, and contact the IBM Support Center for assistance

### Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0, FRXMSTR1, FRXUPRE0

|            |                                                                                                                |
|------------|----------------------------------------------------------------------------------------------------------------|
| <b>020</b> | <b>Internal error indicating a problem in processing an IMS Database Recovery Facility WTO capture service</b> |
|------------|----------------------------------------------------------------------------------------------------------------|

### Explanation

IMS Database Recovery Facility ends abnormally. Acquire an SVC dump of the IMS Database Recovery Facility master address space, and contact IBM Software Support for assistance.

### Module

FRXUPRE0

|            |                                                                                                           |
|------------|-----------------------------------------------------------------------------------------------------------|
| <b>021</b> | <b>Internal error indicating a problem in initializing the IMS Database Recovery Facility to HPIC API</b> |
|------------|-----------------------------------------------------------------------------------------------------------|

### Explanation

IMS Database Recovery Facility ends abnormally. If you are recovering databases or areas with the IC() keyword, refer to the data set identified in the ICEPRINT DD statement of the address space procedure for IMS High Performance Image Copy error messages. Take the action indicated by the messages.

## Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0,  
FRXVSTA0, FRXVSTA1

---

|            |                                                                                                                                   |
|------------|-----------------------------------------------------------------------------------------------------------------------------------|
| <b>022</b> | <b>Internal error indicating a problem in initializing the IMS Database Recovery Facility to HPPC or DEDB Pointer Checker API</b> |
|------------|-----------------------------------------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility ends abnormally. If you are recovering full function database data sets with the PC() keyword, refer to the data set identified in the PROCCTL DD statement of the address space procedure for IMS High Performance Pointer Checker error messages.

If you are recovering fast path area data sets with the PC() keyword, refer to the data set identified in the MSGOUT DD statement of the address space procedure for HP DEDB Pointer Checker error messages. Take the action indicated by the messages.

## Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0,  
FRXVSTA0

---

|            |                                                                                                     |
|------------|-----------------------------------------------------------------------------------------------------|
| <b>023</b> | <b>Internal error indicating a problem in ending the IMS Database Recovery Facility to HPIC API</b> |
|------------|-----------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility ends abnormally. If you are recovering databases or areas with the IC() keyword, refer to the data set identified in the ICEPRINT DD statement of the address space procedure for IMS High Performance Image Copy error messages. Take the action indicated by the messages.

## Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0,  
FRXVRCV0

---

|            |                                                                                                                             |
|------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>024</b> | <b>Internal error indicating a problem in ending the IMS Database Recovery Facility to HPPC or DEDB Pointer Checker API</b> |
|------------|-----------------------------------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility ends abnormally. If you are recovering full function database data sets with the PC() keyword, refer to the data set identified in the PROCCTL DD statement of the address space procedure for IMS High Performance Pointer Checker error messages.

If you are recovering fast path area data sets with the PC() keyword, refer to the data set identified in the MSGOUT DD statement of the address space procedure for HP DEDB Pointer Checker error messages. Take the action indicated by the messages.

## Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0,  
FRXVRCV0

---

|            |                                                                                                             |
|------------|-------------------------------------------------------------------------------------------------------------|
| <b>025</b> | <b>Internal error indicating a problem in processing an IMS Database Recovery Facility to HPIC API call</b> |
|------------|-------------------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility ends abnormally. If you are recovering databases or areas with the IC() keyword, refer to the data set identified in the ICEPRINT DD statement of the address space procedure for IMS High Performance Image Copy error messages. Take the action indicated by the messages.

## Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0

---

|            |                                                                                                                                     |
|------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>026</b> | <b>Internal error indicating a problem in processing an IMS Database Recovery Facility to HPPC or DEDB Pointer Checker API call</b> |
|------------|-------------------------------------------------------------------------------------------------------------------------------------|

---

## Explanation

IMS Database Recovery Facility ends abnormally. If the JCL is correct and there are no missing or incorrect DD statements, take one of the following actions.

- If you are recovering full function database data sets with the PC() keyword, refer to the data set identified in the PROCCTL DD statement of the address space procedure for IMS High Performance Pointer Checker error messages.
- If you are recovering fast path area data sets with the PC() keyword, refer to the data set identified in the MSGOUT DD statement of the address space procedure for HP DEDB Pointer Checker error messages. Take the action indicated by the messages.

## Module

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0

---

|            |                             |
|------------|-----------------------------|
| <b>027</b> | <b>IMS and DBDLIB error</b> |
|------------|-----------------------------|

---

**Explanation**

IAU=yes and both IMS and DBDLIB cannot be specified for a subordinate address space.

**Module**

FRXSDR00

**029 IMS and DBDLIB error****Explanation**

A Log or Change Accumulation data set that is registered in the RECON could not be allocated. Depending on the reason, either message FRD4212I or FRD4210I is issued with this abend and contains the data set name that could not be allocated. IMS Database Recovery Facility requires the data set to be allocated in order to complete the database recovery.

**Module**

FRXYALLO

**02A IMS DD Missing Error****Explanation**

Integrated Auxiliary Utility processing was requested, but the IMS DD the utilities require is missing. This abend is preceded by message FRD6144E.

**Module**

FRXMPSB0 and FRXMDBD0.

**02B DD Mismatch Error****Explanation**

Integrated Auxiliary Utility processing was requested but the data sets allocated to the same DDs in the master and subordinate address spaces do not match. This abend is preceded by FRD6146E and indicates the specific DD in error.

**Module**

FRXMSTR1, FRXSDR20

**02C Allocation error****Explanation**

IMS Database Recovery Facility attempted to dynamically allocate a data set on behalf of IAU processing, but the request to the DYNALLOC service failed. This abend will have been preceded by messages FRD9003A and FRD4100I

**Module**

FRXIVCN0, FRXIOCN0, FRXIOSM0, FRXIVSM0, FRXTSIB0, FRXTSPRO

**02D LIU Initialization Error****Explanation**

LIU@GOPT(xxxx) was coded in SYSIN or the in FRXDRFnn PROCLIB startup member, but Library Integrity Utility initialization failed with RC=16.

**System programmer response**

Look at the FABLnnnnE message that Library Integrity Utility has written to the SYSLOG to determine the reason for the failed initialization.

**Module**

FRXVSTA0

**02E LIU Initialization Error****Explanation**

LIU@GOPT (xxxx) was coded in SYSIN or in FRXDRFnn the PROCLIB startup member, but the Library Integrity Utility load module was not found in the STEPLIB DD concatenation. This abend is accompanied by message FRD9008E.

**System programmer response**

Add the correct Library Integrity Utility load library to the STEPLIB concatenation in all JCL procedures, including the MAS batch job JCL, the RSS JCL procedure DRFPROC and the UAS for IMS Index Builder or DFSPREC0 procedure set in the DRFIAX parameter. Note that all STEPLIB DDs in all procedures must have identical concatenations. See the descriptions for the STEPLIB and JOBLIB DDs for further details.

**02F LIU Initialization Error****Explanation**

LIU@GOPT (xxxx) has been coded in SYSIN or in FRXDRFnn the PROCLIB startup member, but the version of the load module in the Library Integrity Utility library concatenated to the STEPLIB DD is lower than the supported version. This abend is accompanied by FRD9009E.

**System programmer response**

Change the Library Integrity Utility library to a supported version such as V2R2.

|                                                                                                                                      |                                    |
|--------------------------------------------------------------------------------------------------------------------------------------|------------------------------------|
| <b>Module</b><br>FRXVSTA0                                                                                                            |                                    |
| <b>030</b>                                                                                                                           | <b>LIU Termination Error</b>       |
| <b>Explanation</b><br>Library Integrity Utility encountered error during termination with RC=16.                                     |                                    |
| <b>System programmer response</b><br>Examine the FABLnnnnE message written by the Library Integrity Utility to the job log.          |                                    |
| <b>Module</b><br>FRXVSTA0                                                                                                            |                                    |
| <b>031</b>                                                                                                                           | <b>LIU Recovery Error</b>          |
| <b>Explanation</b><br>Library Integrity Utility encountered an error during the recovery phase with RC=16.                           |                                    |
| <b>System programmer response</b><br>Examine the FABLnnnnE message that was written by the Library Integrity Utility to the job log. |                                    |
| <b>Module</b><br>FRXVSTA0                                                                                                            |                                    |
| <b>032</b>                                                                                                                           | <b>LIU commit processing error</b> |
| <b>Explanation</b><br>Library Integrity Utility encountered error during commit phase with RC=16.                                    |                                    |
| <b>System programmer response</b><br>Examine the FABLnnnnE message written by the Library Integrity Utility to the job log.          |                                    |

|                                                                                                                                                                                                                   |                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------|
| <b>Module</b><br>FRXVSTA0                                                                                                                                                                                         |                                                                 |
| <b>033</b>                                                                                                                                                                                                        | <b>Error in SLB (System Level Backup) API call.</b>             |
| <b>Explanation</b><br>A call to the SLB API ended in error.                                                                                                                                                       |                                                                 |
| <b>System programmer response</b><br>Acquire and SVC dump of the abending address space, whether it is the IMS Database Recovery master address space (MAS) or the recovery sort subordinate (RSS) address space. |                                                                 |
| <b>Module</b><br>FRXICTL0, FRXIRTH0, FRXVSTA0                                                                                                                                                                     |                                                                 |
| <b>034</b>                                                                                                                                                                                                        | <b>Error in HPIO API call when HPIO(WRITE(Y) was specified.</b> |
| <b>Explanation</b><br>A call to the HPIO API ended in error.                                                                                                                                                      |                                                                 |
| <b>System programmer response</b><br>Acquire an SVC dump of the IMS Database Recovery Facility recovery sort subordinate (RSS) address space, and contact IBM Software Support for further assistance.            |                                                                 |
| <b>Module</b><br>FRXSDR00, FRXI0CN0, FRXIVCN0                                                                                                                                                                     |                                                                 |

## Reason codes for logic error abend ABENDU0385

IMS Database Recovery Facility detected a logic error. Reason codes further identify the type of error:

|                                                                                                                                                                      |                                  |                                                                                                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>001</b>                                                                                                                                                           | <b>Invalid AWE function code</b> | <b>Module</b><br><br>FRXCAMG0, FRXCBDM0, FRXICLI0, FRXIRTH0, FRXLMRG0, FRXLPD00, FRXLSRT0, FRXLSRT1, FRXLSRT2, FRXLST0, FRXMSTR0, FRXMSTR1, FRXPDIS0, FRXPDSS0, FRXPSDS0, FRXQBUFO, FRXRCTLO, FRXRDTH0, FRXTAS00, FRXTAU00, FRXUORM0, FRXVCMD0, FRXWSPL0 |
| <b>Explanation</b><br><br>An IMS Database Recovery Facility thread attempted to process an unknown AWE function request. Register 14 contains the AWE function code. |                                  | <b>002</b><br><br><b>Data set allocation parameter error</b>                                                                                                                                                                                             |

**Explanation**

IMS Database Recovery Facility detected a data set allocation parameter error. Register 14 contains the address of the FRXRVR control block.

**Module**

FRXYALLO

**003 Control function error****Explanation**

IMS Database Recovery Facility control pipe routine detected an unknown function request from the IMS Control Region.

**Module**

FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0, FRXRAUX0, FRXVRCV0, FRXVSTA0, FRXPDIR0, FRXPDSR0, FRXPDSR0

**006 Invalid storage size****Explanation**

IMS Database Recovery Facility detected an unusable storage size for an obtain storage request.

**Module**

FRXPPIB0, FRXYALLO

**009 Invalid SSID detected****Explanation**

IMS Database Recovery Facility identified an invalid SSID during log buffering. Register 14 contains the address of the AWE function code.

**Module**

FRXQBUFO

**00A Invalid record detected****Explanation**

IMS Database Recovery Facility detected an invalid record.

**Module**

FRXPDIR0, FRXPDSR0, FRXPDSR0, FRXRBUFO, FRXUORM0

**00B Invalid data buffer detected****Explanation**

IMS Database Recovery Facility detected a buffer with invalid data.

**Module**

FRXICAI0, FRXCBDM0, FRXPDSR0, FRXQBUFO, FRXRBUFO, FRXUORM0

**00D Data set restore internal logic error****Explanation**

IMS Database Recovery Facility data set restore process detected an internal logic error while applying updates from change accumulation and/or logs.

**Module**

FRXIDYN0, FRXIOCN0, FRXIOSM0, FRXIVCN0, FRXIVSM0

**00F Log merge initialization error****Explanation**

IMS Database Recovery Facility encountered an error during log merge initialization. Register 14 contains the AWE function code.

**Module**

FRXLMRG0

**010 Log merge invalid input****Explanation**

IMS Database Recovery Facility detected invalid data passed in as input for log merge. Register 14 contains the AWE function code.

**Module**

FRXLMRG0

**011 Early end of data****Explanation**

IMS Database Recovery Facility detected an unexpected early end-of-data condition. Register 14 contains the AWE function code.

**Module**

FRXLMRG0

**012 Log merge invalid buffer chain**



**Explanation**

IMS Database Recovery Facility detected an invalid mergeid in the log input buffer chain. Register 14 contains the AWE function code.

**Module**

FRXLMRGO

---

**013**                      **Log merge buffer sequence error**

**Explanation**

IMS Database Recovery Facility detected a sequence error in the log buffer input chain. Register 14 contains the AWE function code.

**Module**

FRXLMRGO

---

**014**                      **Log merge invalid mergeid**

**Explanation**

IMS Database Recovery Facility detected an invalid mergeid during end of log merge processing. Register 14 contains the AWE function code.

**Module**

FRXLMRGO

---

**015**                      **Invalid ECB address**

**Explanation**

IMS Database Recovery Facility is unable to get the task ECB address.

**Module**

FRXWSPL0

---

**017**                      **Subordinate address space sort error**

**Explanation**

IMS Database Recovery Facility encountered an error during processing in the subordinate address spaces.

**Module**

FRXLNOT0, FRXLPD00, FRXLSRT1, FRXLSRT2, FRXLSTS0, FRXTAU00

---

**018**                      **Invalid end of recovery information**

**Explanation**

Invalid end of recovery information is detected by the IMS Database Recovery Facility master address space.

**Module**

FRXPDSR0

---

**019**                      **Error processing IC entry information**

**Explanation**

IMS Database Recovery Facility detected an error processing IC entry information.

**Module**

FRXLBLD0

---

**01A**                      **Invalid parameter detected**

**Explanation**

IMS Database Recovery Facility encountered an invalid parameter during SYSIN parsing. Register 14 contains the address of the parameter table.

**Module**

FRXVADD0

---

**01B**                      **DBRC Authorization Call Error**

**Explanation**

IMS Database Recovery Facility encountered an unexpected error when calling DBRC for database authorization.

**System action**

System terminates.

**Module**

FRXTAU00

---

**01C**                      **IMS Database Recovery Facility initialization error**

**Explanation**

IMS Database Recovery Facility failed to initialize.

**User response**

Contact IBM Software Support.

---

**01D**                      **Cannot recover to specified RCVTIME**

**Explanation**

IMS Database Recovery Facility attempted to recover to a time where an online reorganization was initiated but the online reorganization was not yet active. The target data set allocation information (x'2930' log) is not available, and therefore recovery cannot finish.

**User response**

Use the start time of the online reorganization to bypass online reorganization processing, or specify a time after the online reorganization was active (x'2940' log).

---

## Chapter 23. Batch condition codes

This reference section provides detailed information about condition codes set by the IMS Database Recovery Facility batch processing step. The explanations provided in this reference can help you diagnose, troubleshoot, and solve IMS Database Recovery Facility problems.

**0**

All recovery is successful or VERIFY was successful.

**4**

At least one recovery list entry was successfully recovered, but at least one was not. This code is also set if VERIFY processing encountered errors in finding the recovery information for some items in the recovery list.

**8**

None of the items on the recovery list were successfully recovered or processed with the VERIFY option. This condition code is also set if there were syntax errors in the SYSIN statements preventing the job from running.



---

## Chapter 24. Gathering diagnostic information

Before you report a problem with IMS Database Recovery Facility to IBM Software Support, you need to gather the appropriate diagnostic information.

### Before you begin

For current guidelines for gathering the proper IMS Database Recovery Facility diagnostic documentation when reporting a problem to IBM Software Support, locate the technote entitled *MustGather: Guidelines for IMS Database Recovery Facility customers in gathering the proper diagnostic documentation when reporting a problem to support* on the IBM Support page at the following link:

<http://www-01.ibm.com/support/docview.wss?uid=swg21170819>

More general recommendations are provided in the following procedure:

### Procedure

Provide the following information for all IMS Database Recovery Facility problems:

- A clear description of the problem and the steps that are required to re-create the problem
- All messages that were issued as a result of the problem
- Product release number and the number of the last program temporary fix (PTF) that was installed
- The version of IMS that you are using and the type and version of the operating system that you are using

Provide additional information based on the type of problem that you experienced:

#### **For online abends, provide the following information:**

- A screen capture of the panel that you were using when the abend occurred
- The job log from the TSO session that encountered the abend
- The job log from the server
- A description of the task that you were doing before the abend occurred

#### **For errors in batch processing, provide the following information:**

- The complete job log
- Print output
- Contents of the data sets that were used during the processing



---

## Part 7. Reference: Utilities and reports

Reference information supports the tasks that you must complete to install, customize, and use IMS Database Recovery Facility.

**Topics:**

- [Chapter 25, “Recovery job statistics report,” on page 279](#)
- [Chapter 26, “Sample reports,” on page 283](#)





---

## Chapter 25. Recovery job statistics report

The recovery job statistics report is a stand-alone batch job that can process system management facility (SMF) records and produce a report showing the elapsed time, CPU time, and I/O counts for an IMS Database Recovery Facility (DRF) recovery.

This report includes not only the DRF master job but also all associated address spaces that are spawned by DRF or any of the auxiliary utilities driven by DRF. This report lists each individual address space (jobs and started tasks) that is involved in the recovery, and also provides: 1) overall totals, and 2) totals by address space type.

### Setting up and running the report job

The JCL to produce this report can be found in SBSYSAMP(BSY#RSRP). The job includes comments on how to provide the input and other job customization. After a successful DRF recovery has been completed, you can run this job to produce the report.

The first time this report is run, you must specify a valid SMF data set for the //SMFIN DD statement that contains the SMF records encompassing the DRF recovery job and associated tasks.

You must also specify input parameter EXTRACT-SMF Y to tell the utility to read the records from the //SMFIN DD.

If you code the //SMFDATA DD on this run and specify a permanent data set to be created (for example, DISP=(NEW,CATLG)), then the SMF records that are needed are written to the SMFDATA data set and saved for subsequent runs. This action allows you to:

- Comment out the //SMFIN DD,
- Code the //SMFDATA DD to the data set created in the prior run (for example, DISP=SHR), and
- Set EXTRACT-SMF N on subsequent runs.

This adjustment saves time because the job now reads only those records needed from the SMFDATA data set, and avoids the re-reading of all records in the SMFIN data set.

### Parameter reference for job statistics report

The following parameters are used by the report.

#### EXTRACT-SMF

This parameter indicates whether you want the job to extract the SMF records from either a live SMF data set or an offloaded data set, and then place the needed records in the data sets specified by the SMFDATA DD.

If EXTRACT-SMF=Y, then SMFDATA can specify either a temporary or permanent data set.

If SMFDATA specifies a temporary data set, any re-run of this job must specify EXTRACT-SMF=Y and must specify the SMFIN DD.

If you specify EXTRACT-SMF=Y and specify a new, permanent data set for SMFDATA, the records needed by this job are saved in the SMFDATA data set. Subsequent job runs can specify EXTRACT-SMF=N and the permanent data set for SMFDATA, and avoid re-reading the original SMF data.

#### START-TIMESTAMP

This parameter specifies the starting range that the utility uses to locate the start of the DRF master recovery job, and must be specified.

This parameter limits the records used in the report and can prevent accumulating incorrect jobs when the same DRF job name is used by multiple runs.

Set this parameter to a date/time that is just prior to the DRF master job start time.

Timestamp format:

```
YYYY-MM-DD-HH.MM.SS.HHHHHH
```

## END-TIMESTAMP

This parameter specifies the ending range that the utility uses to locate the end of the DRF master recovery job, and must be specified.

This parameter limits the records used in the report and can prevent accumulating incorrect jobs when the same DRF job name is used by multiple runs.

Set this parameter to a date/time that is just after the DRF master job end time.

Timestamp format:

```
YYYY-MM-DD-HH.MM.SS.HHHHHH
```

## DRF-JOBNAME

This parameter specifies the fully qualified job name of the DRF master job, and is required.

The following parameters are optional but should have the same value specified by the DRF Master job. Otherwise all associated recovery jobs will not be reported and accumulated.

## ASPREF

This parameter specifies the ASPREF() value, which is used as the prefix for the Recovery Sort Subordinate (RSS) address spaces.

## FSPREF

This parameter specifies the FSPREF() value, which is used as the prefix for the Fast Path Secondary Index address spaces.

## PRPREF

This parameter specifies the PRPREF() value, which is used as the prefix for the IMS DFSPREC0 utility HALDB index rebuild address spaces.

## IBPREF

This parameter specifies the IBPREF() value, which is used as the prefix for the IMS Index Builder index rebuild address spaces.

## PCPREF

This parameter specifies the PCPREF() value, which is used as the prefix for the IMS High Performance Pointer Checker (full function) address spaces.

## IBSORT

This parameter specifies the name of the sort started task used by the IMS Index Builder when rebuilding indexes.

This value is specified either in the IMS Index Builder IIURPRMS module, or if not specified, IIUSORTS is used by IMS Index builder.

## IBAPI

This parameter specifies the name of the API started task used by the IMS Index Builder when rebuilding indexes.

This value is specified either in the IMS Index Builder IIURPRMS module, or if not specified, IIUAPIFC is used by IMS Index builder.

## Sample report output

```
Control Cards:
DRF-JOBNAME      TSMXDMXD
ASPREF           RMXD
FSPREF           FMXD
PRPREF           XMXD
IBPREF           BMXD
PCPREF           PMXD
IBSORT           IIUSORTS
FPSORT           IIUAPIFC
EXTRACT-SMF      N
START-TIMESTAMP  2020-07-09-16.48.00.000000
END-TIMESTAMP    2020-07-09-17.00.00.000000
DEBUG
```

Earliest SMF job found Day: 2020.191 Time: 16:42:14  
 Latest SMF job found Day: 2020.191 Time: 16:53:58

| JOB<br>NAME | JES<br>ID | START<br>DATE | START<br>TIME | END<br>DATE | END<br>TIME | ELAPSED<br>TIME | CPU TIME<br>(SECS) | TOTAL<br>EXCP | JOB<br>TYPE |
|-------------|-----------|---------------|---------------|-------------|-------------|-----------------|--------------------|---------------|-------------|
| TSMXDMXD    | S0718228  | 2020.191      | 16:48:17      | 2020.191    | 16:50:39    | 00:02:21.32     | 1.89               | 20908         | DRFMAS      |
| RMXD0001    | S0718230  | 2020.191      | 16:48:30      | 2020.191    | 16:48:33    | 00:00:03.34     | 0.69               | 4205          | ASPREF      |
| RMXD0002    | S0718231  | 2020.191      | 16:48:30      | 2020.191    | 16:48:33    | 00:00:02.53     | 0.59               | 3568          | ASPREF      |
| RMXD0003    | S0718232  | 2020.191      | 16:48:30      | 2020.191    | 16:48:32    | 00:00:02.42     | 0.57               | 3930          | ASPREF      |
| RMXD0004    | S0718233  | 2020.191      | 16:48:30      | 2020.191    | 16:48:33    | 00:00:02.54     | 0.58               | 4077          | ASPREF      |
| IIUAPIFC    | S0718240  | 2020.191      | 16:48:45      | 2020.191    | 16:48:59    | 00:00:14.45     | 0.07               | 370           | FPSORT      |
| IIUAPIFC    | S0718241  | 2020.191      | 16:48:45      | 2020.191    | 16:49:00    | 00:00:15.37     | 0.07               | 370           | FPSORT      |
| IIUAPIFC    | S0718242  | 2020.191      | 16:48:45      | 2020.191    | 16:49:03    | 00:00:17.08     | 0.07               | 370           | FPSORT      |
| IIUAPIFC    | S0718246  | 2020.191      | 16:48:46      | 2020.191    | 16:49:04    | 00:00:16.50     | 0.07               | 370           | FPSORT      |
| IIUAPIFC    | S0718244  | 2020.191      | 16:48:45      | 2020.191    | 16:49:08    | 00:00:22.57     | 0.07               | 371           | FPSORT      |
| IIUAPIFC    | S0718248  | 2020.191      | 16:48:46      | 2020.191    | 16:49:08    | 00:00:20.42     | 0.06               | 371           | FPSORT      |
| IIUAPIFC    | S0718243  | 2020.191      | 16:48:45      | 2020.191    | 16:49:10    | 00:00:24.45     | 0.06               | 371           | FPSORT      |
| IIUAPIFC    | S0718247  | 2020.191      | 16:48:46      | 2020.191    | 16:49:10    | 00:00:22.12     | 0.07               | 372           | FPSORT      |
| IIUAPIFC    | S0718255  | 2020.191      | 16:49:20      | 2020.191    | 16:50:04    | 00:00:43.09     | 0.04               | 182           | FPSORT      |
| IIUAPIFC    | S0718256  | 2020.191      | 16:49:20      | 2020.191    | 16:50:04    | 00:00:43.10     | 0.05               | 185           | FPSORT      |
| IIUAPIFC    | S0718260  | 2020.191      | 16:49:27      | 2020.191    | 16:50:04    | 00:00:37.40     | 0.04               | 182           | FPSORT      |
| IIUAPIFC    | S0718259  | 2020.191      | 16:49:27      | 2020.191    | 16:50:04    | 00:00:37.43     | 0.05               | 185           | FPSORT      |
| IIUAPIFC    | S0718262  | 2020.191      | 16:49:36      | 2020.191    | 16:50:05    | 00:00:29.25     | 0.04               | 183           | FPSORT      |
| IIUAPIFC    | S0718263  | 2020.191      | 16:49:36      | 2020.191    | 16:50:05    | 00:00:29.24     | 0.03               | 182           | FPSORT      |
| IIUAPIFC    | S0718269  | 2020.191      | 16:50:15      | 2020.191    | 16:50:29    | 00:00:12.34     | 0.04               | 182           | FPSORT      |
| IIUAPIFC    | S0718270  | 2020.191      | 16:50:15      | 2020.191    | 16:50:29    | 00:00:12.35     | 0.04               | 182           | FPSORT      |
| FMXD0001    | S0718234  | 2020.191      | 16:48:34      | 2020.191    | 16:49:14    | 00:00:38.35     | 0.48               | 9037          | FSPREF      |
| FMXD0002    | S0718235  | 2020.191      | 16:48:34      | 2020.191    | 16:49:14    | 00:00:38.32     | 0.41               | 8746          | FSPREF      |
| BMXD0001    | S0718236  | 2020.191      | 16:48:34      | 2020.191    | 16:49:13    | 00:00:37.36     | 0.38               | 5043          | IBPREF      |
| BMXD0001    | S0718251  | 2020.191      | 16:49:13      | 2020.191    | 16:50:09    | 00:00:55.57     | 0.61               | 7576          | IBPREF      |
| BMXD0001    | S0718266  | 2020.191      | 16:50:09      | 2020.191    | 16:50:32    | 00:00:23.45     | 0.66               | 7601          | IBPREF      |
| BMXD0002    | S0718237  | 2020.191      | 16:48:34      | 2020.191    | 16:49:19    | 00:00:43.03     | 0.44               | 6544          | IBPREF      |
| BMXD0002    | S0718253  | 2020.191      | 16:49:19      | 2020.191    | 16:50:11    | 00:00:52.11     | 0.62               | 7564          | IBPREF      |
| BMXD0002    | S0718267  | 2020.191      | 16:50:12      | 2020.191    | 16:50:22    | 00:00:09.24     | 0.28               | 5047          | IBPREF      |
| BMXD0003    | S0718238  | 2020.191      | 16:48:34      | 2020.191    | 16:49:13    | 00:00:37.44     | 0.37               | 5154          | IBPREF      |
| BMXD0003    | S0718252  | 2020.191      | 16:49:13      | 2020.191    | 16:49:25    | 00:00:11.53     | 0.27               | 4512          | IBPREF      |
| BMXD0003    | S0718258  | 2020.191      | 16:49:25      | 2020.191    | 16:50:12    | 00:00:47.20     | 0.68               | 7620          | IBPREF      |
| BMXD0003    | S0718268  | 2020.191      | 16:50:12      | 2020.191    | 16:50:12    | 00:00:00.02     | 0.02               | 65            | IBPREF      |
| BMXD0004    | S0718239  | 2020.191      | 16:48:34      | 2020.191    | 16:49:19    | 00:00:43.42     | 0.46               | 6787          | IBPREF      |
| BMXD0004    | S0718254  | 2020.191      | 16:49:19      | 2020.191    | 16:49:59    | 00:00:39.02     | 0.36               | 5269          | IBPREF      |
| BMXD0004    | S0718265  | 2020.191      | 16:49:59      | 2020.191    | 16:50:19    | 00:00:19.39     | 0.48               | 6489          | IBPREF      |
| IIUSORTS    | S0718245  | 2020.191      | 16:48:45      | 2020.191    | 16:49:13    | 00:00:27.23     | 0.16               | 645           | IBSORT      |
| IIUSORTS    | S0718249  | 2020.191      | 16:48:46      | 2020.191    | 16:49:13    | 00:00:25.03     | 0.16               | 645           | IBSORT      |
| IIUSORTS    | S0718257  | 2020.191      | 16:49:20      | 2020.191    | 16:50:04    | 00:00:44.23     | 0.17               | 645           | IBSORT      |
| IIUSORTS    | S0718261  | 2020.191      | 16:49:27      | 2020.191    | 16:50:05    | 00:00:37.08     | 0.16               | 646           | IBSORT      |
| IIUSORTS    | S0718264  | 2020.191      | 16:49:36      | 2020.191    | 16:50:05    | 00:00:29.37     | 0.16               | 645           | IBSORT      |
| IIUSORTS    | S0718271  | 2020.191      | 16:50:15      | 2020.191    | 16:50:29    | 00:00:13.50     | 0.17               | 645           | IBSORT      |
| PMXD8228    | S0718229  | 2020.191      | 16:48:26      | 2020.191    | 16:48:29    | 00:00:02.52     | 0.15               | 2225          | PCPREF      |
| TOTAL-->    |           |               |               |             |             |                 | 12.84              | 140266        | <--TOTAL    |

