IBM IMS Recovery Solution Pack for z/OS
2.1

IMS Database Recovery Facility Extended Functions User's Guide

IBM

SC27-8442-02

Note:

Before using this information and the product it supports, read the information in "Notices" on page 191.

Third Edition (June 2022)

This edition applies to Version 2.1 of IBM IMS Recovery Solution Pack for z/OS: IMS Database Recovery Facility Extended Functions (program number 5655-ISR) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-8442-01.

Contents

About this information

IBM® IMS Recovery Solution Pack for z/OS®: IMS Database Recovery Facility Extended Functions (also referred to as IMS Database Recovery Facility Extended Functions) is an IMS tool that you can use to help ensure that your database environment is recoverable before you have to do a recovery.

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

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 IBM 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 Extended Functions
- Install and operate IMS Database Recovery Facility Extended Functions
- Customize your IMS Database Recovery Facility Extended Functions environment
- Diagnose and recover from IMS Database Recovery Facility Extended Functions problems
- Use IMS Database Recovery Facility Extended Functions 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 Documentation 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. Extended Functions overview

IMS Database Recovery Facility Extended Functions inspects information that is stored in the RECON data sets, system catalogs, and in the repositories of IMS Tools Knowledge Base. Extended Functions detects problems that can affect database recovery and will ensure that your IMS database environment is recoverable.

The topics in this section provide you with an overview of Extended Functions:

Topics:

- Chapter 1, “Extended Functions overview,” on page 3
- Chapter 2, “Extended Functions scenarios,” on page 15

Chapter 1. Extended Functions overview

IMS Database Recovery Facility Extended Functions is used at both local and disaster recovery sites for identifying problems that affect can database recovery. At the disaster recovery site, the functions also assist you in ensuring that all of the required resources are present and help you prepare the RECON data sets to perform recoveries.

Extended Functions inspects information that is stored in the RECON data sets, system catalogs, and in the repositories of IMS Tools Knowledge Base.

Extended Functions detects problems that can affect database recovery. This tool provides you the ability to create clean recovery points across multiple IMS systems and to perform synchronized log switches.

Extended Functions also offers features to determine valid database recovery points, makes sure that all of the resources that are necessary to perform a recovery are present, and provides you the ability to manage the RECON data sets to ensure that the data sets match the disaster recovery environment.

Topics:

What's new in Extended Functions

This topic summarizes the technical changes for this edition.

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

Revision markers follow these general conventions:

- Only technical changes are marked; style and grammatical changes are not marked.
- If part of an element, such as a paragraph, syntax diagram, list item, task step, or figure is changed, the entire element is marked with revision markers, even though only part of the element might have changed.
- If a topic is changed by more than 50%, the entire topic is marked with revision markers (so it might seem to be a new topic, even though it is not).

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

SC27-8442-02 - June 2022

- Updated information about hardware and software requirements.

SC27-8442-01 - November 2018

- PI93260 - update SOURCE parameter information.
- Note concerning securing IMS commands that are issued through Extended Functions:

Note: RACFCHK and security options in the RECONID are completely independent processes. Only RECONID security options are used for IMS Database Recovery Facility Extended Functions.

When issuing commands, Extended Functions performs security checking based on the definition in the RECONID record. The security checks are performed against the RACF® security class specified in the RECONID record. The security processing normally performed by IMS Tools Online System Interface (RACFCHK=*value*) is bypassed.

SC27-8442-00 - October 2016

- Maintenance updates for IMS Recovery Solution Pack 2.1.
- Information pertaining to the inclusion of IMS Recovery Expert as a product belonging to the IMS Recovery Solution Pack 2.1.
- Information pertaining to the inclusion of IMS Solution Packs: Data Sensor as a product belonging to the IMS Recovery Solution Pack 2.1.
- Information about using IMS Tools Setup for initial product installations, first-time users, and product evaluations.

Extended Functions terminology

IMS Database Recovery Facility Extended Functions 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.

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.

IMS command (IMSCMD)

This feature allow you to issue IMS commands in batch.

You can view the output of the commands that is returned in a report file.

IMS Database Recovery Facility

IMS database tool that can recover databases using Full Recovery, Time Stamp Recovery, or Point In Time Recovery.

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.

RECON cleanup (RCU)

This feature allows you to prepare a set of RECON data sets for your disaster recovery environment by modifying the contents of the RECONs to match their DR site.

Recovery health check (HCHECK)

The HCHECK function consists of many different options to check and verify the recoverability of your environment.

The majority of these options interrogate and verify information in the RECON data sets, and compare the information in the RECONs with information that is outside of the RECONs (like information that is in the system catalog and in other places).

You can specify each piece of information that you want checked. And you can exclude the pieces of information that you do not want checked.

HCHECK uses information from:

- RECON data sets.
- IMS Recovery Expert repository for System Level Backups (SLBs).
- MVS™ catalog.

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.

Recovery point create (RPCR)

This feature allows you to create a recovery point for one or more databases by issuing synchronized /DBRECOVERY or /DBDUMP commands for the databases, waiting for the databases to be deallocated on all of your IMS systems, and then optionally restarting them.

You can also optionally perform a synchronized log switch on all active IMS systems in the group with RPCR.

Recovery point identification (RPID)

This feature interrogates information in the RECON data sets to determine common recovery points for one or more databases.

Recovery time range (RTR)

RPID input parameter containing two times within which the search for recovery time spans and recovery points is bounded.

Recovery time span (RTS)

Span of time within which recovery points can be selected and to which a database or group of databases can be recovered.

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:

- RECON data sets.
- IMS Recovery Expert repository for System Level Backups (SLBs).
- MVS catalog.

Extended Functions features and benefits

By using IMS Database Recovery Facility Extended Functions, you can save time and resources by detecting and correcting problems that can affect database recoveries before you need to recover your databases.

Extended Functions helps you perform database recoveries at the local and disaster recovery site by assisting you in many aspects of IMS database recovery tasks by providing the following features:

- IMS Command Interface
- Health Checker
- Recovery Point Creation
- Recovery Point Identification
- Verify Recovery Assets
- RECON Data Set Clean up
- IMS Database Recovery Facility JCL Generation

The tool also provides a process for creating a copy of your live RECON data sets. The copied data sets are accessed instead of your live RECON data sets, which eliminates I/O against your live RECON data sets.

IMS Command interface

The IMS command interface (IMSCMD) allows you to execute IMS type 1 AOI commands in a batch environment. IMSCMD uses the IMS Tools Online System Interface for entering commands and must first be installed in each IMS control region in which commands are issued. IMSCMD can be used in conjunction with other functions of the tool. The output of the commands is written to a report for viewing.

Health Checker

The Health Checker function (HCHECK) analyzes information in the RECON data sets and identifies problems which could impact IMS database recovery.

HCHECK currently checks for over 20 different conditions which could impact IMS database recovery. Health Check uses information from the RECON data sets and the z/OS catalog when looking for potential problems.

If IMS Recovery Expert is installed, Health Check also uses information from the Recovery Expert repository to include System Level Backups in its analysis and verification.

Recovery Point Creation

The Recovery Point Creation function (RPCR) provides the ability to create recovery points for one or more databases by deallocating the database using /DBR or /DBD command.

This process causes the allocation records in the RECON data sets to be updated, indicating a period of time when the database is not allocated by any IMS subsystem, thus creating a valid recovery point.

When complete, RPCR also optionally restarts the databases to bring them back into their original state. RPCR also has the ability to issue a /SWI LOG command across connected IMS systems to create archived logs that are associated with the RPCR time.

Recovery Point Identification

The Recovery Point Identification function (RPID) analyzes records in the RECON data sets and locates valid recovery time spans for one or more databases.

A Recovery Time Span (RTS) is a period of time within which a database is not allocated and may be recovered. RPID analyzes database allocation records in the RECON data sets to identify RTSs to which one or more databases can be recovered using timestamp recovery (TSR).

RPID also analyzes database recovery records in the RECONs to identify Recovery Point timestamps to which individual databases can be recovered using TSR. These recovery points include times when a database was the object of:

- A batch IC run
- Online IC run (in cases where the databases are not truly allocated)
- A complete CA run
- A recovery run (full or partial)
- An offline reorganization run

All recovery points are recorded in RECON recovery records.

Run times of HALDB online reorganizations and IMS Online Reorganization Facility are also analyzed; for these RPID indicates whether the databases were truly allocated and thus whether the run time is a valid TSR recovery point.

Verify Recovery Assets

The Verify Recovery Assets function (VERIFY) examines the RECON data sets and the IMS Recovery Expert repository to identify the recovery assets that are required to recover one or more databases.

If IMS Recovery Expert is installed, VERIFY also examines the Recovery Expert repository to identify and verify System Level Backups.

VERIFY ensures that the necessary assets exist and that they are valid before performing a recovery. Specifically, the VERIFY function performs one or all of the following functions:

- Lists the recovery assets that are required to recover the specified databases
- Allocates all recovery assets in order to ensure that they exist
- Opens all recovery assets to ensure that they are valid

RECON Data Set Clean up

The RECON Data Set Clean up function (RCU) is used to prepare a set of RECON data sets for your disaster recovery environment.

RCU uses a copy of your RECON data sets and alters them to match the recovery environment at the disaster recovery site. These RECONs can then be used for a disaster recovery test or in the event of an actual disaster.

IMS Database Recovery Facility JCL Generation

The IMS Database Recovery Facility JCL generation function allows you to easily create and run IMS Database Recovery Facility JCL for performing IMS database recoveries and creating incremental image copies.

This function parses the information from the Recovery Point Identification report and presents it in an easy to use ISPF interface.

Extended Functions components

IMS Database Recovery Facility Extended Functions consists of several components that work together to provide the product features.

The main components of Extended Functions include:

BPE architecture

The Extended Functions product is built on the Base Primitive Environment (BPE) architecture.

BPE provides most of the underlying address space services such as dispatcher services, tracing, diagnostics, and storage services.

The other components use these services when performing their tasks.

The main address space

The main address space controls job step initialization and setup, drives the requested functions, and performs address space clean up and termination.

This component parses all input and prepares the environment for processing the specified functions.

The main address space drives each function by performing the following steps:

- Parse all function control cards and initialize the function environment.
- Invoke the function and allow it to process.
- Check for successful or abnormal completion.
- Clean up the function environment.
- Terminate the function.

Functions

Each function is processed by a separate set of execution routines.

These functions are driven by the main address space and process the requested action, as specified by the control card input.

Each function generates its own report and error messages.

Report Writer

The report writer provides an interface for writing reports to the IMS Tools Knowledge Base repository, to report data sets, or to SYSOUT.

Other components make requests of the report writer to generate report output. Based on JCL and control card specifications, the report writer writes data to all necessary locations.

ISPF Interface

The Extended Functions ISPF interface allows you to view the data from the Recovery Point Identification (RPID) report and generate IMS Database Recovery Facility JCL for performing database recoveries and creating incremental image copies.

Extended Functions architecture and process flow

IMS Database Recovery Facility Extended Functions operates as a batch job in a single address space.

The job can be set up to run in a job scheduler and perform required functions on a regular basis. A single Extended Functions job can consist of multiple job steps. Each job step can process a separate function or a single job step can process multiple functions.

Within a single job step, the order in which the functions are processed is controlled by the tool. The functions are processed consecutively in a predefined order. No parallel processing of functions is performed.

The functions are processed in the following order:

- RECON cleanup
- Recovery Point Create

- Recovery Point Identification
- Verify
- Health Checker
- IMS Command

If you specify a function multiple times in the same job step, all occurrences of that function are processed consecutively before the next function is attempted.

If you want to control the order in which the functions are processed, you must use multiple job steps. For example, within a single job:

- STEP1 might issue IMS commands that display and stop certain IMS resources.
- STEP2 might then execute the RPCR process to create a clean recovery point for one or more databases.
- STEP3 might then issue IMS commands to restart those resources.

Extended Functions uses the IMS Tools Knowledge Base or repository for the following purposes:

- Output from reports that are generated by Extended Functions can be stored in an output repository.

  IMS Tools Knowledge Base has an interface to store and retrieve output reports.

  You can view these reports at a later time, using the IMS Tools Knowledge Base ISPF interface.
- You can also process your RPID reports with the RPID ISPF interface.
- IMS Tools Knowledge Base is used to store RECONID records that describe the IMS environments against which Extended Functions can process.

You control Extended Functions processing by specifying input parameters. You can specify the following 2 types of input parameters:

Configuration parameters

These parameters specify overall job configuration settings.

These parameters are not directly related to any one feature but they control address space setup and processing.

Function control cards

The function control cards drive the processing that is performed by Extended Functions for this run.

These control cards specify the functions that are performed and the information that controls their processing.

The following diagram illustrates the general process flow for a Extended Functions environment that includes input sources, the BPE address space where the functions process, and the other components with which the Extended Functions interacts.

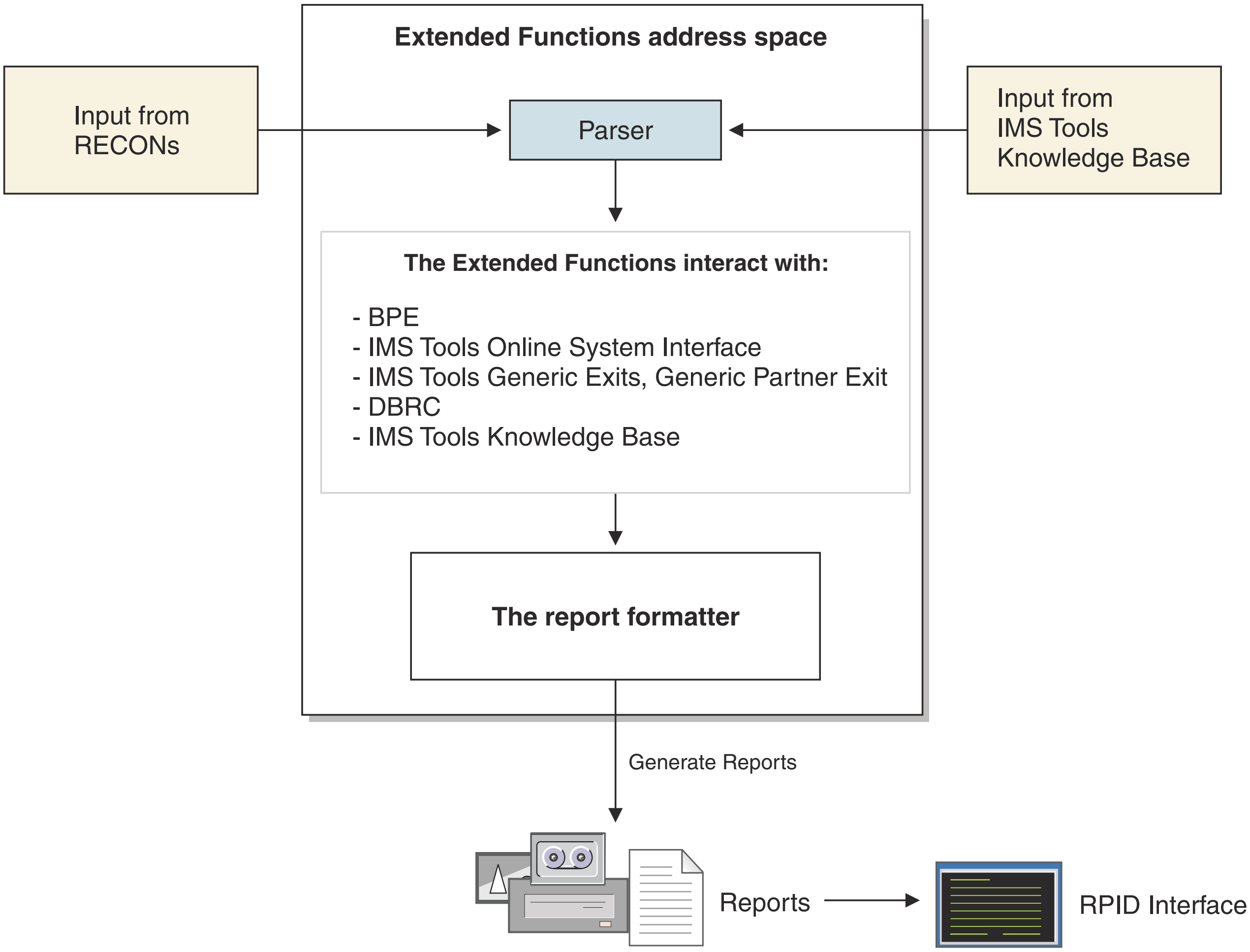

Extended Functions address space
Input from RECONs
Parser
Input from IMS Tools Knowledge Base
The Extended Functions interact with:
- BPE
- IMS Tools Online System Interface
- IMS Tools Generic Exits, Generic Partner Exit
- DBRC
- IMS Tools Knowledge Base
The report formatter
Generate Reports
Reports
RPID Interface

Figure 1. Extended Functions architecture and process flow

Hardware and software requirements

IMS Database Recovery Facility Extended Functions 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 IBM IMS Recovery Solution Pack for z/OS*, GI13-4315.

Hardware requirements

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

Installation software requirements

IMS Recovery Solution Pack requires z/OS 2.3 (5650-ZOS) or later.

Operational software requirements

An operational requisite is defined as a product that is required and must be present or a product that is not required but should be present on the system in order for this product to operate all or some of its functions.

IMS Database Recovery Facility Extended Functions operates with the following IMS versions:

- IMS 15.1 (5635-A06) or later
- IMS Database Value Unit Edition 15.1 (5655-DS5) or later

Common tool requirements

The following common tools, which are provided in IMS Tools Base, are used by IMS Database Recovery Facility Extended Functions:

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

These common tools are not distributed with IMS Recovery Solution Pack but are instead distributed with IMS Tools Base. IMS Tools Base must be installed and configured before IMS Database Recovery Facility Extended Functions can be used.

Some functions and features that require IMS Tools Base include the following:

- Saving reports in IMS Tools Knowledge Base repository
- IMS Tools Online System Interface for issuing IMS commands
- IMS Tools Generic Exits for pausing BMPs and issuing IMS commands

Conditional operational requirements

A conditional operational requisite identifies the following product that is not required for the basic function but is needed at run time for the tool to utilize specific functions.

- The Interactive System Productivity Facility (ISPF), which is included in z/OS for processing Recovery Point Identification (RPID) through TSO terminal.

Service updates and support information

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

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

IBM Support: IMS Recovery Solution Pack for z/OS

Product documentation and updates

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

Information on the web

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

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

The IMS Tools Product Documentation web page includes:

- Links to IBM Documentation for the user guides ("HTML")
- PDF versions of the user guides ("PDF")
- Program Directories for IMS Tools products
- 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 is important in helping us provide the most accurate and highest quality information. If you have any comments about this or any other IMS Tools information, you can take one of the following actions:

- Click the Feedback button at the top of the IBM Documentation topic that you are commenting on.
- Click the Contact Us tab at the bottom of any IBM Documentation topic.
- Send an email to ibmdocs@us.ibm.com. Be sure to include the book title, topic or section title, specific text, and your comment.

To help us respond quickly and accurately, please include as much information as you can about the content you are commenting on, where we can find it, and what your suggestions for improvement might be.

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. Extended Functions scenarios

IMS Database Recovery Facility Extended Functions can be used before you need to perform a recovery, to help you to ensure that your database environment is recoverable.

The scenarios are presented by function.

Topics:

HCHECK scenarios

To interrogate DBRC to determine if a database is recoverable, use the HCHECK function to help you make this determination without manually listing all of the necessary DBRC components.

Some of these scenarios might be applicable to your environment.

Scenario 1: Determine the statuses of all image copies related to a specific group of databases

1. Run the HCHECK function with the ICALL option to perform all of the following tasks:

   a. Check whether any image copies are needed or recommended, or whether no image copies or System Level Backups exist for a specific DBDS in the database list.

   b. Generate a report with the image copy status of each DBDS.

Scenario 2: Determine if the databases in your list are available

1. Run the HCHECK function with the BACKOUT option to determine if any of the databases in the list need to be backed out.

   A report is generated with the status of each database in the list that requires a backout.

2. Run the HCHECK function with the PROAUTH option to determine if any of the databases in the list have prohibit authorization turned on and therefore, cannot be accessed.

   A report is generated with the authorization status of each database in the list.

3. Run the HCHECK function with the HALDBNOTINIT option to determine if any of the HALDB databases in the list are not initialized and therefore, cannot be accessed.

   A report is generated with the initialization status of each HALDB partition in the list.

4. Run the HCHECK function with the NOADS option to determine if any of the Fast Path DEDBs in the list are undefined or unavailable.

   A report is generated with the availability status of each Fast Path area in the list.

5. Run the HCHECK function with the ICNEED option to determine if any of the databases in the list require an image copy and are therefore unavailable until an image copy or System Level Backup is taken.

RPCR scenarios

The RPCR function can assist you in creating a valid recovery point.

When attempting to create a valid recovery point during which one or more databases are not being updated, perform the following tasks:

- Manually /DBR or /DBD a single database or group of databases
- Manually start the databases after all of the databases were stopped to create a recovery point during which the databases were not allocated to an online subsystem.

  However, this manual process is error prone because, not only would you have to manually verify that each IMS command is done correctly in all IMS subsystems, you would also manually have to verify that the databases were not started or allocated before all of the databases in the list are deallocated (both batch and online).

  This manual processing could be inconvenient and this processing can cause the databases to remain offline for some amount of time.

The IMS Database Recovery Facility Extended Functions RPCR function can help you create a valid recovery point easily, without manually invoking the IMS commands and simultaneously verify that the DBDSs were not reallocated by any other subsystem or batch job during the deallocation process.

RPCR also pauses any BMPs that have the databases allocated to allow the deallocation of all of the databases in the list to occur simultaneously.

Here are some scenarios that might be applicable to your environment:

Scenario 1: Determine if a large group of allocated databases can be deallocated

In this scenario, you deallocate a large group of databases that are allocated and updated by several online subsystems to create a valid recovery point.

1. Run the RPCR function with PARTIAL(N) to perform all of the following tasks:
   a. Quickly pause any BMPs that have reached a checkpoint or prevent any BMPs that have not yet started during the process.
   b. Successfully deallocate all of the databases in the list for all of the IMS subsystems that have the databases allocated.
   c. Verify that no other jobs or subsystems have accessed the databases during the time it took for the RPCR deallocation to complete.
   d. Immediately restart the databases in their appropriate online systems with little or no impact to the online systems.
   e. Generate a report with the timestamp of the last database that was deallocated by RPCR.

      This timestamp can be used as a common recovery point for all of the databases in the list.

      If any of the databases in the list could not be deallocated, the entire process fails.

      The report indicates which databases caused the process to fail.

Scenario 2: Determine if a small group of unrelated databases can be deallocated

In this scenario, you deallocate a small group of databases that are unrelated to each other.

- Run the RPCR function with the PARTIAL(Y) option to quickly deallocate the databases in the list for all IMS subsystems that have the databases allocated.

  A report is generated with the timestamp that the last database was successfully deallocated by RPCR.

  If any of the databases in the list could not be deallocated, the report indicates which databases failed and which databases were successful.

RPID scenarios

To check whether a single database or group of databases can be recovered to a common recovery point can be a complicated and timely process.

The IMS Database Recovery Facility Extended Functions RPID function can help you to determine whether databases can be recovered to a common recovery point easily without manually listing and comparing all of the DBRC history for each database in the list.

These scenarios might be applicable to your environment:

Scenario 1: Determine a common recovery point for several databases within an application

In this scenario, you determine a common recovery point for several databases within an application for a specific time period.

1. Run the RPID function with a RANGE to perform all of the following tasks:
    a. Determine when all of the databases in the list were deallocated at a common point in time.
    b. Generate a report with the individual database recovery points and a list of common recovery points for all of the databases in the list.

       Any common timestamp within the recovery times that are listed in the report can be used as a valid recovery point for all databases.

Scenario 2: Determine a recovery point for a single database for a specific time period

In this scenario, you determine if all data set assets that are needed to recover a database to a specific timestamp (TSR) can be allocated or opened successfully.

1. Run the RPID function with a RANGE to perform all of the following tasks:
    a. Determine when the database in the list was deallocated.
    b. Generate a report with the individual database recovery points; common recovery points will be identical because just one database was specified.

       Any timestamp within the recovery times that are listed in the report can be used as a valid recovery point for that database.

VERIFY scenarios

You have many methods to choose from to verify assets before a recovery of one or more databases. However, the VERIFY function can help you verify assets before a recovery with one step.

The following scenarios might be applicable to your environment.

Scenario 1: Determine if all assets necessary for a current time DB recovery can be allocated or opened

In this scenario, you determine if all of the data set assets that are needed to recover a database to the current time can be allocated or opened successfully.

1. Run the VERIFY function with TYPE(OPEN) to perform all of the following tasks:
    a. List, allocate, and open all of the assets that are needed to recover a database to the current time.
    b. Generate a report with the status of each asset that is necessary for the recovery.

Scenario 2: Determine if all necessary assets for a TSR can be allocated or opened

In this scenario, you determine if all data set assets that are needed to recover a database to a specific timestamp (TSR) can be allocated or opened successfully.

1. Run the VERIFY function with PARTIAL(Y) to perform all of the following tasks:
    a. Quickly deallocate the databases in the list for all IMS subsystems that have the databases allocated using the RPCR function.
    b. Generate a VERIFY report with the timestamp that the last database was successfully deallocated by RPCR.

       If any of the databases in the list could not be deallocated, the RPCR report will indicate which databases failed and which databases were successful.

Scenario 3: Determine if all necessary assets for a PITR recovery can be allocated or opened

In this scenario, you determine if all of the data set assets that are needed to recover a database to a specific point in time (PITR) can be allocated or opened successfully.

1. Run the VERIFY function with TYPE(OPEN), RCVTIME(timestamp), and RCVTYPE(PITR) to perform all of the following tasks:
    a. List, allocate, and open all of the assets that are needed to recover a database to a specific point in time (PITR).
    b. Generate a report with the status of each asset that must be used in the PITR.

RCU scenarios

IMS Database Recovery Facility Extended Functions can help you determine a valid cleanup time with several functions that are available.

The following scenarios might be applicable to your environment.

Scenario 1: Databases can remain offline during image copies

Perform the steps in the following list to perform an RCU when your databases can remain offline during image copies.

1. Determine the time period that you are going to schedule the Disaster Recovery (DR) backup for.

   Schedule the DR backup at a time with the least amount of system activity.
2. Run the RPCR function to issue the /DBR command on your databases.

   Specify STARTDB(N).

   The RPCR function process occurs quickly.
3. Create batch image copies for all of the databases that are to be recovered at your DR site.
4. Run the IMSCMD function to issue the /STA DB command on your databases.
5. Run the RCU function with a cleanup time (a timestamp from the last batch image copy).

Scenario 2: Databases cannot remain offline during image copies

Perform the steps in the following list to perform an RCU when your databases cannot remain offline during image copies.

1. Determine the time period that you are going to schedule the DR backup for.

   Schedule the DR backup at a time with the least amount of activity.
2. Run the RPCR function to issue the /DBR and /STA commands on your databases.

The RPCR process occurs quickly and should have minimal impact on your online system.

3. Create concurrent image copies for all of the databases that are to be recovered at your DR site.
4. Run the RPCR function again to issue the /DBR and /STA commands on your databases.

   This action returns a single timestamp that is the time that the last database was /DBR'd.

   This timestamp that can be used as your cleanup time in the next step.
5. Run the RCU function with a cleanup time that is the timestamp from last RPCR.

Scenario 3: 24 x 7 environment– databases cannot be quickly /DBR'd and /STA'd

Perform the following steps to perform an RCU when your databases cannot be quickly DB recovered (by using the /DBR command) and started (by using the /STA DB command).

1. Determine the time period that you are going to schedule the DR backup for.

   Schedule the DR backup at a time with the least amount of activity.
2. Create concurrent image copies for all of the databases that are to be recovered at your DR site.
3. Run the RCU function with a cleanup time that is the timestamp from last concurrent image copy stop time.

Part 2. Configuring Extended Functions

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

Before continuing with product and environment configuration, you must first complete the SMP/E installation as outlined in the *Program Directory for IBM IMS Recovery Solution Pack for z/OS*, GI13-4315 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 3, “Configuration processes,” on page 23
- Chapter 4, “Configuration references,” on page 35

Chapter 3. Configuration processes

The following topics help you complete IMS Database Recovery Facility Extended Functions installation and configure the tool to your environment.

Before continuing with this section, make sure that you have performed the SMP/E installation as outlined in the *Program Directory for IBM IMS Recovery Solution Pack for z/OS*, GI13-4315 and then performed the post-installation steps as outlined in the *IMS Recovery Solution Pack: Overview and Customization*, SC27-8440.

Topics:

RECONID locate processing

Before Extended Functions can begin processing, it must locate the RECONID member and be able to retrieve it from the repository.

Information gathered from the RECONID member

The RECONID member contains information that IMS Database Recovery Facility Extended Functions needs in order to process.

This information includes:

1. The eight-byte external name associated with the RECONID member.
2. The RECON1, RECON2, and RECON3 data set names.
3. The XCF group name that is used to communicate with IMS Tools Online System Interface in the IMS control regions.
4. The IMSPLEX name as defined to the RECON data sets, if present.
5. The RACF security class to be used for securing IMS commands that are issued through Extended Functions.

   Note: When issuing commands, Extended Functions performs security checking based on the definition in the RECONID record. The security checks are performed against the RACF security class specified in the RECONID record.

   The security processing normally performed by IMS Tools Online System Interface (RACFCHK=*value*) is bypassed.

RECONID locate process flow

The ways that the RECONID member can be specified, and the order that is used to locate the RECONID member are outlined in the next list.

When the RECONID member is located, the search terminates (in other words, subsequent steps are performed only until the RECONID member is located).

1. If you specify the RECONID() parameter in either the IMSRECFG= PROCLIB member or the IMSREIN DD, this RECONID() parameter value is used to look up the RECONID member in the repository.

   If this RECONID member is not located, the job step terminates.
2. If the RECON1 DD is allocated in the JCL, the data set name that is associated with the DD is retrieved.

A lookup is performed for that data set name to locate the RECONID member.

If no RECONID member with this RECON1 data set name is located, the job step terminates.

3. If the IMSDALIB DD is allocated in the JCL, a BLDL request is performed for member RECON1.

   If found, the RECON1 member is loaded and the RECON1 data set name is retrieved.

   A lookup is performed for that data set name to locate the RECONID member.

   If no RECONID member with this RECON1 data set name is located, the job step terminates.

4. A load is issued for member RECON1 which will attempt to load the member from JOBLIB, STEPLIB, or the link-pack area.

   If loaded successfully, the RECON1 data set name is retrieved.

   A lookup is performed for that data set name to locate the RECONID member.

   If no RECONID member with this RECON1 data set name is located, the job step terminates.

If, after attempting these steps, Extended Functions is unable to locate the RECONID member, the job terminates with error messages and an ABEND.

TIME and RANGE parameter processing

Many of the functions provided by IMS Database Recovery Facility Extended Functions allow or require you to specify a single timestamp or a timestamp range for processing.

There are two parameters that are used by many of the Extended Functions to specify the timestamp requirements:

- TIME() allows you to specify a single timestamp.
- RANGE() allows you to specify (or compute) a begin and end timestamp.

  The RANGE timestamp forms the boundary period that a function uses in its processing.

TIME parameter reference

The TIME parameter is used by any Extended Function that requires only a single timestamp.

TIME parameter format

You can specify the format of the timestamp value in either compressed or punctuated format.

The timestamp value must be enclosed in single quotes.

```
TIME(timestamp)
```

Timestamp format reference: “Timestamp format reference” on page 26.

TIME parameter example

An example of the TIME parameter is:

```
TIME('21/252-08.24.45.7 -8:00')
```

RANGE parameter reference

The RANGE parameter is used by any Extended Function that requires both a begin and end timestamp.

There are three different formats that you can use to specify this parameter.

RANGE parameter format 1

RANGE parameter format 1 allows you to specify a single base timestamp.

This base timestamp is then used to compute the begin and end timestamps, based on the direction, amount, and scale values.

```
RANGE(timestamp, direction, amount, scale)
```

RANGE specification reference:

timestamp

This value specifies the base timestamp that is used to compute the begin and end timestamps.

The timestamp can be expressed in either compressed or punctuated format.

Timestamp format reference: "Timestamp format reference" on page 26.

direction

This value specifies how the amount value is applied (added, subtracted) to the base timestamp when computing the begin and end timestamps.

Acceptable values are: + (plus), - (minus), and +- (plus,minus).

amount

This value specifies the number of seconds, minutes, hours, or days associated with the specified scale unit.

This value is added to or subtracted from the base timestamp value to compute the begin and end timestamps.

scale

This value specifies the time unit that is used to compute the begin and end timestamps.

Acceptable values for scale are SECS, MINS, HRS, or DAYS.

The scale value determines how the amount value is interpreted when computing the begin and end timestamps (seconds, minutes, hours, or days).

The allowed values for amount depend on the scale unit that is specified:

- SECS (1-59 seconds)
- MINS (1-59 minutes)
- HRS (1-23 hours)
- DAYS (1-366 days)

RANGE format 1 examples:

```
HCHECK(MINBAT RANGE('21.322/14:03:45.738290',-,11,HRS))
```

```
HCHECK(MINBAT RANGE('21.322/08:10:56.878818',+,35,MINS))
```

```
HCHECK(MINBAT RANGE('21.322/09:10:56.878618',+-,3,DAYS))
```

The begin and end timestamps are computed as follows:

- If direction is specified as **+** (plus), the base timestamp that is specified becomes the begin timestamp.

  The end timestamp is then computed by adding the amount value to the base timestamp value.
- If direction is specified as **-** (minus), the base timestamp that is specified becomes the end timestamp.

  The begin timestamp is then computed by subtracting the amount value from the base timestamp value.
- If direction is specified as **+-** (plus,minus), the begin timestamp is computed by subtracting the amount value from the base timestamp value.

  The end timestamp is computed by adding the amount value to the base timestamp value.

RANGE parameter format 2

RANGE parameter format 2 allows you to specify a begin timestamp.

The end timestamp is then computed based on the execution time.

```
RANGE(begin-timestamp)
```

Timestamp format reference: “Timestamp format reference” on page 26.

RANGE format 2 examples:

```
HCHECK(MINBAT RANGE('21.329/07:02:16.7'))
HCHECK(MINBAT RANGE('2021.043/08:30:00'))
```

RANGE parameter format 3

RANGE parameter format 3 allows you to specify both a begin and end timestamp.

```
RANGE(begin-timestamp,end-timestamp)
```

Timestamp format reference: “Timestamp format reference” on page 26.

RANGE format 3 examples:

```
HCHECK(MINBAT RANGE('21.329/07:02:16.7','21.330/07:02:16.8'))
HCHECK(MINBAT RANGE('2021.043/08:30:00','2021.045/14:50:00'))
```

Timestamp format reference

You can specify timestamp values in any format that is recognizable to DBRC.

For detailed information about the formats that are recognizable to IMS Database Recovery Control (DBRC), see 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.

Additionally, timestamps must always be enclosed within single quotation marks.

In the following timestamp formats, 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|thmiju [offset]'
```

Note: In the punctuated timestamp format, the delimiter character (|) can be any non-numeric character delimiter (including blank) with the exception of the single quotation mark (').

Compressed timestamps follow this format:

```
'yydddhhmmssthmiju [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, 2021 and 21 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 hours.)

  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 `CHANGE.RECON TIMEZONE(`*`NNNNN, offset`*`)` command. For example (defining RECON command and usage in a punctuated timestamp):

  ```
  CHANGE.RECON TIMEZONE(ABC, +11:45)
  ```

  ```
  ’2021 045 16.24.45.7 ABC'
  ```

  You can use any symbolic string such as GMT (Greenwich Mean Time), PDT (Pacific Daylight Time), EST(Eastern Standard Time), or even XYZ.

  The symbolic offset string (ABC in the example) has a limit of 5 characters.

  Note: Symbolic offset is appropriate only for IMS Database Recovery Facility. Symbolic offset is not supported by IMS Database Recovery Facility Extended Functions.

Timestamp examples

Here are 3 examples of compressed timestamps:

```
'21252082445712345'
'21252082445712-0800'
'212521624457+0'
```

Here are 5 examples of punctuated time stamps:

```
'21.252/08:24:45.712345'
'21.252 08:24:45.712 -8'
'21/252-08.24.45.71 -8:00'
'2021 252 16.24.45.7 +0'
'2021 252 16.24.45.7'
```

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 yesterday 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 -7 hours.

  In this example, in order to recover to the correct point in time, you must use -8 hours as the offset in the timestamp.

Processing System Level Backup (SLB) image copies

IMS Database Recovery Facility Extended Functions 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 database data sets (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 programing interface (API) is provided that allows Extended Functions 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 Extended Functions 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 Extended Functions 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 Extended Functions 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 Extended Functions 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 30

Controlling the use of SLBs

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

- When USESLBIC(Y) is specified, Extended Functions 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, Extended Functions 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 Extended Functions 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:

- An SLB image copy record exists in the RECON data sets, and
- An Extended Functions 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 Extended Functions jobs execute with USESLBIC(Y), the jobs should continue to run with this setting.

Using SLBs as recovery assets

When USESLBIC(Y) is specified, Extended Functions 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 Extended Functions 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 for managing API environments

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.

Reference: SLB API activation utility commands

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 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    = 03/17/21
 16:23:12 BSY0567I -
 16:23:12 BSY0567I - Active BSY#API environment:
 16:23:12 BSY0567I - Created by TSMXD on 2021-03-21 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.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 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    = 03/17/21
 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    = 03/17/21
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 2021-03-21 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    = 03/17/21
16:23:12 BSY0567I -
16:23:12 BSY0567I - Active BSY#API environment:
16:23:12 BSY0567I - Created by TSMXD on 2021-03-21 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 BSY0002I - IBM IMS Recovery Solution Pack for z/OS complete. RC=000.
```

Chapter 4. Configuration references

The following topics provides reference information to help you complete IMS Database Recovery Facility Extended Functions installation and configure the tool to your environment.

Topics:

Constructing an Extended Functions job

A complete set of sample JCL that you need to run Extended Functions can be found in the product SAMPLIB (SFRXSAMP) in member FRXIVPR1.

Member FRXIVPR1 contains the steps to drive every function of Extended Functions. You can use that job as a skeleton to build the function that you require. The JCL contains descriptive and instructive comments.

EXEC statement and parameters

The following example EXEC statement runs Extended Functions:

```
//STEP1    EXEC PGM=IROMAIN,
           PARM=(‘TYPE=IRE,IMSRECFG=IRECFGXX,BPECFG=BPECFG')
```

IROMAIN is the main driver program for the Extended Functions.

The following reference provides descriptions for the job parameters (PARM=):

TYPE=

Use this parameter to indicate the type of address space that is used.

The only value for this parameter is IRE.

This parameter is optional and its default is IRE.

IMSRECFG=

Use this parameter to specify the name of the Extended Functions configuration member that is located in the JCL procedure library (PROCLIB).

- You can use (and modify) SAMPLIB member IROCFG01.

  This member contains default settings that can be used by all Extended Functions jobs.
- Alternatively, you can create a custom configuration member specific to this job only.
- The member that you specify is read and parsed by Extended Functions.
- Use the configuration member to specify configuration parameters only. No function control cards are allowed in this member.

  Refer to “Extended Functions configuration parameters” on page 38.
- This parameter is optional and has no default. However, if IMSRECFG is omitted, you must specify a value for the ITKBSRVR control statement in the IMSREIN DD input statement.

  The ITKBSRVR control statement identifies the IMS Tools Knowledge Base server name (XCF group name) that is associated with your IMS Tools Knowledge Base environment.

BPECFG=

Use this keyword to specify the name of the BPE configuration member that is located in the JCL procedure library (PROCLIB).

If specified, the member is read and parsed by BPE.

This keyword is optional and has no default.

Input DD statements

The following DD statements are allowed or required as indicated for specifying Extended Functions run input.

IMSDALIB

Use this optional DD statement to point to the dynamic allocation load library that contains the dynamic allocation member for your RECON1 data set.

This data set is used by Extended Functions to locate the RECON1 dynamic allocation member in order to retrieve the RECON1 data set name when the RECON1 data set name is not specified by any other method.

The complete steps for locating the RECON1 data set name are specified in "RECONID locate processing" on page 23.

IMSREIN

Use this DD statement to specify configuration parameters and function control cards that drive Extended Functions processing.

This DD is required.

The parameters that can be specified in IMSREIN are described in "Extended Functions configuration parameters" on page 38.

PROCLIB

Use this DD statement to specify the JCL procedure library (PROCLIB) that contains the members that you specified with the IMSRECFG= or BPECFG= parameters on the MVS parameter string.

If you specified either of these parameters, this DD is required. Otherwise, this DD is optional.

RECON1 / RECON2 / RECON3

You can use these DD statements to specify the names of the RECON1, RECON2, and RECON3 data sets.

These DD statements are optional but, you must specify the RECON1 data set name in some manner.

The complete steps for locating the RECON1 data set name are described in "RECONID locate processing" on page 23.

STEPLIB / JOBLIB

Use this DD statement to specify the load libraries that are necessary to access the Extended Functions code.

This specification must also include the load libraries for:

- IMS Tools Online System Interface
- IMS Tools Knowledge Base
- IMS load library

These concatenated libraries must be APF-authorized.

Output DD statements

The following output DD statements are allowed or required as indicated for specifying the location of outputs that are produced by running Extended Functions.

IREDIAG

Use this DD statement for debugging purposes.

This DD is currently used by the RECON copy function as the output data set for IDCAMS and DSPURXRT.

The characteristics of this data set are:

```
RECFM=FBA,LRECL=133
```

This DD is optional.

RPTBSYOU

This DD is used by the IMS Recovery Expert integrated product to put out information regarding use of System Level Backups as recovery assets.

Several types of information are written to this DD, including error messages, operational, and execution information.

The characteristics of this data set are:

```
RECFM=FBA,LRECL=133
```

This DD is optional. If you do not specify it, and it is required by IMS Recovery Expert, this DD is dynamically allocated to SYSOUT=*

RPTSUMM

This DD is used by Extended Functions to put out information to a general purpose DD.

Several summary reports are written to this DD that contain operational and execution information.

The characteristics of this data set are:

```
RECFM=FBA,LRECL=133
```

This DD is optional. If you do not specify it, this DD is dynamically allocated to SYSOUT=*.

SYSPRINT

This DD statement is written to by some internal functions that are used by Extended Functions.

The characteristics of this data set are:

```
RECFM=FBA,LRECL=133
```

This DD is optional. If you do not specify it, this DD is dynamically allocated to SYSOUT=*.

Extended Functions report DDs

Each Extended Functions has its own report DD that must be specified when that function is requested.

If the function-specific DD is not present then the report DD that is associated with that function is dynamically allocated to SYSOUT=*.

The characteristics of these data sets are:

```
RECFM=FBA,LRECL=133
```

The report DD names for each function are listed in the following table.

Table 1. Report DD names for the Extended Functions

Report DD Name	Function
RPTRPCR	Recovery Point Create (RPCR)
RPTRPID	Recovery Point Identify (RPID)

Table 1. Report DD names for the Extended Functions (continued)

Report DD Name	Function
RPTVERFY	Verify (VERIFY)
RPTRCU	RECON Cleanup Utility (RCU)
RPTHCHEK	Health Checker (HCHECK)
RPTIMSCM	IMS Command (IMSCMD)

Rules for Extended Functions PROCLIB members

IMS Database Recovery Facility Extended Functions can use PROCLIB members in order to share BPE configuration and Extended Functions configuration options among several jobs.

This data set should be a standard IMS-type PROCLIB data set.

Adhere to the following rules when you set up these PROCLIB members.

- The rightmost 8 columns of each line are ignored and can be used for sequence numbers or any other notation that you might need. For example, if your LRECL is 80, columns 73 to 80 are ignored.
- You can add blanks between words for readability, but you cannot add them within words.
- You can code multiple statements on one line.
- Use commas or spaces to delimit keywords.
- Use an asterisk (*) or a pound sign (#) in the first column of a line to identify the line as a comment.
- Comments can also be included between pairs of /* and */ characters.
- Code values in uppercase only.
- Comments can be in mixed case.
- Any values that have imbedded blanks must be enclosed within single quotation marks.

Extended Functions configuration parameters

Configuration parameters are used to control the overall operation and environment of IMS Database Recovery Facility Extended Functions.

You can specify configuration parameters in one of two places:

- The configuration member specified by the IMSRECFG parameter in the Extended Functions job EXEC statement.

  The IMSRECFG parameter specifies the name of the Extended Functions configuration member that is located in the JCL procedure library (PROCLIB).
- The IMSREIN input DD statement in the Extended Functions job.

  The IMSREIN input DD statement specifies configuration parameters and function control cards that drive Extended Functions processing.

  Note: If you specify a configuration parameter in both the IMSREIN DD statement and member IMSRECFG, the value that is specified in the IMSREIN DD statement overrides the value that is specified in member IMSRECFG.

Note: Configuration parameters are processed before any function control statements.

Configuration parameter reference

ITKBSRVR

This parameter identifies the IMS Tools Knowledge Base server name (XCF group name) that is associated with your IMS Tools Knowledge Base environment.

The parameter value is the 8-character XCF group name that is associated with the IMS Tools Knowledge Base server.

This parameter is required and has no default value.

ITKBSRVR syntax:

```
    ITKBSRVR( ITKB-XCF-group )
```

RECONID

This parameter identifies the RECONID member that is obtained from the repository in order to retrieve environmental information.

This is an 8-character name that is assigned to the RECONID member as defined in the repository.

This parameter is optional but, if you do not supply it, the RECON1 data set name must be available either through the JCL (allocated to the RECON1 DD statement), or by the DFSMDA dynamic allocation member in the //STEPLIB or //IMSDALIB.

RECONID syntax:

```
    RECONID( reconid-member-name )
```

REPORT

Use this parameter to specify your preferences for the reports that are created by Extended Functions.

REPORT syntax:

```
     REPORT(ITKBOUT( YES | Y | NO | N )
             PAGEBR( YES | Y | NO | N )
            TIMEFMT( LOCAL | DATA | UTC )
            DATEFMT( YYYYDDD | MMDDYYYY | DDMMYYYY | YYYYMMDD )
           )
```

REPORT parameter reference:

ITKBOUT(YES | Y | NO | N)

Use this parameter to specify whether the reports that are generated by Extended Functions are placed into the IMS Tools Knowledge Base output repository.

The default is ITKBOUT(YES).

PAGEBR(YES | Y | NO | N)

Use this parameter to specify whether the reports that are generated by Extended Functions include page breaks and headers that make the reports easier to interpret when they are printed.

If your reports will be viewed online only, PAGEBR(NO) is recommended.

The default is PAGEBR(NO).

TIMEFMT(LOCAL | UTC | DATA)

Use this parameter to specify whether timestamps that are displayed on your reports are in local time (LOCAL), global time (UTC), or the time that is associated with the data (DATA).

For TIMEFMT(DATA), if the processed data (such as, log records) are from a location other than the current location, the timestamps that are displayed are from the location where the data was created.

The default is TIMEFMT(LOCAL).

DATEFMT(*YYYYDDD* | *MMDDYYYY* | *DDMMYYYY* | *YYYYMMDD*)

Use this parameter to specify the format of the date that is displayed on all of your report headings.

- The *YYYY* variable represents the value for the year.
- The *MM* variable represents the value for the month.

- The *DD* or *DDD* variable represents the value for the day.

The default is DATEFMT(*MMDDYYYY*).

TEMPDSN

Use this parameter to control the allocation of temporary data sets that might be needed by internal services.

If you do not specify this parameter, the system defaults are used to allocate all necessary temporary data sets.

TEMPDSN syntax:

```
TEMPDSN( TUNIT( unit-name )
         TVOLSER( volser )
         TSTORCLASS( SMS-storage-class )
         TMGMTCLASS( SMS-management-class )
         TDATACLASS( SMS-data-class )
       )
```

TEMPDSN parameter reference:

TUNIT(*unit-name*)

Use this parameter to specify the DASD unit that is used to allocate temporary data sets.

TVOLSER(*volser*)

Use this parameter to specify the DASD volume serial that is used to allocate temporary data sets.

TSTORCLASS(*SMS-storage-class*)

TMGMTCLASS(*SMS-management-class*)

TDATACLASS(*SMS-data-class*)

Use these three SMS parameters to specify the SMS classes that are used to allocate temporary data sets.

USESLBIC

Use this parameter to determine whether IMS Recovery Expert system level backups are considered recovery assets in the HCHECK and VERIFY functions.

Note: It is strongly suggested that configuration parameter USESLBIC is set to match the same parameter in IMS Database Recovery Facility.

USESLBIC syntax:

```
USESLBIC(NO | N | YES | Y)
```

The default is USESLBIC(NO).

Support functions

The support functions DBLIST, RECONCOPY, and ALTRECON provide information and services that are used by other Extended Functions when those functions are processed.

The support functions do not operate stand-alone, but perform in conjunction with the other Extended Functions.

Topics:

- “DBLIST processing” on page 41
- “RECONCOPY processing” on page 44
- “ALTRECONS processing” on page 46

DBLIST processing

Use the DBLIST parameter to identify the databases that Extended Functions is to process.

Before Extended Functions invokes any specified function, the database list is pre-processed to validate and expand the parameters into a simple list of databases, database data sets, areas, and partitions that will be processed by the functions.

Database list processing takes the input that is specified by the DBLIST subparameters (as explained in the subsequent topic about target database list processing), and accesses the RECON data sets to resolve the list into individual elements.

All of the DBLIST subparameters (such as, DB(), AREA(), and other subparameters) allow wildcard characters, such as, * and %, in the names, following standard IMS rules.

Each subparameter is processed, as described in the next subtopic and an entry is created in the list. This process ensures that there are no duplicate entries in the list.

The DBLIST parameter is used only in conjunction with other functions; it does not operate alone.

The following Extended Functions require the use of the DBLIST parameter:

- HCHECK (Recovery health checker)
- RPCR (Recovery point create)
- RPID (Recovery point identification)
- VERIFY (Verify recovery assets)

You can specify the DBLIST parameter only once per job step. All of the functions that are specified in the job step will process the same set of databases, as specified by DBLIST().

Extended Functions DBLIST process flow:

- Access the RECON data sets.
- Validate the DBLIST.
- Expand the DBLIST as needed.
- Pass the individual databases to the functions.

You can specify the databases using the following subparameters:

- DB - A single or multiple database
- DBDS - A single or multiple database and database data set
- AREA - A single or multiple fast path DEDB and AREA pair
- PART - A single or multiple HALDB master and partition pair
- DBDSGRP - A single or multiple RECON DBDS group
- REVCOVGRP - A single or multiple RECON recovery group
- CAGRP - A single or multiple RECON CA group

Note: All of the values that can be specified, as shown in the next example, allow the standard use of wildcard characters (for example, * and %).

DBLIST syntax

```
DBLIST( DB( dbname, dbname, … )
        DBDS( ( dbname,dbdsname),(dbname,dbdsname),… )
        AREA( ( dbname,areaname),(dbname,areaname),… )
        PART( ( dbname,partname),(dbname,partname),… )
        DBDSGRP( recon-dbds-group,… )
        CAGRP( recon-ca-group,… )
        RECOVGRP( recon-recovery-group,… )
      )
```

DBLIST parameter reference

DB(*dbname,dbname,...*)

Use this parameter to specify one or more full function, fast path, or HALDB master databases.

You cannot specify HALDB partition names.

You can specify multiple names separated by commas or blanks.

Example of DBLIST(DB()):

```
        DBLIST(DB(PARTSDB,PAYROLL,ACCT%))
```

DBDS((*dbname,dbdsname*),(*dbname,dbdsname*),...)

Use this subparameter to specify one or more pairs of database names (DBD) followed by the database data set name (DD name).

You can specify only full function and fast path databases.

You can specify one or more pairs of databases but, you must specify them in pairs.

You must specify the pairs within single parenthesis, separated by commas or blanks.

You cannot specify HALDB master and partition names in this field.

Example of DBLIST(DBDS())

```
        DBLIST(DBDS((DBHDOJ01,DJOHDG1O),( ACCTA,*)))
```

AREA((*dbname,areaname*),(*dbname,areaname*),...)

Use this subparameter to specify one or more pairs of fast path database names (DEDB) followed by the area name.

You can specify one or more pairs but, you must specify them in pairs.

You must specify the pairs within single parenthesis, separated by commas or blanks.

Example of DBLIST(AREA())

```
        DBLIST(AREA((DEDBDD01,DD01AR0)))
```

PART((*dbname,partname*),(*dbname,partname*),...)

Use this subparameter to specify one or more pairs of HALDB master database names followed by the partition name.

You can specify one or more pairs but, you must specify them in pairs.

You must specify the pairs within single parenthesis, separated by commas or blanks.

Example of DBLIST(PART())

```
        DBLIST(PART((MASTER4,PART41),(MASTER4,PART43),( MASTER7,*)))
```

DBDSGRP(*recon-dbds-group,...*)

Use this subparameter to specify one or more RECON DBDS groups.

You can specify multiple RECON DBDS group names separated by commas or blanks.

Example of DBLIST(DBDSGRP())

```
        DBLIST(DBDSGRP(DBDSGRP1,DBDSGRP2))
```

CAGRP(*recon-ca-group,...*)

Use this subparameter to specify one or more RECON change accumulation groups.

You can specify multiple RECON change accumulation group names separated by commas or blanks.

Example of DBLIST(CAGRP())

```
        DBLIST(CAGRP(CAGRP01,CAGRP02))
```

RECOVGRP(*recon-recovery-group,...*)

Use this subparameter to specify one or more RECON recovery groups.

You can specify multiple RECON recovery group names separated by commas or blanks.

Example of DBLIST(RECOVGRP())

```
        DBLIST(RECOVGRP(RECOVG01,RECOVG02))
```

Target database list processing

Extended Functions processes the database list by using the subparameters as described in the following sections.

DB() subparameter processing

The DB() subparameter allows for a series of one or more full-function (FF), fast path (FP), or HALDB master database names.

This list of database names is processed using the DBRC API and any database that matches a name in the list (allowing for wildcards) is processed.

For any database names in the list that are defined in the RECONs, an entry is added to the list.

DBDS() subparameter processing

The DBDS() subparameter allows for a series of one, or more pairs of full-function (FF) or fast path (FP) database names followed by the associated database data set (for example, DD name or AREA).

Each entry must be paired but each piece can specify wildcard characters.

The list of databases is processed using the DBRC API and the database entries, along with all related database data sets, are returned.

A list entry is created for each FFDB+DBDS or FPDB+AREA entry that matches the input list.

If a database that matches the input is located but is not a full function or fast path database, the entry is not added to the list.

AREA() subparameter processing

This subparameter allows for a series of one or more pairs of fast path database names followed by the associated area name.

Each entry must be paired but each piece can specify wildcard characters.

The list of databases is processed using the DBRC API and the database entries, along with all related areas are returned.

A list entry is created for each of the DEDB AREA+DDname entries that match the input list.

If a database that matches the input is located, but is not a fast path database, the entry is not added to the list.

PART() subparameter processing

This subparameter allows for a series of one or more pairs of HALDB master database names, followed by the associated partition name.

Each entry must be paired but each piece can specify wildcard characters.

The list of databases is processed using the DBRC API and the database entries, along with all of the related areas are returned.

A list entry is created for each of the Partition+DDname entries that match the input list.

If a database that matches the input is located, but it is not a HALDB database, the entry is not added to the list.

DBDSGRP() subparameter processing

This subparameter allows for a series of one or more RECON DBDS group names.

This list is processed using the DBRC API and all of the databases that are associated with the specified groups are returned.

A list entry is created for each DB+DD (or AREA, or PART) entry in the RECON group.

CAGRP() subparameter processing

This subparameter allows for a series of one or more RECON Change Accumulation group names.

This list is processed using the DBRC API and all of the databases that are associated with the specified groups are returned.

A list entry is created for each DB+DD (or AREA or PART) entry in the RECON group.

RECOVGRP() subparameter processing

This subparameter allows for a series of one or more RECON Recovery group names.

This list is processed using the DBRC API and all of the databases that are associated with the specified groups are returned.

A list entry is created for each DB+DD (or AREA or PART) entry in the RECON group.

RECONCOPY processing

The RECON copy (RECONCOPY) support function creates a copy of the RECON data sets that are used by IMS Database Recovery Facility Extended Functions processing, instead of the live RECON data sets.

This process uses the DBRC BACKUP.RECON RECON1 command, so that whichever copy of the RECON data set is COPY1 is used. This process then makes a second and third copy and then allocates these three data sets to the RECON1, RECON2, and RECON3 DD statements.

Note: If you do not specify either the ALTRECONS or RECONCOPY parameter, the Extended Functions will access the live RECON data sets. RECONCOPY is mutually exclusive with ALTRECONS.

The RECONCOPY support function is used only in conjunction with other Extended Functions; it does not operate alone.

Not all of the Extended Functions can operate on RECON copies. The following Extended Functions cannot use RECON copies:

- Recovery Point Create (RPCR)
- IMS Commands (IMSCMD)

The following Extended Functions can use RECON copies:

- Recovery point identification (RPID)
- RECON cleanup (RCU)

  Note: The RECON cleanup (RCU) function requires the use of RECONCOPY. The RCU function will not process without RECONCOPY.
- Recovery health checker (HCHECK)
- Verify recovery assets (VERIFY)

RECONCOPY syntax

```
RECONCOPY( DSNPREF( dsn-prefix )
           REUSE( YES | NO | Y | N )
           DELETE( YES | NO | Y | N )
           COPY1VOL( volser-list )
```

```
               COPY2VOL( volser-list )
               COPY3VOL( volser-list )
               SPACE( space-parameters )
               STORCLASS( SMS-storage-class )
               MGMTCLASS( SMS-management-class )
               DATACLASS( SMS-data-class )
             )
```

RECONCOPY parameter reference

The only required parameter is DSNPREF.

If you do not specify the allocation type parameters (for example, SPACE and VOLSER), their values are obtained from the existing RECON1 data set that is used as a model.

DSNPREF(*dsn-prefix*)

Use this parameter to specify the pattern to use to create the prefix for the RECON copy data sets.

The prefix length is limited to 38 characters.

The RECON copy data set names will consist of this prefix with .C1 C2, or .C3 as the suffix.

This parameter is required and there is no default.

The prefix allows for several special character strings that cause Extended Functions to substitute values into the prefix. These special character strings are:

&date

Substitute a date stamp in the format *Dyyyyddd* within the name.

&time

Substitute a timestamp in the format *Thhmmss* within the name.

&jobnm

Substitute the *job name* within the name.

&stepnm

Substitute the *jobstep* name within the name.

REUSE(Y | YES | N | NO)

Use this parameter to indicate whether to reuse an existing RECON copy data set, if one is found, with the same name as the one that is being created.

A specification of YES or Y causes the process to delete and redefine the existing data sets.

A specification of NO or N causes the process to terminate when one or more of the data sets already exist.

The default for REUSE is REUSE(NO).

DELETE(Y | YES | N | NO)

Use this parameter to indicate whether the RECON copy data sets are deleted at job step termination.

The default is DELETE(NO).

COPY1VOL(*volser-list*), COPY2VOL(*volser-list*), COPY3VOL(*volser-list*)

Use these parameters to specify a list of one-to-five volsers on which to allocate the specified RECON copy data set.

- Use the COPY1VOL parameter to specify the list of volsers for RECON COPY1.
- Use COPY2VOL to specify the list for RECON COPY2.
- Use COPY3VOL to specify the list for RECON COPY3.

SPACE(*space-parameters*)

Use this parameter to specify the space characteristics for the RECON copy data sets.

The values that you specify for this parameter must adhere to the SPACE() parameter that are allowed by IDCAMS as follows:

- SPACE(CYLINDERS(*primary, secondary*))

- SPACE(TRACKS(*primary, secondary*))
- SPACE(RECORDS(*primary, secondary*))

The abbreviations that are allowed are: CYL, TRK, and REC.

STORCLASS(*SMS-storage-class*)

MGMTCLASS(*SMS-management-class*)

DATACLASS(*SMS-data-class*)

Use the three SMS parameters to specify the various SMS classes that are used to allocate COPY1, COPY2, and COPY3 of the RECON data sets.

ALTRECONS processing

The alternate RECONS (ALTRECONS) support function identifies a preexisting copy of the RECON data sets that are used by IMS Database Recovery Facility Extended Functions processing instead of the live RECON data sets.

This process reallocates the RECON1, RECON2, and (optionally) RECON3 DD statements to the specified alternate RECONS data sets.

Note: If you do not specify either the ALTRECONS or RECONCOPY parameter, the Extended Functions will access the live RECON data sets. RECONCOPY is mutually exclusive with ALTRECONS.

The ALTRECONS support function is used only in conjunction with other Extended Functions; it does not operate alone.

Not all of the Extended Functions can operate on the preexisting alternate RECONS. The following Extended Functions cannot use alternate RECONs:

- Recovery Point Create (RPCR)
- IMS Commands (IMSCMD)
- RECON cleanup (RCU)

The following Extended Functions can use alternate RECONS:

- Recovery point identification (RPID)
- Recovery health checker (HCHECK)
- Verify recovery assets (VERIFY)

ALTRECONS syntax

```
ALTRECONS(
          ALTRECON1(dsn)
          ALTRECON2(dsn)
          ALTRECON3(dsn)
          )
```

ALTRECONS parameter reference

The only required parameters are ALTERCON1 and ALTRECON2.

ALTRECON3 is optional.

ALTRECON1

The RECON data set that is reallocated to the RECON1 DD.

Required.

ALTRECON2

The RECON data set that is reallocated to the RECON2 DD.

Required.

ALTRECON3

The RECON data set that is reallocated to the RECON3 DD.

Optional.

Extended Functions sample summary report

When any of the IMS Database Recovery Facility Extended Functions runs, it creates a summary report that is written to the RPTSUMM output DD.

This summary report contains overall system operation information and reflects the configuration and function control cards and other processing options.

- If the RPTSUMM DD is not specified in the JCL, it is dynamically allocated to SYSOUT=*.
- Report output can be written to SYSOUT or to a data set.
- The characteristics of the output are:

```
RECFM=FBA,LRECL=133
```

- Report output can also be written to the IMS Tools Knowledge Base repository by specifying the REPORT(ITKBOUT(YES)) keyword and parameter.

Note: REPORT(ITKBOUT(YES) is the default.

Sample Summary Report

The next figure shows a sample SUMMARY report.

The format of this sample report has been modified to fit the publication formatting.