| JOB<br>TYPE | CPU TIME<br>(SECS) | TOTAL<br>EXCP |
|-------------|--------------------|---------------|
| ASPREF      | 2.43               | 15780         |
| FSPREF      | 0.89               | 17783         |
| PRPREF      | 0.00               | 0             |
| IBPREF      | 5.63               | 75271         |
| PCPREF      | 0.15               | 2225          |
| IBSORT      | 0.98               | 3871          |
| FPSORT      | 0.87               | 4428          |



---

## Chapter 26. Sample reports

The following topics contain examples of the types of reports which are generated by the IMS Database Recovery Facility.

### Topics:

- [“Output report content” on page 283](#)
- [“Sample reports processing” on page 284](#)
- [“Example report: Image copy or incremental image copy output” on page 284](#)
- [“Example report: Point-in-time recovery” on page 290](#)
- [“Example report: Full point-in-time recovery” on page 295](#)
- [“Example report: Verify of the point-in-time recovery” on page 298](#)
- [“Example report: VERIFY\(ALLOC\) recovery” on page 302](#)
- [“Example report: VERIFY\(OPEN\) recovery for a point-in-time” on page 305](#)
- [“Example report: Full recovery using environment control statements” on page 308](#)
- [“Example report: Full recovery with IC\(\) and PC\(\) utilities” on page 311](#)

---

### Output report content

For the IMS Database Recovery Facility, an appropriate output report displays the results of the processing performed. This report is written to the REPORT DD data set.

The contents of the report include the following items:

- Images of each input control statement.
  - List of the input commands with an indication of any syntactically incorrect statements or statements that are incorrectly placed and were, therefore, ignored.
  - Summary of processing performed for each database data set, which includes:
    - Identification, type, and status of database data sets recovered or copied.
    - Number of records written to each DBDS.
    - Number of IC, CA, and log records read.
- Note:** The image copy record read count is a number followed by a "T" if the image copy is restored by DFSMSdss. The count represents the number of tracks restored by DFSMSdss. Otherwise, the count represents the number of records read from a non-DFSMSdss image copy data set.
- The messages are written to the SYSPRINT DD statement of the IMS Database Recovery Facility MAS or RSS, or both, but not to the REPORT.
  - Status of recovery completion provided for each item in the recovery list.
  - Other report items include:
    - Number and attributes of image copy and log data sets read.
    - Destination of output: DBDS.
    - Recovery processing options in effect.
    - Number and IDs of unit of recovery (UOR) or unit of work (UOW) instances at PITR time (optional).

**Note:** Inflight UOR has the same meaning as UOW

- Messages related to SORT are written to the SYSOUT DD statement of the corresponding the IMS Database Recovery Facility RSSs.

The report might contain one or more sub-reports. The sub-reports correspond to the processing output created by the IMS Database Recovery Facility process. The list of sub-reports includes the following:

- Log data set activity report:
  - Number and attributes of image copy and log data sets read.
- The open UOW report for PITR processing: When you invoke a PITR, there is a possibility that there are open UOWs. The changes associated with the open UOW are not applied to the database because the changes were not committed at the recovery time. A list of the open UOW instances at the recovery time is shown in the output report generated by the IMS Database Recovery Facility as-is information after a point-in-time recovery (PITR) on inflight UORs that update data sets that are physically or logically related to the data sets being recovered.

## Sample reports processing

The example reports are a result of the combined effects of several components.

- FRXDRFxx member statements.
- The SYSIN control statements.
- Recovery processing activity.

Although the following reports have many similarities in format, note the following differences:

- PITR recoveries include a trailing "PITR Open UOW/UOR Report" following the other sections; non-PITR recoveries do not include this section.
- Recoveries specifying VERIFY contain the value "N/A" for statistics in the "Facility Summary Report," zeros for statistics in the "Data Set I/O Report," "No open UOWs..." in the "PITR Open UOW/UOR Report"; actual recoveries contain valid statistics in these reports.

The following is an example of the SYSPRINT output, which displays the FRXDRFxx statements in effect when the report examples were created:

```
FRD6133I DRF STARTUP PARAMETER MEMBER FRXDRF01 SELECTED
FRD7201I READNUM(3,3)
FRD7201I SORTPARM(NUM(9),HIPRMAX(OPTIMAL),MAINSIZE(5),
FRD7201I ASPREF(IDRF),AVGLEN(10000),FILSZ(30000))
FRD7201I SPSIZE(1000)
```

## Example report: Image copy or incremental image copy output

This report example shows the image copy or incremental image copy output.

### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I DBDSL101(DISP(NEW))
FRD7201I OUTPUT(ICR)
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=DRBXAG1)
FRD7201I UTILGBL(COMP=Y,DSN(&ICHLQ.&DBD.&DDN.),UNIT(3390),-
FRD7201I ICBUF(15),VOLSER(222222),COMPRTN(FABJCMP3))
FRD7201I ADD DB(DEDBJN23)
FRD7201I IC(ICHLQ(TEMP),SPACE(CYL,1,1)) -
FRD7201I DBATRB(DBDSL(101))
FRD7201I START ERROR(CONT)
```

### DATABASE RECOVERY FACILITY RECOVERY PARAMETERS

```
PROCESS : RCVTOIC          RECOVERY OPTIONS
SOURCE  : PRI              RCVTIME : Not Specified
READNUM : 03,03            TYPE    : Full
ERROR   : CONT             CHECK   : Yes
RCVTOKEN: DRF03            TIME FMT: LCL
DRFPROC : DRFV3            LBI     : No
LOGNUM  : 03,03            ICNUM   : 03,03
OUTPUT  : ICR              TAPECHK : No
CATDS   : No
SPSIZE  : 1024             CACHE   : No
```

### ----- Sort-Related Parameters -----

```
NUM      : 3                MAINSIZE: 30
FILSZ    : 21000            HIPRMAX : OPTIMAL
DYNALLOCC: N/A , N/A
AVGLEN   : 256
ASPREF   : DRFI
ASGNAME  : N/A
```

### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | IC  | Records Read | LOG  | Records Written | Subord. Reg Name | STC # | Final Status          |
|---------------|--------------|------|-----|--------------|------|-----------------|------------------|-------|-----------------------|
| DEDBJN23      | DB23AR0      | 1    | 180 | 0            | 1227 | 0               | DRFI0001         | 385   | No errors encountered |
| DEDBJN23      | DB23AR1      | 2    | 21  | 0            | 1226 | 0               | DRFI0002         | 386   | No errors encountered |
| DEDBJN23      | DB23AR2      | 3    | 147 | 0            | 0    | 0               | DRFI0003         | 387   | No errors encountered |
| DEDBJN23      | DB23AR3      | 4    | 147 | 0            | 0    | 0               | DRFI0001         | 385   | No errors encountered |
| DEDBJN23      | DB23AR4      | 5    | 231 | 0            | 0    | 0               | DRFI0002         | 386   | No errors encountered |
| DEDBJN23      | DB23AR5      | 6    | 99  | 0            | 0    | 0               | DRFI0003         | 387   | No errors encountered |

```

DEDBJN23 DB23AR6 7 165 0 0 0 DRFI0001 385 No errors encountered
DEDBJN23 DB23AR7 8 21 0 0 0 DRFI0002 386 No errors encountered
FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 01/14/2020 Time: 16:31 Page: 2

```

#### DATABASE RECOVERY FACILITY UTILITY REPORT

| Database | DDN     | Database Data Set Name | IC | PC/DP | IB  | PR  | Utility | Final Status |
|----------|---------|------------------------|----|-------|-----|-----|---------|--------------|
| DEDBJN23 | DB23AR0 | IMSTESTL.DB23AR0       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR1 | IMSTESTL.DB23AR1       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR2 | IMSTESTL.DB23AR2       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR3 | IMSTESTL.DB23AR3       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR4 | IMSTESTL.DB23AR4       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR5 | IMSTESTL.DB23AR5       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR6 | IMSTESTL.DB23AR6       | 00 | N/A   | N/A | N/A |         |              |
| DEDBJN23 | DB23AR7 | IMSTESTL.DB23AR7       | 00 | N/A   | N/A | N/A |         |              |

Final Return (RC) and Reason (RSN) Codes

| ---IC--- |     | ---PC--- |     | ---DP--- |     | ---IB--- |     | ---PR--- |     | ---LIU--- |     |
|----------|-----|----------|-----|----------|-----|----------|-----|----------|-----|-----------|-----|
| RC       | RSN | RC       | RSN | RC       | RSN | RC       | RSN | RC       | RSN | RC        | RSN |
| 00       | 00  | N/A      | N/A | N/A      | N/A | N/A      | N/A | N/A      | N/A | N/A       | N/A |

```

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 01/14/2020 Time: 16:31 Page: 2

```

#### DATABASE RECOVERY FACILITY IC / ICR OUTPUT REPORT

| Database Name | DD/Area Name | IC / ICR Data Set Name | Compr Count | Volume Serial | File Seq. | IC Type | IC Time           | Compr Ratio | Compr Routine |
|---------------|--------------|------------------------|-------------|---------------|-----------|---------|-------------------|-------------|---------------|
| DEDBJN23      | DB23AR0      | TEMP.DEDBJN23.DB23AR0  | 178         | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 2.5 FABJCMPS  |
| DEDBJN23      | DB23AR1      | TEMP.DEDBJN23.DB23AR1  | 19          | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 13.1 FABJCMPS |
| DEDBJN23      | DB23AR2      | TEMP.DEDBJN23.DB23AR2  | 145         | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 4.6 FABJCMPS  |
| DEDBJN23      | DB23AR3      | TEMP.DEDBJN23.DB23AR3  | 145         | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 4.6 FABJCMPS  |
| DEDBJN23      | DB23AR4      | TEMP.DEDBJN23.DB23AR4  | 229         | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 5.9 FABJCMPS  |
| DEDBJN23      | DB23AR5      | TEMP.DEDBJN23.DB23AR5  | 97          | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 8.9 FABJCMPS  |
| DEDBJN23      | DB23AR6      | TEMP.DEDBJN23.DB23AR6  | 163         | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 6.9 FABJCMPS  |
| DEDBJN23      | DB23AR7      | TEMP.DEDBJN23.DB23AR7  | 19          | 222222        | 00001     | Con.    | 20.013 11:16:53.8 | -0800       | 12.6 FABJCMPS |

```

FRD0000I IMS RECOVERY SOLUTION PACK V1R1 : IMS Database Recovery Facility Date: 01/14/2020 Time: 16:31 Page: 4

```

#### DATABASE RECOVERY FACILITY DATA SET I / O REPORT

Recover to point: Not Specified

| Image Copy Data Set Name           | Volume Serial | IC DS Read Count | IC Type | Time Stamp Range |             |
|------------------------------------|---------------|------------------|---------|------------------|-------------|
|                                    |               |                  |         | 1st Record       | Last Record |
| IMSVS.DEDBJN23.DB23AR0.IC.IC111249 | 222222        | 180              | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR1.IC.IC111249 | 222222        | 21               | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR2.IC.IC111249 | 222222        | 147              | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR5.IC.IC111250 | 222222        | 99               | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR3.IC.IC111249 | 222222        | 147              | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR4.IC.IC111249 | 222222        | 231              | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR6.IC.IC111250 | 222222        | 165              | STD     |                  |             |
| IMSVS.DEDBJN23.DB23AR7.IC.IC111250 | 222222        | 21               | STD     |                  |             |

| Change Accum Data Set Name               | Volume Serial | CA DS Read Count | Time Stamp Range |             |
|------------------------------------------|---------------|------------------|------------------|-------------|
|                                          |               |                  | 1st Record       | Last Record |
| No data available for this type data set |               |                  |                  |             |

| Log Data Set Name                    | Volume Serial | Log DS Read Count | IMS SYSID | Time Stamp Range         |                            |
|--------------------------------------|---------------|-------------------|-----------|--------------------------|----------------------------|
|                                      |               |                   |           | 1st Record               | Last Record                |
| IMSVS.RLDSP.IMS1.D20013.T1116000.V00 | 000000        | 344               | IMS1      | 2020.013 11:16:00.085956 | -02020.013 11:16:09.347812 |
| Prilog: 2020.013 11:13:43.900000     |               |                   |           |                          |                            |
| IMSVS.RLDSP.IMSA.D20013.T1117015.V00 | 000000        | 817               | IMSA      | 2020.013 11:17:01.514320 | -02020.013 11:17:49.476351 |
| Prilog: 2020.013 11:14:53.000000     |               |                   |           |                          |                            |
| IMSVS.RLDSP.IMSA.D20013.T1116576.V03 | 000000        | 0                 | IMSA      | 2020.013 11:17:00.048405 | -02020.013 11:17:01.514320 |
| Prilog: 2020.013 11:14:53.000000     |               |                   |           |                          |                            |
| IMSVS.RLDSP.IMSA.D20013.T1115306.V00 | 000000        | 817               | IMSA      | 2020.013 11:15:30.696015 | -02020.013 11:16:57.653660 |
| Prilog: 2020.013 11:14:53.000000     |               |                   |           |                          |                            |
| IMSVS.RLDSP.IMSA.D20013.T1116576.V02 | 000000        | 0                 | IMSA      | 2020.013 11:16:57.653660 | -02020.013 11:17:00.048405 |
| Prilog: 2020.013 11:14:53.000000     |               |                   |           |                          |                            |
| IMSVS.RLDSP.IMS1.D20013.T1115211.V00 | 000000        | 475               | IMS1      | 2020.013 11:15:21.111820 | -02020.013 11:16:00.085956 |
| Prilog: 2020.013 11:13:43.900000     |               |                   |           |                          |                            |
| IMSVS.RLDSP.IMS1.D20013.T1116093.V00 | 000000        | 0                 | IMS1      | 2020.013 11:16:09.347812 | -02020.013 11:16:53.864245 |
| Prilog: 2020.013 11:13:43.900000     |               |                   |           |                          |                            |

Below is an explanation of the image copy or incremental image copy output example. There are 6 sections to the report:

### • 1 - COMMANDS / CONTROL STATEMENTS

This section displays the images of the control statements read from the SYSIN file. Six control statements were read from the SYSIN file.

### • 2 - RECOVERY PARAMETERS

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

1. Explicitly coded in the SYSIN control statements.

– These values can be seen in the "Control Statements" part of the report:

**PROCESS: RCVTOIC**

A TOIC-type recovery was selected.

- This parameter was not specified. The recovery facility default parameter was used.

**SOURCE: PRI**

The primary log and image copy data sets are to be used as the source of input.

- This parameter was not specified. The recovery facility default parameter was used.

**READNUM: 03,03**

The maximum number of devices to be used is 03. Of these, up to 03 can be tape devices.

- The maximum number of total devices and the number of allowable tape devices were specified in the FRXDRFxx member.

**ERROR: CONT**

Should an error be encountered on one database during recovery, processing should continue for other databases.

- This parameter was not specified. The recovery facility default parameter was used.

**RCVTOKEN: DRF03**

The recovery token used during recovery is "DRF03."

**DRFPROC : DRFV3**

The procedure for the RSS is contained in member DRFV3 in the PROCLIB PDS.

**RCVTIME: Not Specified**

A recovery time was not specified.

**TYPE: Full**

A full recovery was requested.

- This parameter was not specified. The recovery facility default parameter was used.

**CHECK: YES**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDSs within that recovery group need to be included in the list.

- This parameter was not specified. The recovery facility default parameter was used.

**TIME FMT: LCL**

Time format is local.

**LBI: No**

Large Block Interface processing was not requested.

**LOGNUM: 06,20**

The maximum number of tape drives used for reading change accumulation data sets is 6. Of these, up to 20 can be read instances which can be initiated in parallel.

**OUTPUT: ICR**

The output type is incremental image copy.

**SPSIZE: 1024**

The SPSIZE specified is 1024.

**ICNUM: 06,20**

The maximum number of tape drives used for reading image copy data sets is 6. Of these, the maximum number of read instances that can be initiated in parallel is 20

**TAPECHK: No**

The availability status of tape devices before DBDS allocation is not checked.

- This parameter was not specified. The recovery facility default parameter was used.

**CACHE: No**

No image copy data sets are located on a VTS device.

- This parameter was not specified. The recovery facility default parameter was used.



### • **3 - SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database.

The three databases included in the recovery consisted of 5 database data sets. For each of these data sets, the following information is listed:

**Database Name**

The database name is listed.

**DD/AREA Name**

The DD name or Area name associated with each image copy data set is listed.

**DSID**

The data set ID number is listed.

**Records Read**

The number of records read from the image copy, change accumulation and log data sets for application in the recovery are listed. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg Name**

The name of the started task which was responsible for sorting the records being written for the indicated data set is listed.

**STC**

The started task control number.

**Final Status**

The status of the conclusion of the recovery is listed.

### • **4 - UTILITY REPORT**

**Database Name**

The database name is listed.

**DDN**

The DDNAME is listed.

**Database Data Set Name**

The database data set name is listed.

**IC**

The IMS HP Image Copy auxiliary utility return code is listed.

**PC/DP**

The IMS HP Pointer Checker/IMS HP Change Accumulation Utility DEDB Pointer Checker auxiliary utility return code is listed.

**IB**

The IMS Index Builder auxiliary utility return code is listed.

**PR**

The DFSPRECO auxiliary utility return code is listed.

### • **5 - IC / ICR OUTPUT REPORT**

**Database Name**

The database name is listed.

**DD/Area Name**

The DD name or Area name associated with each image copy data set is listed.

**IC / ICR Data Set Name**

The output image copy or incremental image copy data set name is listed.

**Compr Count**

The number of records written to the data set. If a compression routine is in use, it reflects the number of records resulting from compression.

**Volume Serial**

The volume serial number upon which the data set was written.

If there are multiple volumes, then there is one report line for each volume. If this is the case, the same data set name is listed on each succeeding report line.

**File Seq.**

The file sequence number for each data set.

**IC Type**

The image copy type is listed, where Bat. is for batch, Con. is for concurrent, and UIC for user image copy.

**IC Time**

The image copy time recorded in the image copy record in the RECONs. If the image copy is not registered because NOTIFY(N) was specified, then this field is left blank.

**Compr Ratio**

If an image copy compression routine is used, the compression ratio is listed.

**Compr Routine**

If an image copy compression routine is used, the name of the routine is listed.

• **6 - DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to Point" in this section.

The data set I/O report is divided into three sections:

**1. Image Copy Data Set Name**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**DS Count**

The number of records read from the data set is listed. If Flash Copy, the read count will be 'N/A'.

**IC Type**

The image copy type is listed. Allowable values for this field include:

*Table 6. Valid values for IC Type*

| <b>IC Type listed</b> | <b>Description</b>                                                                                                                                                                                                                          |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| STD                   | Standard Image Copy                                                                                                                                                                                                                         |
| IC2                   | Image Copy 2                                                                                                                                                                                                                                |
| IC EXT                | Extended Image Copy                                                                                                                                                                                                                         |
| HISAM UNL             | HISAM Unload                                                                                                                                                                                                                                |
| Time Stamp Range      | Although this heading appears, this field has no significance for image copy data sets.                                                                                                                                                     |
| SLB-CIC-O *           | The System Level Backup is concurrent and is offloaded to a sequential device. For Recovery, VERIFY(ALLOC) and VERIFY(OPEN), the offload data set names are the ones operated and reported on; not the data set name in the RECON IC record |
| SLB-CIC-F *           | The System Level Backup is concurrent and is on a fast-replication capable volume.                                                                                                                                                          |

Table 6. Valid values for IC Type (continued)

| IC Type listed | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SLB-BAT-O *    | The System Level Backup is batch and is offloaded to a sequential device. For Recovery, VERIFY(ALLOC) and VERIFY(OPEN), the offload data set names are the ones operated and reported on; not the data set name in the RECON IC record                                                                                                                                                                                                                                                   |
| SLB-BAT-F *    | The System Level Backup is batch and is on a fast-replication capable volume.                                                                                                                                                                                                                                                                                                                                                                                                            |
| SLB-RECON *    | This IC type indicates that the data set name on the report is symbolic only and represents the fact that a RECON IC record exists for the SLB. Because this data set name is symbolic only, it cannot be allocated or opened. This data set name is always displayed for VERIFY(LIST). If USESLBIC(Y) is set, and either VERIFY(ALLOC) or VERIFY(OPEN) is specified, and the SLB has been offloaded from the fast-replication device, then the offload data sets display if they exist. |

**Note:** \* If IMS Recovery Expert is available and USESLBIC(YES) is specified, the VERIFY option causes IMS Recovery Expert to write an image copy record to the RECON which represents the System Level Backup. The image copy record receives a data set name in the form of SLB. Ixxxxx . Dxxxxx . dbdname . ddname, and this image copy record is used for Recovery and Recovery with the VERIFY option.

If there is already an image copy record for the System Level Backup, or if no System Level Backup is available relative to the specified RCVTIME parameter, no image copy record is written to the RECON.

If USESLBIC(N) is in effect, then there is no attempt to operate on this symbolic data set name. The report reflects N/A in the Status column.

## 2. Change Accumulation Data Set Name

For each change accumulation data set used during the recovery, the following information is listed:

### Change Accum Data Set Name

The data set name is listed.

### Volume Serial

The volume or volumes upon which the data sets were found.

### DS Count

The number of records read from the data set is listed.

### Time Stamp Range

Although this heading appears, this field has no significance for image copy data sets, except when a System Level Backups was used for the recovery.

When SLBs are being processed, this header is changed to 'SLB IC DSN RECON.' The data set name from the image copy record representing the SLB is listed underneath.

If there is an SLB Offload data set, it is listed in the Image Copy Data Set Name column so that the two can be correlated.

To determine whether the Offload data set or the SLB on the fast-replication volume was used in the recovery, refer to the 'SLB IC DSN RECON' column. If the fast-replication volume was used, then the SLB image copy data set name is preceded with the characters '->'.

## 3. Log Data Set Name

For each log data set used during the recovery, the following information is listed:

### Log Data Set Name

The data set name is listed.

### Volume Serial

The volume or volumes upon which the data sets were found.

### DS Count

The number of records read from the data set is listed.

### IMS SYSID

The SYSID associated with the IMS which created the log is listed.

### Time Stamp Range

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

### PRILOG:

The PRILOG time listed in RECON for the log is listed.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.

## Example report: Point-in-time recovery

This report example shows the batch invocation of a PITR (point-in-time recovery).