```
IMS Recovery Solution Pack V2.1.0 (5655-V86)
IMS DATABASE FACILITY EXTENDED FUNCTIONS SUMMARY REPORT   Run Time: 11/30/2021 09:20
********************************************************************************
-> MVSP - MVS PARM= STRING
********************************************************************************
TYPE=IRE,BPECFG=BPECONFG,IMSRECFG=IRECFG01
********************************************************************************
-> CFGP - IMSRECFG= CONFIGURATION MEMBER
********************************************************************************
ITKBSRVR(QAIMSRE1)
********************************************************************************
-> REIN - IMSREIN INPUT MEMBER
********************************************************************************
DBLIST(CAGRP(CAFF,CAFP),DB(F101P1,H201P3))
********************************************************************************
-> PARM - ACTIVE CONFIGURATION PARMS
********************************************************************************
  IMSRECFG = IRECFG01
  ITKBSRVR = QAIMSRE1
  RECONID  = Q91ARCNS
  TEMPDSN:
   TUNIT   =
   TVOLSER =
   TSTCLASS=
   TMGCLASS=
   TDTCLASS=
  REPORT:
   ITKBOUT = YES
   PAGEBR  = NO
   TIMEFMT = LOCAL
   DATEFMT = MMDDYYYY
********************************************************************************
-> RCID - ACTIVE RECONID RECORD
********************************************************************************
  RECONID  = Q91ARCNS
   TOIXCFGR= TOIORF
   IMSPLEX =
   CMDCLASS= #NONE#
   CMDTYPE = NONE
   RECON1  = IMSTOOL.IMSRE.QA.Q91A.RECON1
   RECON2  = IMSTOOL.IMSRE.QA.Q91A.RECON2
   RECON3  = IMSTOOL.IMSRE.QA.Q91A.RECON3
********************************************************************************
-> RCGR - DBLIST() RECON GROUP EXPANSIONS
********************************************************************************
INPUT:  CAGRP(CAFF    )
   GROUP: NAME=CAFF
     MEMBER: DB=F202P1   DD=F202P11
     MEMBER: DB=F202P1   DD=F202P12

INPUT:  CAGRP(CAFP    )
   GROUP: NAME=CAFP
     MEMBER: DB=D2V1P1   DD=D2V1P11
     MEMBER: DB=D2V2P1   DD=D2V2P11
********************************************************************************
-> DBLS - LIST OF DATABASES PROCESSED
********************************************************************************
DATABASE    AREA/PART     DD             TYPE   DSN          PART-TYPE    OLR-PARTNER
--------------------------------------------------------------------------------------
D2V1P1      D2V1P11       D2V1P11        FP     IMSTOOL.IMSRE.QA.Q91A.D2V1P11.D2V1P11A
D2V2P1      D2V2P11       D2V2P11        FP     IMSTOOL.IMSRE.QA.Q91A.D2V2P11.D2V2P11A
F101P1                    F101P11        FF     IMSTOOL.IMSRE.QA.Q91A.F101P1.F101P11
F202P1                    F202P11        FF     IMSTOOL.IMSRE.QA.Q91A.F202P1.F202P11
F202P1                    F202P12        FF     IMSTOOL.IMSRE.QA.Q91A.F202P1.F202P12
********************************************************************************
-> ERLS - DBLIST() ERRORS
********************************************************************************
DATABASE    DD/AREA/PART    ERROR
--------------------------------------------------------------------------------
H201P3                      ENTRY NOT DEFINED TO RECON
H201P3                      DATABASE HAS NO DEFINED DD/AREA/PART
```

Figure 2. Sample Extended Functions Summary report

Descriptions of summary report sections

The major sections of the summary report are described in this list:

MVSP - MVS PARM= STRING

This report section shows the value of the PARM= string that was specified on the EXEC statement for this job step.

If the PARM= string was specified, it shows the BPE and Extended Functions configuration members that were used.

CFGP - IMSRECFG= CONFIGURATION MEMBER

This report section shows the contents of the Extended Functions configuration member, if specified.

This member is read by Extended Functions and any parameters that are specified will be in effect for this run.

This member is specified using the IMSRECFG= keyword on the EXEC statement PARM= string.

REIN - IMSREIN INPUT MEMBER

This report section shows the contents of the IMSREIN DD.

This data is read by Extended Functions and any parameters that are specified will be in effect for this run.

PARM - ACTIVE CONFIGURATION PARMS

This report section shows the settings for the configuration parameters that were used for this job run.

RCID - ACTIVE RECONID RECORD

This report section shows the RECONID member that was used for this job run.

RCGR - DBLIST() RECON GROUP EXPANSIONS

This report section shows any DBLIST(CAGRP(), DBDSGRP() or RECOVGRP()) input and their expansions.

Report headings and descriptions:

INPUT:

This field indicates the specific input that was specified.

GROUP:

This field shows the CA group name.

MEMBER:

This field shows the database name (DB=) and DD name (DD=) for each member of the group.

DBLS - LIST OF DATABASES PROCESSED

This report section shows the databases that were processed in this run.

Report headings and descriptions:

DATABASE

This column specifies the database name.

AREA/PART

This column specifies the area name for fast path databases, or the partition name for HALDB databases.

This column is blank for full function databases.

DD

This column specifies the DD name that is associated with this area, partition, or full function database.

TYPE

This column specifies the type of database.

Possible values are:

- FF for full function databases.
- FP for fast path areas.
- HALD for HALDB partitions.

DSN

This column specifies the data set name that is associated with this area, partition, or full function database.

PART-TYPE

For HALDB partitions, this column specifies the type of data set.

Possible values are:

- INDEX for the index portion of the HALDB
- ILE for the index list element (or ILDS) portion of the HALDB
- DATA for the data portion of the HALDB

OLR-PARTNER

For HALDB partitions that are OLR capable, this column specifies the OLR partner DD name that is associated with the partition.

ERLS - DBLIST() ERRORS

This report section shows the databases that were specified in the DBLIST, or expanded based on the DBLIST but were found to be in error.

Report headings and descriptions:

DATABASE

This column shows the database name.

DD/AREA/PART

This column shows the DD name, area name for fast path databases, or the partition name for HALDB databases.

ERROR

This column indicates the specific error that was found.

The possible values are:

- DATABASE HAS NO DEFINED DD/AREA/PART
- ENTRY NOT DEFINED TO RECON
- HALDB PART SPECIFIED WHERE MASTER REQUIRED

Part 3. Using Extended Functions

Each IMS Database Recovery Facility Extended Function function performs a specific operation. You need to understand each function and the operation of each function to fully operate Extended Functions.

Topics:

Chapter 5. IMS Command (IMSCMD) function

The IMS command function (IMSCMD) provides a batch interface for IMS Type-1 AOI commands in IMS Database Recovery Facility Extended Function.

Topics:

- “IMSCMD overview” on page 53
- “IMSCMD input” on page 53
- “IMSCMD usage examples” on page 54
- “IMSCMD output” on page 55

IMSCMD overview

The IMSCMD function issues the commands that you specify to one or more active IMS subsystems. IMS Type-1 AOI commands all begin with a slash (/) (for example, /DBR).

IMSCMD function uses the IMS Tools Online System Interface to issue IMS commands. For this function to operate, IMS Tools Online System Interface must be installed and active on the IMS systems to which this command is issued.

The available target IMS systems are determined by the IMS Tools Online System Interface XCF group in the RECONID member that is used by the job.

You can issue the IMSCMD function to all of the IMS systems that are connected to the IMS Tools Online System Interface XCF group, or to individual IMS systems, by using the SYSTEMS subparameter on the IMSCMD function specification.

Use of either the ALTRECONS or the RECONCOPY supporting function has no effect on the IMSCMD function. Any commands that you issue run under the control of the IMS control region and therefore, those commands access the RECON data sets that are allocated to that IMS subsystem.

Similarly, use of the DBLIST supporting function has no effect on the IMSCMD function. Any commands that you issue must be specified completely and no database list processing is involved.

Note: When issuing commands, Extended Functions performs security checking based on the definition in the RECONID record. The security checks are performed against the RACF security class specified in the RECONID record.

The security processing normally performed by IMS Tools Online System Interface (RACFCHK=*value*) is bypassed.

IMSCMD input

This topic provides syntax reference information for the IMS Command function (IMSCMD).

IMSCMD function syntax

```
IMSCMD(COMMAND(command)
       SYSTEMS(systems-list)
        )
```

Parameter reference

COMMAND

Use this required parameter to specify a single IMS Type-1 AOI command to be issued. Specify the IMS commands as you would normally; starting with a slash (/) and following the same syntax as the command follows when it is issued manually. For example:

```
IMSCMD(COMMAND(/STA DB F101P11 H1V4S12))
```

Note: The *command* value can be a maximum of 80 characters.

SYSTEMS

Use this optional parameter to specify one or more active IMS systems to which the command is issued. If you omit this parameter, the command is issued to all of the active IMS systems in the IMS Tools Online System Interface XCF group. If a system that is specified in the list is not active, the fact that the system was inactive is noted in the output, but the command is still issued for any active systems in the Tools Online System Interface XCF group.

When you use the GLOBAL parameter on IMS commands, the SYSTEMS parameter is not allowed. If you specify both the GLOBAL and SYSTEMS parameters, an error message is displayed and the IMS command is not performed. The GLOBAL parameter requires internal resource lock manager (IRLM) in order to run the commands on all of the sharing online subsystems.

Notes on using the IMSCMD function

The IMSCMD(COMMAND(*command*)) function accepts a maximum of 80 characters for the *command* value. If greater than 80 characters are specified, then the job terminates and the message BPE0003E is displayed.

As a workaround, you can issue multiple IMSCMDs in one job step. For example:

```
//IMSREIN DD*
IMSCMD(COMMAND(/STA DB H1V4P1 H1V4S1 H1V4S2 H1V4P11 H1V4P12 H1V4P13))
IMSCMD(COMMAND(/STA DB H1V4S11 F101P11 H1V4S12))
```

IMSCMD usage examples

You can use the IMS Command function (IMSCMD) to alter or display the status of IMS resources across all of the IMS systems in an IMSplex or data sharing environment.

Example 1: Specifying multiple parameters

You can specify multiple IMSCMD parameters within the same job step. For example, to display the status of a series of databases, programs and transactions across the IMSplex, you would specify the following input:

```
    IMSCMD(COMMAND(/DIS DB FFDB01 FFDB02))
    IMSCMD(COMMAND(/DIS PGM FFPGM01 FFPGM02))
    IMSCMD(COMMAND(/DIS TRAN FFTRAN01 FFTRAN02))
```

Example 2: Displaying the same set of IMS resources

If you wanted to display the same set of IMS resources for just the IMS1 system, you would specify the following input:

```
    IMSCMD(COMMAND(/DIS DB FFDB01 FFDB02) SYSTEMS(IMS1))
    IMSCMD(COMMAND(/DIS PGM FFPGM01 FFPGM02) SYSTEMS(IMS1))
    IMSCMD(COMMAND(/DIS TRAN FFTRAN01 FFTRAN02) SYSTEMS(IMS1))
```

IMSCMD output

The IMS Command function (IMSCMD) output is written to the RPTIMSCM DD.

If you do not specify the RPTIMSCM DD in the JCL, it is dynamically allocated to SYSOUT=*. Report output can be written to SYSOUT or to a data set. The characteristics of the output are RECFM=FBA,LRECL=133. You can also have the report output written to the IMS Tools Knowledge Base if you specify the REPORT(ITKBOUT(YES)) keyword and parameter.

Note: REPORT(ITKBOUT(YES) is the default.

The following figure shows a sample IMSCMD report output.

```
IMS Recovery Solution Pack V2.1.0 (5655-V86)
IMS DRF Extended Functions IMS COMMAND REPORT         Run Time: 11/06/2021 08:57




********************************************************************************
-> Invocation Parameters
********************************************************************************

  Command: /DIS DB DI21PART
  Systems: ALL



********************************************************************************
-> IMS Command Output
********************************************************************************



-->   IMSID: Q91A  COMMAND: /DIS DB DI21PART

 D71 DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
D01 DI21PART                                   UP   STOPPED, NOTOPEN, NOTINIT
X99 *21310/085729*



********************************************************************************
-> IMS Command Output
********************************************************************************



-->   IMSID: Q91B  COMMAND: /DIS DB DI21PART

 D71 DATABASE  TYPE  TOTAL UNUSED  TOTAL UNUSED ACC  CONDITIONS
D01 DI21PART                                   UP   STOPPED, NOTOPEN, NOTINIT
X99 *21310/085729*
```

Figure 3. Sample IMSCMD report output

The major report sections are described under their headings in the following list:

Invocation Parameters
: This section of the report shows the IMS command that the IMSCMD function issued and the systems to which the IMS command was issued. This sample report shows that the /DIS DB DI21PART command was issued to all of the active IMS systems.

IMS Command Output
: This section of the report is repeated for each IMS system to which the IMS command was issued. This section also contains all of the command output. This sample report shows that the command was issued to IMSIDs, Q91A and Q91B because they were all of the active IMS systems in the Tools Online System Interface XCF group.

Chapter 6. Health Checker (HCHECK) function

The Health Checker function (HCHECK) provides an interface to the RECONs that identifies a number of issues that can impact IMS database recovery in IMS Database Recovery Facility Extended Function.

Topics:

HCHECK overview

The HCHECK function checks information in the RECON data sets that is associated with the IMS database subsystem to identify potential problems with the recoverability of one or more databases. HCHECK uses information from the RECON data sets, the IMS Recovery Expert repository, and the MVS catalog.

The HCHECK function requires that you specify the target databases using the DBLIST supporting function.

The HCHECK function can use the RECON copy data sets that are created by the RECONCOPY supporting function, or the data sets identified on the ALTRECONS supporting function.

Note: Use either RECONCOPY or ALTRECONS in conjunction with HCHECK to prevent input/output (I/O) to the live RECON data sets.

Processing System Level Backup Image Copies

The RECON can contain Image Copy records that represent System Level Backups (SLB). The following considerations pertain to using HCHECK to process SLBs.

- The data set name for the SLB Image Copy is symbolic only and is formatted as follows:

  ```
  SLB.Ixxxxx.Dxxxxx.dbdname.ddname
  ```

  These data sets are never cataloged, nor does any function try to allocate or open them.
- When USESLBIC(N) is specified, RECON Image Copy records that represent SLBs are treated by HCHECK no differently than any other Image Copy records with one exception: HCHECK(VERIFY) does not verify as to whether they are cataloged.

  You can use the VERIFY function with the LIST option to determine if there are SLB Image Copy records in the RECON.

  IMS Database Recovery Facility VERIFY, IMS Database Recovery Facility Extended Functions VERIFY, and IMS Database Recovery Facility Extended Functions HCHECK functions all recognize these data set names as symbolic only.
- When USESLBIC(Y) is specified, and whenever necessary, IMS Database Recovery Facility Recovery, IMS Database Recovery Facility VERIFY, and IMS Database Recovery Facility Extended Functions VERIFY functions issue a **NOTIFY.IC** command to DBRC for the most recent SLB that exists relative to the function's specified time parameter (for example, RCVTIME or "current").

  HCHECK(VERIFY) does not issue a **NOTIFY.IC** because SLB.*Ixxxxx*.*Dxxxxx*.*dbdame*.*ddname* data sets are never cataloged and so do not apply to its function. Instead, HCHECK(VERIFY) lists on the report that there is a symbolic SLB Image Copy record in the RECON and that it is not cataloged.
- It is a best practice to delete any Image Copy records that represent deleted SLBs.

Otherwise, the HCKECK functions of ICNONE, ICFREQ, MINBAT, and MINONL would consider these IC records as representing a usable backup. This correlates with the fact that IMS Database Recovery Facility Recovery and IMS Database Recovery Facility VERIFY would also consider these records as valid and attempt to use the deleted SLBs.

- Another possible situation occurs when an SLB exists, yet there is no RECON IC record that represents it.

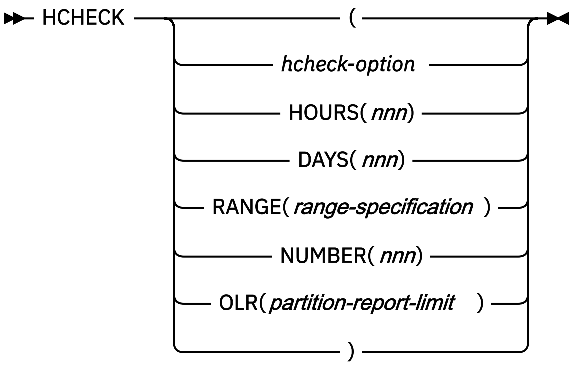
  The VERIFY function with any of its options (LIST, ALLOC, OPEN) causes representative IC records to be written to the RECON.

  HCKECK functions ICNONE, ICFREQ, MINBAT, and MINONL consider the corresponding SLBs as potentially valid backups. However, they do not write NOTIFY.IC records to the RECON because it is not necessary relative to their function.

HCHECK input

Run the Health Checker function by running HCHECK.

HCHECK function syntax



Parameter reference

hcheck-option
Use this positional parameter to specify the HCHECK option to be processed. The list of valid options is described in the following table. This parameter must be the first parameter that you specify on the HCHECK keyword.

Use the following HCHECK parameters to specify the time frame range, or number that HCHECK uses for various options. Not all HCHECK options allow these parameters. Also, these parameters are mutually exclusive.

The valid list of HCHECK options and the parameters that are allowed is shown in the following section (HCHECK options reference).

HOURS(*nnn*)
Use the HOURS parameter to specify the number of hours to be used to limit the HCHECK option.

HOURS allows an integer between 1 and 999.

There is no default for the HOURS parameter.

DAYS(*nnn*)
Use the DAYS parameter to specify the number of days to be used to limit the HCHECK option.

DAYS allows an integer between 1 and 999.

There is no default for the DAYS parameter.

RANGE(*range-specification*)

Use the RANGE parameter is used to specify the beginning and ending time frame (or range) to be used to limit the HCHECK option.

There is no default for the RANGE parameter.

For more detailed information about the format of the RANGE specification, see "TIME and RANGE parameter processing" on page 24.

NUMBER(*nnn*)

Use the NUMBER parameter to specify the number to be used to limit the HCHECK option.

NUMBER allows an integer between 1 and 999.

There is no default for the NUMBER parameter.

OLR(*partition-report-limit*)

Use the OLR parameter to limit which partitions are reported on.

The following values are valid for *partition-report-limit*:

BOTH

Reports on all partitions, whether active or inactive. This is the default value.

ACTIVE

Reports only the currently active partitions.

INACTIVE

Reports only the currently inactive partitions.

Options reference

The following table shows all of the valid HCHECK options, their abbreviations, acceptable parameters, and a brief description of each option. A more detailed description of each option appears after this table.

Table 2. The valid HCHECK options, abbreviations, parameters, and descriptions

Option	Abbreviations	Parameters	Description
ICALL	IA, ICA	N/A	Produces the ICNONE, ICNEED, and ICREC reports.
ICNONE	IO, ICNO	N/A	DBs for which no ICs or SLBs exist.
ICNEED	IE, ICNE	N/A	DBs for which ICs are needed.
ICREC	IR, ICR	N/A	DBs for which ICs are recommended.

ICFREQ	IF, ICF	HOURS()
		DAYS()
		RANGE()
			DBs for which no ICs or SLBs exist within the specified time.

MINBAT	MB, MINB	HOURS()
		DAYS()
		RANGE()
			DBs for which no batch ICs or SLBs exist within the specified time.

MINONL	MO, MINO	HOURS()
		DAYS()
		RANGE()
			DBs for which no online ICs or SLBs exist within the specified time.

EEQE	EQ, E	N/A	DBs that have error queue elements (EEQEs).
DBNOTCA	DC	N/A	DBs that are not in a CA group.
HALDBDISABLEDPART	HD, HALDBD	N/A	HALDBs that have disabled partitions.
HALDBNOTINIT	HN, HALDBN	N/A	HALDBs that are not initialized.
NOADS	NA	N/A	Produces the NOADSDEFINE and NOADSAVAIL reports.
NOADSDEFINE	ND, NOADSD	N/A	DEDBs that have no area data sets defined.
NOADSAVAIL	NV, NOADSA	N/A	DEDBs that have no area data sets available.

Table 2. The valid HCHECK options, abbreviations, parameters, and descriptions (continued)

Option	Abbreviations	Parameters	Description
BACKOUT	BO, B	N/A	DBs that show backout needed.
MINCARECS	MC. MINC	NUMBER()	DBs that do not have the minimum number of CA records available.
SECLOG	SL, SECL	N/A - global	No valid corresponding SECLOG entry for a PRILOG entry.
SECSLD	SS, SECS	N/A - global	No valid corresponding SECSLDS entry for a PRISLDS entry.

CARUN	CR, CAR,CARU	HOURS()
		DAYS()
		RANGE()
			CA group does not have a valid CARUN within specified time.

PROAUTH	PA, PR	N/A	DBs that indicate that prohibit authorization is on.

PISEC	PS, PI	HOURS()
		DAYS()
		RANGE()
			Primary IC with no corresponding valid secondary IC within that specified time.

VERIFY	VF, VER, VFY	N/A	Ensures that all recovery assets are cataloged.

You can specify just one of the HCHECK options within each HCHECK command. However, you can specify the HCHECK command any number of times within a single JCL input stream.

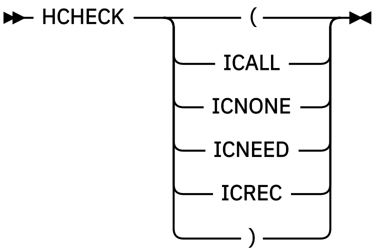
Also, you can specify the HCHECK option using the fully-qualified option, or any of the acceptable abbreviations that are displayed in the previous table.

For those options that require additional parameters, the HCHECK command is marked in error and is not processed if you do not specify the required additional parameters.

The following sections describe the HCHECK options and their relationships in more detail.

Image copy needed or recommended

Syntax for the ICALL, ICNONE, ICNEED, and ICREC options:



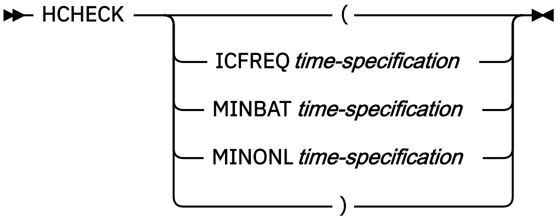
ICALL
Use this option to specify that the ICNONE, ICNEED. and ICREC options are to be processed.

ICNONE
Use this option to check for database data sets for which no image copy or System Level Backup has been taken.

ICNEED
Use this option to check for database data sets for which an image copy is needed.

ICREC
Use this option to check for database data sets for which an image copy is recommended.

No Batch or Online Image Copy within Specified Time Frame


HCHECK
(
ICFREQ *time-specification*
MINBAT *time-specification*
MINONL *time-specification*
)

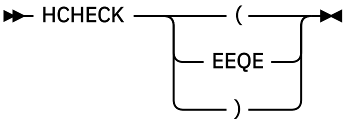
ICFREQ
Use this option to specify that a check is performed for valid Image Copies or SLBs within the specified time frame.

MINBAT
Use this option to specify that a check is performed for valid batch image copies or System Level Backups within the specified time frame.

MINONL
Use this option to specify to check for valid online image copies or System Level Backups within the specified time frame.

Databases that have extended error queue elements (EEQEs)

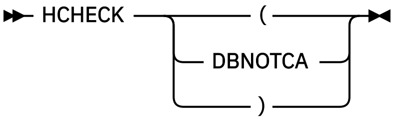
Syntax for the EEQE option:



EEQE
Use this option to check for databases that have EEQEs, indicating that recovery might be needed.

Databases not in a change accumulation group

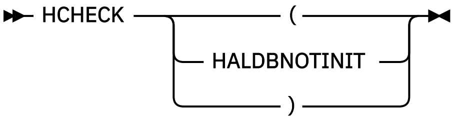
Syntax for the DBNOTCA option:



DBNOTCA
Use this option to check for databases that are not included in any change accumulation group. Databases that are not in change accumulation groups might adversely affect database recovery.

HALDB not initialized

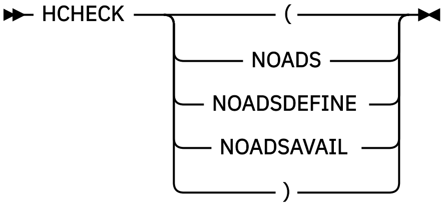
Syntax for the HALDBNOTINIT option:



HALDBNOTINIT
Use this option to check for High Availability Large Databases (HALDBs) that have not been initialized. These databases are currently unusable.

No area data set defined, no area data set available

Syntax for the NOADS, NOADSDEFINE, and NOADSAVAIL options:



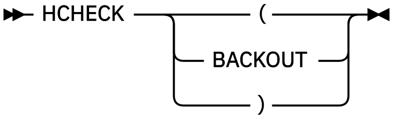
NOADS
Use this option to specify that the NOADSDEFINE and NOADSAVAIL options are processed. Use the NOADSDEFINE option to check for fast path databases that have no area data sets defined.

NOADSDEFINE
Use this option to check for fast path databases that have no area data sets defined.

NOADSAVAIL
Use this option to check for fast path databases that have no currently available area data sets.

Backout needed

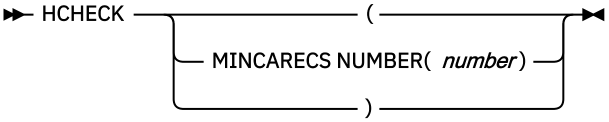
Syntax for the BACKOUT option:



BACKOUT
Use this option to check for databases marked as backout needed.

Minimum change accumulation records not available

Syntax for the MINCARECS option:

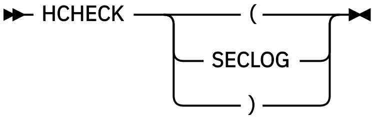


MINCARECS
Use this option to check for databases that do not have the specified number of change accumulation data sets available.

NUMBER(*number*)
Use this option keyword to specify the minimum acceptable available change accumulation data sets.

No valid corresponding SECLOG entry for the PRILOG record

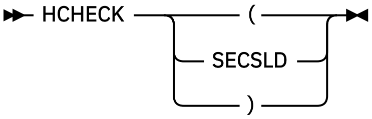
Syntax for the SECLOG option:



SECLOG
Use this option to check for databases that have a valid PRILOG entry, but no corresponding SECLOG record.

No valid corresponding SECSLDS entry for PRISLDS record

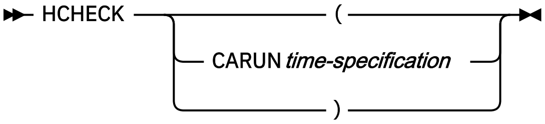
Syntax for the SECSLD option:



SECSLD
Use this option to check for databases that have a valid PRISLDS entry, but no corresponding SECSLDS record.

Change accumulation (CA) group does not have a valid CA run within the specified time range

Syntax for the CARUN option:

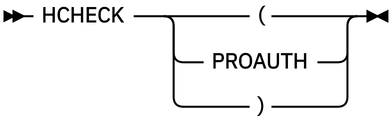


CARUN
Use this option to check for databases that do not have a valid change accumulation within the specified time frame.

time-specification
Use this required option keyword to specify the time frame to check. Specify the *time-specification* with either the HOURS(), DAYS(), or RANGE() keyword. You must specify one of the HOURS(), DAYS(), or RANGE() keywords or the HCHECK CARUN function will be rejected. For a description of HOURS(), DAYS(), or RANGE(), see "TIME and RANGE parameter processing" on page 24.

Database is in prohibit authorization state

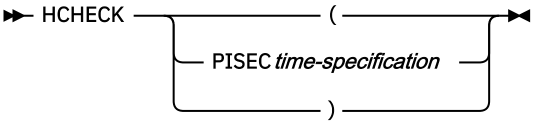
Syntax for the PROAUTH option:



PROAUTH
Use the PROAUTH option to check for databases that are in prohibit authorization state.

Primary image copy with no corresponding valid secondary image copy within the specified time range

Syntax for the PISEC option:



PISEC

Use this option to check for databases that have a valid primary image copy, but no valid secondary image copy within the specified time frame.

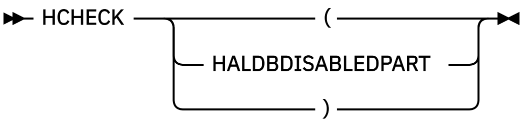
time-specification

Use this required option keyword to specify the time frame to check.

Specify the *time-specification* with either the HOURS(), DAYS(), or RANGE() keyword. You must specify one of the HOURS(), DAYS(), or RANGE() keywords or the HCHECK CARUN function will be rejected. For a description of HOURS(), DAYS(), or RANGE(), see "TIME and RANGE parameter processing" on page 24.

Disabled HALDB partition

Syntax for the HALDBDISABLEDPART option:

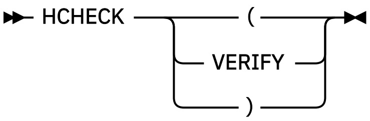


HALDBDISABLEDPART

Use this option to check for High Availability Large Databases (HALDBs) that have disabled partitions.

Verify that recovery assets are cataloged

Syntax for the VERIFY option:



VERIFY

Use this option to verify that the recovery assets associated with the selected databases are cataloged. All of the following recovery assets that are registered with DBRC or are in the IMS Recovery Expert repository are verified:

- Database data set names
- Image copy data set names
- System Level Backup data set names
- Log data set names
- Change accumulation data set names

HCHECK(VERIFY) has no time oriented parameter. It verifies only whether the latest Image Copy is in the catalog or not.

In the case of SLBs, the data set name in the Image Copy record is symbolic only. Therefore, it is the offload data set names that are being verified.

Non-cataloged offload data sets are not listed on the HCHECK(VERIFY) report. Only the fact that they are not cataloged is indicated. In order to determine which specific offload data set is not cataloged, use the VERIFY(ALLOC) or VERIFY(OPEN) function.

HCHECK usage examples

You can set up the Health Check function (HCHECK) in several different ways.

You can set up HCHECK to run on a scheduled basis to periodically check for problems that can affect database recovery. You can also use HCHECK at the Disaster Recovery (DR) site to verify that your DR site is recoverable. You must specify the options to be checked individually, each with its own HCHECK keyword. You can although, specify multiple HCHECK keywords within the same job step.

Example 1: Identify problems related to image copies

To identify problems that are related to image copies, specify the following control cards:

To report on any databases that do not have a batch or online image copy within the last 14 days, issue the first line in the next example.

To report on any databases for which one of the following conditions are true, issue the second line that is shown in the next example:

- No image copy has been taken
- An image copy is needed
- An image copy is recommended

```
HCHECK(ICFREQ DAYS(14))
HCHECK(ICALL)
```

Example 2: Identify problems with partitions or unavailable areas

To identify problems that are related to disabled partitions, partitions that are not initialized, or unavailable areas, specify these control cards:

```
HCHECK(HALDBDISABLEDPART)
HCHECK(HALDBNOTINIT)
HCHECK(NOADS)
```

Example 3: Identify databases with potential backout or authorization problems

To identify databases with potential backout or authorization problems, use this input:

```
HCHECK(BACKOUT)
HCHECK(PROAUTH)
```

HCHECK output

The HCHECK function report is written to the RPTHCHEK output DD.

If you do not specify the RPTHCHEK DD in the JCL , it will be dynamically allocated to SYSOUT=*. You can also have the output report written to SYSOUT, or to a data set. The characteristics of the output are RECFM=FBA,LRECL=133. Or, you can have the output report written to the IMS Tools Knowledge Base by specifying the REPORT(ITKBOUT(YES)) keyword and parameter.

Note: REPORT(ITKBOUT(YES) is the default.

The following figure shows a sample HCHECK report:

```
IMS Recovery Solution Pack V2.1.0 (5655-V86)
IMS DATABASE FACILITY EXTENDED FUNCTIONS HEALTH CHECKER REPORT             Run Time: 11/05/2021 12:30



********************************************************************************
-> Invocation Parameters
********************************************************************************

FUNCTION           NUMBER DAYS   HOURS  START DATE/TIME                     END DATE/TIME
----------------------------------------------------------------------------------------------------------------
CARUN                                   2021.309 10:01:01.100000 -08:00     2021.309 10:02:02.100000 -08:00
ICNEED



********************************************************************************
-> Database List
********************************************************************************

DATABASE    AREA/PART     DD             TYPE   DSN                                          PART-TYPE    OLR-PARTNER
-----------------------------------------------------------------------------------------------------------------------
F201P1                    F201P11        FF     IMSTOOL.IMSRE.QA.Q91A.F201P1.F201P11
H101P3      H101P31       H101P31X       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.X00001           INDEX
H101P3      H101P31       H101P31L       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.L00001           ILE
H101P3      H101P31       H101P31A       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.A00001           DATA
H101P3      H101P32       H101P32X       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.X00002           INDEX
H101P3      H101P32       H101P32L       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.L00002           ILE
H101P3      H101P32       H101P32A       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.A00002           DATA
H101P3      H101P33       H101P33X       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.X00003           INDEX
H101P3      H101P33       H101P33L       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.L00003           ILE
H101P3      H101P33       H101P33A       HALD   IMSTOOL.IMSRE.QA.Q91A.H101P3.A00003           DATA


********************************************************************************
-> Health Checker Report
********************************************************************************

DBD      DDN/AREA TYPE FUNCTION           HEALTH CHECKER MESSAGE(S)  * = "WITHIN SPECIFIED RANGE"
----------------------------------------------------------------------------------------------------------------
F201P1   F201P11  FF   CARUN              CA group has no valid CA run *
H101P31  H101P31A HALD ICNEED             Image Copy is Needed
H101P33  H101P33A HALD ICNEED             Image Copy is Needed
```

Figure 4. Sample HCHECK output report

The major report sections are described here with their headings:

Invocation Parameters

This section of the report shows the options that were used by HCHECK processing. In this example you can see that the CARUN and ICNEED options were specified. Any parameters that you specified for the options are also displayed. In the example of CARUN, a range was specified.

Database List

This section of the report shows the databases that were processed in this run. The report headings and descriptions are listed here:

DATABASE

This column lists the names of the databases that were processed.

AREA/PART

This column lists the area names for fast path databases or the partition names for HALDB databases. This column is blank for full-function databases.

DD

This column shows the DD name that is associated with this area, partition, or full-function database.

TYPE

This column lists the type of databases that were processed. Valid values are FF for full-function databases, FP for fast path areas, and HALD for HALDB partitions.

DSN

This column lists the data set names that are associated with this area, partition, or full function database.

PART-TYPE

For HALDB partitions, this column specifies the type of data set. Valid values are INDEX for the index portion of the HALDB, ILE for the index list element (or ILDS) portion of the HALDB, or DATA for the data portion of the HALDB.

OLR-PARTNER

For HALDB partitions that are IMS Online Reorganization (OLR) capable, this column specifies the OLR partner DD name that is associated with the partition.

Health Checker Report

This section includes a line for any condition that HCHECK identifies as a problem. The report headings and their descriptions are:

DBD

This column specifies the database name for which the condition was identified. If the database is a HALDB type, this column indicates the partition name.

DDN/AREA

This column specifies the DD name (for full-function or HALDB databases), or the fast path area name for which the condition was identified.

TYPE

This column indicates the type of database that was processed. Valid values are FF (for full-function database), FP (for Fast Path area), and HALD (for HALDB partition).

FUNCTION

This column indicates the HCHECK option that is associated with the condition.

HEALTH CHECKER MESSAGE(S)

This column indicates the condition that was identified by HCHECK. Possible values and their meanings are listed in the next table. An asterisk in the message column indicates that the condition is associated with a specified range.

Table 3. Health checker messages and their issuing functions

Message	Function
No Image Copy has been taken	ICNONE
No Image Copy taken, but SLB exists. Note: This includes SLBs for which there is no Image Copy record in the RECON.	ICNONE
Image Copy is Needed	ICNEED
Image Copy is Recommended	ICREC
No Batch Image Copy taken *	MINBAT
No Batch IC taken, but SLB exists *	MINBAT
No Online Image Copy taken *	MINONL
No Online IC taken, but SLB exists *	MINONL
DB has EEQE, so recovery may be needed	EEQE
DB is not in a Change Accumulation group	DBNOTCA
HALDB is not initialized	HALDBNOTINIT
No Area Data Set is defined	NOADSDEFINE
No Area Data Set is available	NOADSAVAIL
One or more Area Data Sets are unavailable	NOADSAVAIL
Backout is needed	BACKOUT
Minimum CA records not available *	MINCARECS
No corresponding SECLOG for PRILOG	SECLOG
No corresponding SECSLDS for PRISLDS	SECSLD

Table 3. Health checker messages and their issuing functions (continued)

Message	**Function**
CA group has no valid CA run *	CARUN
HALDB has Prohibit Authorization status	PROAUTH
Primary IC has no valid secondary IC *	PISEC
HALDB partition is disabled	HALDBDISABLEDPART
DB Recovery Asset(s) not catalogued - SLB	VERIFY
DB Recovery Asset(s) not catalogued	VERIFY
LOG Recovery Asset(s) not catalogued	VERIFY
CA Recovery Asset(s) not catalogued	VERIFY
Recovery Asset(s) not catalogued	VERIFY

Chapter 7. Recovery Point Creation (RPCR) function

The Recovery Point Creation (RPCR) function allows you to create recovery points for one or more databases in IMS Database Recovery Facility Extended Function.

Topics:

RPCR overview

Recovery Point Creation (RPCR) sets the allocation records in the RECON data sets to be updated, indicating a period of time where the database is not allocated by an IMS subsystem, and thereby creating a valid recovery point.

Creating a valid recovery point

The following items are performed to ensure that a valid database recovery point is created when RPCR runs. In this list, references to *database* include databases, partitions, and areas.

- Issues a CHANGE.DB NOAUTH command for each database in the list to set prohibit authorization.
- Issues an ENQ for each database to communicate with BMPs and CICS®/ODBA applications.
- Issues a /DIS DB command to obtain the status of each database.
- Checks if any BMPs are currently holding any of the databases in the list. If a BMP is currently holding a database that is in the list, RPCR issues a WTOR and waits for *n* number of seconds as specified on the WAITTIME() parameter. If you respond to the outstanding WTOR, the check is performed again and the BMP list is refreshed. This process continues until the wait time expires, until all of the BMPs have been paused or ended, or until you cancel the RPCR operation. If you specified PARTIAL(Y), RPCR processes those databases that are not allocated at the end of the WAITTIME. If you specified PARTIAL(N), RPCR fails if any databases are allocated at the end of the WAITTIME.
- Issues synchronized /DBR or /DBD commands for the databases across all systems on which the database is allocated.
- Dynamically allocates all database data sets for all databases in the list with DISP=OLD to prevent allocation.
- Checks for new allocation records in the RECON data sets to ensure that a database is not allocated after the /DBR or /DBD and before all other databases in the list that were deallocated by RPCR. This check is done to ensure that all of the databases that were deallocated by RPCR have a common recovery point.
- Optionally, issues a /SWI OLDS command on all IMS subsystems on which databases were deallocated.
- Dynamically deallocates all database data sets that were previously allocated as DISP=OLD.
- Issues a CHANGE.DB AUTH command to reset prohibit authorization.
- Optionally restarts the databases, if it was already started before RPCR.
- Issues a DEQ command for each previously issued ENQ to allow BMPs and CICS/ODBA applications to continue.

Conditions and dependencies

- RPCR uses the IMS Tools Online System Interface to issue all IMS commands.

The XCF group name that is used for IMS Tools Online System Interface must be unique within each RECONID and the IMS online subsystems that must share the same RECONs to avoid conflicts with IMS Tools Online System Interface and RPCR.

- The RPCR function requires that you specify the target databases using the DBLIST supporting function.
- RPCR cannot be used in combination with either the RECONCOPY or the ALTRECONS supporting function.

  If either RECONCOPY or ALTERCONS is specified with RPCR, the job step terminates.

Automatic pausing of BMPs to allow RPCR to temporarily deallocate databases

When RPCR attempts to issue a /DBR or /DBD command for a database, an active BMP or a CICS/ODBA thread can prevent successful deallocation. To prevent that, install the BMP interface and the CICS/ODBA interface.

Installing the Extended Functions BMP interface and the CICS/ODBA interface will pause these applications at checkpoint time to allow RPCR to complete. Installing the IMS RE BMP interface and the CICS/ODBA interface is highly recommended.

If the BMP interface is installed, any active BMPs that reference databases that are specified in the DBLIST at the time RPCR is invoked, will be paused at the next checkpoint and continued after RPCR processing completes.

If a BMP cannot be paused because it has not reached a checkpoint or the BMP Interface was not installed or used by the active BMP, RPCR displays a list of BMPs so that you can respond accordingly.

If a BMP list is displayed, all full-function and HALDB BMPs are displayed first, followed by any FastPath BMPs.

Alternative to using Recovery Point Create (RPCR)

An alternative to using the Recovery Point Create (RPCR) command is to use the Create Recovery Point function in the IMS Recovery Expert tool.

This function is driven from an application profile and makes use of the IMS QUIESCE command. This function accomplishes the same result but does not require the overhead of taking the database offline and having to unallocate and reallocate the database.

For information on this function, refer to the topic "Recovery using application operations" in the *IMS Recovery Solution Pack for z/OS: IMS Recovery Expert User's Guide*.

RPCR input

Run the Recovery Point Create (RPCR) function by running RPCR.

RPCR function syntax

```
    RPCR(  CMDTYPE(DBR | DBD)
           SYNCLOG(YES | Y | NO | N)
           STARTDB(YES | Y | NO | N)
           PARTIAL(YES | Y | NO | N)
           WAITTIME(0 - 1800)
        )
```

Parameter reference

The RPCR parameters are described here:

CMDTYPE(DBR | DBD)
Use the CMDTYPE parameter to specify whether RPCR will issue a synchronized /DBR or a synchronized /DBD command to deallocate the databases. The ability to issue synchronized /DBR and /DBD commands is provided through IMS Tools Online System Interface.

If you specify CMDTYPE(DBR), a synchronized /DBR is issued for all databases in the DBLIST.

If you specify CMDTYPE(DBD), a synchronized /DBD is issued for all full-function databases in the DBLIST. Because there is no /DBD command for fast path databases, a synchronized /DBR will be issued for all fast path databases. The default is CMDTYPE(DBR).

SYNCLOG(YES | Y | NO | N)

Use the SYNCLOG parameter to specify whether RPCR will issue a /SWI OLDS command on all connected IMS systems when the RPCR is completed. Performing a synchronized log switch will assist in creating a coordinated set of archived logs associated with RPCR.

The default is SYNCLOG(YES).

STARTDB(YES | Y | NO | N)

Use the STARTDB parameter to specify whether RPCR should start the databases when RPCR completes.

If you specify STARTDB(Y), the databases are put back into their original state before RPCR. For example, if a database was not allocated to the online system prior to RPCR, the STARTDB(Y) parameter will be ignored for that database.

If you specify STARTDB(NO), the databases are left in their /DBR or /DBD state.

The default is STARTDB(YES).

PARTIAL(YES | Y | NO | N)

Use the PARTIAL parameter to specify whether RPCR will attempt to create a recovery point for some of the specified databases, even though RPCR was unable to create a recovery point for other databases.

If you specify PARTIAL(YES), RPCR will attempt to create a recovery point for as many databases in the DBLIST as possible. If one or more of the databases cannot be deallocated, RPCR is performed for those databases that can be deallocated.

If you specify PARTIAL(NO), RPCR will return an error indicating that it is unable to create a recovery point for at least one database.

The default is PARTIAL(NO).

WAITTIME(YES | Y | NO | N)

The WAITTIME parameter specifies the maximum amount of time, in seconds, that RPCR should wait when attempting to deallocate the databases in the DBLIST. If RPCR is unable to deallocate any database in the DBLIST before the specified time has expired, it will return an error message and RPCR for that database will fail. A value of zero indicates that there is no limit to the amount of time RPCR will wait for the database to become deallocated.

Recommendation: Install the BMP Pausing interface that is provided with IMS RE so that RPCR can pause any BMPs that have access to the databases in the DBLIST and prevent any new BMPs from starting during the RPCR process. The BMP Pausing interface will temporarily halt processing of a BMP after the next checkpoint to allow the /DBR or /DBD command to complete. Also consider installing the BMP Pausing interface as an SMP/E usermod so that all BMP JCL does not need to be modified.

The default is WAITTIME(300), which is 300 seconds (or 5 minutes).

RPCR usage examples

The Recovery Point Creation function (RPCR) uses several different interfaces to perform its functions.

The interfaces that RPCR uses require specific installation procedures. The interfaces are:

- IMS Tools Online System Interface
- BMP Pausing interface
- ODBA/CICS interface

Example 1: /DBR, perform a /SWI OLDS, and start 3 databases

When the following example code is run, RPCR attempts to:

- /DBR three databases within two minutes (120 seconds) only if all of the databases can be deallocated
- Perform a /SWI OLDS to switch the online log data sets
- Start the databases after RPCR only if they have been deallocated by RPCR