```
FRD0000I  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   05/08/2020 10:19                                     Page 1

  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   C O M M A N D S / C O N T R O L   S T A T E M E N T S

FRD7201I  ADD DB(DIVNTZ02 DHVNTZ02 DXVNTZ02)
FRD7201I  START ERROR(CONT) READNUM(2) RCVTIME('20128101651' PITR)

  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   R E C O V E R Y   P A R A M E T E R S

PROCESS : RCVDBDS                RECOVERY OPTIONS
SOURCE  : PRI                    RCVTIME : 2020.128 17:16:51.000000 -07:00
READNUM : 02,03                  TYPE    : PITR                      LOGNUM : 02,03      ICNUM : 02,03
ERROR   : CONT                   CHECK    : Yes
RCVTOKEN: DRF                    TIME FMT: LCL

DRFPROC : TESTDRF                LBI      : No                      SPSIZE  :      1000

----- Sort-Related Parameters -----
NUM      :      9                MAINSIZE:      5                      AVGRLEN :      10000
FILSZ    :    30000              HIPRMAX : OPTIMAL                    ASPREF  :      IDRF
DYNALLOC:      3 , N/A

  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   S U M M A R Y   R E P O R T

Database DD/Area DSID      Records Read  Records Subord. Final
Name     Name     IC      CA      LOG   Written Reg Name  Status
-----
DHVNTZ02 HIDAM     1        49      0    2043     49  DRFI0001  No errors encountered
DHVNTZ02 HIDAM2    2        49      0      0     49  DRFI0002  No errors encountered
DIVNTZ02 DBHVSAM1  1         3      0      0      3  DRFI0003  No errors encountered
DIVNTZ02 DBHVSAM2  2         4      0    210     4  DRFI0001  No errors encountered
DXVNTZ02 XDLBT04I  1         4      0      0      4  DRFI0002  No errors encountered
FRD0000I  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   05/08/2020 10:19                                     Page 1

  D A T A B A S E   R E C O V E R Y   F A C I L I T Y   D A T A   S E T   I / O   R E P O R T

Recover to point: 2020.128 17:16:51.000000

Image Copy Data Set Name      Volume  IC DS  IC  ----- Time Stamp Range -----
-----,-----,-----,----- Serial  Read Count  Type  1st Record      Last Record

IMSVS.DIVNTZ02.DBHVSAM1.IC.IC101443  222222      3  STD
IMSVS.DHVNTZ02.HIDAM2.IC.IC101518    222222     49  STD
IMSVS.DHVNTZ02.HIDAM.IC.IC101518     222222     49  STD
IMSVS.DIVNTZ02.DBHVSAM2.IC.IC101443  222222      4  STD
IMSVS.DXVNTZ02.XDLBT04I.IC.IC101552  222222      4  STD

Change Accum Data Set Name      Volume  CA DS  ----- Time Stamp Range -----
-----,-----,-----,----- Serial  Read Count  1st Record      Last Record

IMSVS.CAGRP2.CA.CA022201          222222      23
IMSVS.CAGRP1.CA.CA022501          222222      24
```

| Log Data Set Name                                                                         |                           | Volume | Log DS           | IMS      | Time Stamp Range                    |                     |
|-------------------------------------------------------------------------------------------|---------------------------|--------|------------------|----------|-------------------------------------|---------------------|
| -----                                                                                     |                           | Serial | Read Count       | SYSID    | 1st Record                          | Last Record         |
| IMSVS.RLDSP.IMS1.D20128.T1013276.V00                                                      |                           | 000000 | 2537             | IMS1     | 2020.128 17:13:27.6                 | 2020.128 17:17:14.7 |
| Prilog: 2020.128 17:13:27.6                                                               |                           |        |                  |          |                                     |                     |
| FRD0000I D A T A B A S E R E C O V E R Y F A C I L I T Y                                  |                           |        | 05/08/2020 10:19 |          |                                     | Page 1              |
| D A T A B A S E R E C O V E R Y F A C I L I T Y P I T R O P E N U O W / U O R R E P O R T |                           |        |                  |          |                                     |                     |
| RCVTIME: 2020.128 17:16:51.000000 -07:00                                                  |                           |        |                  |          |                                     |                     |
| PSB                                                                                       | ----- Recovery Token----- | Prior  | Database         |          | Duration of                         | Log Record          |
|                                                                                           | Subsys #Sched #Commits    | Chkpid | Name             | DDname   | Open UOW                            | Count               |
| PLVAPZ12                                                                                  | IMS1 00000001 00000000    | N/A    | DXVNTZ02         | XDLBT04I | 2020.128 17:16:30.4 00:00:20.585240 | 600                 |
|                                                                                           |                           |        | DHVNTZ02         | HIDAM    |                                     | 3426                |

Below is an explanation of the Batch Invocation of a PITR example. There are 5 sections to the report:

#### • **COMMANDS / CONTROL STATEMENTS**

This section displays the images of the control statements read from the SYSIN file:

- Two control statements were read from the SYSIN file.
- Recovery was requested for three databases.
- The ERROR(CONT), READNUM(2), RCVTIME('20128101651' PITR) parameters were specified.

**Note:** No time zone was included in the RCVTIME; in this case, the RECTIME is assumed to be local time.

#### • **RECOVERY PARAMETERS**

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

1. The IMS Database Recovery Facility defaults.
2. Explicitly coded in the FRXDRFxx member.
  - Although the values explicitly coded in the FRXDRFxx member do not appear in the report, these values are associated with the FRD7201I messages available in the SYSPRINT output from the recovery.
3. Explicitly coded in the SYSIN control statements.
  - These values can be seen in the "Control Statements" part of the report:

##### **PROCESS: RCVDBDS**

A DBDS-type recovery was selected.

- This parameter was not specified. The recovery facility default parameter was used.

##### **SOURCE: PRI**

The primary log and image copy data sets are to be used as the source of input for the recovery.

- This parameter was not specified. The recovery facility default parameter was used.

##### **READNUM: 02,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 02 can be tape devices.

- The maximum number of total devices was specified in the FRXDRFxx member. The number of allowable tape devices was specified in the SYSIN control statements.

##### **ERROR: CONT**

Should an error be encountered on one database during recovery, processing should continue for other databases.

- This parameter was not specified. The recovery facility default parameter was used.

##### **RCVTOKEN: DRF**

The recovery token used during recovery is "DRF."

- This parameter was not specified. The recovery token used during a batch recovery is the job name.

**RCVTIME: 2020.128 17:16:51.000000 -07:00**

A recovery time of 2020.128 17:16:51.000000 -07:00 is indicated.

- It has been converted to UTC from the initial specification of 20128101651.

**TYPE: PITR**

A point-in-time recovery was requested.

**CHECK: YES**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDSs within that recovery group need to be included in the list.

- This parameter was not specified. The recovery facility default parameter was used.

• **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database.

The three databases included in the recovery consisted of 5 database data sets. For each of these data sets, the following information is listed:

**Database Name**

The database name is listed.

**DD/AREA Name**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. Had any fast path databases been included, the associated Area name or names would have been listed.

**DSID**

The data set id number is listed.

**Records Read**

The number of records read from the image copy, change accumulation and log data sets for application in the recovery are listed. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg Name**

The name of the started task which was responsible for sorting the records being written for the indicated data set is listed.

**Final Status**

The status of the conclusion of the recovery is listed.

• **DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to Point" in this section.

The data set I/O report is divided into three sections:

**1. Image Copy**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set is listed. If Flash Copy, the read count will be 'N/A'.

**IC Type**

The image copy type is listed. Allowable values for this field include:

**STD**

Standard Image Copy

**IC2**

Image Copy 2

**IC EXT**

Extended Image Copy

**HISAM UNL**

HISAM Unload

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**2. Change Accum**

For each change accum data set used during the recovery, the following information is listed:

**Change Accum Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set is listed.

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**3. Log**

For each log data set used during the recovery, the following information is listed:

**Log Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set is listed.

**IMS SYSID**

The SYSID associated with the IMS which created the log is listed.

**Time Stamp Range**

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

**PRILOG:**

The PRILOG time listed in RECON for the log is listed.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.

**• PITR OPEN UOW / UOR REPORT**

This report includes information about PSBs for which units of work were active at the time selected for the point-in-time recovery. The PSBs might be divided into the following two categories:

- PSBs for long records were encountered signifying that a unit of work was active at the recovery time that included updates to the databases being recovered. Furthermore, because the unit of work had not reached a sync point at the recovery point in time, the log records are not applied to the database, nor are they included in the report statistics. In these cases, the report will include information identifying the databases and the active unit of recovery for this category of PSB.
- PSBs which were active scheduled at the recovery time, but for which no log records were found indicating activity related to the recovered databases in the active unit of recovery. These PSBs are

included because the IMS Database Recovery Facility cannot determine that any database activity has occurred after the point of recovery in the subject unit of recovery, or in subsequent units of recovery. In these cases, the report will include information identifying the active unit of recovery, but will not have any information for any associated databases. All columns in the report pertaining to database information will contain the value N/A.

The following information is included:

**RCVTIME:**

As was seen in earlier sections of the report, a point-in-time recovery was specified. The Recover to Point is listed in this section.

**PSB**

The PSBs are listed which contain activity related to the databases being recovered. For each PSB, there might be multiple columns, one column per database in the recovery list.

For each PSB, the following are listed:

- The 16 character Recovery Token for the PSB is listed. The recovery token consists of three subparts:

**Subsys**

The Subsystem ID.

**#Sched**

The Schedule Number for this particular PSB.

**#Commits**

The number of commits represented by this recovery token.

- In this example, the number of commits is zero, which indicates that this recovery token represents the beginning of the program execution. Had the program taken syncpoints, the #Commits value would reflect the number of syncpoints taken.

**Prior Chkpid**

The checkpoint id which is associated with the recovery token.

- Since the recovery token represents the beginning of execution and not a syncpoint, there is no Chkpid present.

**Time Opened**

This value represents the time at which this unit of recovery began.

- In this example, Time opened represents the beginning of program execution. Had checkpoints been taken, the Time opened would indicate the time that the Prior Chkpid had been taken.

**Duration of Open UOW**

This value represents the elapsed time between the Time Opened and the Recover to point time.

For each database within the PSB, the following is listed:

**Database Name**

The database name is listed.

**DDname**

The DDname (or Area name) associated with the database is listed.

**Log Record Count**

The number of log records for this database which were read, but not used as input to the recovery is listed.



## Example report: Full point-in-time recovery

This report example shows the batch invocation of full PITR (point-in-time recovery) of both the production and duplicate database.

FRD0000I DATABASE RECOVERY FACILITY 11/22/2020 18:59 Page 1

### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I DBDSN001('NEWDS.DBOVLFP.C.TESTDUP')
FRD7201I DBDSN002('IMSTEST')
FRD7201I DBDSN003('IMSTEST',,%,DATE)
FRD7201I DBDSL001(DISP(OLD))
FRD7201I DBDSL002(DISP(NEW),DBN(MYDB1))
FRD7201I OUTPUT(BOTH)
FRD7201I LCLTIME(Y)
FRD7201I ADD DB(DBOVLFP.C) DBATRB(DBDSN(001) DBDSL(001))
FRX7201I ADD AREA(DEDBDD01 DD01AR0) DBATRB(DBDSN(002) DBDSL(002))
FRX7201I ADD AREA(DEDBJN22 DB22AR2) DBATRB(DBDSN(003) DBDSL(002))
FRX7201I START ERROR(STOP) RCVTIME('2020.056 10:57:28.00')
```

### DATABASE RECOVERY FACILITY RECOVERY PARAMETERS

```
PROCESS : RCVBOTH
SOURCE : PRI
READNUM : 03,03
ERROR : STOP
RCVTOKEN: DRF1
DRFPROC : DRFS
RCVTIME : 2020.056 10:57:28.00
TYPE : TSR
CHECK : Yes
TIME FMT : LCL
LBI : No
RCVCOPY OPTIONS
LOGNUM : 03,03
ICNUM : 03,03
OUTPUT : Pro,Dup
SPSIZE : 1024
CACHE : No
```

### ----- Sort-Related Parameters -----

```
NUM : 1
FILSZ : 21000
DYNALLO: SYSDALL , N/A
MAINSIZE: 30
HIPRMAX : OPTIMAL
AVGRLN : 256
ASPREF : DRFI
```

### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | IC   | Records Read CA | Log  | Records Written | Subord. Reg Name | Final Status           |
|---------------|--------------|------|------|-----------------|------|-----------------|------------------|------------------------|
| DBOVLFP.C     | VLOSAM01     | 1    | 1001 | 11              | 1006 | 1001            | DRFS0001         | Delete/define complete |
| DEBDD01       | DD01AR0      | 1    | 72   | 5               | 211  | 5               | DRFS0002         | Delete/define complete |
| DEBDBJN22     | DB22AR2      | 3    | 15   | 2               | 911  | 725             | DRFS0003         | Delete/define complete |

FRD0000I DATABASE RECOVERY FACILITY 11/22/2020 18:59 Page 2

### DATABASE RECOVERY FACILITY RCV COPY REPORT

| Database Name | DD / Area Name | New Data Set Name        | Records copied | Volume Serial |
|---------------|----------------|--------------------------|----------------|---------------|
| DBOVLFP.C     | VLOSAM01       | NEWDS.DBOVLFP.C.TESTDUP  | 1001           | 222222        |
| DEBDD01       | DD01AR0        | IMSTEST.DD01AR0          | 5              | 222222        |
| DEBDBJN22     | DB22AR2        | IMSTEST.DB00AR2.D2020056 | 725            | 222222        |

FRD0000I DATABASE RECOVERY FACILITY 11/22/2020 18:59

### DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: 2020.056 10:57:28.000000

| Image Copy Data Set Name             | Volume Serial | IC DS Read Count | IC Type | Time Stamp Range       |
|--------------------------------------|---------------|------------------|---------|------------------------|
|                                      |               |                  |         | 1st Record Last Record |
| IMSVS.DBOVLFP.C.VLOSAM01.IC.IC105637 | 333333        | 1001             | STD     |                        |
| IMSVS.DEBDD01.DD01AR0.IC.IC105637    | 333333        | 72               | IC EXT. |                        |
| IMSVS.DEBDBJN22.DB22AR2.IC.IC105627  | 333333        | 15 T             | IC2     |                        |

| Change Accum Data Set Name | Volume Serial | CA DS Read Count | Time Stamp Range                        |
|----------------------------|---------------|------------------|-----------------------------------------|
|                            |               |                  | 1st Record Last Record                  |
| IMSVS.CADSN.D20056.V00     | 000000        | 18               | 2020.056 10:56:04.6 2020.056 10:59:08.3 |

| Log Data Set Name                    | Volume Serial | Log DS Read Count | IMS SYSID | Time Stamp Range                        |
|--------------------------------------|---------------|-------------------|-----------|-----------------------------------------|
|                                      |               |                   |           | 1st Record Last Record                  |
| IMSVS.SLDSP.IMS1.D20056.T1056046.V00 | 000000        | 1222              | IMS1      | 2020.056 10:56:04.6 2020.056 10:59:08.3 |
| Prilog: 2020.056 10:56:04.6          |               |                   |           |                                         |
| IMSVS.RLDSP.IMS1.D20056.T1056046.V00 | 000000        | 106               | IMS1      | 2020.056 10:56:04.6 2020.056 10:59:08.3 |
| Prilog: 2020.056 10:56:04.6          |               |                   |           |                                         |

Below is an explanation of the batch invocation of full recovery example. There are 4 sections to the report:

#### • **COMMANDS / CONTROL STATEMENTS**

This section displays the images of the control statements read from the SYSIN file:

- Two control statements were read from the SYSIN file.
- Recovery was requested for three databases.
- The ERROR(CONT) parameter was specified.

#### • **RECOVERY PARAMETERS**

This section lists the parameters used to direct the recovery. The recovery parameters were supplied through one of the following three mechanisms, in ascending order of selection priority:

1. The IMS Database Recovery Facility defaults.
2. Explicitly coded in the FRXDRExx member.
  - Although the values explicitly coded in the FRXDRExx member do not appear in the report, these values are associated with the FRD7201I messages available in the SYSPRINT output from the recovery.
3. Explicitly coded in the SYSIN control statements.
  - These values can be seen in the "Control Statements" part of the report:

**PROCESS: RCVBOTH**

A DBDS-type recovery was selected.

- This parameter was not specified. The recovery facility default parameter was used.

**SOURCE: PRI**

The primary log and image copy data sets are to be used as the source of input for the recovery.

- This parameter was not specified. The recovery facility default parameter was used.

**READNUM: 03,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 03 can be tape devices.

- The maximum number of total devices and the number of allowable tape devices were specified in the FRXDRExx member.

**ERROR: CONT**

Should an error be encountered on one database during recovery, processing should continue for other databases.

- This parameter was not specified. The recovery facility default parameter was used.

**RCVTOKEN: DRFI**

The recovery token used during recovery is "DRF."

- This parameter was not specified. The recovery token used during a batch recovery is the job name.

**RCVTIME: Not Specified**

A recovery time was not specified.

- There is no default for the recovery time.

**TYPE: Full**

A full recovery was requested.

- This parameter was not specified. The recovery facility default parameter was used.

**CHECK: Yes**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDSs within that recovery group need to be included in the list.

- This parameter was not specified. The recovery facility default parameter was used.

**LCLTIME: Yes**

Local time stamps will be used in messages and reports.

• **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database.

The three databases included in the recovery consisted of 5 database data sets. For each of these data sets, the following information is listed:

**Database Name**

The database name is listed.

**DD/Area Name**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. Had any fast path databases been included, the associated Area name or names would have been listed.

**DSID**

The data set id number is listed.

**Records Read**

The number of records read from the image copy, change accumulation, and log data sets for application in the recovery are listed. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg name**

The name of the started task which was responsible for sorting the records being written for the indicated data set is listed.

**Final Status**

The status of the conclusion of the recovery is listed.

**• DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to Point" in this section.

The data set I/O report is divided into three sections:

**1. Image Copy**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set is listed.

**IC Type**

The image copy type is listed. Allowable values for this field include:

**STD**

Standard Image Copy

**IC2**

Image Copy 2

**IC EXT**

Extended Image Copy

**HISAM UNL**

HISAM Unload

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**2. Change Accum**

For each change accum data set used during the recovery, the following information is listed:

**Change Accum Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

### Read Count

The number of records read from the data set is listed. If Flash Copy, the read count will be 'N/A'.

### Time Stamp Range

Although this heading appears, this field has no significance for change accumulation data sets.

## 3. Log

For each log data set used during the recovery, the following information is listed:

### Log Data Set Name

The data set name is listed.

### Volume Serial

The volume or volumes upon which the data sets were found.

### Read Count

The number of records read from the data set is listed.

### IMS SYSID

The SYSID associated with the IMS which created the log is listed.

### Time Stamp Range

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

### PRILOG:

The PRILOG time listed in RECON for the log is listed.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.

## Example report: Verify of the point-in-time recovery

This report example below shows batch invocation of a verify of the Pitr (point-in-time recovery).

```
FRD0000I  DATABASE  RECOVERY  FACILITY    05/08/2020 10:19                      Page 1

  DATABASE  RECOVERY  FACILITY  COMMANDS / CONTROL  STATEMENTS

FRD7201I ADD DB(DIVNTZ02 DHVNTZ02 DXVNTZ02)
FRD7201I START ERROR(CONT) READNUM(2) RCVTIME('20128101651' PITR) VERIFY

  DATABASE  RECOVERY  FACILITY  RECOVERY  PARAMETERS

PROCESS : VERIFY                RECOVERY OPTIONS
SOURCE  : PRI                   RCVTIME  : 2020.128 17:16:51.000000 -07:00
READNUM : 02,03                 TYPE     : PITR
ERROR   : CONT                  CHECK    : Yes
RCVTOKEN: DRF
DRFPROC : TESTDRF              LBI      : No                      SPSIZE   : 1000

----- Sort-Related Parameters -----
NUM      : 9                    MAINSIZE: 5                      AVGRLEN : 10000
FILSZ    : 30000                HIPRMAX : OPTIMAL                ASPREF  : IDRF
DYNALLOC : 3 , N/A

  DATABASE  RECOVERY  FACILITY  SUMMARY  REPORT

Database Name  DD/Area Name  DSID  IC  Records Read  LOG  Records Written  Subord. Reg Name  Final Status
DHVNTZ02      HIDAM         1        N/A  N/A          N/A  N/A            N/A            Verify list complete
DHVNTZ02      HIDAM2        2        N/A  N/A          N/A  N/A            N/A            Verify list complete
DIVNTZ02      DBHVSAM1      1        N/A  N/A          N/A  N/A            N/A            Verify list complete
DIVNTZ02      DBHVSAM2      2        N/A  N/A          N/A  N/A            N/A            Verify list complete
DXVNTZ02      XDLBT04I      1        N/A  N/A          N/A  N/A            N/A            Verify list complete
FRD0000I  DATABASE  RECOVERY  FACILITY    05/08/2020 10:19                      Page 1

  DATABASE  RECOVERY  FACILITY  DATA  SET  I / O  REPORT

Recover to point: 2020.128 17:16:51.000000

Image Copy Data Set Name  Volume Serial  IC DS  IC Type  Time Stamp Range
-----
IMSVS.DHVNTZ02.HIDAM.IC.IC101518 222222 0 STD 1st Record Last Record
IMSVS.DHVNTZ02.HIDAM2.IC.IC101518 222222 0 STD
IMSVS.DIVNTZ02.DBHVSAM1.IC.IC101443 222222 0 STD
```

```

IMSVS.DIVNTZ02.DBHVSAM2.IC.IC101443      222222      0      STD
IMSVS.DXVNTZ02.XDLBT04I.IC.IC101552      222222      0      STD

Change Accum Data Set Name                Volume   CA DS   ----- Time Stamp Range -----
-----,-----,-----,-----,----- Serial   Read Count   1st Record   Last Record

No data available for this type data set

Log Data Set Name                        Volume   Log DS   IMS   ----- Time Stamp Range -----
-----,-----,-----,-----,----- Serial   Read Count   SYSID  1st Record   Last Record

IMSVS.SLDSP.IMS1.D20128.T1013276.V00      000000      0   IMS1  2020.128 17:13:27.6  2020.128 17:17:14.7
Prilog: 2020.128 17:13:27.6
IMSVS.RLDSP.IMS1.D20128.T1013276.V00      000000      0   IMS1  2020.128 17:13:27.6  2020.128 17:17:14.7
Prilog: 2020.128 17:13:27.6
FRD0000I D A T A B A S E   R E C O V E R Y   F A C I L I T Y   05/08/2020 10:19                      Page 1

D A T A B A S E   R E C O V E R Y   F A C I L I T Y   P I T R   O P E N   U O W / U O R   R E P O R T

RCVTIME: 2020.128 17:16:51.000000 -07:00

PSB      ----- Recovery Token-----   Prior   Database   Duration of   Log Record
Subsys   #Sched   #Commits   Chkpid   Name        DDname    Time Opened   Open UOW     Count

No open UOWs encountered for this recovery

```

Below is an explanation of the Batch Invocation of a Verify of the PITR example. There are 5 sections to the report:

### • **COMMANDS / CONTROL STATEMENTS**

This section displays the images of the control statements read from the SYSIN file.

Two control statements were read from the SYSIN file. Recovery was requested for three databases. The ERROR(CONT), READNUM(2), RCVTIME('20128101651' PITR), VERIFY parameters were specified.

**Note:** No time zone was included in the RCVTIME; in this case, the RECTIME is assumed to be local time.

### • **RECOVERY PARAMETERS**

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

1. The IMS Database Recovery Facility defaults
2. Explicitly coded in the FRXDRFxx member.
  - Although the values explicitly coded in the FRXDRFxx member do not appear in the report, these values are associated with the FRD7201I messages available in the SYSPRINT output from the recovery.
3. Explicitly coded in the SYSIN control statements.
  - These values can be seen in the "Control Statements" part of the report:

#### **PROCESS: VERIFY**

A VERIFY was selected.

- This parameter was not specified. The recovery facility default parameter was used.

#### **SOURCE: PRI**

The primary log and image copy data sets are to be used as the source of input for the recovery.

- This parameter was not specified. The recovery facility default parameter was used.

#### **READNUM: 02,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 02 can be tape devices.

- The maximum number of total devices was specified in the FRXDRFxx member. The number of allowable tape devices was specified in the SYSIN control statements.

#### **ERROR: CONT**

Should an error be encountered on one database during recovery, processing should continue for other databases.

- This parameter was not specified. The IMS Database Recovery Facility default parameter was used.

**RCVTOKEN: DRF**

The recovery token used during recovery is "DRF."

- This parameter was not specified. The recovery token used during a batch recovery is the job name.

**RCVTIME: 2020.128 17:16:51.000000 -07:00**

A recovery time of 2020.128 17:16:51.000000 -07:00 is indicated.

- It has been converted to UTC from the initial specification of 20128101651.

**TYPE: PITR**

A PITR was requested.

**CHECK: Yes**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDSs within that recovery group need to be included in the list.

- This parameter was not specified. The recovery facility default parameter was used.

• **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database.

The three databases included in the recovery consisted of 5 database data sets. For each of these data sets, the following information is listed:

**Database Name**

The database name is listed.

**DD/Area**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. If any fast path databases had been included, the associated Area name or names would have been listed.

**DSID**

The data set id number is listed.

Because this invocation of the recovery is a "verify," the image copy, change accumulation and log data sets were never actually opened or processed. Consequently, a value of N/A is listed for those entries which are based upon actual data processing, such as:

**Records Read**

The number of records read. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg Name**

The name of the started task responsible sorting the records.

**Final Status**

The status of the conclusion of the recovery.

• **DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to point" in this section.

The data set I/O report is divided into three sections:

**1. Image Copy**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set is zero. If Flash Copy, the read count will be 'N/A'.

**IC Type**

The image copy type is listed. Allowable values for this field include:

**STD**

Standard Image Copy

**IC2**

Image Copy 2

**IC EXT**

Extended Image Copy

**HISAM UNL**

HISAM Unload

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**2. Change Accum**

For each change accumulation data set used during the recovery, the following information is listed:

**Change Accum Data Set Name**

No change accum data sets were used.

**Volume Serial**

This field is intentionally blank.

**Read Count**

This field is intentionally blank.

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**3. Log**

For each log data set used during the recovery, the following information is listed:

**Log Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set is zero.

**IMS SYSID**

The SYSID associated with the IMS which created the log is listed.

**Time Stamp**

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

**Prilog:**

The PRILOG time listed in RECON for the log is listed.

Because no processing has occurred, the recovery has not determined whether the RLDS or the SLDS is available for use. Both are listed in the report.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.

• **PITR OPEN UOW / UOR REPORT**

Because no log records have been read, there is no data available for this report.

## Example report: VERIFY(ALLOC) recovery

This report example shows the batch invocation of a VERIFY(ALLOC) recovery.

```
FRD0000I  DATABASE RECOVERY FACILITY  11/22/2020 18:59                                     Page 1

DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

FRD7201I OUTPUT(PRO)
FRD7201I LCLTIME(Y)
FRD7201I ADD DB(DHVNTZ02 DIVNTZ02)
FRX7201I ADD DBDS(DXVNTZ02 XDLBT04I)
FRX7201I START VERIFY(ALLOC) ERROR(STOP)

DATABASE RECOVERY FACILITY RECOVERY PARAMETERS

PROCESS : VERIFY(ALLOC)          RECOVERY OPTIONS
SOURCE  : PRI                    RCVTIME  : Not Specified
READNUM : 03,03                  TYPE     : Full
ERROR   : STOP                   CHECK    : Yes
RCVTOKEN: DRF1                   LCLTIME  : Yes
DRFPROC : DRFS                   LBI     : No
                                OUTPUT   : Pro
                                SPSIZE   : 1024
                                CACHE    : No

----- Sort-Related Parameters -----
NUM      : 1                      MAINSIZE: 30
FILSZ    : 21000                  HPRMAX  : OPTIMAL
DYNALLOC : SYSDDL , N/A          AVGRLEN : 256
                                ASPREF   : DRFI

DATABASE RECOVERY FACILITY SUMMARY REPORT

Database DD/Area DSID      Records Read  Records Subord. Final
Name      Name      ID      IC      CA      Log   Written Reg Name  Status
-----
DHVNTZ02  HIDAM      1      N/A     N/A     N/A     N/A     N/A     Verify alloc failure
DHVNTZ02  HIDAM2     2      N/A     N/A     N/A     N/A     N/A     Verify alloc complete
DIVNTZ02  DBHVSAM1   1      N/A     N/A     N/A     N/A     N/A     Verify alloc complete
DIVNTZ02  DBHVSAM2   2      N/A     N/A     N/A     N/A     N/A     Verify alloc complete
DXVNTZ02  XDLBT04I   N/A     N/A     N/A     N/A     N/A     N/A     Verify alloc complete
FRD0000I  DATABASE RECOVERY FACILITY  11/22/2020 18:59                                     Page 2

DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: Not Specified

Image Copy Data Set Name      Volume IC DS IC  Time Stamp Range
----- Serial Read Count Type 1st Record Last Record
IMSVS.DHVNTZ02.HIDAM.IC.IC105637 333333 0 STD ALLOC RC = 4
IMSVS.DHVNTZ02.HIDAM2.IC.IC105637 333333 0 STD ALLOC RC = 0
IMSVS.DIVNTZ02.DBHVSAM1.IC.IC105627 333333 0 STD ALLOC RC = 0
IMSVS.DIVNTZ02.DBHVSAM2.IC.IC105627 333333 0 STD ALLOC RC = 0
IMSVS.DXVNTZ02.XDLBT04I.IC.IC105647 333333 0 STD ALLOC RC = 0

Change Accum Data Set Name      Volume CA DS  Time Stamp Range
----- Serial Read Count 1st Record Last Record
No data available for this type data set

Log Data Set Name      Volume Log DS IMS  Time Stamp Range
----- Serial Read Count SYSID 1st Record Last Record
IMSVS.SLDSP.IMS1.D20056.T1056046.V00 000000 1222 IMS1 2020.056 10:56:04.6 2020.056 10:59:08.3
Prilog: 2020.056 10:56:04.6
IMSVS.RLDSP.IMS1.D20056.T1056046.V00 000000 106 IMS1 2020.056 10:56:04.6 2020.056 10:59:08.3
Prilog: 2020.056 10:56:04.6
```

Below is an explanation of the Batch Invocation of a VERIFY(ALLOC) recovery. There are 4 sections to the report:

- **COMMANDS / CONTROL STATEMENTS**

This section displays the images of the control statements read from the SYSIN file.

Five control statements were read from the SYSIN file. Recovery was requested for four databases. The VERIFY(ALLOC) and ERROR(STOP) parameters were specified.

- **RECOVERY PARAMETERS**

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

1. The IMS Database Recovery Facility defaults
2. Explicitly coded in the FRXDRFxx member.



- Although the values explicitly coded in the FRXDRFxx member do not appear in the report, these values are associated with the FRD7201I messages available in the SYSPRINT output from the recovery.

### 3. Explicitly coded in the SYSIN control statements.

- These values can be seen in the "Control Statements" part of the report:

#### **PROCESS: VERIFY(ALLOC)**

A VERIFY(ALLOC) was selected.

- This parameter was not specified. The recovery facility default parameter was used.

#### **SOURCE: PRI**

The primary log and image copy data sets are to be used as the source of input for the recovery.

- This parameter was not specified. The recovery facility default parameter was used.

#### **READNUM: 03,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 03 can be tape devices.

- The maximum number of total devices and the number of allowable tape devices were specified in the FRXDRFxx member.

#### **ERROR: STOP**

Should an error be encountered during recovery, processing should terminate.

#### **RCVTOKEN: DRF1**

The recovery token used during recovery is "DRF1."

- This parameter was not specified. The recovery token used during a batch recovery is the job name.

#### **RCVTIME: Not specified**

A recovery time was not specified.

#### **TYPE: Full**

A full recovery was requested.

#### **CHECK: Yes**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDSs within that recovery group need to be included in the list.

- This parameter was not specified. The recovery facility default parameter was used.

### • **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database.

The three databases included in the recovery consisted of 5 database data sets. For each of these data sets, the following information is listed:

#### **Database Name**

The database name is listed.

#### **DD/Area**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. If any fast path databases had been included, the associated Area name or names would have been listed.

#### **DSID**

The data set id number is listed.

Because this invocation of the recovery is a "verify," the image copy, change accumulation and log data sets were never actually opened or processed. Consequently, a value of N/A is listed for those entries which are based upon actual data processing, such as:

**Records Read**

The number of records read. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg Name**

The name of the started task responsible sorting the records.

**Final Status**

The status of the conclusion of the recovery.

**• DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to point" in this section.

The data set I/O report is divided into three sections:

**1. Image Copy**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

A value of N/A is issued in the field if CATDS(Y) and VERIFY(ALLOC) or VERIFY(OPEN) are specified.

**Read Count**

The number of records read from the data set is zero.

**IC Type**

The image copy type is listed. Allowable values for this field include:

**STD**

Standard Image Copy

**IC2**

Image Copy 2

**IC EXT**

Extended Image Copy

**HISAM UNL**

HISAM Unload

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**2. Change Accum**

For each change accumulation data set used during the recovery, the following information is listed:

**Change Accum Data Set Name**

No change accum data sets were used.

**Volume Serial**

This field is intentionally blank.

**Read Count**

This field is intentionally blank.

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**3. Log**

For each log data set used during the recovery, the following information is listed:

### Log Data Set Name

The data set name is listed.

### Volume Serial

The volume or volumes upon which the data sets were found.

A value of N/A is issued in the field if CATDS(Y) and VERIFY(ALLOC) or VERIFY(OPEN) are specified.

### Read Count

The number of records read from the data set is zero. If Flash Copy, the read count will be 'N/A'.

### IMS SYSID

The SYSID associated with the IMS which created the log is listed.

### Time Stamp

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

### Prilog:

The PRILOG time listed in RECON for the log is listed.

Because no processing has occurred, the recovery has not determined whether the RLDS or the SLDS is available for use. Both are listed in the report.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.

## Example report: VERIFY(OPEN) recovery for a point-in-time

This output report example shows a VERIFY(OPEN) recovery for PITR.

One of the image copies was successfully allocated and opened. The other allocate and open attempt experienced an OPEN failure with the return code of 8.

```
DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS
FRD7201I ADD DBDS(PDHDOKA PDHDOKAM PDHDOKA PDHDOKAN)
FRD7201I START ERROR(CONT) VERIFY(OPEN) READNUM(2) RCVTIME('20203203957' PITR)

DATABASE RECOVERY FACILITY RECOVERY PARAMETERS
PROCESS : VERIFY(OPEN)          RECOVERY OPTIONS
SOURCE  : PRI                   RCVTIME : 2020.203 20:39:57.000000 -07:00
READNUM : 02,03                 TYPE    : PITR
ERROR   : CONT                  CHECK   : Yes
RCVTOKEN: DRF
DRFPROC : DRFS                  LBI     : No                SPSIZE  : 1024        CACHE  : No

----- Sort-Related Parameters -----
NUM      : 3                    MAINSIZE: 30                AVGRLEN : 256
FILSZ    : 21000                HIPRMAX : OPTIMAL          ASPREF  : DRFI
DYNALLOC : N/A , N/A

DATABASE RECOVERY FACILITY SUMMARY REPORT
Database DD/Area DSID          IC Records Read LOG Records Subord. Final
Name      Name   ID          Name CA  Count CA   Type Written Reg Name Status
-----
PDHDOKA   PDHDOKAM 1          0      0    61468 1355 DRFI0001 Verify open complete
PDHDOKA   PDHDOKAN 4          0      0    36391 1483 DRFI0002 Verify open failure
.
.
.

Image Copy Data Set Name          Volume IC DS IC  Time Stamp Range
----- Serial Read Count Type 1st Record Last Record
IMSVS.PDHDOKA.PDHDOKAM.IC.IC2027002 222222 OPEN RC = 0
IMSVS.PDHDOKA.PDHDOKAN.IC.IC2027002 222222 OPEN RC = 8

Change Accum Data Set Name          Volume Log DS IMS  Time Stamp Range
----- Serial Read Count SYSID 1st Record Last Record
IMSVS.CADS.IMS1.D20203.T2025002.V01 SCR04 OPEN RC = 0
Prilog: 2020.203 20:25:58.9

Log Data Set Name          Volume Log DS IMS  Time Stamp Range
----- Serial Read Count SYSID 1st Record Last Record
```

This next report example below shows the same VERIFY(OPEN) recovery for Pitr. This time the log data set receives an ALLOC failure with a return code of 12. Notice how that forces each database data set Final Status to be marked, Verify alloc failure.

#### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I ADD DBDS(PDHDOKA PDHDOKAM PDHDOKA PDHDOKAN)
FRD7201I START ERROR(CONT) VERIFY(OPEN) READNUM(2) RCVTIME('20203203957' Pitr)
```

#### DATABASE RECOVERY FACILITY RECOVERY PARAMETERS

```
PROCESS : VERIFY(OPEN)          RECOVERY OPTIONS
SOURCE  : PRI                   RCVTIME : 2020.203 20:39:57.000000 -07:00
READNUM : 02,03                 TYPE    : Pitr
ERROR   : CONT                  CHECK   : Yes
RCVTOKEN: DRF
DRFPROC : DRFS                  LBI     : No                      SPSIZE  : 1024          CACHE  : No
```

#### ----- Sort-Related Parameters -----

```
NUM      : 3                     MAINSIZE: 30                     AVGRLEN : 256
FILSZ    : 21000                 HIPRMAX : OPTIMAL                ASPREF  : DRFI
DYNALLOC : N/A , N/A
```

#### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | IC | Records | Read CA | LOG  | Records Written | Subord. Reg Name     | Final Status |
|---------------|--------------|------|----|---------|---------|------|-----------------|----------------------|--------------|
| PDHDOKA       | PDHDOKAM     | 1    | 0  | 0       | 61468   | 1355 | DRFI0001        | Verify alloc failure |              |
| PDHDOKA       | PDHDOKAN     | 4    | 0  | 0       | 36391   | 1483 | DRFI0002        | Verify alloc failure |              |

| Image Copy Data Set Name            | Volume Serial | IC DS Read Count | IC Type | Time Stamp Range          |
|-------------------------------------|---------------|------------------|---------|---------------------------|
| IMSVS.PDHDOKA.PDHDOKAM.IC.IC2027002 | 222222        |                  |         | 1st Record<br>OPEN RC = 0 |
| IMSVS.PDHDOKA.PDHDOKAN.IC.IC2027002 | 222222        |                  |         | 1st Record<br>OPEN RC = 0 |

| Change Accum Data Set Name                                         | Volume Serial | Log DS Read Count | IMS SYSID | Time Stamp Range          |
|--------------------------------------------------------------------|---------------|-------------------|-----------|---------------------------|
| IMSVS.CADS.IMS1.D20203.T2025002.V01<br>Prilog: 2020.203 20:25:58.9 | SCR04         |                   |           | 1st Record<br>OPEN RC = 0 |

| Log Data Set Name                                                    | Volume Serial | Log DS Read Count | IMS SYSID | Time Stamp Range          |
|----------------------------------------------------------------------|---------------|-------------------|-----------|---------------------------|
| IMSVS.RLDSP. IMS1.D20203.T2027127.V01<br>Prilog: 2020.203 20:25:58.9 | SCR03         |                   |           | 1st Record<br>Last Record |

ALLOC RC = 12

Below is an explanation of the Batch Invocation of a VERIFY(OPEN) recovery. There are 3 sections to the report:

#### • **COMMANDS / CONTROL STATEMENTS**

This section displays the images of the control statements read from the SYSIN file.

Two control statements were read from the SYSIN file. Recovery was requested for four databases. The ERROR(CONT), VERIFY(OPEN), READNUM(2), and RCVTIME('20203203957' Pitr) parameters were specified.

#### • **RECOVERY PARAMETERS**

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

1. The IMS Database Recovery Facility defaults
2. Explicitly coded in the FRXDRFxx member.
  - Although the values explicitly coded in the FRXDRFxx member do not appear in the report, these values are associated with the FRD7201I messages available in the SYSPRINT output from the recovery.
3. Explicitly coded in the SYSIN control statements.
  - These values can be seen in the "Control Statements" part of the report:

#### **PROCESS: VERIFY(OPEN)**

A VERIFY(OPEN) was selected.

- This parameter was not specified. The recovery facility default parameter was used.

**SOURCE: PRI**

The primary log and image copy data sets are to be used as the source of input for the recovery.

- This parameter was not specified. The recovery facility default parameter was used.

**READNUM: 02,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 02 can be tape devices.

- The maximum number of total devices was specified in the FRXDRFxx member. The number of allowable tape devices was specified in the SYSIN control statements.

**ERROR: CONT**

Should an error be encountered on one database during recovery, processing should continue for other databases.

- This parameter was not specified. The IMS Database Recovery Facility default parameter was used.

**RCVTIME: 2020.203 20:39:57.000000**

A recovery time of 2020.203 20:39:57.000000 is indicated.

- It has been converted to UTC from the initial specification of 20203203957.

**TYPE: PITR**

A PITR was requested.

**CHECK: Yes**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDSs within that recovery group need to be included in the list.

- This parameter was not specified. The recovery facility default parameter was used.

**RCVTOKEN: DRF**

The recovery token used during recovery is "DRF."

- This parameter was not specified. The recovery token used during a batch recovery is the job name.

• **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database.

The four databases included in the recovery consisted of two database data sets. For each of these data sets, the following information is listed:

**Database Name**

The database name is listed.

**DD/Area**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. If any fast path databases had been included, the associated Area name or names would have been listed.

**DSID**

The data set id number is listed.

Because this invocation of the recovery is a "verify," the image copy, change accumulation and log data sets were never actually opened or processed. Consequently, a value of N/A is listed for those entries which are based upon actual data processing, such as:

**Records Read**

The number of records read. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

## Subord. Reg Name

The name of the started task responsible sorting the records.

## Final Status

The status of the conclusion of the recovery.

## Example report: Full recovery using environment control statements

This report example below shows the batch invocation of full recovery using SYSIN-supplied environment control statements.

```
FRD0000I  DATABASE RECOVERY FACILITY  10/03/2020 09:24                               Page 1

  DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

FRD7201I  SORTPARM(NUM(9) HIPRMAX(OPTIMAL) MAINSIZE(5) ASPREF(IDRF) -
FRD7201I  AVGLEN(10000) FILSZ(30000) DYNALOC(3))
FRD7201I  READNUM(01,03) DRFPROC(TESTDRF)
FRD7201I  SPSIZE(1000)
FRD7201I  ADD DBDS(POHIDKA POHIDKAA POHIDKC POHIDKCA -
FRD7201I  POHIDKD POHIDKDA)
FRD7201I  START ERROR(STOP) READNUM(2) VERIFY

  DATABASE RECOVERY FACILITY RECOVERY PARAMETERS

PROCESS : VERIFY                RECOVERY OPTIONS
SOURCE  : PRI                   RCVTIME : Not Specified
READNUM : 02,03                 TYPE    : Full
ERROR   : STOP                  CHECK    : Yes
RCVTOKEN: DRF
DRFPROC : TESTDRF              LBI      : No                      SPSIZE   : 1000

----- Sort-Related Parameters -----
NUM      : 9                    MAINSIZE: 5                      AVGLEN  : 10000
FILSZ    : 30000               HIPRMAX : OPTIMAL                ASPREF  : IDRF
DYNALLOC : 3 , N/A

  DATABASE RECOVERY FACILITY SUMMARY REPORT

Database DD/Area DSID      Records Read  Records Subord.. Final
Name     Name      IC      CA      LOG   Written Reg Name  Status
-----
POHIDKA  POHIDKAA   1      N/A    N/A    N/A    N/A    N/A    No errors encountered
POHIDKC  POHIDKCA   1      N/A    N/A    N/A    N/A    N/A    No errors encountered
POHIDKD  POHIDKDA   1      N/A    N/A    N/A    N/A    N/A    No errors encountered
FRD0000I  DATABASE RECOVERY FACILITY  10/03/2020 09:24                               Page 1

  DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: Not Specified

Image Copy Data Set Name      Volume IC DS IC  Time Stamp Range
Serial      Read Count  Type 1st Record      Last Record
-----
IMSVS.POHIDKA.POHIDKAA.IC.IC124722 222222 0 STD
IMSVS.POHIDKC.POHIDKCA.IC.IC124723 222222 0 STD
IMSVS.POHIDKD.POHIDKDA.IC.IC124724 222222 0 STD

Change Accum Data Set Name      Volume CA DS CA  Time Stamp Range
Serial      Read Count  Type 1st Record      Last Record
-----
No data available for this type data set

Log Data Set Name      Volume Log DS IMS  Time Stamp Range
Serial      Read Count SYSID 1st Record      Last Record
-----
IMSVS.RLDSP.IMS1.D20252.T1245071.V00 SCR03 0 IMS1 2020.252 19:45:07.1 2020.252 19:50:07.9
Prilog: 2020.252 19:44:32.1
IMSVS.RLDSP.IMS1.D20252.T1245071.V01 SCR03 0 IMS1 2020.252 19:50:07.9 2020.252 19:50:10.9
Prilog: 2020.252 19:44:32.1
IMSVS.RLDSP.IMS1.D20252.T1245071.V02 SCR03 0 IMS1 2020.252 19:50:10.9 2020.252 19:50:13.5
Prilog: 2020.252 19:44:32.1
IMSVS.RLDSP.IMS1.D20252.T1250135.V01 SCR03 0 IMS1 2020.252 19:50:13.5 2020.252 19:50:44.0
Prilog: 2020.252 19:44:32.1
```

Below is an explanation of the Batch Invocation of Full Recovery Using SYSIN-Supplied Environment Control Statements example:

### • COMMANDS / CONTROL STATEMENTS

This section displays the images of the control statements read from the SYSIN file:

- The first four statements are Environment Control Statements and override comparable parameters included in the FRXDRFxx member.
- The last three statements are Recovery Control Statements.

### • RECOVERY PARAMETERS

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

**PROCESS: VERIFY**

A VERIFY recovery process was selected.

**SOURCE: PRI**

The Primary log and image copy data sets are to be used.

**READNUM: 02,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 02 can be tape devices. These numbers were specified as (3,3) in the FRXDRFxx member, but were overridden to (1,3) by the environment control statements in SYSIN. The number of tapes was further overridden by the START recovery control statement.

**ERROR: STOP**

Should an error be encountered during recovery, processing should terminate.

**RCVTOKEN: DRF**

The recovery token used during recovery is "DRF."

**RCVTIME: Not Specified**

A recovery time was not specified.

**TYPE: Full**

A full recovery was requested.

**CHECK: Yes**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDs within that recovery group need to be included in the list.

**DRFPROC: TESTDRF**

The procedure for the RSS is contained in member TESTDRF in the proclib PDS.

**LB I: No**

Large Block Interface processing was not requested.

**SPSIZE: 1000**

The SPSIZE specified is 1000. This number was specified as 1024 in the FRXDRFxx member, but was overridden by SYSIN.

**NUM: 9**

Num was specified as 9. This number was specified as 3 in the FRXDRFxx member, but was overridden by SYSIN.

**MAINSIZE: 5**

Mainsize was specified as 5. This number was specified as 30 in the FRXDRFxx member, but was overridden by SYSIN.

**AVGRLN: 10000**

Avgrlen was specified as 10000. This number was specified as 256 in the FRXDRFxx member, but was overridden by SYSIN.

**FILSZ: 30000**

Filsz was specified as 30000. This number was specified as 21000 in the FRXDRFxx member, but was overridden by SYSIN.

**HIPRMAX: OPTIMAL**

Hiprmax was specified as Optimal. Both the FRXDRFxx member and SYSIN specified the save value.

**ASPREF: IDRF**

Aspref was specified as IDRF. This value was specified as DRFI in the FRXDRFxx member, but was overridden by SYSIN.

**DYNALLOC: 3 , N/A**

The number of dynamically allocated SORTWORK data sets is 3, as specified in SYSIN. The unit associated with the dynamic allocation was not specified, and is not available for the report. The Sort installation value for unit will be used by the SORT utility.

• **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database:

**Database Name**

The database name is listed.

**DD/Area Name**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. Had any fast path databases been included, the associated Area name or names would have been listed.

**DSID**

The data set id number is listed.

**Records Read**

The number of records read from the image copy, change accumulation and log data sets for application in the recovery are listed. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg Name**

The name of the started task which was responsible for sorting the records being written for the indicated data set is listed.

**Final Status**

The status of the conclusion of the recovery is listed.

• **DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to point" in this section. The data set I/O report is divided into three sections:

**1. Image Copy**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set with intent to be used as input is listed. If Flash Copy, the read count will be 'N/A'.

**IC Type**

The image copy type is listed. Allowable values for this field include:

**STD**

Standard Image Copy

**IC2**

Image Copy 2

**IC EXT**

Extended Image Copy

**HISAM UNL**

HISAM Unload

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**2. Change Accum**

For each change accum data set used during the recovery, the following information is listed:

**Change Accum Data Set Name**

No change accum data sets were used.



**Volume Serial**

This field is intentionally blank.

**Read Count**

This field is intentionally blank.

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**3. Log**

For each log data set used during the recovery, the following information is listed:

**Log Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set with intent to be used as input is listed.

**IMS SYSID**

The SYSID associated with the IMS which created the log is listed.

**Time Stamp Range**

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

**Prilog:**

The PRILOG time listed in RECON for the log is listed.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.

## Example report: Full recovery with IC() and PC() utilities

This report example shows the output from a batch invocation of full recovery with integrated auxiliary processing utilities (IC() and PC() utilities).

### DATABASE RECOVERY FACILITY COMMANDS / CONTROL STATEMENTS

```
FRD7201I XCFGROUP(TOIV11)
FRD7201I SORTPARM(ASPREF(AS))
FRD7201I UTILGBL(IBPREF(IB),PRPREF(PR),PCJOBNM(PC),ICNDX(Y),HALDB(BOTH))
FRD7201I REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=TEMP.QA)
FRD7201I DRFIAX(DRFIB)
FRD7201I ADD DB(F1V4P1 H1V4P1) -
FRD7201I IB (BLD_PRIMARY,BLD_SECONDARY(ALL)) -
FRD7201I IC(COMP(N),-
FRD7201I ICCAT(Y),-
FRD7201I ICNMRULE(Y),-
FRD7201I ICHLQ(TEMP.QA),-
FRD7201I DATACLAS(SMS),-
FRD7201I UNIT(3390),-
FRD7201I SPACE(CYL,10,1,RLSE)) -
FRD7201I PC(DUMPFOR=FORMAT,PRINTDATA=YES)
FRD7201I START READNUM(2)
```

### DATABASE RECOVERY FACILITY RECOVERY PARAMETERS

```
PROCESS : RCVDBDS          RECOVERY OPTIONS
SOURCE  : PRI              RCVTIME : Not Specified
READNUM : 02,03           TYPE      : Full
ERROR   : STOP            CHECK     : Yes
RCVTOKEN: DRFIBJOB        TIME FMT : LCL
DRFPROC : DRFRSS          LBI       : No
LOGNUM  : 02,03           ICNUM     : 02,03
OUTPUT  : Prio            TAPECHK   : No
CATDS   : Default
SPSIZE  : 1024            CACHE     : No