```
    DBLIST ( DB (dbd1 dbd2 dbd3) )
    RPCR   ( CMDTYPE(DBR)
             SYNCLOG(YES)
             STARTDB(YES)
             PARTIAL(NO)
             WAITTIME(120)
           )
```

Example 2: /DBD the databases that can be deallocated, then leave databases in the same state

When the following example code is run, RPCR attempts to:

- /DBD any of the two databases within five minutes (300 seconds)
- Leave the affected databases in the same state as the state that was performed

```
    DBLIST ( DB (dbd1 dbd2 dbd3) )
    RPCR   ( CMDTYPE(DBD)
             SYNCLOG(NO)
             STARTDB(NO)
             PARTIAL(YES)
             WAITTIME(300)
           )
```

RPCR output

The RPCR function report is written to the RPTRPCR output DD.

If you do not specify the RPTRPCR DD in the JCL, it is dynamically allocated to SYSOUT=*. Report output can be written to SYSOUT or to a data set. The characteristics of the output are RECFM=FBA,LRECL=133. You can also write the report output to the IMS Tools Knowledge Base if you specify the REPORT(ITKBOUT(YES)) keyword and parameter.

REPORT(ITKBOUT(YES) is the default.

Report

The following figure shows a sample RPCR report:

```
IMS Recovery Solution Pack V2.1.0 (5655-V86)
IMS DATABASE FACILITY EXTENDED FUNCTIONS RECOVERY POINT CREATE REPORT       Run Time: 11/08/2021 11:56




*********************************************************************************
-> Invocation Parameters
*********************************************************************************

      CMDTYPE:   DBR                       SYNCLOG:   NO
      STARTDB:   YES                       PARTIAL:   YES
      WAITTIME:  00000010




*********************************************************************************
-> Database List
*********************************************************************************

DATABASE    AREA/PART     DD             TYPE   DSN                                            PART-TYPE    OLR-PARTNER
------------------------------------------------------------------------------------------------------------------------
F101P1                    F101P11        FF     IMSTOOL.IMSRE.QA.Q91A.F101P1.F101P11
F101X1                    F101X11        FF     IMSTOOL.IMSRE.QA.Q91A.F101X1.F101X11
F201P1                    F201P11        FF     IMSTOOL.IMSRE.QA.Q91A.F201P1.F201P11
F202P1                    F202P11        FF     IMSTOOL.IMSRE.QA.Q91A.F202P1.F202P11
F202P1                    F202P12        FF     IMSTOOL.IMSRE.QA.Q91A.F202P1.F202P12


*********************************************************************************
-> Recovery Point Create Report
*********************************************************************************

DATABASE  AREA/PART TYPE   +--- RPCR TIME (UTC) ----+  +-- RPCR TIME (LOCAL) ---+   +- INFORMATIONAL MESSAGES ---------+
------------------------------------------------------------------------------------------------------------------------
F101P1              IMS                                                              FAILED   DB IS CURRENTLY BEING USED
F101X1              IMS                                                              FAILED   DB IS CURRENTLY BEING USED
F201P1              IMS     2021.312 19:56:18.2 +00:00  2021.312 11:56:18.2 -08:00   SUCCESS
F202P1              IMS     2021.312 19:56:18.2 +00:00  2021.312 11:56:18.2 -08:00   SUCCESS
```

Figure 5. Sample RPCR report

The RPCR report contains the following sections:

Invocation Parameters

This section of the report shows the options that were used for this run of RPCR process.

CMDTYPE

This field indicates whether a DBR or DBD command was requested.

STARTDB

This field indicates whether the databases will be restarted after RPCR is complete.

WAITTIME

This field indicates the value specified for the WAITTIME() parameter.

SYNCLOG

This field indicates whether a synchronized log switch will be performed after RPCR is complete.

PARTIAL

This field indicates whether a partial RPCR is allowed.

Database List

This section of the report shows the databases that were processed in this run.

DATABASE

This column specifies the database name.

AREA/PART

This column specifies the area name for fast path databases or the partition name for HALDB databases. This column is blank for full function databases.

DD

This column specifies the DD name that is associated with this area, partition, or full function database.

TYPE

This column specifies the type of database that was processed. Valid values are FF for full function databases, FP for fast path areas, and HALD for HALDB partitions.

DSN

This column specifies the data set name that is associated with this area, partition, or full function database.

PART-TYPE

For HALDB partitions, this column specifies the type of data set. Valid values are INDEX for the index portion of the HALDB, ILE for the index list element (or ILDS) portion of the HALDB, or DATA for the data portion of the HALDB.

OLR-PARTNER

For HALDB partitions that are OLR-capable, this column specifies the OLR partner DD name that is associated with the partition.

Recovery Point Create Report

This section of the report includes a line for each database, area, and partition that was processed and indicates the final status.

DATABASE

This column specifies the name of the databases that were processed.

AREA/PART

This column specifies the fast path area name or HALDB partition name that is associated with the database. If the database is full function, this column is blank.

TYPE

This column indicates the type of database that was processed. Valid values are IMS for full function database, FP for fast path area, and HALDB for HALDB partition.

RPCR TIME (UTC)

This column indicates the RPCR time in coordinated universal time (UTC) format. If RPCR failed, this column will be blank.

RPCR TIME (LOCAL)

This column indicates the RPCR time in local format. If RPCR failed, this column will be blank.

INFORMATIONAL MESSAGES

This column indicates the final disposition for the database. The valid messages are shown in the following table.

Table 4. RPCR report informational messages and their descriptions

Message text	Description
INVALID TOKEN FOR FUNCTION	Invalid function token passed to RPCR. Contact IBM Software Support.
TOKEN NOT FOR SAME TCB	Invalid TCB token passed to RPCR. Contact IBM Software Support.
UNABLE TO JOIN XCF GROUP	An attempt to join the XCF group for IMS Tools Online System Interface has failed. Ensure that you have specified the correct XCF group in the RECONID record and the IMS subsystems were initialized correctly with IMS Tools Online System Interface.
TARGET SYSTEM NOT AVAILABLE	An attempt to issue an IMS command for one or more target IMS systems has failed. Ensure that the IMS system is still active.
UNABLE TO OBTAIN RECON INFO	RPCR was unable to access the RECON data sets to obtain RECON information. Review the job log for additional messages that indicate the specific error.
UNABLE TO GET ONLINE DB STATUS	RPCR was unable to access the RECON data sets to obtain database information. Review the job log for additional messages that indicate the specific error.

Table 4. RPCR report informational messages and their descriptions (continued)

Message text	Description
UNABLE TO SWITCH LOG	RPCR was unable to complete the /SWI OLDS command processing. Review the job log and the IMS messages for additional messages that indicate the specific error.
UNABLE TO DBR A DB	RPCR was unable to complete the /DBR command processing for one or more databases. Review the job log and the IMS messages for additional messages that indicate the specific error.
UNABLE TO DBD A DB	RPCR was unable to complete the /DBD command processing for one or more databases. Review the job log and the IMS messages for additional messages that indicate the specific error.
UNABLE TO START DB	RPCR was unable to start one or more databases. Review the job log and the IMS messages for additional messages that indicate the specific error.
OPTIONS CONFLICT WITH DB TYPE	The RPCR options that were specified conflict with database type. Contact IBM Software Support.
GETMAIN REQUEST FAILED	Unable to acquire storage. Ensure that the region size for the Extended Functions job is adequate and that storage is available.
NAME TOKEN SERVICES FAILED	An error was encountered using MVS name token services. Contact IBM Software Support.
UNABLE TO LOAD PROGRAM	RPCR was unable to load a required program. Ensure that Extended Functions has been correctly installed and that all required load modules are included in the JOBLIB or STEPLIB.
DYNAMIC ALLOCATION FAILED	Dynamic allocation failed for one or more database data sets. Review the job log for details on the specific data set. This normally indicates that a job or user outside of IMS has the data set in use.
UNABLE TO LOCATE SUBTASK SVCS	Unable to locate subtask services. Contact IBM Software Support.
SVCS NOT AVAILABLE FOR SUBTASK	Subtask services are unavailable. Contact IBM Software Support.
ERROR STARTING SERVICE SUBTASK	Unable to start the subtask services. Contact IBM Software Support.
IMS ENVIRONMENTAL ERROR	RPCR encountered an error within the IMS environment. contact IBM Software Support.
NOT APF AUTHORIZED	The RPCR program is not running in an APF authorized environment. Ensure that the load libraries in the JOBLIB or STEPLIB are APF authorized.
MISSING IMS DD	The specific RPCR options that were specified require the IMS DD to be specified. Add the appropriate IMS DD concatenation.

Table 4. RPCR report informational messages and their descriptions (continued)

Message text	Description
INCORRECT DSNAME FOR RECON	The data set name that is associated with the RECON that is being accessed does not match the data set name that was passed to RPCR. Ensure that the same RECON data sets are accessed by the Extended Functions job (JCL / RECONID) and the online IMS subsystem.
DB IN USE FOR TIMEOUT DURATION	One or more databases were in use for the timeout duration, and RPCR failed. If you specified PARTIAL(Y), some databases might have succeeded. If you specified PARTIAL(N), RPCR failed.
DB NOT PROCESSED, PARTIAL(N)	One or more databases were in use for the timeout duration and PARTIAL(N) was specified. All other databases in the list were skipped.
DB IS NOT REGISTERED	One or more databases that were passed to RPCR were not registered to DBRC. Ensure that all databases are registered.
DB TYPE IS UNSUPPORTED	The database type is either unsupported by RPCR or cannot be determined.
UNABLE TO LOCATE DBD	RPCR could not locate the DBD in the IMS DD concatenation. Ensure that the correct blocks are located in the IMS DD concatenation.
UNABLE TO LOCATE PART INFO	RPCR could not locate information for a HALDB partition. Ensure that the correct blocks are located in the IMS DD concatenation.
DUPLICATE DBD/PART IN DBLIST	A duplicate database or partition was passed to RPCR in the DBLIST. Correct the DBLIST.
ALLOCATED AFTER DBR	RPCR has determined that one or more of the databases processed were allocated after they were processed by the /DBR command and before all other databases in the list were processed by the /DBR command. The RPCR process must fail because it cannot ensure that a common recovery point was created for all of the databases.
CMDTYPE(DBD) NOT VALID FOR DB	Internal error, contact IBM Software Support.
DB TYPE NOT PROVIDED	Internal error, contact IBM Software Support.
DB DSLIST NOT PROVIDED	Internal error, contact IBM Software Support.
DB IS CURRENTLY BEING USED	One or more databases were in use for the timeout duration, and RPCR failed.

Chapter 8. Recovery Point Identification (RPID) function

The Recovery Point Identification function (RPID) will help you identify common recovery time spans for databases or help you identify a valid recovery point for a single database in IMS Database Recovery Facility Extended Function.

Topics:

- "RPID overview" on page 77
- "RPID input" on page 79
- "RPID usage examples" on page 79
- "RPID output" on page 79

RPID overview

The RPID function can be used to analyze the RECON data sets and locate valid recovery time spans for one or more databases without the need to manually compare DBRC histories for each database.

Recovery time span (RTS)

A recovery time span (RTS) is a period of time during which a database is not allocated and, therefore, the database can be recovered. Although valid recovery times might exist during which the database is allocated, RPID does not currently analyze log records and therefore, does not report on these recovery times.

The following sequence of events illustrates the concept of a simple RTS:

Event 1:

Database X was started with the /STA command at Time 1:

```
21.320 08:53:37.924206-0800
```

Allocated => 21.320 08:53:37.924206-0800 to 21.320 08:54:16.730540-0800

Event 2:

Database X was unallocated with the /DBR command at Time 2:

```
21.320 08:54:16.730541-0800
```

Event 3:

Database X was started with the /STA command at Time 3:

```
21.320 08:59:01. 75235-0800
```

Time Span => 21.320 08:54:16.730541-0800 to 21.320 08:59:01.75235-0800

The sequence of these three events shows that:

- Database X was successfully started and allocated at Time 1
- The database remained allocated until it was unallocated at Time 2
- Database X was successfully started again at Time 3

This sequence shows that an RTS exists between Time 2 and Time 3. The database can be recovered successfully to any timestamp within this RTS.

Timestamp recovery (TSR)

RPID analyzes database allocation records in the RECON data sets to identify RTSs to which one or more databases can be recovered by using timestamp recovery (TSR). RPID also analyzes database recovery records in the RECONs to identify recovery point timestamps to which individual databases can be recovered by using TSR.

Recovery point

A recovery point is a timestamp that can be used as input to IMS Database Recovery Facility, or in some cases to other database recovery utilities such as, the standard IMS Recovery utility (DFSURDB0), as a time to which a database can be restored.

These recovery points include times when a database was the object of any of the following utility runs, all of which are recorded in RECON recovery records:

- A batch image copy (IC) run
- An online IC run (in cases where the databases are not truly allocated)
- A complete change accumulation (CA) run
- A recovery run (Full or Partial), or an offline reorganization run

HALDB IMS OLRs and IMS Online Reorganization Facility

The run times of HALDB IMS Online Reorganizations (IMS OLRs) and IMS Online Reorganization Facility are also analyzed.

For IMS OLR and IMS Online Reorganization Facility, RPID indicates whether the databases were truly allocated and therefore, whether the run time is a valid TSR point.

HALDB partitions that have been part of an IMS Online Reorganization have two sets of DBDSs, that are known as A-J and M-V.

At any point in time, either set might be active and the other set might be inactive (an IMS OLR is not in progress), or either set might be a shadow (an IMS OLR is in progress).

Both sets of DBDSs are recognized as one logical partition when RPID analyzes their associated RECON records.

Recovery points available for recovery

When RPID lists the recovery time spans it also indicates specific recovery points (timestamps) that are available for recovery including:

- Times of complete change accumulation runs.
- Times of batch image copy runs, and times of online image copy runs when the databases are not allocated.
- Times of offline reorganizations and online reorganizations when the databases are not allocated. In the second case (OLR), RPID indicates whether there was a subsequent image copy during the recovery time range.
- Times of database recovery, either full or partial recoveries.

Conditions and dependencies

- To manually create a recovery time span or recovery point for a TSR for all of the databases in a group, run the Recovery Point Create (RPCR) function that is described in Recovery Point Creation function (RPCR).
- The RPID function requires that you specify the target databases by using the DBLIST supporting function.

- The RPID function can use the RECON copy data sets that are created by the RECONCOPY supporting function, or the data sets identified on the ALTRECONS supporting function.

  Note: Use either RECONCOPY or ALTRECONS in conjunction with RPID to prevent input/output (I/O) to the live RECON data sets.

RPID input

Run the Recovery Point Identification function by running RPID.

RPID function syntax

```
RPID( RANGE(range-specification) )
```

Parameter reference

The RPID parameters are described here:

RANGE(*range-specification*)

Use the required RANGE parameter to specify the beginning and ending time frame (or the time range) that is used by RPID to locate recovery time spans. There is no default for this keyword.

The format of the range specification can be found in the "TIME and RANGE parameter processing" on page 24.

RPID usage examples

Use the Recovery Point Identification function (RPID) to identify valid recovery time spans that occur between specified time spans for a single or group of databases.

You can use the RPID function to identify common recovery time spans for the following database types:

- A single database that is not associated with or related to any other database
- A group of databases that logically belong together (for example, a CAGRP, a RECOVGRP, or other DB groupings)

Example of identifying valid recovery times

To identify valid recovery time spans that occurred between 21.291 06:30:01.1 and 21.291 11:30:01.1 for one HDAM database and one HIDAM database with a secondary index and an external logical, you would specify these control cards:

```
DBLIST ( DB(HDAM1 HIDAMDB1 HIDAMIX PRIMESX EXTLOGDB) )
RPID   ( RANGE (‘21.291 06:30:01.1',‘21:291 11:30:01.1') )
```

In this example, RPID determines a separate recovery time span for each database that you specified in the DBLIST. This recovery time span is the time frame in which a database was not allocated during the specified range. RPID also determines recovery time spans that are common to all of the databases in the DBLIST. This time frame is a period during which all of the databases were not allocated at the same time during the specified range.

RPID output

The output of the RPID function is written to the RPTRPID DD.

If you do not specify the RPTRPID DD in the JCL, it is dynamically allocated to SYSOUT=*. You can write the RPID report output to SYSOUT or to a data set. The characteristics of the output are RECFM=FBA,LRECL=133. You can write the report to the IMS Tools Knowledge Base repository by specifying the REPORT(ITKBOUT(YES)) keyword and parameter.

The default is REPORT(ITKBOUT(YES)).

The RPID report consists of three report sections. The following figure shows the first two sections of the RPID report:

```
IMS DATABASE FACILITY EXTENDED FUNCTIONS V2.1.0 (5655-V86)
IMS DATABASE FACILITY EXTENDED FUNCTIONS RECOVERY POINT ID REPORT          Run Time: 10/08/2021 19:26

********************************************************************************
-> Invocation Parameters:
********************************************************************************


Time Format           : UTC
Time Range            : 19.283 14:15:40.1+0000 to 21.364 14:15:40.1+0000



********************************************************************************
-> Database List
********************************************************************************

DATABASE    AREA/PART     DD             TYPE   DSN                                           PART-TYPE    OLR-PARTNER
-------------------------------------------------------------------------------------------------------------------------
DBOHIDK5    POHIDKA       POHIDKAA       HALD   IMSTESTS.DBOHIDK5.A00001                      DATA        POHIDKAM
DBOHIDK5    POHIDKA       POHIDKAB       HALD   IMSTESTS.DBOHIDK5.B00001                      DATA        POHIDKAN
DBOHIDK5    POHIDKA       POHIDKAL       HALD   IMSTESTS.DBOHIDK5.L00001                      ILE
DBOHIDK5    POHIDKA       POHIDKAM       HALD   IMSTESTS.DBOHIDK5.M00001                      DATA        POHIDKAA
DBOHIDK5    POHIDKA       POHIDKAN       HALD   IMSTESTS.DBOHIDK5.N00001                      DATA        POHIDKAB
DBOHIDK5    POHIDKA       POHIDKAX       HALD   IMSTESTS.DBOHIDK5.X00001                      INDEX       POHIDKAY
DBOHIDK5    POHIDKA       POHIDKAY       HALD   IMSTESTS.DBOHIDK5.Y00001                      INDEX       POHIDKAX
DBOHIDK5    POHIDKB       POHIDKBA       HALD   IMSTESTS.DBOHIDK5.A00002                      DATA        POHIDKBM
DBOHIDK5    POHIDKB       POHIDKBB       HALD   IMSTESTS.DBOHIDK5.B00002                      DATA        POHIDKBN
DBOHIDK5    POHIDKB       POHIDKBL       HALD   IMSTESTS.DBOHIDK5.L00002                      ILE
DBOHIDK5    POHIDKB       POHIDKBM       HALD   IMSTESTS.DBOHIDK5.M00002                      DATA        POHIDKBA
DBOHIDK5    POHIDKB       POHIDKBN       HALD   IMSTESTS.DBOHIDK5.N00002                      DATA        POHIDKBB
DBOHIDK5    POHIDKB       POHIDKBX       HALD   IMSTESTS.DBOHIDK5.X00002                      INDEX       POHIDKBY
DBOHIDK5    POHIDKB       POHIDKBY       HALD   IMSTESTS.DBOHIDK5.Y00002                      INDEX       POHIDKBX
DBOHIDK5    POHIDKC       POHIDKCA       HALD   IMSTESTS.DBOHIDK5.A00003                      DATA        POHIDKCM
DBOHIDK5    POHIDKC       POHIDKCB       HALD   IMSTESTS.DBOHIDK5.B00003                      DATA        POHIDKCN
DBOHIDK5    POHIDKC       POHIDKCL       HALD   IMSTESTS.DBOHIDK5.L00003                      ILE
DBOHIDK5    POHIDKC       POHIDKCM       HALD   IMSTESTS.DBOHIDK5.M00003                      DATA        POHIDKCA
DBOHIDK5    POHIDKC       POHIDKCN       HALD   IMSTESTS.DBOHIDK5.N00003                      DATA        POHIDKCB
DBOHIDK5    POHIDKC       POHIDKCX       HALD   IMSTESTS.DBOHIDK5.X00003                      INDEX       POHIDKCY
DBOHIDK5    POHIDKC       POHIDKCY       HALD   IMSTESTS.DBOHIDK5.Y00003                      INDEX       POHIDKCX
DBOHIDK5    POHIDKD       POHIDKDA       HALD   IMSTESTS.DBOHIDK5.A00004                      DATA        POHIDKDM
DBOHIDK5    POHIDKD       POHIDKDB       HALD   IMSTESTS.DBOHIDK5.B00004                      DATA        POHIDKDN
DBOHIDK5    POHIDKD       POHIDKDL       HALD   IMSTESTS.DBOHIDK5.L00004                      ILE
DBOHIDK5    POHIDKD       POHIDKDM       HALD   IMSTESTS.DBOHIDK5.M00004                      DATA        POHIDKDA
DBOHIDK5    POHIDKD       POHIDKDN       HALD   IMSTESTS.DBOHIDK5.N00004                      DATA        POHIDKDB
DBOHIDK5    POHIDKD       POHIDKDX       HALD   IMSTESTS.DBOHIDK5.X00004                      INDEX       POHIDKDY
DBOHIDK5    POHIDKD       POHIDKDY       HALD   IMSTESTS.DBOHIDK5.Y00004                      INDEX       POHIDKDX
DBOVLFPC    VLOSAM01      VLOSAM01       FF     DBOVLFPC.XABCD01O
DBVHDJ05    PVHDJ5A       PVHDJ5AA       HALD   IMSTESTS.DBVHDJ05.A00001                      DATA
DBVHDJ05    PVHDJ5A       PVHDJ5AB       HALD   IMSTESTS.DBVHDJ05.B00001                      DATA
DBVHDJ05    PVHDJ5A       PVHDJ5AL       HALD   IMSTESTS.DBVHDJ05.L00001                      ILE
DBVHDJ05    PVHDJ5B       PVHDJ5BA       HALD   IMSTESTS.DBVHDJ05.A00002                      DATA
DBVHDJ05    PVHDJ5B       PVHDJ5BB       HALD   IMSTESTS.DBVHDJ05.B00002                      DATA
DBVHDJ05    PVHDJ5B       PVHDJ5BL       HALD   IMSTESTS.DBVHDJ05.L00002                      ILE
DBVHDJ05    PVHDJ5C       PVHDJ5CA       HALD   IMSTESTS.DBVHDJ05.A00003                      DATA
DBVHDJ05    PVHDJ5C       PVHDJ5CB       HALD   IMSTESTS.DBVHDJ05.B00003                      DATA
DBVHDJ05    PVHDJ5C       PVHDJ5CL       HALD   IMSTESTS.DBVHDJ05.L00003                      ILE
DBVHDJ05    PVHDJ5D       PVHDJ5DA       HALD   IMSTESTS.DBVHDJ05.A00004                      DATA
DBVHDJ05    PVHDJ5D       PVHDJ5DB       HALD   IMSTESTS.DBVHDJ05.B00004                      DATA
DBVHDJ05    PVHDJ5D       PVHDJ5DL       HALD   IMSTESTS.DBVHDJ05.L00004                      ILE
DEDBDD01    DD01AR0       DD01AR0        FP     DD01AR0
DEDBJN24    DB24A000      DB24A000       FP     DB24A000
DEDBJN24    DB24A001      DB24A001       FP     DB24A001
DEDBJN24    DB24A239      DB24A239       FP     DB24A239
DHVNTZ02    HIDAM         HIDAM          FF     DHVNTZ02.FKXXI01E
DHVNTZ02    HIDAM2        HIDAM2         FF     DHVNTZ02.FKXXI02E
DIVNTZ02    DBHVSAM1      DBHVSAM1       FF     DIVNTZ02.FJXXS01K
DIVNTZ02    DBHVSAM2      DBHVSAM2       FF     DIVNTZ02.FJXXS01E
DXVNTZ02    XDLBT04I      XDLBT04I       FF     DXVNTZ02.FKXXSIXK
D2XHDJ05    P2XHDJA       P2XHDJAA       HALD   IMSTESTS.D2XHDJ05.A00001                      DATA
D2XHIDK5    P2XIDKA       P2XIDKAA       HALD   IMSTESTS.D2XHIDK5.A00001                      DATA
```

Figure 6. Example of an Extended Functions Recovery Point ID report

The RPID report contains the following sections:

Invocation Parameters

This section of the report shows the options that RPID used for its processing.

Time Format

This field indicates whether the timestamps that are displayed in the report are in Local or UTC (Coordinated Universal Time) format. Specify this parameter on the REPORT keyword by using the TIMEFMT() parameter.

Time Range

This field displays the formatted time range that you specified on the RPID RANGE() parameter.

Database List

This section of the report shows the databases that were processed in this RPID run.

DATABASE

This column specifies the name of the database that was processed.

AREA/PART

This column specifies the area name for fast path databases or the partition name for HALDB databases that were processed. This column is blank for full function databases.

DD

This column specifies the DD name that is associated with this area, partition, or full function database.

TYPE

This column specifies the type of database that was processed. Valid values are FF for full function databases, FP for fast path areas, and HALD for HALDB partitions.

DSN

This column specifies the data set name that is associated with this area, partition, or full function database.

PART-TYPE

For HALDB partitions, this column specifies the type of data set that was processed. Valid values are INDEX for the index portion of the HALDB, ILE for the index list element (or ILDS) portion of the HALDB, or DATA for the data portion of the HALDB.

OLR-PARTNER

For HALDB partitions that are IMS Online Reorganization (OLR) capable, this column specifies the OLR partner DD name that is associated with the partition.

Recovery Time Spans and Points for Time Stamp Recovery

The next section shows the following information:

- Recovery Time Spans that are common to all database entries
- Recovery Time Spans for Individual database entries
- Recovery Points (Utility End Times) for individual database entries

The following figure shows an example Recovery Time Spans and Points for Time Stamp Recovery:

```
********************************************************************************
-> Recovery Time Spans and Points for Time Stamp Recovery
********************************************************************************

--> Recovery Time Spans Common to All Entries in the DBLIST:

19.283 14:15:40.100000+0000 to 21.139 10:54:27.712196+0000
21.139 11:10:19.946777+0000 to 21.364 14:15:40.100000+0000


--> Recovery Time Spans for Individual Entries in the DBLIST:

DBD=POHIDKA   DDN=POHIDKAN DB_Type=HALDB-Part Partner_DD=POHIDKAB Master_DB=DBOHIDK5
19.283 14:15:40.100000+0000 to 21.139 10:59:31.214366+0000 Last_IMSID=n/a
21.139 11:00:17.329938+0000 to 21.139 11:00:38.972179+0000 Last_IMSID=IMS1
21.139 11:10:19.562406+0000 to 21.364 14:15:40.100000+0000 Last_IMSID=IMS1

DBD=POHIDKA   DDN=POHIDKAM DB_Type=HALDB-Part Partner_DD=POHIDKAA Master_DB=DBOHIDK5
19.283 14:15:40.100000+0000 to 21.139 10:59:31.192224+0000 Last_IMSID=n/a
21.139 11:00:17.329938+0000 to 21.139 11:00:38.951749+0000 Last_IMSID=IMS1
21.139 11:10:19.478393+0000 to 21.364 14:15:40.100000+0000 Last_IMSID=IMS1

DBD=POHIDKA   DDN=POHIDKAB DB_Type=HALDB-Part Partner_DD=POHIDKAN Master_DB=DBOHIDK5
19.283 14:15:40.100000+0000 to 21.139 10:59:31.214366+0000 Last_IMSID=n/a
21.139 11:00:17.329938+0000 to 21.139 11:00:38.972179+0000 Last_IMSID=IMS1
21.139 11:10:19.562406+0000 to 21.364 14:15:40.100000+0000 Last_IMSID=IMS1

DBD=POHIDKA   DDN=POHIDKAA DB_Type=HALDB-Part Partner_DD=POHIDKAM Master_DB=DBOHIDK5
19.283 14:15:40.100000+0000 to 21.139 10:59:31.192224+0000 Last_IMSID=n/a
21.139 11:00:17.329938+0000 to 21.139 11:00:38.951749+0000 Last_IMSID=IMS1
21.139 11:10:19.478393+0000 to 21.364 14:15:40.100000+0000 Last_IMSID=IMS1

DBD=POHIDKB   DDN=POHIDKBN DB_Type=HALDB-Part Partner_DD=POHIDKBB Master_DB=DBOHIDK5
19.283 14:15:40.100000+0000 to 21.139 10:59:23.973009+0000 Last_IMSID=n/a
21.139 11:00:17.329938+0000 to 21.139 11:00:42.276666+0000 Last_IMSID=IMS1
21.139 11:10:19.690115+0000 to 21.364 14:15:40.100000+0000 Last_IMSID=IMS1


--> Recovery Points (Utility End Times) for Individual Entries in the DBLIST:

DBD=POHIDKA   DDN=POHIDKAN DB_Type=HALDB-Part Partner_DD=POHIDKAB Master_DB=DBOHIDK5

DBD=POHIDKA   DDN=POHIDKAM DB_Type=HALDB-Part Partner_DD=POHIDKAA Master_DB=DBOHIDK5

DBD=POHIDKA   DDN=POHIDKAB DB_Type=HALDB-Part Partner_DD=POHIDKAN Master_DB=DBOHIDK5
21.139 10:58:40.700316+0000 --                             Batch-IC
21.139 11:11:19.946777+0000 --                             Batch-IC
21.139 11:11:32.500000+0000 -- 21.139 11:11:32.500000+0000 Full-Recovery
21.139 11:12:26.800000+0000 -- 21.139 11:12:26.800000+0000 Full-Recovery

DBD=POHIDKA   DDN=POHIDKAA DB_Type=HALDB-Part Partner_DD=POHIDKAM Master_DB=DBOHIDK5
21.139 10:58:39.740236+0000 -- 00.000 00:00:00.000000+0000 Batch-IC
21.139 11:11:19.946777+0000 --                             Batch-IC
21.139 11:11:32.500000+0000 -- 21.139 11:11:32.500000+0000 Full-Recovery
21.139 11:12:26.800000+0000 -- 21.139 11:12:26.800000+0000 Full-Recovery
21.130 11:12:26.812300+0000 -- 21.130 11:12:26.800000+0000 Concurrent-IC

DBD=POHIDKB   DDN=POHIDKBN DB_Type=HALDB-Part Partner_DD=POHIDKBB Master_DB=DBOHIDK5

DBD=POHIDKB   DDN=POHIDKBM DB_Type=HALDB-Part Partner_DD=POHIDKBA Master_DB=DBOHIDK5

DBD=POHIDKB   DDN=POHIDKBB DB_Type=HALDB-Part Partner_DD=POHIDKBN Master_DB=DBOHIDK5
21.139 10:58:42.471480+0000 -- 00.000 00:00:00.000000+0000 Batch-IC
21.139 11:11:32.500000+0000 -- 21.139 11:11:32.500000+0000 Full-Recovery         Alloc-Overlap
21.139 11:12:26.800000+0000 -- 21.139 11:12:26.800000+0000 Full-Recovery         Alloc-Overlap

DBD=POHIDKB   DDN=POHIDKBA DB_Type=HALDB-Part Partner_DD=POHIDKBM Master_DB=DBOHIDK5
21.139 10:58:41.545885+0000 -- 00.000 00:00:00.000000+0000 Batch-IC
21.139 11:11:32.500000+0000 -- 21.139 11:11:32.500000+0000 Full-Recovery
21.139 11:12:26.800000+0000 -- 21.139 11:12:26.800000+0000 Full-Recovery

DBD=POHIDKC   DDN=POHIDKCN DB_Type=HALDB-Part Partner_DD=POHIDKCB Master_DB=DBOHIDK5

DBD=POHIDKC   DDN=POHIDKCM DB_Type=HALDB-Part Partner_DD=POHIDKCA Master_DB=DBOHIDK5
```

Figure 7. Example Recovery Time Spans and Points for Time Stamp Recovery report

The Recovery Time Spans and Points for Time Stamp Recovery report contains the following sections:

Recovery Time Spans Common to All Entries in the DBLIST:

This section of the report lists all of the recovery time spans (RTSs) that are common for all of the databases in the DBLIST. The RTSs that are shown are based on allocation ranges bounded by the RANGE parameter input. If more than one common RTS is found, each one is listed. If no common RTS was found, no lines are printed under this heading. Each entry shows the beginning and ending timestamp that is associated with the common recovery time span.

When you review an RTS, any timestamp can be used on a Time Stamp Recovery job for recovering any and all of the databases on the DBLIST, including the beginning and the ending of the RTS. When you review this information, if it appears that the RTSs are more limited than what you expected, you can find the reasons for the limitation in the Utility End Times section of the report. For example, if any of the databases were reorganized and no image copy was taken after the reorganization, the end of the RTS will be the time immediately before the reorganization job started.

Recovery Time Spans for Individual Entries in the DBLIST:

This section of the report lists the RTSs for individual databases in the DBLIST, which are based on allocation ranges bounded by the RANGE parameter input. The last IMSID to allocate the database just before the beginning of the RTS is also listed.

If you recognize the Last_IMSID field as a DL/I batch job, the time to the left of the two dashes (--) is the log start time for the batch job, and the time to the right of the two dashes (--) is the log end time for the batch job.

If the database is a HALDB partition, the partner partition, and HALDB master database name are listed.

Any timestamp that is bounded by and includes the beginning and ending timestamps of the RTS is a valid recovery TSR time for the individual database. When RPID calculates the Common Recovery Time Spans, it assumes that if one of the HALDB partners is allocated the other HALDB partner is also allocated. And so, a TSR that is using a recovery point can recover both of the partner partitions.

Recovery Points (Utility End Times) for Individual Entries in the DBLIST:

This section of the report lists the subsequent recovery points (that is, the utility run and end times within the RANGE specification for individual entries in the DBLIST).

The following table lists the utility information that can be displayed:

Table 5. RPID DB recovery points or utility end times for DBLIST entries

What is displayed in the report	Meaning
Batch-IC	Batch image copy
Concurrent-IC	Concurrent image copy
User-IC	User image copy
SMS-IC-DB-Exclusive	SMS image copy – database exclusive
SMS-IC-DB-Shared	SMS image copy – database shared
Full-Recovery	Full recovery
Partial-Recovery	Partial recovery
Offlin-REORG+SubseqIC	Offline reorganization and subsequent image copy
Offlin-REORG-SubseqIC	Offline reorganization and no subsequent image copy
Online-REORG+IC	Online reorganization and image copy
Online-REORG-IC	Online reorganization and no subsequent image copy
Full-Rcvr+SubseqIC	Full recovery and subsequent image copy
Full-Rcvr-SubseqIC	Full recovery and no subsequent image copy

Table 5. RPID DB recovery points or utility end times for DBLIST entries (continued)

What is displayed in the report	Meaning
Partial-Rcvr+SubseqIC	Partial recovery and subsequent image copy
Partial-Rcvr-SubseqIC	Partial recovery and no subsequent image copy
Complete-Change-Accum	Complete change accumulation

RPID ISPF interface for viewing RPID reports and generating JCL statements

The RPID ISPF interface plays an important role for viewing RPID reports and generating JCL for a recovery. The RPID ISPF interface allows you to easily select database components and recovery times. After specifying database components and recovery times, you can use the RPID ISPF interface to generate IMS Database Recovery Facility control statements. For viewing RPID reports and generating control statements, you must have an understanding of the IMS Recover Point Identification ISPF interface.

Chapter 9. Verify Recovery Assets (VERIFY) function

The VERIFY function ensures that the necessary data sets exist and that they are usable before performing a recovery with IMS Database Recovery Facility Extended Function.

Topics:

VERIFY overview

The Verify Recovery Assets function (VERIFY) examines the RECON data sets and the IMS Recovery Expert repository to identify the recovery assets that are required to recover one or more databases.

Specifically, the VERIFY function performs one or all of the following functions:

- Lists or prints the recovery assets that are required to recover the specified databases
- Allocates all recovery assets to ensure that they exist
- Opens all recovery assets to ensure that they are valid

The VERIFY function requires that you specify the target databases by using the DBLIST supporting function.

The VERIFY function can use the RECON copy data sets that are created by the RECONCOPY supporting function, or the data sets identified on the ALTRECONS supporting function.

Note: Use either RECONCOPY or ALTRECONS in conjunction with VERIFY to prevent input/output (I/O) to the live RECON data sets.

Processing System Level Backup Image Copies

The RECON can contain Image Copy records that represent System Level Backups (SLB). The following considerations pertain to using VERIFY to process SLBs.

- The data set name for the SLB Image Copy is symbolic only and is formatted as follows:

  ```
  SLB.Ixxxxx.Dxxxxx.dbdname.ddname
  ```

  These data sets are never cataloged, nor does any function try to allocate or open them.
- When USESLBIC(N) is specified, RECON Image Copy records that represent SLBs are treated by VERIFY no differently than any other Image Copy records with one exception: VERIFY does not verify as to whether they are cataloged.

  You can use the VERIFY function with the LIST option to determine if there are SLB Image Copy records in the RECON.

  IMS Database Recovery Facility VERIFY, IMS Database Recovery Facility Extended Functions VERIFY, and IMS Database Recovery Facility Extended Functions HCHECK functions all recognize these data set names as symbolic only.
- When USESLBIC(Y) is specified, and whenever necessary, IMS Database Recovery Facility Recovery, IMS Database Recovery Facility VERIFY, and IMS Database Recovery Facility Extended Functions VERIFY functions issue a **NOTIFY.IC** command to DBRC for the most recent SLB that exists relative to the function's specified time parameter (for example, RCVTIME or "current").
- It is a best practice to delete any Image Copy records that represent deleted SLBs.

Otherwise, the VERIFY function would consider these Image Copy records as representing a usable backup. This correlates with the fact that IMS Database Recovery Facility Recovery and IMS Database Recovery Facility VERIFY would also consider these records as valid and attempt to use the deleted SLBs.

- Another possible situation occurs when an SLB exists, yet there is no RECON IC record that represents it.

  The VERIFY function with any of its options (LIST, ALLOC, OPEN) cause representative Image Copy records to be written to the RECON.
- If available, IMS Database Recovery Facility Recovery uses the fast-replication volume(s) for recovery. If fast-replication volume(s) are not available, IMS Database Recovery Facility Recovery uses the offload data sets.

  If there are offload data set(s), they are displayed on the report. This does not indicate that they will be used for recovery. It just indicates that they are available if needed.

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 Solution Pack for z/OS: IMS Recovery Expert User's Guide*.

VERIFY input

Run the Verify Recovery Assets (VERIFY) function by running VERIFY.

VERIFY function syntax

```
    VERIFY( TYPE(LIST | ALLOC | OPEN)
            TIME(timestamp)
            RCVTYPE(TSR | PITR)
            SOURCE(PRI | SEC)
          )
```

Parameter reference

The VERIFY parameters are described in the following list:

TYPE(LIST |ALLOC | OPEN)

Use the TYPE parameter to specify the verification level that VERIFY will perform.

TYPE(LIST) provides a list of all log, image copy, and change accumulation data sets that are required for recovery. The list contains information that is extracted from DBRC about each data set.

TYPE(ALLOC) provides the same detail as TYPE(LIST), but TYPE(ALLOC) also allocates each data set with a deferred mount for tape data sets to verify the existence and physical accessibility of each data set.

TYPE(OPEN) provides the same detail as TYPE(LIST) and TYPE(ALLOC) but TYPE(OPEN) also opens each data set to verify its availability and usability.

The default is TYPE(LIST).

TIME(*timestamp*)

The TIME parameter provides a way to obtain information about recovery assets for a timestamp recovery, including recoveries to any prior point in time. TIME specifies the timestamp value that determines how much log and change accumulation data would be processed during recovery. Only log data sets that contain data from before the recovery timestamp will be reported. Only change accumulation data sets from before an allocation boundary, as defined by DBRC, will be reported. TIME() is an optional parameter. If TIME() is not specified, the VERIFY function will process recovery assets for performing a full recovery.

You must supply the timestamp value within an interval where the databases and fast path areas in the input list are unallocated if you specified RCVTYPE(TSR). The timestamp value can be any earlier point-in-time if RCVTYPE(PITR) is specified.

The format of the timestamp specification can be found in "TIME and RANGE parameter processing" on page 24.

RCVTYPE(TSR | PITR)

Use the RCVTYPE parameter to specify the type of recovery that is being requested.

RCVTYPE is an optional parameter.

TSR is the default parameter if TIME() is specified. TSR specifies that recovery assets verification for a timestamp recovery is to be performed. The timestamp must be such that no ALLOC records in the RECON span the timestamp that is supplied with TIME(). All RECON ALLOC records that have an ALLOC time that earlier than the recovery time must have one of the following specifications:

DEALLOC time:

The DEALLOC time must be earlier than the recovery time.

STOP time:

The associated PRILOG record must have a STOP time that is earlier than the recovery time. The associated PRILOG record is the one with a START time equal to the START time of the PRILOG START time specified in the ALLOC record.

Use the PITR parameter to specify that recovery assets verification for a timestamp recovery to any prior point in time (Point-In-Time Recovery) is to be performed. At the timestamp that is specified, database data sets and areas can be in any allocation state as recorded in the RECON data sets. Log data sets that span allocation intervals for the database data sets and FastPath areas in the input list, up to and including the specified recovery time, are reported. If PITR is specified, the timestamp that is supplied can be any value before the current time, but after the run time of a valid image copy that is registered with DBRC.

SOURCE(PRI | SEC)

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

If SOURCE(PRI) is specified, primary image copy and log data sets are used as the source for the verify process except for any that are marked as invalid in the RECON data sets. The status, as recorded in the RECON data sets, of the primary image copy and log data sets is reported.

If SOURCE(SEC) is specified, secondary image copy and log data sets are verified as the source for the recovery process. The status, as recorded in the RECON data sets, of the secondary image copy and log data sets is reported. The default is SOURCE(PRI).

VERIFY usage example

Use the Verify Recovery Assets function (VERIFY) to verify the availability of recovery assets.

You can set up VERIFY to run on a scheduled basis to periodically check for problems that might affect database recovery. VERIFY can also be used at the disaster recovery (DR) site to ensure that your DR site is recoverable. All recovery assets that are recorded in DBRC must be usable.

If you are attempting to recover a set of databases to a specific time, and you want to identify any problems related to image copies, change accumulation data sets, and log data sets that will be used as input to that recovery, specify the following control cards to report on all of the assets for all of the databases in the DBLIST.

Example

The following example uses primary log records and primary image copy data sets for verification. It allocates or opens all of the assets to verify that the data sets are usable. It reports on only those assets that are recorded before the timestamp.