----- Sort-Related Parameters -----
NUM      : 5              MAINSIZE: 30
FILSZ    : 210000         HIPRMAX : OPTIMAL
DYNALLOC : N/A , N/A     AVGRLEN : 256
                                ASPREF : AS
                                ASGNAME : N/A
```

### DATABASE RECOVERY FACILITY SUMMARY REPORT

| Database Name | DD/Area Name | DSID | IC | Records Read CA | LOG | Records Subord. Written Reg Name | STC # | Final Status          |
|---------------|--------------|------|----|-----------------|-----|----------------------------------|-------|-----------------------|
| F1V4P1        | F1V4P11      | 1    | 3  | 0               | 0   | 3 AS912501                       | 9127  | No errors encountered |
| F1V4P1        | F1V4P12      | 2    | 3  | 0               | 0   | 3 AS912502                       | 9128  | No errors encountered |

|         |          |   |     |   |   |     |          |      |                       |
|---------|----------|---|-----|---|---|-----|----------|------|-----------------------|
| H1V4P11 | H1V4P11A | 1 | 315 | 0 | 0 | 315 | AS912503 | 9129 | No errors encountered |
| H1V4P11 | H1V4P11B | 6 | 315 | 0 | 0 | 315 | AS912504 | 9130 | No errors encountered |
| H1V4P12 | H1V4P12A | 1 | 315 | 0 | 0 | 315 | AS912505 | 9131 | No errors encountered |
| H1V4P12 | H1V4P12B | 6 | 315 | 0 | 0 | 315 | AS912501 | 9127 | No errors encountered |
| H1V4P13 | H1V4P13A | 1 | 315 | 0 | 0 | 315 | AS912502 | 9128 | No errors encountered |
| H1V4P13 | H1V4P13B | 6 | 315 | 0 | 0 | 315 | AS912503 | 9129 | No errors encountered |

#### DATABASE RECOVERY FACILITY UTILITY REPORT

| Database | DDN      | Database Data Set Name         | IC  | PC/DP | IB  | PR  | Utility | Final Status |
|----------|----------|--------------------------------|-----|-------|-----|-----|---------|--------------|
| F1V4P1   | F1V4P11  | DBGT00L.QA.Q11A.F1V4P1.F1V4P11 | 00  | 00    | N/A | N/A |         |              |
| F1V4P1   | F1V4P12  | DBGT00L.QA.Q11A.F1V4P1.F1V4P12 | 00  | 00    | N/A | N/A |         |              |
| H1V4P11  | H1V4P11A | DBGT00L.QA.Q11A.H1V4P1.A00001  | 00  | 00    | N/A | N/A |         |              |
| H1V4P11  | H1V4P11B | DBGT00L.QA.Q11A.H1V4P1.B00001  | 00  | 00    | N/A | N/A |         |              |
| H1V4P12  | H1V4P12A | DBGT00L.QA.Q11A.H1V4P1.A00002  | 00  | 00    | N/A | N/A |         |              |
| H1V4P12  | H1V4P12B | DBGT00L.QA.Q11A.H1V4P1.B00002  | 00  | 00    | N/A | N/A |         |              |
| H1V4P13  | H1V4P13A | DBGT00L.QA.Q11A.H1V4P1.A00003  | 00  | 00    | N/A | N/A |         |              |
| H1V4P13  | H1V4P13B | DBGT00L.QA.Q11A.H1V4P1.B00003  | 00  | 00    | N/A | N/A |         |              |
| F1V4X1   | F1V4X11  | (Index Database)               | 00  | 00    | 00  | N/A |         |              |
| F1V4S1   | F1V4S11  | (Index Database)               | 00  | 00    | 00  | N/A |         |              |
| F1V4S2   | F1V4S21  | (Index Database)               | 00  | 00    | 00  | N/A |         |              |
| H1V4P11  | PRIMINDX | DBGT00L.QA.Q11A.H1V4P1.X00001  | N/A | N/A   | 00  | N/A |         |              |
| H1V4P12  | PRIMINDX | DBGT00L.QA.Q11A.H1V4P1.X00002  | N/A | N/A   | 00  | N/A |         |              |
| H1V4P13  | PRIMINDX | DBGT00L.QA.Q11A.H1V4P1.X00003  | N/A | N/A   | 00  | N/A |         |              |
| H1V4P11  | ILDS/ILE | DBGT00L.QA.Q11A.H1V4P1.L00001  | N/A | N/A   | 00  | N/A |         |              |
| H1V4P12  | ILDS/ILE | DBGT00L.QA.Q11A.H1V4P1.L00002  | N/A | N/A   | 00  | N/A |         |              |
| H1V4P13  | ILDS/ILE | DBGT00L.QA.Q11A.H1V4P1.L00003  | N/A | N/A   | 00  | N/A |         |              |
| H1V4S21  | H1V4S21A | (Partition Index Database)     | 00  | N/A   | 00  | N/A |         |              |
| H1V4S11  | H1V4S11A | (Partition Index Database)     | 00  | N/A   | 00  | N/A |         |              |
| H1V4S12  | H1V4S12A | (Partition Index Database)     | 00  | N/A   | 00  | N/A |         |              |
| H1V4S13  | H1V4S13A | (Partition Index Database)     | 00  | N/A   | 00  | N/A |         |              |

Final Return (RC) and Reason (RSN) Codes

| ---IC--- | ---PC--- | ---DP--- | ---IB--- | ---PR--- | ---LIU--- |
|----------|----------|----------|----------|----------|-----------|
| RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN   | RC RSN    |
| 00 00    | 00 00    | N/A N/A  | 00 N/A   | N/A N/A  | N/A N/A   |

#### DATABASE RECOVERY FACILITY DATA SET I/O REPORT

Recover to point: Not Specified

| Image Copy Data Set Name           | Volume Serial | IC DS Read Count | IC Type | Time Stamp Range       |
|------------------------------------|---------------|------------------|---------|------------------------|
| DBGT00L.QA.Q11A.F1V4P1.F1V4P11.IC1 | DRF004        | 3                | STD     | 1st Record Last Record |
| DBGT00L.QA.Q11A.F1V4P1.F1V4P12.IC1 | DRF004        | 3                | STD     |                        |
| DBGT00L.QA.Q11A.H1V4P1.A00001.IC1  | DRF004        | 315              | STD     |                        |
| DBGT00L.QA.Q11A.H1V4P1.B00001.IC1  | DRF004        | 315              | STD     |                        |
| DBGT00L.QA.Q11A.H1V4P1.A00002.IC1  | DRF004        | 315              | STD     |                        |
| DBGT00L.QA.Q11A.H1V4P1.B00002.IC1  | DRF004        | 315              | STD     |                        |
| DBGT00L.QA.Q11A.H1V4P1.A00003.IC1  | DRF004        | 315              | STD     |                        |
| DBGT00L.QA.Q11A.H1V4P1.B00003.IC1  | DRF004        | 315              | STD     |                        |

| Change Accum Data Set Name | Volume Serial | CA DS Read Count | Time Stamp Range       |
|----------------------------|---------------|------------------|------------------------|
|                            |               |                  | 1st Record Last Record |

No data available for this type data set

| Log Data Set Name | Volume Serial | Log DS Read Count | IMS SYSID | Time Stamp Range       |
|-------------------|---------------|-------------------|-----------|------------------------|
|                   |               |                   |           | 1st Record Last Record |

No data available for this type data set

Below is an explanation of the Batch Invocation of Full Recovery with Integrated Auxiliary Processing Utilities example:

### • **COMMANDS / CONTROL STATEMENTS**

This section displays the images of the control statements read from the SYSIN file:

- The first five statements are Environment Control Statements and override comparable parameters included in the FRXDRFxx member.
- The rest of the statements are Recovery Control Statements.

### • **RECOVERY PARAMETERS**

This section lists the parameters used to direct the recovery. The recovery parameters were supplied via one of the following three mechanisms, in ascending order of selection priority:

#### **PROCESS: RCVDBDS**

A RCVDBDS recovery process was selected.

#### **SOURCE: PRI**

The Primary log and image copy data sets are to be used.

#### **READNUM: 02,03**

The maximum number of total devices to be used during recovery is 03. Of these, up to 02 can be tape devices.

**ERROR: CONT**

Should an error be encountered during recovery, processing should continue for other databases.

**RCVTOKEN: DRF3111**

The recovery token used during recovery is "DRF3111."

**DRFPROC: DRFV32**

The procedure for the RSS is contained in member DRFV32 in the PROCLIB PDS.

**RCVTIME: Not Specified**

A recovery time was not specified.

**TYPE: Full**

A full recovery was requested.

**CHECK: Yes**

A check will be performed to ensure that, should a DBDS be in a recovery group, all of the DBDs within that recovery group need to be included in the list.

**TIME FMT: LCL**

Time format is local.

**LB I: No**

Large Block Interface processing was not requested.

**SPSIZE: 1000**

The SPSIZE specified is 1000. This number was specified as 1024 in the FRXDRFxx member, but was overridden by SYSIN.

**NUM: 3**

Num was specified as 3. This number was specified as 3 in the FRXDRFxx member, but was overridden by SYSIN.

**MAINSIZE: 100**

Mainsize was specified as 100. This number was specified as 100 in the FRXDRFxx member.

**AVGLEN: 1024**

Avglen was specified as 1024. This number was specified as 1024 in the FRXDRFxx member.

**FILSZ: 400000**

Filsz was specified as 400000. This number was specified as 400000 in the FRXDRFxx member.

**HIPRMAX: OPTIMAL**

Hiprmax was specified as Optimal. Both the FRXDRFxx member and SYSIN specified the save value.

**ASPREF: FRXI**

Aspref was specified as FRXI. This value was specified as FRXI in the FRXDRFxx member, but was overridden by SYSIN.

**DYNALLOC: N/A , N/A**

The number of dynamically allocated SORTWORK data sets is N/A, as specified in SYSIN. The unit associated with the dynamic allocation was not specified, and is not available for the report. The Sort installation value for unit will be used by the SORT utility.

- **SUMMARY REPORT**

This section presents statistics and summarizes the results of the recovery for each of the data sets associated with each database:

**Database Name**

The database name is listed.

**DD/Area Name**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. Had any fast path databases been included, the associated Area name or names would have been listed.

**DSID**

The data set id number is listed.

**Records Read**

The number of records read from the image copy, change accumulation and log data sets for application in the recovery are listed. If Flash Copy, the read count will be 'N/A'.

**Records Written**

The number of records written to the database data set is listed.

**Subord. Reg Name**

The name of the started task which was responsible for sorting the records being written for the indicated data set is listed.

**Final Status**

The status of the conclusion of the recovery is listed.

**• DATABASE RECOVERY FACILITY UTILITY REPORT**

This section summarizes the results of the integrated auxiliary utilities:

**Database**

The database name is listed.

**DD/Area**

The DD name associated with each data set is listed. In this case, all of the databases were full-function. Had any fast path databases been included, the associated Area name or names would have been listed.

**Database Data Set Name**

The database data set name associated with each DD or area are listed.

**IC**

Return code from the Image copy utility.

**PC/DP**

Return code from the pointer checker or DEDB pointer checker utility.

**IB**

Return code from the Build Index function of FPA and the Index builder utility.

**PR**

Return code from the DFSPRECO utility.

**Utility Final Status**

The utility final status is listed.

**Final Return (RC) and Reason (RSN) Codes**

Final Return and Reason Codes from the Image copy, pointer checker, DEDB pointer checker, the Build Index function of FPA, Index builder, or DFSPRECO utilities.

**• DATA SET I/O REPORT**

As was seen in earlier sections of the report, no recovery time was specified. This fact is reflected in the "Recover to point" in this section. The data set I/O report is divided into three sections:

**1. Image Copy**

For each image copy data set used during the recovery, the following information is listed:

**Image Copy Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set with intent to be used as input is listed. If Flash Copy, the read count will be 'N/A'.

**IC Type**

The image copy type is listed. Allowable values for this field include:

**STD**

Standard Image Copy

**IC2**

Image Copy 2

**IC EXT**

Extended Image Copy

**HISAM UNL**

HISAM Unload

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**2. Change Accum**

For each change accum data set used during the recovery, the following information is listed:

**Change Accum Data Set Name**

No change accum data sets were used.

**Volume Serial**

This field is intentionally blank.

**Read Count**

This field is intentionally blank.

**Time Stamp Range**

Although this heading appears, this field has no significance for image copy data sets.

**3. Log**

For each log data set used during the recovery, the following information is listed:

**Log Data Set Name**

The data set name is listed.

**Volume Serial**

The volume or volumes upon which the data sets were found.

**Read Count**

The number of records read from the data set with intent to be used as input is listed.

**IMS SYSID**

The SYSID associated with the IMS which created the log is listed.

**Time Stamp Range**

The range of times which are covered by the log is listed. The time stamp on the first and last records in the log is listed.

**Prilog:**

The PRILOG time listed in RECON for the log is listed.

The difference between the log records read and the log records applied values in the recovery report are due to the update records read that generate multiple output records. The IMS Database Recovery Facility creates multiple update records for expedited processing when certain internal conditions are recognized.



---

## Part 8. Reference: Base Primitive Environment (BPE)

The topics in this section provide you with technical references for the Base Primitive Environment (BPE).

**Topics:**

- [Chapter 27, “BPE commands,” on page 319](#)
- [Chapter 28, “BPE messages and codes,” on page 325](#)





---

## Chapter 27. BPE commands

IMS Database Recovery Facility Base Primitive Environment (BPE) provides a set of commands that you can use to manage resources in the IMS Database Recovery Facility environment.

Topics:

- [“Specifying BPE command parameters” on page 319](#)
- [“BPE TRACETABLE commands” on page 319](#)

---

### Specifying BPE command parameters

IMS Database Recovery Facility BPE commands enable you to display and update trace table resources that IMS Database Recovery Facility BPE manages.

#### About this task

Some resource types are defined and owned by the IMS Database Recovery Facility BPE itself, and are known as system resource types. Other resource types that are defined and owned by the IMS Database Recovery Facility are known as component resource types or user-product resource types.

The IMS Database Recovery Facility BPE commands allow you to restrict the resource types upon which a command operates to either those owned by the IMS Database Recovery Facility BPE, or to those owned by the IMS Database Recovery Facility itself. This is done through the OWNER keyword on commands that support OWNER.

#### Procedure

- Use OWNER(BPE) to restrict the command operation to resource types that the IMS Database Recovery Facility BPE owns and defines (system resource types).
- Use OWNER(DRF) to restrict the command operation to resource types that the IMS Database Recovery Facility address defines and owns (component resource types).

---

### BPE TRACETABLE commands

The TRACETABLE resource type refers to the internal IMS Database Recovery Facility BPE-managed trace tables that are defined either by IMS Database Recovery Facility BPE (for example: DISP, CBS, STG, LATC), or by IMS Database Recovery Facility.

Two command verbs operate on the TRACETABLE resource type:

#### DISPLAY

Display trace level and number of trace table pages of specified trace tables.

#### UPDATE

Update trace level attributes of specified trace tables.

### DISPLAY TRACETABLE command

You can use the DISPLAY TRACETABLE command to display the current attribute settings for the requested trace tables.

The following diagram shows the syntax for the DISPLAY TRACETABLE command:



## OWNER(BPE | DRF)

An optional parameter that specifies the owner of the trace table type or types about which you want attributes displayed. You can specify one of the following values:

### BPE

For all IMS Database Recovery Facility BPE-defined trace table types

### DRF

For IMS Database Recovery Facility-defined trace table types

The OWNER parameter acts as a filter to help you select which trace tables you want to display. For example, you could specify NAME(\*) OWNER(DRF) to display all of the IMS Database Recovery Facility-defined trace table types (INIT, DRF).

You could specify NAME(\*) OWNER(BPE) to display all of the IMS Database Recovery Facility BPE-defined trace table types.

If OWNER is omitted, then both the IMS Database Recovery Facility BPE and the IMS Database Recovery Facility component trace tables might be displayed (depending on the tables specified on NAME).

## DISPLAY TRACETABLE command output

The DISPLAY TRACETABLE command output consists of a header line, one line per selected trace table, and one message BPE0032I line that indicates that the command has completed.

The following example shows the output from the DISPLAY TRACETABLE command.

```
BPE0030I TABLE OWNER LEVEL #PAGES
BPE0000I DISP BPE HIGH 12
BPE0000I STR SMDC MEDIUM 8
BPE0032I DISPLAY TRACETABLE COMMAND COMPLETED
```

These columns are in the DISPLAY TRACETABLE output:

### TABLE

Specifies the name of the trace table type about which information is being displayed on the current row.

### OWNER

Specifies the IMS Database Recovery Facility component that owns the trace table (BPE or the IMS Database Recovery Facility).

### LEVEL

Specifies the current level setting of the trace table. A trace table's level determines the volume of trace data that is collected. These levels are possible:

#### NONE

No tracing is being done into the table.

#### ERROR

Only traces for error or exception conditions are being made into the table.

#### LOW

Only major event trace entries are made into the table.

#### MEDIUM

Major event trace entries and some minor event trace entries are made into the table.

#### HIGH

All trace entries are made into the table.

#### INACTV

The trace table is inactive and cannot be used. This status occurs only when BPE was unable to get any storage for the trace table. No tracing will be done for the indicated table type, and you cannot change the level for the trace table with the UPDATE TRACETABLE command. You must restart the address space in order to use the trace table again.

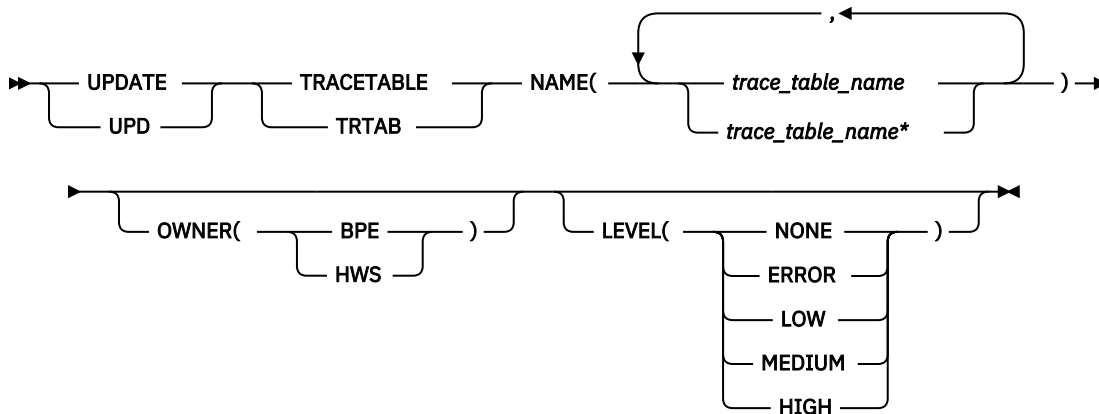
## #PAGES

Specifies the number of 4 KB (4096 byte) pages that are allocated for the trace table type.

## UPDATE TRACETABLE command

You can use the UPDATE TRACETABLE command to change the trace level setting for the requested trace tables.

The following diagram shows the syntax for the UPDATE TRACETABLE command:



The following list describes the parameters of UPDATE TRACETABLE:

### UPDATE | UPD

A required parameter that specifies that the action against the trace table is to update its attributes.

### TRACETABLE | TRTAB

A required parameter that specifies that the resource type being acted upon is an IMS Database Recovery Facility BPE-managed trace table.

### NAME(trace\_table\_name)

A required parameter that specifies the name of the trace table type or types that you want to update. You can specify a single trace table name or a list of trace table names separated by commas. Trace table names can contain wildcard characters. Trace table names can be IMS Database Recovery Facility BPE-defined trace tables or IMS Database Recovery Facility-defined trace tables.

The following IMS Database Recovery Facility BPE-defined trace table types are available:

#### AWE

Asynchronous work element (AWE) trace table

#### CBS

Control block services trace table

#### CMD

Command trace table

#### DISP

Dispatcher trace table

#### ERR

IMS Database Recovery Facility BPE Error trace table

#### LATC

Latch trace table

#### MISC

Miscellaneous trace table that is used only by IMS Service for trap traces

#### SSRV

System services trace table

#### STG

Storage service trace table

## **USRX**

User exit routine trace table

You can update IMS Database Recovery Facility-defined trace tables only for IMS Database Recovery Facility address spaces.

## **OWNER(BPE | DRF)**

An optional parameter that specifies the owner of the trace table type or types that you want to update. You can specify one of the following values:

### **BPE**

For IMS Database Recovery Facility BPE-defined trace table types

### **DRF**

For IMS Database Recovery Facility-defined trace table types

The OWNER parameter acts as a filter to help you select the trace tables that you want to update. For example, you could specify NAME(\*) OWNER(DRF) to update all of the IMS Database Recovery Facility-defined trace table types.

You could specify NAME(\*) OWNER(BPE) to update all of the IMS Database Recovery Facility-defined trace table types.

If OWNER is omitted, then both IMS Database Recovery Facility BPE and IMS Database Recovery Facility trace tables might be updated (depending on the tables specified on NAME).

## **LEVEL(level)**

An optional parameter that sets the new tracing level for the specified trace tables. If LEVEL is omitted, the level of the specified trace tables is not changed. These levels are possible:

### **NONE**

No tracing is being done into the table.

### **ERROR**

Only traces for error or exception conditions are being made into the table.

### **LOW**

Only major event trace entries are made into the table.

### **MEDIUM**

Major event trace entries and some minor event trace entries are made into the table.

### **HIGH**

All trace entries are made into the table.

**Important:** You cannot change the level for the trace table type ERR. IMS Database Recovery Facility BPE forces the level to HIGH to ensure that error diagnostics are captured. Any level that you specify for the ERR trace table is ignored.

## **UPDATE TRACETABLE command output**

The UPDATE TRACETABLE command output consists of message BPE0032I, which indicates that the command has completed.

An example of the UPDATE TRACETABLE command output message is:

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
```

### **Example: UPDATE TRACETABLE command LEVEL(HIGH)**

Update the level of the IMS Database Recovery Facility BPE dispatcher trace table (DISP) to HIGH.

#### **Example command:**

```
F DRF1,UPDATE TRACETABLE NAME(DISP) LEVEL(HIGH)
```

**Example output:**

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
```

**Example: UPDATE TRACETABLE command LEVEL(MEDIUM)**

Update the level of all the IMS Database Recovery Facility trace tables to MEDIUM.

**Important:** You cannot change the level for the trace table type ERR - even when using a wildcard character to select all tables with a given owner, as in the following command example. The IMS Database Recovery Facility BPE forces the level to HIGH to ensure that error diagnostics are captured.

**Example command:**

```
F DRF1,UPD TRTAB NAME(*) OWNER(DRF) LEVEL(MEDIUM)
```

**Example output:**

```
BPE0032I UPDATE TRACETABLE COMMAND COMPLETED
```

---

## Chapter 28. BPE messages and codes

This section describes the messages and user abend codes that are issued by Base Primitive Environment (BPE) in IMS Database Recovery Facility.

### Topics:

- [“BPE messages” on page 325](#)
- [“BPE service return codes” on page 342](#)
- [“BPE user abend codes” on page 350](#)

---

### BPE messages

This topic provides detailed information about messages issued by the Base Primitive Environment (BPE). The explanations and user responses provided in this reference can help you diagnose, troubleshoot, and solve IMS Database Recovery Facility problems.

Base Primitive Environment messages adhere to the following format:

```
BPEnnnnx
```

where:

#### **BPE**

Indicates that the message was issued by Base Primitive Environment

#### **nnnn**

Indicates the message identification number

#### **x**

Indicates the severity of the message:

##### **A**

Indicates that operator intervention is required before processing can continue.

##### **E**

Indicates that an error occurred, which might or might not require operator intervention.

##### **I**

Indicates that the message is informational only.

##### **W**

Indicates that the message is a warning to alert you to a possible error condition.

Each message also includes the following information:

#### **Explanation:**

The Explanation section explains what the message text means, why it occurred, and what its variables represent.

#### **System action:**

The System action section explains what the system will do in response to the event that triggered this message.

#### **User response:**

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

#### **Module**

The Module section indicates which module or modules are affected.

#### **Message Variables**

In the message text, there can be lowercase variables (for example, xxx...). The variables represent values when the message appears such as:

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

---

## **BPE0000I**      *displayoutput*

### **Explanation**

Message BPE0000I is the message number associated with output generated by the DISPLAY command verb for IMS Database Recovery Facility BPE resources.

In the message text, *displayoutput* is output text from the display verb. Typically, when a display command is issued against a resource, the output will consist of header lines, and then one or more data lines containing information about the resources being displayed.

### **System action**

None.

### **User response**

None. This message is informational.

### **Module**

FRXCTRA\$, FRXCURF\$

---

## **BPE0001E**      **BPE INITIALIZATION ERROR IN MODULE *module* details**

### **Explanation**

An error occurred in early IMS Database Recovery Facility BPE initialization. This is a two-line message. Each line begins with the message number.

In the message text:

#### ***module***

The module detecting the error

#### ***details***

A one-line explanation of the type of error that was detected

#### ***version***

The hexadecimal invalid version number from the definition module

The *details* line of the BPE0001E message further explains the error detected, and can be one of the following:

- **MODULE *module* HAS AN INVALID VERSION NUMBER *version***

An IMS Database Recovery Facility BPE definition module had an invalid version number. IMS

Database Recovery Facility BPE uses version numbers as a consistency check to ensure that the definition modules being loaded match the version of IMS Database Recovery Facility BPE that is running.

In the message text:

#### ***module***

The IMS Database Recovery Facility BPE definition module in which the invalid version was detected

#### ***version***

The hexadecimal invalid version number from the definition module

- **MODULE *module* IS LINKED REENTRANT - IT MUST BE NON-REENTRANT**

An IMS Database Recovery Facility BPE definition module was loaded into key zero storage. IMS Database Recovery Facility BPE definition modules are modules that contain (data control blocks), and thus must be loaded into storage that is in the same storage key in which IMS Database Recovery Facility BPE is running. The most common cause for this error message is linking a definition module as reentrant, which loads it into key zero storage. IMS Database Recovery Facility BPE definition modules should be link-edited as non-reentrant.

In the message text:

#### ***module***

The IMS Database Recovery Facility BPE definition module that was loaded in key zero storage

- **UNKNOWN ERROR, MODULE RC=*rc***

An internal error occurred that IMS Database Recovery Facility BPE does not recognize.

In the message text:

#### ***rc***

The return code from the failing module

- **ERROR LOADING MODULE *module* BPELOAD RC=*rc***

Load failed for a module.

In the message text:

#### ***module***

The name of a module that could not be loaded

#### ***rc***

The return code from the IMS Database Recovery Facility BPE load service, BPELOAD



- **ERROR LOADING MODULES *module...* BPELOAD RC=*rc***

Load failed for several modules.

In the message text:

***module***

The name of the first of several modules that could not be loaded

***rc***

The return code from the IMS Database Recovery Facility BPE load service, BPELOAD

- **UNABLE TO CREATE *threadtype* THREAD, BPETHDCR RC=*rc***

A request to create an IMS Database Recovery Facility BPE thread (internal unit of work) failed.

In the message text:

***threadtype***

A 4-character name of the thread type that could not be created

***rc***

The return code from the thread create service, BPETHDCR

- **MODULE *module* IS NOT A VALID *type* DEFINITION MODULE**

IMS Database Recovery Facility BPE encountered an error with an internal IMS Database Recovery Facility BPE definition module. IMS Database Recovery Facility BPE uses definition modules to construct its execution environment. If a definition module is not correct, IMS Database Recovery Facility BPE cannot build the appropriate environment.

In the message text:

***module***

The name of an IMS Database Recovery Facility BPE definition module that is in error

***type***

The type of definition module in error

- **UNABLE TO GET NECESSARY STORAGE, BPEGETM RC=*rc***

IMS Database Recovery Facility BPE could not obtain the required storage.

In the message text:

***rc***

The return code from the BPE GETMAIN service, BPEGETM

- **UNABLE TO GET STORAGE FOR *blocktype* BLOCK, BPECBGET RC=*rc***

IMS Database Recovery Facility BPE could not obtain storage for a required control block.

In the message text:

***blocktype***

A 4-character name of an internal BPE control block that could not be obtained

***rc***

The return code from the BPE control block get service, BPECBGET

- **ERROR READING PROCLIB DATA SET, BPERDPDS RC=*rc***

IMS Database Recovery Facility BPE could not read a PROCLIB DD data set. This message follows message BPE0002E, which provides further details on the specific data set and member that could not be read.

In the message text:

***rc***

The return code from the IMS Database Recovery Facility BPE partitioned data set reading service, BPERDPDS

- **ERROR PARSING *datasettype*, BPEPARSE RC=*rc***

An error was detected while trying to parse a configuration data set. This message follows message BPE0003E, which provides further details on the error.

In the message text:

***data\_set\_type***

A description of the type of data set that was being parsed. This can be one of the following:

- **BPE CONFIG MEMBER:** The main IMS Database Recovery Facility BPE configuration PROCLIB member specified on the BPECFG parameter of the startup JCL or procedure.
- **component USER EXIT LIST MEMBER:** The user exit list PROCLIB member for either IMS Database Recovery Facility BPE or IMS Database Recovery Facility. This is the user exit list member specified on the EXITMBR statement in the BPE configuration PROCLIB member.

***rc***

The return code from the IMS Database Recovery Facility BPE parsing service, BPEPARSE.

- **INITIALIZATION FAILED FOR *service*, RC=*rc***

IMS Database Recovery Facility BPE was unable to initialize an IMS Database Recovery Facility BPE system service.

In the message text:

***service***

The name of the IMS Database Recovery Facility BPE service that failed initialization. Values for this field include:

- **BPE RESMGR:** An error occurred when IMS Database Recovery Facility BPE tried to establish a resource manager routine to clean up global resources used by IMS Database Recovery Facility BPE. The return code in this message is the return code from the MVS RESMGR macro call that failed.
- **USER EXITS:** An error occurred while trying to load user exit modules. This message might be preceded by other error messages that indicate the problem. The return code from the module that loads the user exits (FRDUXRF\$) is in the *rc* field in this message.
- **BPE SVC:** An error occurred when IMS Database Recovery Facility BPE tried to initialize the BPE SVC routine. IMS Database Recovery Facility BPE provides an internal SVC routine for use by IMS Database Recovery Facility BPE and IMS Database Recovery Facility components. The SVC is installed dynamically when an IMS Database Recovery Facility BPE address space is started. This error message is issued when the IMS Database Recovery Facility BPE SVC could not be installed.

If the *rc* field in the message is X'00000020', you are probably trying to start the address space on an MVS system that is not at the required maintenance level. The MVS system must be at least at MVS SP 4.3.0 or higher. If the MVS system is at MVS SP 5.2.0 or below, you must have MVS APARs OW13312 and OW13315 applied.

#### ***rc***

The return code from the lower level initialization module that encountered the error.

#### • **INVALID CALLABLE SERVICE CODE *code* IN MODULE *module***

IMS Database Recovery Facility BPE detected an invalid user exit callable service code in the indicated module. Both IMS Database Recovery Facility BPE and IMS Database Recovery Facility can define callable services that user exits can use. Each callable service has a callable service code used to request the service. This error indicates that there is a definition error in the specified module.

In the message text:

#### ***code***

The callable service code in error, in hexadecimal.

#### ***module***

The name of the callable service module that had the invalid callable service code defined.

#### **System action**

Abend U3400 with subcode 05 follows this message. The address space ends.

#### **User response**

If the error described in this message is caused by environmental conditions (for example, insufficient storage or modules missing from your STEPLIB data set), correct the indicated problem and restart the address space. Otherwise, save any dump and SYSLOG information, and contact IBM Software Support.

#### **Module**

FRXAWIO\$, FRXCBI0\$, FRXCMDI\$, FRXDSIO\$, FRXHTIO\$, FRXINIT\$, FRXPCFG\$, FRXRV1I\$, FRXSTIO\$, FRXTRIO\$, FRXUXIO\$, FRXXMOD\$, FRXXSYS\$

---

|                 |                                    |
|-----------------|------------------------------------|
| <b>BPE0002E</b> | <b>ERROR READING <i>ddname</i></b> |
|                 | <b>MEMBER <i>member</i></b>        |
|                 | <b><i>details</i></b>              |

---

#### **Explanation**

An error occurred trying to read a partitioned data set member. This is a two-line message. Each line begins with the message number.

In the message text:

#### ***ddname***

The DD name of the data set being read

#### ***member***

The member name of the data set being read

#### ***details***

A one-line explanation of the type of error detected

The *details* line of the BPE0002E message further explains the error detected, and can be one of the following:

#### • **OPEN FAILED FOR DATA SET**

The IMS Database Recovery Facility BPE PDS read service could not open the data set for reading.

#### • **DATA SET RECORDS ARE NOT FIXED FORMAT**

The data set specified records that were not in fixed format. The BPE PDS read service requires fixed-format data sets.

#### • **MEMBER NOT FOUND IN DATA SET**

The indicated member was not in the partitioned data set.

#### • **BSAM READ FAILED READING MEMBER**

An error occurred during the reading of the data set member.

• **UNABLE TO OBTAIN SUFFICIENT STORAGE FOR MEMBER TO BE READ**

The PDS read service could not obtain the storage necessary to read the data set member.

**System action**

The BPE PDS read service does not take any further action after issuing this message. The caller of the service might provide additional diagnostic messages or end the address space.

**User response**

Correct the error as indicated in this message.

**Module**

FRXRDPD\$

|                 |                                                                                                                                                                           |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>BPE0003E</b> | <b>AN ERROR OCCURRED PARSING</b><br><i>description</i><br><b>AT LINE</b> <i>line</i> , <b>CHARACTER</b> <i>char</i><br><b>FAILING TEXT:</b> <i>text</i><br><i>details</i> |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Explanation**

An error occurred in the IMS Database Recovery Facility BPE parsing service. This is a four-line message. Each line begins with the message number.

In the message text:

**description**

A text description of what was being parsed.

**line**

The line number of the data where the error occurred. For data that is not line-oriented, the line number is omitted.

**char**

The position of the character on the line where the error was detected. For data that is not line-oriented, *char* identifies the position of the character within the input data.

**text**

Up to 16 characters of the text where the error was detected.

**details**

A one-line explanation of the type of error detected.

The *details* line of the BPE0003E message further explains the error detected, and can be one of the following:

• **INVALID KEYWORD DETECTED**

The parser found an unknown keyword in the input data.

• **UNKNOWN POSITIONAL PARAMETER**

The parser found a positional parameter in the input data when one was not expected.

• **"=" ENCOUNTERED WHEN "(" EXPECTED**

The parser found an equal sign in the input data when a left parenthesis was expected.

• **EARLY END OF INPUT DATA**

The input data ended before the parser found all the required data.

• **KEYWORD ENCOUNTERED WHEN VALUE EXPECTED**

The input data contained a keyword when the parser expected a value.

• **NUMERIC VALUE OUTSIDE OF LEGAL RANGE**

A numeric value was outside the allowed range for the parameter.

• **DECIMAL NUMBER CONTAINED NONDECIMAL DIGITS**

A decimal number contained a non-decimal character.

• **HEXADECIMAL NUMBER CONTAINED NONHEX DIGITS**

A hexadecimal number contained a non-hexadecimal character.

• **UNKNOWN KEYWORD VALUE DETECTED**

The parser found a parameter that could be one of a set of keyword values. The parameter was not one of the values in the set.

• **DUPLICATE KEYWORD PARAMETER DETECTED**

The parser found a nonrepeatable keyword more than once in the input data.

• **A REQUIRED PARAMETER WAS OMITTED**

A required parameter was not found in the input data.

• **CHARACTER VALUE WAS TOO LONG FOR PARAMETER**

The character value specified was too long for the parameter field.

**System action**

The IMS Database Recovery Facility BPE parsing service does not take any further action after issuing this message. The caller of the service can provide additional diagnostic messages or end the address space.

### User response

Correct the error as indicated in the BPE0003E message.

### Module

FRXPARS\$

---

**BPE0004I      BPECFG= NOT SPECIFIED ON  
STARTUP PARMS - DEFAULTS  
BEING USED**

---

### Explanation

IMS Database Recovery Facility BPE could not find the specification for the IMS Database Recovery Facility BPE configuration data set in the startup parameters.

### System action

The address space continues initialization. BPE runs with defaults for all parameters in the IMS Database Recovery Facility BPE configuration data set.

### User response

If the BPE configuration parameter defaults are acceptable, no action is required. Otherwise, create a BPE configuration member in the data set pointed to by the PROCLIB DD statement and specify `BPECFG=member_name` in the startup parameters of the job that starts the address space.

### Module

FRXPCFG\$

---

**BPE0005I      UNKNOWN component TRACE  
TYPE type IN PROCLIB MEMBER  
member- IGNORED**

---

### Explanation

A TRCLEV= statement was found in the IMS Database Recovery Facility BPE configuration PROCLIB data set member for an unknown trace table type.

In the message text:

#### **component**

The name of the component that was specified on the TRCLEV statement (BPE or FRX) being parsed

#### **type**

The type of the trace table that was unknown

#### **member**

The member name of the PROCLIB DD data set that was being processed

### System action

The address space initialization continues. IMS Database Recovery Facility BPE ignores the invalid trace specification.

### User response

Correct the TRCLEV statement.

### Module

FRXPCFG\$

---

**BPE0006I      aaaa bbbb TCB ABEND cddd-  
eeeeeee, THD = ffff retrystatus  
MODULE ID = gggggggggggggggggg  
EP = hhhhhhhh  
PSW = iiiiii iiiiii OFFSET = jjjjjj  
R00-03 kkkkkkkk kkkkkkkk  
kkkkkkkk kkkkkkkk  
R04-07 kkkikkkkk kkkkkkkk  
kkkkkkkk kkkkkkkk  
R08-11 kkkkkkkk kkkkkkkk  
kkkkkkkk kkkkkkkk  
R12-15 kkkkkkkk kkkkkkkk  
kkkkkkkk kkkkkkkk  
R12-15 kkkkkkkk kkkkkkkk  
kkkkkkkk kkkkkkkk**

---

### Explanation

This seven-line message is printed when an abend occurs. It documents the environment at the time of abend. Each line begins with the message number.

In the message text:

#### **aaaa**

The owning component of the TCB that abended (BPE or FRX).

#### **bbbb**

The 4-character TCB type of the TCB that abended.

#### **c**

Whether the abend is a system abend or a user abend. S is for a system abend; U is for a user abend.

#### **dddd**

The abend code. For system abends, this is a 3-character hexadecimal code. For user abends, this is a 4-character decimal code.

#### **eeeeeee**

The value in register 15 at the time of abend. For some abends, this is the abend subcode.

#### **ffff**

The 4-character IMS Database Recovery Facility BPE thread (internal unit of work) name of the thread that was running when the abend occurred.

If the thread type cannot be determined, this field is set to question marks (????).

#### **retrystatus**

Whether or not the abend is being retried that is, whether the system is attempting to recover from the abend). If the abend is being retried, *retrystatus* is the character string "(RETRYING)". If the abend is not being retried, *retrystatus* is blank.

#### **gggg...**

The abending module's name from its module ID, if it can be determined.

#### **hhhhhhhh**

The entry point address of the abending module, if it can be determined.

#### **iiiiii iiiiii**

The PSW contents at the time of abend.

#### **jjjjjj**

The offset within the abending module in which the abend occurred, if it can be determined.

#### **kkkkkkkk**

The contents of the registers at the time of abend.

#### **System action**

If the abend is being retried (*retrystatus* is "(RETRYING)" in the first line of the message), the system attempts to recover from the abend and the address space continues to function.

If the abend is not being retried, the action taken after this message depends on whether the abending TCB is considered a critical TCB to the address space. If it is not critical, the TCB is ended, but the address space continues execution. If it is critical, the address space is ended abnormally.

#### **User response**

Save any dump and SYSLOG information, and contact IBM Software Support.

#### **Module**

FRXINIT\$, FRXSYES\$

---

### **BPE0007I      FRD BEGINNING PHASE 1 OF SHUTDOWN**

#### **Explanation**

IMS Database Recovery Facility BPE is beginning the first phase of shutting down the address space.

#### **System action**

The address space enters the first phase of ending, in which all of the IMS Database Recovery Facility TCBs are ended.

#### **Module**

FRXSYTR\$

---

### **BPE0008I      FRD BEGINNING PHASE 2 OF SHUTDOWN**

#### **Explanation**

IMS Database Recovery Facility BPE is beginning the second phase of shutting down the address space.

#### **System action**

The address space enters the second phase of ending, in which all of the IMS Database Recovery Facility BPE system TCBs are ended.

#### **Module**

FRXSYTR\$

---

### **BPE0009I      FRD SHUTDOWN COMPLETE**

#### **Explanation**

Shutdown of the address space completed. IMS Database Recovery Facility BPE is returning to MVS.

#### **System action**

The address space ends normally.

#### **Module**

FRXSYTR\$

---

### **BPE0010I      PSW AND REGISTERS AT ABEND ARE NOT AVAILABLE**

#### **Explanation**

An abend occurred and the IMS Database Recovery Facility BPE system ESTAE routine received control with no SDWA available. The ESTAE is unable to provide diagnostic information normally obtained from the SDWA, such as PSW and register contents at abend.

#### **System action**

BPE continues to process the abend with limited capability because of the lack of the SDWA.

#### **Module**

FRXSYES\$

---

### **BPE0011E      ABEND IN BPE SYSTEM ESTAE ROUTINE (FRDSYES\$)**

## Explanation

An abend occurred in the IMS Database Recovery Facility BPE system ESTAE module itself while it was processing a prior abend.

## System action

This message should be followed by an MVS symptom dump on the MVS console. Additionally, a SYS1.LOGREC entry is generated for the abend. The IMS Database Recovery Facility BPE ESTAE module attempts to recover from the abend and continues processing the original abend.

## Module

FRXSYES\$

---

**BPE0012E      BPE ETXR UNABLE TO FIND *block*  
BLOCK FOR TCB AT *address***

## Explanation

The IMS Database Recovery Facility BPE end of task exit routine (ETXR) was called when a TCB in the address space ended. The routine tried to process ending of the task, but was not able to locate a required IMS Database Recovery Facility BPE control block. This is probably caused by internal control block errors or overlays within the address space.

In the message text:

### ***block***

The name of the IMS Database Recovery Facility BPE control block that could not be found

### ***address***

The address of the ending TCB

## System action

IMS Database Recovery Facility BPE attempts to clean up the TCB. However, without the required control blocks, the cleanup might not be successful, and other TCBs in the address space might not end.

## User response

If the address space appears hung (is not processing or does not end), cancel the address space with a dump, and contact IBM Software Support.

## Module

FRXSYET\$

---

**BPE0013E      VERSION MISMATCH BETWEEN  
BPE AND FRD  
BPE MODULE VERSION IS  
*bver.brel.bptrel***

**FRD WAS ASSEMBLED AT BPE  
VERSION *pver.prel.pptrel***

## Explanation

The IMS Database Recovery Facility BPE version on which IMS Database Recovery Facility was built does not match the version of the IMS Database Recovery Facility BPE modules that were loaded.

This can occur, for example, if IMS Database Recovery Facility was assembled at one IMS Database Recovery Facility BPE version and the your STEPLIB data set contained IMS Database Recovery Facility BPE modules at a different IMS Database Recovery Facility BPE level.

This is a three-line message. Each line begins with the message number.

In the message text:

### ***bver***

The version number of IMS Database Recovery Facility BPE modules

### ***brel***

The release number of IMS Database Recovery Facility BPE modules

### ***bptrel***

The point-release number of IMS Database Recovery Facility BPE modules

### ***pver***

The version number of IMS Database Recovery Facility BPE macros at which IMS Database Recovery Facility was assembled

### ***prel***

The release number of IMS Database Recovery Facility BPE macros at which IMS Database Recovery Facility was assembled

### ***pptrel***

The point-release number of IMS Database Recovery Facility BPE macros at which IMS Database Recovery Facility was assembled

## System action

IMS Database Recovery Facility BPE abends during early initialization with abend U3400, subcode X'09'.

## User response

This problem is probably caused by a mismatch with your STEPLIB data set. Ensure that IMS Database Recovery Facility is using the correct version of the your STEPLIB data set.

## Module

FRXINIT\$

**BPE0014E            ABEND IN RECOVERY ROUTINE**

**Explanation**

An IMS Database Recovery Facility BPE recovery routine (BRR) that was intended to provide recovery for abends in a section of code itself encountered an abend. (A BRR is an internal recovery routine established by either IMS Database Recovery Facility BPE or IMS Database Recovery Facility to protect a functional area. BRRs run in an MVS ESTAE environment and attempt to recover from abends that occur in mainline code.)

**System action**

The BPE ESTAE routine treats this abend as if the recovery routine indicated that it could not recover, and continues abend processing. Usually, this results in the abnormal end of the address space; however, if there were other BRRs established when the abend occurred, recovery of the original abend still might occur.

**User response**

Save any dump and SYSLOG information, and contact IBM Software Support.

**Module**

FRXSYES\$

**BPE0015I            UNKNOWN *component* EXIT TYPE  
                      *type* IN EXIT LIST PROCLIB  
                      MEMBER *member*- IGNORED**

**Explanation**

While processing a user exit list PROCLIB member, IMS Database Recovery Facility BPE encountered an EXITDEF statement that specified a user exit type that was not defined to IMS Database Recovery Facility BPE. The exit definition for the indicated type is ignored.

In the message text:

***component***

The name of the owning component of the user exit list member that was being processed (for example, BPE or FRX)

***type***

The up-to-8 character exit type name that was undefined

***member***

The user exit list PROCLIB member name

**System action**

The user exit definition is ignored. Processing of the user exit list PROCLIB member continues.

**User response**

Examine the indicated user exit list PROCLIB member and correct the EXITDEF statement for the indicated user exit type.

**Module**

FRXUXRF\$

**BPE0016I            ERROR LOADING *componenttype*  
                      EXIT module (service RC=*rc*)**

**Explanation**

While processing a PROCLIB member for a user exit list, IMS Database Recovery Facility BPE was unable to load a user exit that was specified on an EXITDEF statement.

In the message text:

***component***

The name of the owning component of the user exit list member that was being processed (for example, BPE or FRX).

***type***

The up-to-8 character exit type name of the exit that could not be loaded.

***module***

The load module name of the exit that could not be loaded.

***service***

The name of the failing service that prevented the exit from being loaded. The most likely failing service is BLDL, which is the MVS service that IMS Database Recovery Facility BPE uses to locate the user exit module to load. This usually means that BLDL could not find the indicated exit module in the STEPLIB or JOBLIB data set for the job.

***rc***

The 4-digit hexadecimal return code from the failing service.

**System action**

The user exit module is ignored. Processing of the PROCLIB member continues. If this error occurs in the initial PROCLIB member processing during early address space initialization, IMS Database Recovery Facility BPE abends with abend U3400, subcode 5. If this error occurs during REFRESH USEREXIT command processing, the command is ignored, and no changes are made to the user exit environment.

**User response**

Examine the indicated PROCLIB member for the user exit list and correct the EXITDEF statement for the indicated user exit type and module. If the BPE0016I message was issued during address space initialization, restart the address space. If the message was issued in response to a REFRESH USEREXIT command, reissue the command.

**Module**

FRXUXRF\$

---

|                 |                                                                                            |
|-----------------|--------------------------------------------------------------------------------------------|
| <b>BPE0017I</b> | <b>MULTIPLE <i>type</i> STATEMENTS IN <i>member</i> PROCLIB MEMBER - LAST WILL BE USED</b> |
|-----------------|--------------------------------------------------------------------------------------------|

---

**Explanation**

While processing a PROCLIB member, BPE encountered multiple statements of a specific type when only one was expected. BPE uses the last statement of the duplicated type; the others are ignored.

In the message text:

***type***

A short description of the type of statement that was duplicated

***member***

The PROCLIB member name

**System action**

Processing continues.

**User response**

No response is required. However, you might want to examine the indicated PROCLIB member to ensure that the values specified on the final statement are correct.

**Module**

FRXTRIO\$, FRXUXRF\$

---

|                 |                                                                                        |
|-----------------|----------------------------------------------------------------------------------------|
| <b>BPE0018I</b> | <b>DUPLICATE EXIT <i>module</i> SPECIFIED FOR <i>component type</i> EXIT - IGNORED</b> |
|-----------------|----------------------------------------------------------------------------------------|

---

**Explanation**

While processing a PROCLIB member for a user exit list, IMS Database Recovery Facility BPE found the same user exit module specified more than once in a single exit list (EXITS parameter) on an EXITDEF statement. Only the first instance of the module is in

effect; all subsequent specifications of the module are ignored.

In the message text:

***module***

The name of the user exit module that was specified more than once

***component***

The name of the owning component of the user exit list member that was being processed (BPE or FRX)

***type***

The up-to-8 character exit type name of the exit being processed

**System action**

The duplicate user exits after the first occurrence are ignored. Processing of the user exit PROCLIB continues.

**User response**

No response is required. However, you might want to examine the EXITDEF statement to insure that the exits specified are listed in the correct order for your installation. User exits are called in the order they are listed on the EXITS statement.

**Module**

FRXUXRF\$

---

|                 |                                                                                   |
|-----------------|-----------------------------------------------------------------------------------|
| <b>BPE0019E</b> | <b><i>component type</i> USER EXIT<br/>MODULE <i>module</i> ABEND <i>code</i></b> |
|-----------------|-----------------------------------------------------------------------------------|

---

**Explanation**

An abend occurred while a user exit module was in control. "In control" means that IMS Database Recovery Facility BPE gave control to the exit. However, the abend does not have to be in the exit module itself. This message is also issued if the exit called another module, which then abended.

In the message text:

***component***

The name of the owning component of the user exit type being called (BPE or IMS Database Recovery Facility).

***type***

The up-to-8 character exit type name of the exit that abended.

***module***

The load module name of the exit that abended.



**code**

The abend code. For system abends, the format of code is Sxxx, where xxx is the 3-digit abend code in hexadecimal. For user abends, the format of code is Udddd, where dddd is the 4-digit abend code in decimal.

**System action**

IMS Database Recovery Facility BPE protects all user exits with a recovery routine. When a user exit abends, IMS Database Recovery Facility BPE attempts to recover from the abend and continue processing. Recovery actions include the following:

- The first time a specific user exit module abends after it is loaded, IMS Database Recovery Facility BPE issues an SDUMP to dump the address space for the abend. For abends after the first abend, BPE does not dump the address space again; however, it does generate a SYS1.LOGREC entry for each abend occurrence.
- If the number of abends for a specific user exit module reaches or exceeds the abend limit value for the exit's user exit type, no further calls are made to the exit until it is refreshed. The abend limit for an exit type is specified by the ABLIM parameter on the EXITDEF statement.

After a user exit abends, IMS Database Recovery Facility BPE attempts to call any other exit modules that were specified after the failing exit in the EXITS exit list.

**User response**

Examine the SDUMP, the SYS1.LOGREC entries, or both to determine the problem with the user exit. Correct the problem with the exit, relink the exit to the job's libraries, and issue the REFRESH USEREXIT command. the REFRESH USEREXIT command loads a copy of the corrected exit and causes IMS Database Recovery Facility BPE to resume calling the exit module whenever exits of its type are invoked.

**Module**

FRXUXCL\$

|                 |                                                                               |
|-----------------|-------------------------------------------------------------------------------|
| <b>BPE0020I</b> | <b>ABEND LIMIT REACHED FOR<br/>component type USER EXIT<br/>MODULE module</b> |
|-----------------|-------------------------------------------------------------------------------|

**Explanation**

The indicated user exit module reached the abend limit for the exit type as specified in the ABLIM parameter on the EXITDEF statement.

In the message text:

**component**

The name of the owning component of the user exit type being called (BPE or FRX)

**type**

The up-to-8 character exit type name of the exit that abended

**module**

The load module name of the exit that abended

**System action**

No further calls are made to the indicated user exit module until the next refresh of the exit. Refreshing the user exit resets the abend count for the exit to zero, allowing it to be called again.

**User response**

Examine the SDUMP, the SYS1.LOGREC entries, or both to determine the problem with the user exit. Correct the problem with the exit, relink the exit to the job's libraries, and issue the REFRESH USEREXIT command. The REFRESH USEREXIT command loads a copy of the corrected exit and causes BPE to resume calling the exit module whenever exits of its type are invoked.

**Module**

FRXUXCL\$

|                 |                                                                           |
|-----------------|---------------------------------------------------------------------------|
| <b>BPE0021E</b> | <b>ABEND code IN BPE SVC INIT<br/>MODULE FRXSVC1\$, PSW=psw1<br/>psw2</b> |
|-----------------|---------------------------------------------------------------------------|

**Explanation**

An abend occurred while module FRXSVC1\$ was in control. Module FRXSVC1\$ is the module that initializes the IMS Database Recovery Facility BPE SVC routine (an internal SVC used by BPE and IMS Database Recovery Facility). FRXSVC1\$ processing is protected by an internal ESTAE, which attempts to retry from the abend and clean up any global resources (common storage, MVS Enqueues) that FRXSVC1\$ obtained. Message BPE0021E is issued to alert the operator that an abend occurred.

In the message text:

**code**

The abend code. For system abends, the format of code is Sxxx, where xxx is the 3-digit abend code in hexadecimal. For user abends, the format of code is Udddd, where dddd is the 4-digit abend code in decimal.

**psw1**

The first word of the PSW at abend.

## **psw2**

The second word of the PSW at abend.

### **System action**

IMS Database Recovery Facility BPE collects diagnostic data about the abend, and then resumes execution in a cleanup routine within FRXSVC0\$. This routine attempts to release any global resources that FRXSVC0\$ obtained as a part of its processing. The routine returns to the FRXSVC0\$'s caller, which can choose to continue processing or to end abnormally.

The first time that FRXSVC0\$ abends, its ESTAE takes an SDUMP of the address space, and causes a record to be written to the SYS1.LOGREC data set to document the abend. If FRXSVC0\$ abends a second time or more, its ESTAE does not take another SDUMP. However, it writes a record to SYS1.LOGREC.

### **User response**

Save any dump, SYSLOG, and SYS1.LOGREC information and contact the IBM Software Support.

### **Module**

FRXSVC0\$

---

|                 |                                                        |
|-----------------|--------------------------------------------------------|
| <b>BPE0022E</b> | <b>ABEND code IN BPE SVC PROCESSING, PSW=psw1 psw2</b> |
|-----------------|--------------------------------------------------------|

---

### **Explanation**

An abend occurred during BPE SVC processing. The IMS Database Recovery Facility BPE SVC module (FRXSVC0\$) establishes an ESTAE to protect its processing. This ESTAE attempts to retry from the abend and clean up any global resources (common storage, MVS ENQs) that FRXSVC0\$ obtained. Message BPE0022E is issued to alert the operator that an abend occurred.

In the message text:

#### **code**

The abend code. For system abends, the format of code is Sxxx, where xxx is the 3-digit abend code in hexadecimal. For user abends, the format of code is Udddd, where dddd is the 4-digit abend code in decimal.

## **psw1**

The first word of the PSW at abend.

## **psw2**

The second word of the PSW at abend.

### **System action**

IMS Database Recovery Facility BPE collects diagnostic data about the abend, and then resumes

execution in a cleanup routine within FRXSVC0\$. This routine attempts to release any global resources that FRXSVC0\$ obtained as a part of its processing, and returns to the caller of FRXSVC0\$, which can choose to continue processing or to end abnormally.

The first time that FRXSVC0\$ abends, its ESTAE takes an SDUMP of the address space, and causes a record to be written to the SYS1.LOGREC data set to document the abend. If FRXSVC0\$ abends a second time or more for the same SVC call, its ESTAE does not take another SDUMP. However, it writes a record to SYS1.LOGREC.

### **User response**

Save any dump, SYSLOG, and SYS1.LOGREC information and contact the IBM Software Support.

### **Module**

FRXSVC0\$

---

|                 |                                 |
|-----------------|---------------------------------|
| <b>BPE0023I</b> | <b>command COMMAND REJECTED</b> |
|-----------------|---------------------------------|

---

### **Explanation**

A command was issued, but could not be processed. The command is rejected.

This message might not be issued when a command is rejected. IMS Database Recovery Facility might choose to issue its own message rejecting the command, and might request that IMS Database Recovery Facility BPE not issue the BPE0023I message.

In the message text:

#### **command**

The command that was rejected. If the command that was entered was the MVS stop command (P jobname), then MVS STOP displays as the command. For all other commands, the command verb and resource type are displayed (for example, UPDATE TRACETABLE or REFRESH USEREXIT) .

### **System action**

The command is not processed.

### **Module**

FRXMOD0\$, FRXCMD1\$

---

|                 |                               |
|-----------------|-------------------------------|
| <b>BPE0024E</b> | <b>command COMMAND FAILED</b> |
|-----------------|-------------------------------|

---

### **Explanation**

The processing for a command failed.

This message might not be issued when a command fails. IMS Database Recovery Facility might choose to

issue its own message about a command failure, and might request that IMS Database Recovery Facility BPE not issue the BPE0024E message.

In the message text:

**command**

The command that failed. If the command that was entered was the MVS stop command (P jobname), then MVS STOP displays as the command. For all other commands, the command verb and resource type are displayed (for example, UPDATE TRACETABLE or REFRESH USEREXIT).

**System action**

The command failed.

**Module**

FRXMOD0\$, FRXCMD1\$

---

**BPE0025I            STOP OF FRD IS IN PROGRESS**

**Explanation**

An MVS STOP command was issued for IMS Database Recovery Facility. IMS Database Recovery Facility BPE is processing the stop request.

**System action**

IMS Database Recovery Facility BPE initiates a shutdown of the address space.

**Module**

FRXMOD0\$

---

**BPE0026E            CLEANUP FAILURE RC=rc RSN=rsn  
                         component**

**Explanation**

An error occurred during IMS Database Recovery Facility BPE resource cleanup processing. Some resources might not be properly cleaned up.

IMS Database Recovery Facility BPE establishes a resource manager routine to clean up global resources when an address space using IMS Database Recovery Facility BPE services ends. If the resource manager cannot clean up a particular resource, it issues a BPE0026E message.

In the message text:

**rc**

The return code, if applicable, from the failing service. This code might help to identify the cause of the failure.

**rsn**

The reason code, if applicable, from the failing service.

**component**

A short text string that identifies the component or resource that could not be cleaned up. Possible components are:

- **BPE SVC:** Cleanup failed for the IMS Database Recovery Facility BPE SVC service. The return code and reason code in the message are from the IMS Database Recovery Facility BPE SVC EOMCLEANUP function. This error indicates that some of the SVC functions registered by the ending address space might not have been properly deregistered.
- **ALESERV:** A call to the MVS ALESERV service to obtain the current address space's STOKEN failed. The return code in the message is the return code from the ALESERV macro; the reason code is always zero.

**System action**

IMS Database Recovery Facility BPE resource cleanup processing continues with the next resource, and the address space ends.

**User response**

Depending on the cause of the cleanup failure, IMS Database Recovery Facility BPE might have taken an SDUMP. If this is the case, save the dump, SYSLOG, and SYS1.LOGREC information and contact IBM Software Support. If IMS Database Recovery Facility BPE did not take an SDUMP, obtain the return code, reason code, and component from the BPE0026E message and contact IBM Software Support with this information.

**Module**

FRXRSM0\$

---

**BPE0027E            ABEND code IN BPE RESMGR  
                         PROCESSING, PSW=psw1 psw2**

**Explanation**

An abend occurred during BPE resource manager processing while ending an address space running with IMS Database Recovery Facility BPE services. The IMS Database Recovery Facility BPE resource manager module (FRXRSM0\$) establishes an ESTAE to protect its processing. This ESTAE attempts to retry from the abend and to continue cleaning up global resources. Message BPE0027E is issued to alert the operator that an abend occurred.

In the message text:

**code**

The abend code. For system abends, the format of code is Sxxx, where xxx is the 3-digit abend code in hexadecimal. For user abends, the format of code is Udddd, where dddd is the 4-digit abend code in decimal.

**psw1**

The first word of the PSW at abend.

**psw2**

The second word of the PSW at abend.

**System action**

IMS Database Recovery Facility BPE collects diagnostic data about the abend, and then resumes execution in the main FRXRSM0\$ routine, which attempts to continue cleanup with the next resource after the one that was being processed when the abend occurred.

The first time that FRXRSM0\$ abends, its ESTAE takes an SDUMP of the address space, and causes a record to be written to the SYS1.LOGREC data set to document the abend. If FRXRSM0\$ abends a second time or more, its ESTAE does not take another SDUMP. However, it writes a record to SYS1.LOGREC.

**User response**

Save any dump, SYSLOG, and SYS1.LOGREC information and contact IBM Software Support.

**Module**

FRXRSM0\$

---

|                 |                                                                              |
|-----------------|------------------------------------------------------------------------------|
| <b>BPE0028I</b> | <b>SDUMP FAILED FOR <i>abend</i> ABEND,<br/>RC=<i>rc</i>, RSN=<i>rsn</i></b> |
|-----------------|------------------------------------------------------------------------------|

---

**Explanation**

IMS Database Recovery Facility BPE issued an SDUMP call to MVS to produce a dump of the address space after an abend, but the SDUMP was not successful.

In the message text:

***abend***

The abend code for which the dump was taken. For system abends, the format of code is Sxxx, where xxx is the 3-digit abend code in hexadecimal. For user abends, the format of code is Udddd, where dddd is the 4-digit abend code in decimal.

***rc***

The return code from the MVS SDUMP macro.

***rsn***

The reason code from the MVS SDUMP macro.

**System action**

The SDUMP is skipped.

**User response**

Use the return and reason codes from the MVS SDUMP macro to determine the cause of the SDUMP failure. These return and reason codes are documented in *z/OS MVS Programming: Authorized Assembler Services Reference, Volume 3 (LLA-SDU)*. If appropriate, correct the cause of the failure so that future dumps are not lost.

**Module**

FRXINIT\$, FRXRSM0\$, FRXSVC1\$, FRXSVC0\$,  
FRXSYES\$

---

|                 |                                                       |
|-----------------|-------------------------------------------------------|
| <b>BPE0029I</b> | <b>DAE SUPPRESSED DUMP FOR<br/><i>abend</i> ABEND</b> |
|-----------------|-------------------------------------------------------|

---

**Explanation**

IMS Database Recovery Facility BPE issued an SDUMP call to MVS to produce a dump of the address space after an abend, but the SDUMP was suppressed by MVS dump analysis and elimination (DAE).

IMS Database Recovery Facility BPE recovery routines gather symptom string data related to an abend, and provide this data to MVS when an SDUMP is requested. If DAE is enabled, MVS will suppress duplicate dumps, for example, dumps that have symptom strings identical to previously captured dumps. DAE is controlled through the MVS ADYSETxx PARMLIB member and the MVS SET DAE command. For details on specifying DAE options, see *z/OS MVS Initialization and Tuning Reference*.

In the message text:

***abend***

The abend code for which the dump was taken. For system abends, the format of code is Sxxx, where xxx is the 3-digit abend code in hexadecimal. For user abends, the format of code is Udddd, where dddd is the 4-digit abend code in decimal.

**System action**

The SDUMP is skipped. Note that a BPE-generated dump will be suppressed if its symptom string matches a previous dump, and if the current DAE setting in ADYSETxx is either SUPPRESS OR SUPPRESSALL.

**User response**

None. This message is informational.

**Module**

FRXINIT\$, FRXRSM0\$, FRXSVC1\$, FRXSVC0\$,  
FRXSYES\$

---

**BPE0030I**      *displayheader*

**Explanation**

Message BPE0030I is the message number associated with the header line or lines generated by the DISPLAY command verb for IMS Database Recovery Facility BPE resources.

In the message text, *displayheader* is one or more lines of header information associated with the output from a DISPLAY command.

**Module**

FRXCTRA\$, FRXCURF\$

---

**BPE0031E**      **verb COMMAND IS INVALID**  
**verb resourcetype COMMAND IS INVALID**

**Explanation**

A command was issued that BPE did not recognize. Even if you entered a valid command verb, this message might be generated if you omitted a required resource type or specified a resource type to which the verb does not apply. For example, if you entered:

```
DISPLAY NAME(DISP)
```

You would receive the error message:

```
BPE0031E DISPLAY COMMAND IS INVALID
```

This error occurs because the DISPLAY command requires a resource type on which it is to operate. Similarly, if you entered:

```
REFRESH TRACETABLE NAME(AWE)
```

You would receive the error message:

```
BPE0031E REFRESH TRACETABLE COMMAND IS INVALID
```

This error occurs because the REFRESH command applies to the USEREXIT resource type, not the TRACETABLE resource type.

In the message text:

**verb**

Is the command verb from the command that was issued.

**resourcetype**

Is the resource type from the command that was issued, if present.

**System action**

The command is ignored.

**User response**

Reenter the command with the correct verb, or verb and resource type.

**Module**

FRXCMD0\$, FRXCMD1\$

---

**BPE0032I**      **verb COMMAND COMPLETED**  
**verb resourcetype COMMAND COMPLETED**

**Explanation**

A command completed processing. This message is issued in two cases:

- For commands that generate a variable number of lines of output, such as DISPLAY commands. Message BPE0032I is issued to indicate the end of the command output.
- For commands that generate no other messages. Message BPE0032I is issued to provide feedback that command processing is complete.

Message BPE0032I will not be issued when a command generates a fixed number of lines of output, such as a single response line or error message line.

In the message text:

**verb**

Is the command verb from the command that was issued.

**resourcetype**

Is the resource type from the command that was issued, if present.

**System action**

None.

**User response**

None. This message is informational.

**Module**

FRXCMD1\$

---

**BPE0033E**      **MISSING COMMAND VERB**

**Explanation**

A command was entered that is missing a command verb, which is the first word of the command string. You receive this message only if you enter a command

that consists of only *keyword(value)* parameters. For example, if you enter NAME (DISP) OWNER (BPE), the BPE0033E message is issued. This command does not contain a non-keyword parameter to act as the command verb.

**System action**

The command is ignored.

**User response**

Reenter the command with the proper command verb.

**Module**

FRXCMD0\$

---

**BPE0034E      NO MATCH FOUND FOR  
                  resource type WITH NAME "name"**

**Explanation**

A name or wild card pattern on the NAME parameter of a command did not match any instances of the requested resource type. For example, the command:

```
DISPLAY TRACETABLE NAME (ZZZZ, QQ*)
```

generates a response similar to the following:

```
NO MATCH FOUND FOR TRACETABLE WITH NAME "ZZZZ"
NO MATCH FOUND FOR TRACETABLE WITH NAME "QQ*"
```

In the message text:

**resource type**

Is the resource type specified in the command.

**name**

Is the name or wild card pattern specified on the NAME parameter that did not match any instance of the resource type.

**System action**

If some of the names listed on the NAME parameter of the command matched existing resource instances, the command processes against those instances. If none of the names listed on the NAME parameter matched existing resource instances, the command is ignored.

**User response**

Reenter the command with the proper resource type names.

**Module**

FRXCTRA\$, FRXCURF\$

---

**BPE0035E      INVALID VALUE "value" FOR  
                  KEYWORD keyword**

**Explanation**

The value specified on the indicated keyword was not valid. For example, the command:

```
UPDATE TRACETABLE NAME (DISP) OWNER (ABCD)
LEVEL (GROUND)
```

generates a response similar to the following:

```
INVALID VALUE "ABCD" FOR KEYWORD OWNER
INVALID VALUE "GROUND" FOR KEYWORD LEVEL
```

In the message text:

**value**

The keyword parameter value that is in error.

**keyword**

The keyword whose value is in error.

**System action**

The command is ignored.

**User response**

Reenter the command with the proper value on the indicated keyword.

**Module**

FRXCTRA\$, FRXCURF\$

---

**BPE0036E      INTERNAL COMMAND  
                  PROCESSING ERROR, service  
                  RC=rc**

**Explanation**

An internal error in IMS Database Recovery Facility BPE occurred while processing the command. For details on the failure see the service return code, as indicated by *service* and *rc*, in IMS Database Recovery Facility Service Return Codes.

In the message text:

**service**

Is the name of the failing service that prevented the command from processing.

**rc**

Is the return code from the failing service call.

**System action**

The command is ignored.

**User response**

Some internal errors that result in message BPE0036E could be due to temporary conditions, such as a temporary lack of storage. Other errors indicate possible logic errors within product code. Reenter the command and see if the error reoccurs. If it does, issue a console dump of the address space immediately after entering the command and contact IBM Software Support. To ensure that trace information necessary to the resolution of the problem is available, you can turn on the following IMS Database Recovery Facility BPE traces at LEVEL(HIGH): CMD, SSRV, STG, DISP.

**Module**

FRXCMD0\$, FRXCMD1\$, FRXCTRA\$, FRXCURF\$

---

|                 |                                         |
|-----------------|-----------------------------------------|
| <b>BPE0038E</b> | <b>BPE STACK MANAGER INTERNAL ERROR</b> |
|-----------------|-----------------------------------------|

---

**Explanation**

The IMS Database Recovery Facility BPE stack storage manager was unable to obtain storage for a new stack block.

**System action**

The IMS Database Recovery Facility BPE stack storage manager will issue an SDUMP the first time this condition arises to gather diagnostic data about the problem. The stack manager then waits the caller until a stack block is eventually freed by another thread.

**User response**

This message is issued when storage is not available to satisfy an internal request for stack storage within the IMS Database Recovery Facility address space. IMS Database Recovery Facility BPE will suspend the unit of work that is requesting additional stack storage until such storage becomes available. However, this suspension could lead to reduced function and performance in IMS Database Recovery Facility. You should increase the available region size for the IMS Database Recovery Facility address space and shut it down and restart it as soon as possible. If you continue to get this message, save the SDUMP produced by IMS Database Recovery Facility BPE and contact IBM Software Support.

**Module**

FRXSTKM\$

---

|                 |                                                   |
|-----------------|---------------------------------------------------|
| <b>BPE0039E</b> | <b>BPE STACK MANAGER MULTIPLE INTERNAL ERRORS</b> |
|-----------------|---------------------------------------------------|

---

**Explanation**

The IMS Database Recovery Facility BPE stack storage manager encountered multiple cases where it could not obtain storage for a new stack block.

**System action**

The IMS Database Recovery Facility BPE stack storage manager will issue BPE0038E the first time it is not able to get a new stack block when one is needed. Subsequent failures do not generate additional BPE0038E messages. However, if the stack manager repeatedly cannot get storage, it will eventually issue message BPE0039E, indicating that the storage problem is likely a chronic one, and that action should be taken quickly to resolve the storage shortage. In addition to issuing the BPE0039E message, IMS Database Recovery Facility BPE will also take a second SDUMP after the repeated failures to capture additional diagnostics.

**User response**

This message is issued by the IMS Database Recovery Facility BPE stack manager after repeated failures to obtain storage. It is unlikely that the address space will function well at this point. You should increase the available region size for the address space and immediately shut it down and restart it. If you continue to get this message, save the SDUMP produced by IMS Database Recovery Facility BPE and contact IBM Software Support.

**Module**

FRXSTKM\$

---

|                 |                                             |
|-----------------|---------------------------------------------|
| <b>BPE0041E</b> | <b>UNABLE TO ALLOCATE REQUESTED STORAGE</b> |
|-----------------|---------------------------------------------|

---

**Explanation**

The IMS Database Recovery Facility BPE storage allocation service, BPEGETM, could not allocate storage requested by its caller. This message will only be issued the first time that BPEGETM could not obtain storage.

**System action**

BPEGETM will return an error return code to its caller.

**User response**

It is possible that the storage shortage which lead to the BPE0041E message is simply due to the region size for the address space being set too small. Increase the region size by changing the REGION=

parameter on the address space JCL, and restart the address space.

If the problem persists, save any dumps and contact IBM Software Support. If no dumps were produced for this problem, you should take a console dump of the address space that is getting the BPE0041E message.

## Module

FRXSTG0\$

## BPE service return codes

This reference section provides detailed information about service return codes issued by Base Primitive Environment (BPE). Use this information to help you with troubleshooting.

### BPEATTCH

#### Routing code

Attach a TCB (thread control block).

#### Return Code

##### Explanation

#### X'00000004'

The TCB index passed to the BPEATTCH program was not a valid index.

#### X'00000008'

There is no TCB type of the requested type defined to BPE.

#### X'0000000C'

A new TCB of the indicated type could not be attached because there were no available entries for it in the BPE TCB table. The maximum number of TCBs of this type is already attached.

#### X'00000010'

The MVS ATTACH for the new TCB failed.

#### X'00000014'

The caller is a non-thread running under the parent TCB of the TCB that was requested to be attached. This is not a valid environment for the BPEATTCH call.

#### X'00000020'

Internal processing error: The enqueue of an AWE (asynchronous work element) to the TCB attach processor for the requested TCB type failed.

#### X'00000024'

Internal processing error: The BPEWAIT for a response from the TCB attach processor for the requested TCB failed.

#### X'00000028'

Internal processing error: The routine called by BPEATTCH enqueued an AWE to the TCB attach processor for the requested TCB; however, the DDB address it saved in the AWE was invalid, and the attach processor rejected the attach request.

#### X'0000002C'

Internal processing error: The routine called by BPEATTCH enqueued an AWE to the TCB attach

processor for the requested TCB; however, the TCB attach processor rejected the AWE because it contained an invalid function code.

#### X'00000030'

Internal processing error: BPEATTCH was unable to obtain a DQCB (dispatcher queue control block) for the new TCB -- the BPECBGET call for the DQCB failed.

#### X'000000F8'

The requested function is not present in the BPE system at run time. This is likely due to a macro or module mismatch.

#### X'000000FC'

The parameter list version generated by the macro is not supported by the called service routine. This is probably due to a macro or module mismatch.

### BPEAWSRV

#### Routing code

#### Return Code

##### Explanation

#### X'00000004'

The CREATE request for an AWE server was successful for at least one, but not all of the requested AWE servers.

#### X'00000008'

Unable to create a thread for an AWE server. The Thread CREATE service, BPETHDCR, failed.

#### X'0000000C'

Unable to obtain an AQSB (asynchronous queue server block) for an AWE server being created.

#### X'00000010'

The BPE TCB token that was passed on the TCBTKN parameter for an AWE server CREATE request was not a valid token.

#### X'00000020'

The AQHE (asynchronous queue handler element) passed to create routine was not a valid AQHE. This is likely an internal processing error.



**X'00000024'**

BPEAWSRV could not determine the TCB under which it should create the server. The TCB is determined by the value of TCBTKN on this macro, or by the value of the TCBTYPE on the BPEAWDEF macro when the queue server is defined. If neither of these parameters are coded, then the server create function does not know under what TCB to create the server thread.

**X'00000028'**

If AWEIDX was specified, the value passed was not a valid AWE server index. If TYPE was specified, the type indicated was not a valid AWE server type according to the AWE server definition module. For BPE AWE servers, macro BPEAWIX and module FRXAQHT\$ are probably out of sync. For user-product AWE servers, the user-product type definition macro and module are probably out of sync.

**X'0000002C'**

The NUMTHDS parameter specified an invalid number of threads. It must specify a number from 1 to 255.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPEBPCRE****Routing code**

Create a buffer pool.

**Return Code****Explanation****X'00000004'**

The value specified on the INCREMENT parameter was zero or negative.

**X'00000008'**

The value specified on the MINBUF parameter was zero or negative.

**X'0000000C'**

The value specified on the MAXBUF parameter was less than the MINBUF value.

**X'00000010'**

The value specified on the PRIMARY parameter was negative.

**X'00000014'**

The value specified on the UPPERLIMIT parameter was less than the PRIMARY or MAXIMUM values.

**X'00000018'**

The subpool specified on SP was invalid or unsupported.

**X'0000001C'**

The buffer format specified on the FORMAT parameter is an invalid format for the current version of the buffer create service.

**X'00000020'**

Either the pointer to the 4-character buffer pool type string was zero, or the string was all blanks or all nulls.

**X'00000024'**

A buffer pool of the type specified on the TYPE parameter has already been created. The token of the already-existing pool is returned in the location indicated by the TOKEN parameter.

**X'00000040'**

A BPEGETM call failed for control block storage for the new buffer pool.

**X'00000044'**

A BPEGETM call failed for the primary allocation of buffer storage.

**X'00000048'**

BPELOAD failed for one or more of the BPE buffer manager service modules for the requested pool format.

**X'0000004C'**

A BPELAGET call failed when trying to get the buffer pool create or destroy latch.

**X'00000050'**

An abend occurred in buffer pool create processing. The buffer pool was not created.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPEBPGET****Routing code**

Get a buffer from a buffer pool.

**Return Code****Explanation****X'00000004'**

No buffer could be obtained because the pool is at its upper limit.

**X'00000008'**

No buffer could be obtained because a BPEGETM call for buffer storage failed.

**X'00000020'**

The value specified on the SIZE parameter was zero or negative.

**X'00000024'**

The value specified on the PREFIX was not between 0 and 256.

**X'00000028'**

A nonzero value was specified on the PREFIX parameter, but the buffer pool was not created with FORMAT=PREFIX.

**X'0000002C'**

The address of the word to receive the returned buffer address (BUFFERPTR) was not valid (it was in the first 4K of storage).

**X'00000030'**

The token passed to BPEBPGET was not a valid token for any BPE-managed buffer pool.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPECBGET**
**Routing code**

Get a control block.

**Return Code****Explanation****X'00000004'**

A bad CBTE address was passed to the CB get routine. The get routine validates that the CBTE address it is passed has the character string "CBTE" as the first word of the block. If it does not, this return code is passed back to the caller. This is likely an internal systems error of some kind.

**X'00000008'**

Storage was unavailable to satisfy the request.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPECMSRV**
**Routing code**

Submit a command for processing.

**Return Code****Explanation****X'00000020'**

The command was rejected by the command action routine and was not processed.

**X'00000024'**

The command failed.

**X'00000028'**

Unknown command.

**X'0000002C'**

Missing command verb.

**X'00000040'**

Bad command pointer. The address of the command passed to BPECMSRV was zero.

**X'00000044'**

Bad command length. The length of the command passed to BPECMSRV was zero.

**X'00000048'**

Bad BPEWPRNT exit parameter list length. The length of an exit parameter list passed to BPECMSRV was a zero or negative value.

**X'0000004C'**

The BPECMSRV caller was not running as a BPE thread.

**X'00000060'**

BPE was unable to obtain the necessary working storage to process the command.

**X'00000064'**

BPE was unable to create a new thread to process the command.

**X'00000068'**

BPE could not process the command because a BPEPOST call to start the command processing thread failed.

**X'0000006C'**

BPE could not process the command because a BPELAGET call for a latch to serialize the command failed.

**X'00000070'**

The command action module that processed the command returned an undefined return code to BPE. The result of the command processing is not known.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro to module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro to module mismatch.

---

**BPEGETM****Routing code**

GETMAIN storage.

**Return Code****Explanation****X'00000004'**

An invalid or unsupported subpool was specified.

**X'00000008'**

A zero or negative value length was requested.

**X'00000010'**

Unable to obtain the requested 31-bit storage. The caller requested 31-bit storage, but only 24-bit storage was available.

**X'0000000C'**

Unable to obtain the requested storage (MVS GETMAIN failed).

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPELAGET****Routing code**

Get a latch.

**Return Code****Explanation****X'00000004'**

The latch was not obtained because it was owned in exclusive mode by another caller (WAIT=NO only).

**X'00000008'**

The latch was not obtained because it was owned in shared mode by one or more callers (WAIT=NO only).

**X'0000000C'**

A request for a latch in shared mode was not granted because there was a waiter for the latch in exclusive mode (WAIT=NO only).

**X'00000010'**

The latch was successfully obtained in exclusive mode, and was transferred from another thread to the current thread.

**X'00000020'**

The latch was not obtained because the caller already owns the latch in exclusive mode.

**X'00000024'**

The latch was not obtained because the latch services module was unable to WAIT the caller. This is usually due to an error in the caller's environment: either the caller is not in TCB mode, or the caller's ECB is already in a WAIT state.

**X'00000028'**

A request was not granted for a latch in shared mode because the latch had been transferred to the caller's ECB by another thread; for this reason the request must be in exclusive mode.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPELAREL****Routing code**

Release a latch.

**Return Code****Explanation****X'00000004'**

The latch was not released because it is owned in exclusive mode by the caller.

**X'00000008'**

The latch was not released because it is owned in exclusive mode by another caller.

**X'0000000C'**

The latch was not released because it is owned in shared mode by one or more other callers.

**X'00000010'**

The latch was not released because it is not owned.

**X'00000014'**

The latch was not transferred because the ECB, to which the latch was to be transferred, was waiting to get the latch in shared mode.

**X'00000018'**

The latch was released (MODE=EX), but a protocol error might have occurred. The latch had been transferred to the caller's ECB from a different ECB, but the caller had never accepted the latch.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPELOAD****Routing code**

Load a module.

**Return Code****Explanation****X'00000004'**

For a list-form load (MODLIST), at least one of the modules in the list had an error.

**X'0000000C'**

Specified module was not found.

**X'00000010'**

BLDL for module failed. This error is most likely due to an internal error with BLDL, such as an out-of-storage condition.

**X'00000014'**

Load of module failed. The module was found in the library, but LOAD returned a nonzero return code.

**X'00000028'**

TCB=JSTP required an AWE to be queued to JSTP TCB for LOAD processing, and the AWE enqueue failed.

**X'00000030'**

Internal failure in load processing.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPELOADC****Routing code**

Load/call/delete a module.

**Return Code****Explanation****X'00000004'**

The LOAD and call of the module were successful, but the subsequent DELETE of the module failed. The return code that the module passed back in

register 15 is in the word or register specified by MODRC.

**X'00000008'**

The module specified on EP or EPLOC could not be loaded. The contents of the word or register specified by MODRC is undefined.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPEPARSE****Routing code**

Parse data.

**Return Code****Explanation****X'00000004'**

The parser definition grammar passed on PADEF was not a valid BPEPADEF grammar.

**X'00000008'**

The control block storage passed on CBSTG was not big enough to contain the control blocks that needed to be built to contain the parsed input data. Issue BPEPARSE again with a larger amount of CBSTG storage. Note that the data in the CBSTG area is incomplete, and should not be used.

**X'0000000C'**

The CBSTG address passed to the parsing service was zero.

**X'00000010'**

The input data address passed to the parsing service was zero.

**X'00000014'**

An internal error occurred in the parsing service.

**X'00000040'**

An invalid keyword was detected in the input data.

**X'00000044'**

An unknown positional parameter was encountered in the input.

**X'00000048'**

A keyword parameter was specified with only an equal sign (KEYWORD=value), but the keyword was defined as having a sublist of values. Sublists can only be specified in parenthesis. An equal sign can only be used by itself (i.e., with no parentheses) if a keyword has a single value.

**X'0000004C'**

The input ended before all of a sublist or keyword had been parsed.

**X'00000050'**

A keyword was encountered (KEYWORD or KEYWORD=) when a value was expected.

**X'00000054'**

An input number being parsed was out of the range allowed for its output field length. For decimal numbers, the numbers must be less than or equal to 255, 65535, 16777215, and 2147483647 for 1, 2, 3, and 4 byte fields, respectively. For hexadecimal numbers, the number might not have digits more than 2 times the number of bytes in the output field.

**X'00000058'**

A parameter value defined as decimal contained non-decimal digits.

**X'0000005C'**

A parameter value defined as hex contained non-hex digits.

**X'00000060'**

A parameter value defined as a key value parameter had an unknown key value.

**X'00000064'**

A keyword parameter was seen multiple times and was not defined as being repeatable (REPEAT=YES on BPEPADEF).

**X'00000068'**

A parameter defined with REQUIRED=YES on BPEPADEF was not found in the input data (omitted).

**X'0000006C'**

A character parameter value was longer than the defined output field length, and the parameter definition did not specify TRUNC=YES on BPEPADEF to allow truncation.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPERSDPS****Routing code**

Read a PDS member.

**Return Code****Explanation****X'00000004'**

The member name address passed to the read PDS routine was 0.

**X'00000008'**

The output buffer pointer address passed to the read PDS routine was 0.

**X'0000000C'**

The output buffer length word address passed to the read PDS routine was 0.

**X'00000010'**

The value specified on the RIGHTMAR parameter was either negative or not less than the actual LRECL of the data set.

**X'00000040'**

OPEN failed for the PDS.

**X'00000044'**

The PDS specified was not in fixed format.

**X'00000048'**

The member specified on the MEMBER parameter was not in the PDS.

**X'0000004C'**

A BSAM READ for the specified member failed.

**X'00000050'**

Unable to obtain storage for a buffer to hold the PDS data.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPESVC****Routing code****Return Code****Explanation****X'00000004'**

The BPE SVC routine was unable to GETMAIN storage for an internal work area. R0 = return code from GETMAIN.

**X'00000008'**

The BPE SVC routine could not establish a recovery environment using the MVS ESTAE macro. R0 = return code from ESTAE.

**X'0000000C'**

The BPE SVC routine could not locate its main control block (the BPESVCA block). This is likely due to errors initializing the BPE SVC routine or other BPE system services.

**X'00000010'**

An invalid SVC function code was passed to the BPE SVC routine in register 0. R0 contains the invalid function code.

**X'00000014'**

The BPE SVC function requested requires the caller to be authorized (supervisor state and key 0-7), and the caller was not authorized.

**X'00000018'**

An abend occurred in the BPE SVC routine or in the named function routine. R0 contains the abend code.

**X'0000001C'**

The address for the name of the named function was zero (REGISTER, DEREGISTER, CALL).

**X'00000020'**

The BPE SVC module was unable to get a SYSTEM level ENQ on the resource SYSZBPE SVCNFHTB, which is required in order to serialize the named function hash table. R0 = the return code from the ENQ macro.

**X'00000024'**

The named function indicated could not be found (QUERY, REGISTER, DEREGISTER, CALL).

**X'00000040'**

The PSW key indicated for the named function being registered was not in the range 0-7. Named function routines must run in a system key (REGISTER).

**X'00000044'**

The BPE SVC routine could not obtain storage for an SVC Function Block (SVCF) for the function being registered. R0 = return code from GETMAIN (REGISTER).

**X'00000048'**

The named function being registered is already registered and the REPLACE parameter did not allow it to be replaced (REGISTER).

**X'00000060'**

The named function that was requested to be called requires the caller to be authorized (supervisor state, key 0-7), and the caller was not authorized (CALL).

**X'00000064'**

The BPE SVC routine could not obtain work area storage for the named function routine to be called. R0 = return code from GETMAIN (CALL).

**X'00000080'**

The STOKEN address passed to the EOMCLEANUP function was zero (EOMCLEANUP).

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPETHDCR****Routing code**

Create a thread.

**Return Code****Explanation****X'00000004'**

Create failed: An invalid BPE TCB token was passed on the TCBTKN parameter.

**X'00000008'**

Create failed: A bad TCB index value was passed on the TCBIDX parameter.

**X'0000000C'**

Create failed: A zero routine address was passed on the ROUTINE parameter.

**X'00000010'**

Create failed: A bad TCB table entry address was passed into the thread create routine. The BPETHDCR macro determines the TCBT address based on whichever of the TCBIDX, TCBTKN, or TCBTYPE parameters is specified. Ensure that this parameter is correctly coded.

**X'00000014'**

Create failed: Unable to get storage for a Thread Control Block (THCB) for the thread.

**X'00000018'**

Create failed: Unable to get stack storage for the thread.

**X'0000001C'**

Create failed: The initial POST of the thread failed.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPETIMER****Routing code**

Issue a timer request.

**Return Code****Explanation****X'00000004'**

FUNC=TIMER request was canceled.

**X'00000008'**

The request was canceled because the timer service has been shutdown.

**X'0000000C'**

FUNC=CANCEL request failed because no outstanding timer request with a matching token could be found.

**X'00000010'**

The timer AWE enqueued to the timer service contained an invalid function code.

**X'00000014'**

The FUNC=TIMER request indicated that the expiration value was specified as TOD. The current version of timer services does not support TOD=.

**X'00000018'**

The DINTVL parameter contained a non-decimal digit.

**X'00000020'**

The minutes field in the DINTVL parameter was greater than 59. It must be between 0 and 59.

**X'00000024'**

The seconds field in the DINTVL parameter was greater than 59. It must be between 0 and 59.

**X'00000028'**

The FUNC=TIMER request could not be processed because the timer service could not get storage for a timer queue element (TQUE) for the request.

**X'0000002C'**

The BPETIMER request failed: MVS STIMER service returned a nonzero return code.

**X'00000080'**

Unable to obtain an AWE from control block services (BPECBGET) to initiate timer request. Register 0 contains the return code from BPECBGET. No timer request was issued.

**X'00000084'**

Unable to enqueue initiate timer request. AWE enqueue to timer server failed. R0 contains the return code from BPEAWSRV FUNC=ENQ. Note that the state of the AWE is indeterminate. It is probably on an AWE queue, but it might not be processed.

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**BPETRsrv****Routing code**

Request trace services.

**Return Code****Explanation****X'00000004'**

There is no trace table of the requested type defined to BPE.

**X'00000008'**

Input trace token was not valid or was not for a trace table of the indicated type (FUNC=DELETE, GETUDATA, SETUDATA).

**X'0000000C'**

The trace table specified was not defined as a trace type that supports multiple trace tables. The CREATE and DELETE functions might be issued only for trace types that were defined as multiple types. See BPETRDEF and BPETRIX for further information (FUNC=CREATE and DELETE).

**X'00000010'**

The user data address passed on USERDATA was invalid (less than 00001000) (FUNC=GETUDATA and SETUDATA).

**X'00000014'**

The trace table specified does not support user data (UDATALEN was not specified or specified zero on the trace table's BPETRDEF definition). (FUNC=GETUDATA and SETUDATA)

**X'00000030'**

Unable to get the storage required for a trace table header entry for the new trace table. (FUNC=CREATE)

**X'00000034'**

Unable to get the storage required for the new trace table. (FUNC=CREATE)

**X'000000F8'**

The requested function is not present in the BPE system at execution time. This is likely due to a macro or module mismatch.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

**FRXUXRF\$****Routing code**

Load user exits.

**Return Code****Explanation****X'00000001'**

Couldn't read BPE PROCLIB (Rsn code = BPERDPDS RC)

**X'00000002'**

Couldn't read user-product PROCLIB (Rsn code = BPERDPDS RC)

**X'00000003'**

BPEGETM failed for user product parse data area (Rsn code = BPEGETM RC)

**X'00000004'**

BPEGETM failed for BPE parse data area (Rsn code = BPEGETM RC)

**X'00000005'**

BPEPARSE failed for BPE exit list PROCLIB member (Rsn code = BPEPARSE RC)

**X'00000006'**

BPEPARSE failed for user prod exit list PROCLIB member (Rsn code = BPEPARSE RC)

**X'00000010'**

BPECBGET failed for a UXIB block (Rsn code = BPECBGET RC)

**X'00000011'**

BPELAGET failed for global user exit latch (UXRF\_Lhdr) (Rsn Code = BPELAGET RC)

**X'00000012'**

User exit refresh failed due to abend in refresh processing (Rsn code = abend code)

**X'00000013'**

BPECBGET failed for a static work area for a user exit module (Rsn code = BPECBGET RC)

**X'00000014'**

BPEGETM failed for storage for a user exit module (Rsn code = BPEGETM RC)

**X'00000015'**

BLDL failed for a user exit module (Rsn code = BLDL RC)

**X'00000016'**

LOAD failed for a user exit module (Rsn code = LOAD RC)

---

**FRXSVCIS\$****Routing code**

Initialize the BPE SVC module.

**Return Code****Explanation****X'00000004'**

Unable to load init routine FRXSVCIS\$ to initialize the BPE SVC.

**X'00000008'**

ESTAE create failed for ESTAE for FRXSVCIS\$. R0 = return code from ESTAE.

**X'0000000C'**

BLDL failed for BPE SVC module. R0 = return code from BLDL.

**X'00000010'**

GETMAIN failed for storage for the SVC module. R0 = return code from GETMAIN.

**X'00000014'**

LOAD failed for the SVC module. R0 = return code from LOAD.

**X'00000018'**

Exclusive ENQ on resource name SYSZBPE SVCUPDTE failed. R0 = return code from ENQ.

**X'0000001C'**

Exclusive ENQ on SYSZSVC TABLE failed. R0 = return code from ENQ.

**X'00000020'**

Unable to find MVS ECVTCTBL (CVT to CVTECVT to ECVTCTBL = 0). The MVS system must be at least SP 4.2.0, with APARs OW13312 and OW13315 applied.

**X'00000024'**

GETMAIN failed for BPE Global Function Vector Table (BPEGFVT) block. R0 = return code from GETMAIN.

**X'00000028'**

GETMAIN failed for SVC control area (SVCA) block. R0 = GETMAIN return code.

**X'0000002C'**

SVCUPDTE failed for the BPE SVC. R0 = SVCUPDTE return code.

**X'00000040'**

An ABEND occurred while initializing the SVC. R0 = the abend code.

**X'000000FC'**

The parameter list version generated by the macro is not supported by the called service routine. This is likely due to a macro or module mismatch.

---

## BPE user abend codes

This reference section provides detailed information about user abend codes issued by Base Primitive Environment (BPE). Use this information to help you with troubleshooting.

For each abend code, the following information is provided where applicable:

**Explanation:**

The Explanation section explains what the abend code means, why it occurred, and what its variable entry fields are (if any)



**System Action:**

The System Action section explains what the system will do next

**User Response:**

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

During processing, you might encounter abend codes that are issued by IMS or DFS. For these abend codes, see *IMS Messages and Codes*.

**3400****Explanation**

An error during early IMS Database Recovery Facility BPE initialization prevented IMS Database Recovery Facility BPE services from starting. Register 15 at the time of abend contains a subcode that can be used to determine the reason for the abend. The subcodes and their meanings are described in the following list:

**X'01'**

FRXINIT\$ could not establish an ESTAE recovery routine.

**X'02'**

IMS Database Recovery Facility BPE stack storage services initialization failed.

**X'03'**

A request to obtain a stack block for early IMS Database Recovery Facility BPE initialization failed.

**X'04'**

A lower-level service initialization module could not be loaded.

**X'05'**

A lower-level service initialization module could not complete the initialization of its service.

**X'06'**

An error occurred while parsing the IMS Database Recovery Facility BPE configuration PROCLIB member specified on the BPECFG= parameter in the MVS startup parameters.

**X'07'**

IMS Database Recovery Facility detected a problem during its early initialization and indicated to IMS Database Recovery Facility BPE that it should abend.

**X'08'**

An error occurred creating an MVS name or token for the IMS component address space.

**X'09'**

A mismatch occurred between the BPE version used by IMS Database Recovery Facility and the version of the IMS Database Recovery Facility BPE modules.

**X'0A'**

The IMS Database Recovery Facility BPE SVC routine (FRXSVC0\$) could not be installed.

**X'0B'**

The IMS Database Recovery Facility BPE resource manager (FRXRSM0\$) could not be established.

**System action**

The address space is ended abnormally.

**User response**

Take the appropriate action based on the subcode you received.

**For subcodes X'01' through X'05' and X'07' through X'0B'**, the problem is one of the following:

- An environmental error, such as missing modules in your STEPLIB data set or a storage shortage in the address space.
- An internal error within IMS Database Recovery Facility BPE or IMS Database Recovery Facility.

Messages that further explain the reason for the error might precede this abend. See the job log output from the failed job for explanatory messages.

If you cannot determine and correct an environmental cause for the problem, save any dump and SYSLOG information, and contact the IBM Support Center.

**For subcode X'06'**: Although this abend can result from internal or environmental errors, its most probable cause is a problem with the IMS Database Recovery Facility BPE configuration PROCLIB member. If BPE cannot find or read the member, abend U3400 follows message BPE0002E, which indicates the problem. Correct the problem and restart the IMS Database Recovery Facility address space.

If there is an error parsing the member, this abend follows message BPE0003E, which indicates where in the member the error occurred. Correct the configuration member and restart the IMS Database Recovery Facility address space.

**For subcode X'09'**: This abend occurs when the IMS Database Recovery Facility modules and the IMS Database Recovery Facility BPE modules in your STEPLIB data sets are at different versions. This abend follows message BPE0013E, which indicates the different levels of the BPE modules. Ensure that you are running the correct level of IMS Database Recovery Facility BPE modules.

## Module

FRXINIT\$

### 3401

#### Explanation

An error during the initialization of a new TCB in the IMS Database Recovery Facility address space prevented the TCB from starting. Register 15 at the time of abend contains a subcode that can be used to determine the reason for the abend. Abend code 3401 subcode meanings are described in the following list:

##### X'01'

FRXSYIN\$ could not establish an ESTAE recovery routine.

##### X'02'

A request to obtain a stack block for the TCB initialization failed.

##### X'03'

IMS Database Recovery Facility BPE could not initialize the BPE internal dispatcher for the new TCB.

##### X'04'

A lower-level TCB initialization module could not complete initialization because of an error or unexpected condition.

##### X'05'

IMS Database Recovery Facility BPE could not create an internal server that runs under the new TCB.

##### X'06'

IMS Database Recovery Facility BPE could not load a lower-level TCB initialization module, and therefore could not complete the initialization of the new TCB's environment.

#### System action

If the TCB being created is not a required TCB, the IMS Database Recovery Facility address space continues to function; otherwise, the IMS Database Recovery Facility address space is ended abnormally.

#### User response

The problem is one of the following:

- An internal error within IMS Database Recovery Facility BPE or IMS Database Recovery Facility.
- An environmental error, such as missing modules in your STEPLIB data set, or a storage shortage in the address space.

Messages that further explain the reason for the error might precede this abend. See the job log output from the failed job for explanatory messages.

If you cannot determine and correct an environmental cause for the problem, save any dump and SYSLOG information, and contact IBM Software Support.

## Module

FRXSYIN\$

### 3402

#### Explanation

An error occurred during the ending of a TCB in the IMS Database Recovery Facility address space. Register 15 at the time of the abend contains a subcode that can be used to determine the reason for the abend. Code 3402 subcodes are as follow:

##### X'01'

A routine involved in the cleanup process for the ending TCB encountered an error that prevented the cleanup from completing.

##### X'02'

A routine involved in the cleanup process for the IMS Database Recovery Facility address space encountered an error that prevented the cleanup from completing.

##### X'03'

IMS Database Recovery Facility BPE could not start required processes to end subtask TCBs under the TCB that was ending.

#### System action

If the TCB being ended is not a required TCB, the IMS Database Recovery Facility address space continues to function. If the TCB is required, or if the IMS Database Recovery Facility address space is in the process of being shut down, the IMS Database Recovery Facility address space is ended abnormally.

#### User response

The problem is one of the following:

- An internal error within IMS Database Recovery Facility BPE or IMS Database Recovery Facility.
- An environmental error, such as missing modules in your STEPLIB data set or a storage shortage in the address space.

Messages that further explain the reason for the error might precede this abend. See the job log output from the failed job for explanatory messages.

If you cannot determine and correct an environmental cause for the problem, save any dump and SYSLOG information, and contact IBM Software Support.

**Module**

FRXSYTR\$

---

**4095****Explanation**

IMS Database Recovery Facility BPE issues this abend once for each TCB in the address space when an unrecoverable abend occurs and it is necessary to abnormally end the address space.

**System action**

The IMS Database Recovery Facility address space is ended abnormally.

**User response**

BPE issues this abend after one of the address space's TCBs has an unrecoverable abend. It is used to abnormally end the remaining TCBs; it is not itself part of the original problem. Examine the job log output for the first TCB to abend with a code other than 4095 in order to determine the cause of the subsequent 4095 abends.



---

## Part 9. Reference: General

Reference information supports the tasks that you must complete to install, customize, and use IMS Database Recovery Facility.

**Topics:**

- [Chapter 29, “How to read syntax diagrams,” on page 357](#)



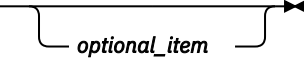
## Chapter 29. How to read syntax diagrams

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

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

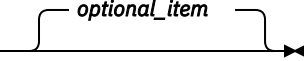
➤ *required\_item* ➤

- Optional items appear below the main path.

➤ *required\_item*  ➤

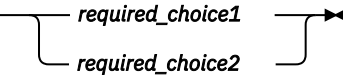
The diagram shows a horizontal line starting with a double arrow pointing right. Below this line, a bracket labeled *optional\_item* connects the line to itself, indicating a path that can be taken or not.

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

➤ *required\_item*  ➤

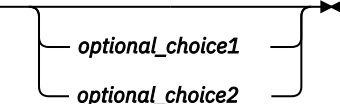
The diagram shows a horizontal line starting with a double arrow pointing right. Above this line, a bracket labeled *optional\_item* connects the line to itself, indicating a path that can be taken or not.

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

➤ *required\_item*  ➤

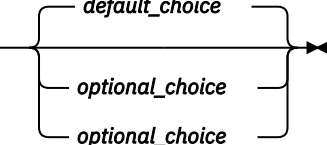
The diagram shows a horizontal line starting with a double arrow pointing right. Below this line, two items, *required\_choice1* and *required\_choice2*, are stacked vertically. A bracket connects the main path to this stack, indicating that one of the choices must be taken.

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

➤ *required\_item*  ➤

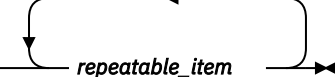
The diagram shows a horizontal line starting with a double arrow pointing right. Below this line, two items, *optional\_choice1* and *optional\_choice2*, are stacked vertically. A bracket connects the main path to this stack, indicating that one of the choices can be taken.

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

➤ *required\_item*  ➤

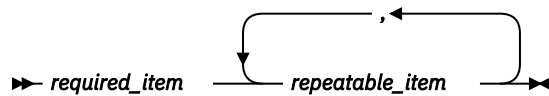
The diagram shows a horizontal line starting with a double arrow pointing right. Above this line, *default\_choice* is shown. Below the line, two items, *optional\_choice* and *optional\_choice*, are stacked vertically. A bracket connects the main path to this stack, indicating that one of the choices can be taken.

- An arrow returning to the left, above the main line, indicates an item that can be repeated.

➤ *required\_item*  ➤

The diagram shows a horizontal line starting with a double arrow pointing right. Above this line, a loop arrow points from the line back to itself, indicating that the item can be repeated. The item *repeatable\_item* is shown on the main path.

If the repeat arrow contains a comma, you must separate repeated items with a comma.



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

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



## Notices

---

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive

Armonk, NY 10504-1785  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

#### COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

#### Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com)<sup>®</sup> 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

## A

- abend reason codes [263](#)
- accessibility
  - overview [10](#)
- ADD IB() command
  - Build Index function of FPA [129](#)
  - IMS Index Builder [124](#)
- administration
  - database checklist [159](#)
- architecture [7](#)

## B

- batch condition codes [273](#)
- batch recovery control statements
  - ADD command [105](#)
  - ADD command examples [109](#)
  - commands [105](#)
  - REMOVE command [111](#)
  - START command [113](#)
  - timestamp format [119](#)
- BPE
  - commands [319](#)
  - messages [325](#)
  - messages and codes [325](#)
  - service return codes [342](#)
  - user abend codes [350](#)
- BPE commands [319](#)
- BPE messages [325](#)
- BPE messages and codes [325](#)
- BPE service return codes [342](#)
- BPE user abend codes [350](#)
- BSYIN control statements [37](#)

## C

- change accumulation (CA) data set [6](#)
- CHANGE.DBDSGRP [147](#)
- coexistence [16](#)
- components [7](#)
- configuration
  - BSYIN control statements [37](#)
  - environmental control statements [40](#)
  - FRXDRFxx [33](#)
  - IB, DFSPREC0, Build Index function of FPA address space [100](#)
  - integrated auxiliary utilities [63](#)
  - JCL [93](#)
  - master address space [93](#)
  - pointer checker address space [101](#)
  - recovery sort subordinate address space [98](#)
  - system level backup API [36](#)
  - system level backup image copies [34](#), [35](#)
- cookie policy [359](#), [361](#)

## D

- DBRC commands
  - CHANGE.DBDSGRP [147](#)
  - INIT.DBDSGRP [147](#)
  - LIST.DBDSGRP [148](#)
  - NOTIFY.RECOV [148](#)
- diagnostic information
  - gathering [275](#)
- disaster recovery site (DR) [6](#)
- documentation
  - accessing [9](#)
  - sending feedback [9](#)

## E

- environmental control member [33](#)
- environmental control statements [40](#)
- example reports [283](#)

## F

- fallback [16](#)
- full recovery [6](#)

## H

- hardware and software requirements [13](#)

## I

- INIT.DBDSGRP [147](#)
- installation
  - coexistence [16](#)
  - fallback [16](#)
  - hardware and software requirements [13](#)
  - migration [16](#)
  - operational requirements [13](#)
  - requirements [13](#)
- integrated auxiliary utilities
  - ADD IB() command [124](#)
  - ADD IC() command [131](#)
  - ADD PC() command [140](#)
  - architecture [9](#), [21](#)
  - capture [29](#)
  - control flow [25](#)
  - data set processing [29](#)
  - RAUX control task [25](#)
  - return code handling [144](#)
  - return codes [27](#)
  - specification [123](#)
  - SYSOUT reports [27](#), [28](#)
  - terminology [19](#)
  - usage [20](#)
  - utility control statement parameters [63](#), [69](#)
  - utility control statements [63](#)

integrated auxiliary utilities (*continued*)

viewing [28](#)

WTO messages [27](#)

integrated auxiliary utility processing

calling IMS HP Image Copy [153](#)

calling IMS HP Pointer Checker [153](#)

calling IMS Index Builder or DFSPREC0 [153](#)

message and report output handling [154](#)

## L

legal notices

cookie policy [359](#), [361](#)

notices [359](#)

programming interface information [359](#)

trademarks [359](#), [360](#)

LIST.DBDSGRP [148](#)

## M

management

assigning security [163](#)

auditability [164](#)

CA input data sets [161](#)

data integrity [163](#)

database availability [163](#)

DBRC controls [160](#)

DEDBs with MADS [163](#)

environment [162](#)

image copy frequency [162](#)

image copy input types [160](#)

input data sets [160](#)

input sources [159](#)

log data sets [161](#)

managing output [162](#)

nonstandard image copy recovery [160](#)

performance [165](#)

recovery from secondary sources [162](#)

restart [164](#)

spill data space [164](#)

stacked image copy output [162](#)

starting database and areas [163](#)

time stamp recovery [169](#)

master address space

DDNAME reference [93](#)

report DDNAME reference [96](#)

messages

abend reason codes [263](#)

BPE messages and codes [325](#)

overview [207](#)

migration [16](#)

## N

notices [359](#)

NOTIFY.RECOV [148](#)

## O

optimization scenarios [173](#)

overview [4](#)

## P

performance

improvement [165](#)

input parallelism [166](#)

virtual tape system [166](#)

point-in-time recovery (PITR) [6](#)

problems

diagnostic information about [275](#)

process flow [7](#)

programming interface information [359](#)

## R

RAUX control task [25](#)

reader comment form [9](#)

reason codes

logic error abend [269](#)

service error abend [263](#)

recovery job statistics report [279](#)

recovery point [6](#)

recovery, perform

building a recovery list [151](#)

define recovery scope [149](#)

defining physical data sets [152](#)

post-recovery activities [154](#)

starting recovery [152](#)

taking databases offline [151](#)

using integrated auxiliary utility processing [152](#)

report, recovery job statistics [279](#)

reports and utilities [277](#)

reports, sample

Full point-in-time recovery [295](#)

Full recovery using environment control statements [308](#)

Full recovery with IC() and PC() utilities [311](#)

Image copy or incremental image copy output [284](#)

Output report content [283](#)

Point-in-time recovery [290](#)

Sample reports processing [284](#)

Verify of the point-in-time recovery [298](#)

VERIFY(ALLOC) recovery [302](#)

VERIFY(OPEN) recovery for a point-in-time [305](#)

requirements [13](#)

restrictions

database and area [14](#)

esoteric name tape drive [15](#)

seasonal time change timestamp [14](#)

## S

sample reports [283](#)

scenarios

basic environment [175–177](#)

incremental image copy, offline DBs [185–189](#)

incremental image copy, online DBs [191–196](#)

optimization [173](#)

rebuild indexes [197–199](#)

recover HALDB, online reorg [201–203](#)

recover to current time [179](#), [180](#)

validate with pointer checking [181](#), [182](#)

screen readers and magnifiers [10](#)

service information [9](#)

statistics report, recovery job [279](#)

- summary of changes [3](#)
- support
  - required information [275](#)
- support information [9](#)
- syntax diagrams
  - how to read [357](#)
- System Level Backup (SLB) [6](#)
- system level backup API activation utility [36](#)
- system level backup image copies [34](#), [35](#)

## T

- technotes [9](#)
- terminology [6](#)
- time stamp recovery
  - RECOVGRP [170](#)
  - with Pitr [169](#)
  - with Pitr and database updates on multiple UOR [171](#)
  - with Pitr and UOR [170](#)
  - without Pitr [169](#)
- timestamp format [119](#)
- timestamp recovery (TSR) [6](#)
- trademarks [359](#), [360](#)
- troubleshooting [205](#)

## U

- usage scenarios
  - optimization [173](#)
- using [103](#)
- UTILGBL [70](#)
- utilities and reports [277](#)
- utility control statements
  - Build Index function of FPA [75](#)
  - IMS HP Image Copy [78](#)
  - IMS HP Pointer Checker [87](#)
  - IMS Index Builder [70](#)
  - UTILGBL [70](#)

## V

- verify recovery assets (VERIFY) [6](#)









Product Number: 5655-ISR

SC27-8441-02