```
DBLIST( DB (DB1  DB2  DB3) )
VERIFY( TYPE(OPEN)
        TIME(timestamp)
        RCVTYPE(TSR)
        SOURCE(PRI)
      )
```

VERIFY output

The output of the VERIFY function is written to the RPTVERFY DD.

If you do not specify the RPTVERFY DD in the JCL, it is dynamically allocated to SYSOUT=*. Report output can be written to SYSOUT or to a data set. The characteristics of the output are RECFM=FBA,LRECL=133. Report output can also be written to the IMS Tools Knowledge Base by specifying the REPORT(ITKBOUT(YES)) keyword and parameter.

The default is REPORT(ITKBOUT(YES)).

The Verify Asset report consists of three main report sections. The following figure shows the sections of the Verify Asset report.

```
IMS DATABASE FACILITY EXTENDED FUNCTIONS V2.1.0 (5655-V86)
IMS DATABASE FACILITY EXTENDED FUNCTIONS VERIFY ASSETS REPORT        Run Time:  11/19/2021 14:35



*******************************************************************************
 -> Invocation Parameters
 *******************************************************************************

 PROCESS : VERIFY(OPEN)         TYPE    : TSR
 SOURCE  : PRI                  RCVTIME : 2021.323 14:01:01.100000 -08:00



 *******************************************************************************
 -> Summary Report
 *******************************************************************************


Database Name  DD/Area Name   DSID  Data Set Type    Status
-------------------------------------------------------------------------------
F2O2P1         F2O2P11         0    Full Function    No errors encountered
F2O2P1         F2O2P12         0    Full Function    No errors encountered



*******************************************************************************
-> Data Set I/O Report
*******************************************************************************

 Recover to point: 2021.323 14:01:01.100000 -08:00

                                            Volume  IC        Status of
Image Copy Data Set Name                    Serial  Type      Allocate/Open
-------------------------------------------------------------------------------
IMSTOOL.IMSRE.QA.Q91A.F2O2P1.F2O2P11.IC1    IMT203  STD       OPEN RC  =   0
IMSTOOL.IMSRE.QA.Q91A.F2O2P1.F2O2P12.IC1    IMT187  STD       OPEN RC  =   0

                                            Volume            Status of
Change Accum Data Set Name                  Serial            Allocate/Open
-------------------------------------------------------------------------------

TEMP.IMSRE.QA.CAF2O2P1.D21323.T125904       IMSTL4            OPEN RC  =   0


                                            Volume  IMS       Status of
Log Data Set Name                           Serial  SYSID     Allocate/Open
-------------------------------------------------------------------------------

TEMP.V00FF02E.IEFRDER4                      IMT178  V00FF02E  OPEN RC =    0
  Prilog: 2021.323 13:01:25.4
```

Figure 8. Example of an IMS Database Recovery Facility Extended Functions Verify Asset report

The major report sections are described here:

Invocation Parameters

This section of the report shows the options that VERIFY used for its processing.

PROCESS

This field indicates what kind of verification is done on the assets (LIST, ALLOC, or OPEN).

SOURCE

This field displays whether to verify primary or secondary image copy records. In the last report sample, the primary image copies are verified.

RCVTIME

This field displays the recovery time during which you want to verify assets for this recovery.

TYPE

This field is the type of recovery, either TSR or PITR, that is requested.

Summary Report

This section of the report displays the databases that were selected for verification and the overall status of each database and DD name. The following list describes the column fields that are displayed:

Database Name
This column specifies the database name.

DD/Area Name
This column specifies the area name for fast path databases or the partition name for HALDB databases. This column is blank for full function databases.

DSID
This column specifies the data set identification number that is associated with this DD or AREA name.

Data Set Type
This column specifies the type of database: full-function, Fast Path, or HALDB.

Status
This column specifies the overall status of all of the assets that were verified.

Data Set I/O Report
This section of the report displays the individual status of each asset that has been processed by VERIFY. It begins by showing the Recover-To Point, which is the value that you specified on the VERIFY keyword. This report consists of three sections. Each section shows a different type of recovery asset:

- Image copy data sets
- Change accumulation data sets
- Log data sets

Image Copy Data Sets

Image Copy Data Set Name
This column specifies the name of the image copy data set for each database data set that was verified.

Volume Serial
This column specifies the volume serial number of the image copy data set for each database data set that was verified.

IC Type
This column specifies the type of image copy for each database data set that was verified. Valid values for this field are shown in the following table:

Table 6. Valid values for IC Type in the Verify Asset report

IC Type listed	Meaning
STD	Standard Image Copy
IC2	Image Copy 2
IC EXT	Extended Image Copy
HISAM UNL	HISAM Unload
SLB-CIC-O *	The System Level Backup is concurrent and is offloaded to a sequential device. For the ALLOC or OPEN process, 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.
SLB-BAT-O *	The System Level Backup is batch and is offloaded to a sequential device. For the ALLOC or OPEN process, 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.

Table 6. Valid values for IC Type in the Verify Asset report (continued)

IC Type listed	Meaning
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).

Note: * If IMS Recovery Expert is available and USESLBIC(YES) is specified, the VERIFY function 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.I`*`xxxx`*`.D`*`xxxx`*`.dbdname.ddname`, and this image copy record is used during VERIFY function processing.

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(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.

Status of Allocate/Open
This column displays the return code from the ALLOC or OPEN process. If the process was successful, the return code is zero.

Change Accumulation Data Sets

Change Accum Data Set Name
This column specifies the name of the change accumulation data set for each database data set that was verified.

Volume Serial
This column specifies the volume serial number of the change accumulation data set for each database data set that was verified.

Status of Allocate/Open
This column shows the return code from the ALLOC or OPEN process. If the process was successful, the return code will be zero.

Log Data Sets

Log Data Set Name
This column specifies the name of the log data set for each database data set that was verified.

Volume Serial
This column specifies the volume serial number of the log data set for each database data set that was verified.

IMS SYSID
This column shows the SYSID that is associated with the IMS that created the log.

Status of Allocate/Open
This column shows the return code from the ALLOC or OPEN process. If the process was successful, the return code is zero.

Chapter 10. RECON cleanup (RCU) function

The RECON cleanup function (RCU) prepares the RECON data sets for recovery at a disaster recovery (DR) site in IMS Database Recovery Facility Extended Function.

Topics:

RCU overview

A copy of the RECON must be made and modified to match the data sets that exist at the recovery site to perform valid recoveries. RCU will create a clean copy of the RECON data sets at a particular point in time for recovery.

In general, a recovery site contains all the data that is needed to restore IMS databases to a known active site time. The following data is needed for recovery:

- Image copy data sets taken at the active site
- IMS log data from the active site (logs, change accumulation data sets, or both)
- A RECON that accurately reflects the activities that have taken place at the active site

Image copies

Database recovery works at the database data set (DBDS) level; each DBDS needs to be restored individually and starts with an image copy. Restoring an image copy creates an exact copy of the DBDS to a known time. Image copies can be concurrent (also called fuzzy) or non-concurrent. Non-concurrent image copies are usually called batch image copies. When a concurrent image copy is used, log data from before the time that the image copy was taken is usually needed. Batch image copies require only the log data that was created after the image copy was taken. Most image copy utilities take a primary and secondary copy (two copies in one pass). Often it is the secondary copy that is shipped to a recovery site. Information about image copy data sets is normally retained by DBRC.

After a DBDS has been restored by using an image copy, you must apply all of the changes that have been made since the image copy was taken. IMS log data sets are read to determine what changes have been made.

IMS log data

IMS keeps a log of all of the changes that are made to databases. This log is written to data sets that are called online log data sets (OLDS), system log data sets (SLDS), and recovery log data sets (RLDS). A SLDS contains all of the log data that was written by IMS. An RLDS is a subset of the SLDS and contains only the log data that is needed for database recovery. Information about log data sets is maintained by DBRC. Like image copies, there can be primary and secondary log data sets. Often the secondary copy is shipped to a recovery site. No IBM recovery utility uses an OLDS as input.

A change accumulation data set also contains log data that is usable by a recovery utility. The Change accumulation (CA) utility reads log data sets and reformats the data to a form that is more efficient for recovery utilities. The CA utility does not create a secondary copy. Information about CA data sets is kept by DBRC.

RECON data sets

RECON data sets must exist at the recovery site. Often, a copy of the RECON data sets that are taken at the active site are shipped to the recovery site. One method of making a copy of the RECON data sets is by using the DBRC BACKUP.RECON command.

Preparing for site recovery

Before a role change takes place, the databases must be restored to a predetermined time that is called the cleanup time. Information in the RECON most likely needs to be changed to conform to this time. RECON records are either deleted or changed to remove information that is later than the selected cleanup time.

RCU leaves the RECON in a real-time situation, as it would be in at cleanup time.

The typical tasks that are performed to prepare the RECON data sets are:

- Closing open PRILOG records
- Deleting PRIOLD records
- Deleting SECOLD records
- Closing open PRISLD records
- Closing open SECSLD records
- Deleting SUBSYS records
- Updating or deleting ALLOC records
- Updating or deleting LOGALL records
- Deleting Image copy records that are past the RECON cleanup time
- Deleting CA records that are past the RECON cleanup time
- Marking CA runs invalid if log data spans past the RECON cleanup time
- Marking DBs as Recov Needed if they are affected by RCU
- Incrementing Recov Needed Counts if they marked for recovery
- Adjusting ICUSED (image copy was used) when one or more image copies are deleted
- Adjusting CAUSED (change accumulation was used) when one or more change accumulations are deleted
- Deleting Batch Backout Needed records
- Optionally marking primary or secondary log records and IC records in ERROR
- Optionally deleting disabled partitions
- Providing detailed reports with applicable warning messages in case manual intervention is required

RCU does not verify the physical existence of any of the data sets that are identified in the RECON.

Clean up time

Before preparing the RECON for site recovery, a clean up time must be determined.

The cleanup time is input to RCU, and a copy of the active RECON data sets are modified to make them conform to this time. All indications of IMS activity equal to or later than the cleanup time are removed from the RECON. The RECON is not modified to appear that all activity ended normally at the cleanup time. Rather, it will appear as if time simply stopped.

Attention: RCU will cause data to be lost after the clean up time. It is not intended to be used to create a mirror-image of the active site. RCU is intended to create a clean copy of the RECON data sets at a particular point in time.

Updated RECON records

The RCU function makes updates to several of the RECON records.

The RCU function takes various actions for all of the records in the RECON.

Database (DB) records:

DB records (including AREA AAUTH) are never deleted. Any authorizations for the database are removed.

Actions on other records might cause the DB record to be updated. All DB records that are changed in any way are reported.

Partition records:

All partitions that are marked as Disabled are deleted from the RECON by a DBRC DELETE.PART command if the DELDISABLEPART(Y) keyword is specified.

If specified, The DBRC DELETE.PART command deletes all of the records that are associated with a partition. The partition that is deleted is reported, but not all of the deleted associated records.

Database data set (DBDS) records:

DBDS records (including AREA RECOV) are never deleted. All EEQEs are removed, and the record is marked as needing recovery.

The Recovery needed count in the DB record is incremented. Actions on other records might cause the DBDS record to be updated. All DBDS records that are changed are reported.

Allocation (ALLOC) records:

All ALLOC records with an allocation time equal to or greater than the cleanup time are deleted. For each ALLOC record that is deleted, the associated LOGALL record is updated.

ALLOC records that are open before the cleanup time (ALLOC records that do not have a deallocation time) remain open so that no deallocation time is added. When an ALLOC record is deleted, the DBDS or AREA RECOV record is marked as needing recovery (if not already marked as such), and the Recov Needed count in the DB record is incremented.

Image copy (IC1 and IC2) records:

Image copy records with a runtime that is equal to or greater than the cleanup time are deleted. Deleting an IC record also causes image copy counts (ICUSED) that are maintained in the DBDS record to be updated, but ICAVAIL is not adjusted.

RCU reports DBDSs that have no image copy.

REORG records:

REORG records with a runtime that is equal to or greater than the cleanup time are deleted. Deleting a REORG record does not affect the IC Needed flag.

The IC Needed flag might be on or off. If a REORG record is not deleted and there are no later image copies, this condition is reported.

Note: Special recovery consideration should be given to the associated records for Primary or Secondary Indexes, ILEs, and External logicals since they might not have been changed by the deletion of the Primary database's REORG record.

RECOV records:

RECOV records with a runtime that is equal to or greater than the cleanup time are deleted.

If the recovery is a timestamp recovery (TSR), the DBDS is marked as needing recovery and the Recov needed count in the DB record is incremented. Both the DBDS or AREA record and the DB record are reported as having been changed.

If the recovery is a TSR and the recover-to time is earlier than the cleanup time, this condition is reported.

Tip: Special recovery consideration should be given to the associated records for Primary or Secondary Indexes, ILEs, and External logicals because they might not have been changed by the deletion of the Primary database's RECOV record.

Change accumulation (CA) records:

CA execution records that include logs that extend past the cleanup time are deleted from the RECON by a DBRC DELETE.CA command. CA STOP TIME is used; not the RUN TIME.

The CA count in the CAGRP record that is used is adjusted accordingly. If the CAGROUP is REUSE, CA execution records that are deleted are not made available.

CA execution records that are deleted are reported. CA Group records that might be changed by the deletion of execution records are not reported.

Subsystem (SSYS) records:

All subsystem records are deleted from the RECON and reported.

Log records (PRILOG, SECLOG, PRISLDS, and SECSLDS):

All log records with a start time that is equal to or greater than the cleanup time are deleted along with their associated LOGALL record. Log data set entries in these log records with a start time equal to or greater than the cleanup time are deleted from the record.

If a log data set entry is deleted, the stop time of the log is set to zero. The log data set count is adjusted accordingly.

A log record with no data set entries can exist when the OLDS has not been archived. Both log records are deleted and their associated LOGALL records that are deleted are reported.

Log allocation (LOGALL) records:

LOGALL records are affected only by changes to other records.

Primary and secondary online log data sets (PRIOLDS and SECOLDS):

All PRIOLDS and SECOLDS records are deleted from the RECON and reported.

Backout records:

All BACKOUT records are deleted from the RECON and reported.

Special considerations for using RCU

Some conditions preclude running RCU and must be corrected before using the function. Other conditions require special consideration after RCU completes. In any case, RCU will either deliver an error message or report this information.

Interim log records:

Interim log records are temporary, they are created during the use of the IMS Log Recovery utility, and they remain in the RECON only while the utility is running.

Therefore, interim log records with a start time earlier than the Clean Up time must be removed either by running the Log Recovery utility or by using the DBRC DELETE.LOG command.

RCU terminates with an error message if any of these interim log records exist.

OLR:

The IMS Online Reorganization utility (OLR) might have been in a suspended or active state. If OLR is running at the same time as the cleanup time, a report will be generated for those databases that were being reorganized.

Manual intervention is required to determine the state of the active or inactive data sets before any recovery of these databases occurs.

RCU will report this information.

RECOV records that do not have a subsequent IC record:

RCU will detect if a RECOV record exists without a subsequent image copy.

RCU will report this information.

REORG records that do not have a subsequent IC record:

RCU will detect if a REORG record exists without a subsequent image copy.

RCU will report this information.

Databases that do not have any image copy records:

RCU will detect if a DBDS record does not have at least one associated IC record. Image copies are required for recovery.

RCU will report this information.

Primary and secondary image copy records:

If you specify that RCU is to mark the primary image copies in error and no corresponding secondary image copy record exists.

RCU will report this information.

Primary and secondary log records:

If you specify that RCU is to mark the primary log records in error and no corresponding secondary log record exists.

RCU will report this information.

RCU input

Run the RCU function by issuing the RECONCOPY supporting function keyword and the RCU function keyword.

RCU function syntax

```
RCU ( TIME(timestamp)
      DELDISABLEDPART(YES | Y | NO | N)
      SOURCE(PRI | SEC | PRIIC | SECIC)
    )
```

Parameter reference

TIME(***timestamp*** **)**

Use the TIME parameter to specify the cleanup time. The TIME parameter is required.

The timestamp variable has no default value.

The format of the timestamp specification can be found in "Timestamp format reference" on page 26.

DELDISABLEPART(Y | YES | N | NO)

Use the optional DELDISABLEPART parameter to specify whether to delete partitions that have been marked as disabled.

The default value is YES.

SOURCE(PRI | SEC | SECIC | SECLOG)

Use the optional SOURCE parameter to specify whether to use only primary or secondary log data sets and image copy data sets for subsequent recovery.

Important: If SOURCE is not specified, RCU uses both primary and secondary logs and image copies.

PRI

Use this option to indicate that only primary logs and primary image copies are used for recovery.

If secondary logs and secondary image copy data sets exist, RCU marks them in error.

SEC

Use this option to indicate that only secondary logs and secondary image copies are to be used for recovery.

In this case, RCU marks the primary logs and primary image copies in error.

If secondary logs and secondary image copies do not exit, this condition is indicated in the Extended Functions RECON Cleanup Report.

SECIC

Use this option to indicate that only secondary image copies are to be used for recovery.

In this case, RCU marks the primary image copies in error whether or not a secondary image copy exists.

If secondary image copies do not exist, this condition is indicated in the Extended Functions RECON Cleanup Report.

Both primary and secondary logs are used.

SECLOG

Use this option to indicate that only secondary logs are to be used for recovery.

In this case, RCU marks the primary logs in error whether or not a secondary log data set exists.

If secondary logs do not exist, this condition is indicated in the Extended Functions RECON Cleanup Report.

Note: If either SEC, SECIC, or SECLOG is specified and secondary data sets do not exist, the RCU function ends with return code = 8 and reason code = C3200001.

RCU usage example

The RCU function can be used to clean up a copy of the active RECON data sets to prepare them for a disaster recovery site.

In the following example, the active RECONs are copied to DRSITE.IMS1.C1 / C2 / C3 data set names, and if the RECON copy data sets already exist, they are reused. Otherwise, the RECON copy data sets are allocated on a non-SMS volume with 300 primary and 10 secondary cylinders of space.

```
RECONCOPY (DSNPREF( DRSITE.IMS1 )
           REUSE( YES )
           SPACE(CYL(300 10))
           DATACLASS(NOSMS)
          )

RCU ( TIME (‘21.291 06:30:01.1') )
```

In the preceding example, RCU cleans up the copied RECON data sets to the specified time.

RCU output

The output of the RCU function is written to the RPTRCU DD.

If you did not specify the RPTRCU DD in the JCL, it is dynamically allocated to SYSOUT=*.
Report output can be written to SYSOUT or to a data set. The characteristics of the output are RECFM=FBA,LRECL=133. Report output can also be written to the IMS Tools Knowledge Base by specifying the REPORT(ITKBOUT(YES)) keyword and parameter.

The default is REPORT(ITKBOUT(YES)).

The RCU report consists of two main report sections. The following figure shows an example of the RCU report.

```
IMS DATABASE FACILITY EXTENDED FUNCTIONS V2.1.0 (5655-V86)
********************************************************************************
-> Invocation Parameters:
********************************************************************************
     TIME:           2021.280 12:27:23.100000 -07:00
     SOURCE:         NONE
     DELDISABLEPART: YES
***********************************************************************
-> Log Record Cleanup Report - Log Records Deleted
***********************************************************************
SSID Name  Log Start Time
---------  -------------------------------
IMZ2       2021.287 20:11:14.500000 -07:00
IMZ1       2021.287 20:11:17.100000 -07:00
JCLBTCH1   2021.287 20:26:47.700000 -07:00
JCLBTCH2   2021.287 20:26:53.200000 -07:00

Record Type: Log                  # Records Deleted:                4

Record Type: Log                  # Records Modified:               0
***********************************************************************
-> Online Log Data Set Cleanup Report - OLDS Records Deleted
***********************************************************************
SSID Name  Primary/Secondary
---------  -----------------
IMZ1       Primary
IMZ2       Primary

Record Type: OLDS                 # Records Deleted:                2
***********************************************************************
-> Subsystem Cleanup Report - SSYS Records Deleted
***********************************************************************
Subsys Name  Subsys Type  Subsys Status
-----------  -----------  -------------
IMZ1         Online       Normal
IMZ2         Online       Normal

Record Type: Subsystem            # Records Deleted:                2
***********************************************************************
-> Backout Record Cleanup Report - Backout Records Deleted
***********************************************************************
Record Type: Backout              # Records Deleted:                0
***********************************************************************
-> Change Accum Execution Cleanup Report - CAs Deleted
***********************************************************************
Record Type: CA Execution         # Records Deleted:                0
***********************************************************************
-> Database/DEDB Cleanup Report - Database(s) Modified
***********************************************************************
DBD/DEDB Name
-------------
CUSTD1
PCUST21
PCUST22
PCUST42

Record Type: DB                   # Records Modified:               4
***********************************************************************
-> Database Data Set Cleanup Report - DBDSs Modified
***********************************************************************
DBD/DEDB Name    DDN/Area Name
-------------    -------------
CUSTD1           CUSTD1
PCUST21          PCUST21A
PCUST21          PCUST21M
PCUST22          PCUST22A
PCUST42          PCUST42A
PCUST42          PCUST42M

Record Type: Database Data Sets   # Records Modified:               6
```

Figure 9. Example of an Extended Functions RECON cleanup report (part 1 of 2)

```
***********************************************************************
-> Alloc Record Cleanup Report - ALLOC Records Deleted
***********************************************************************
DBD/DEDB Name  DDN/Area Name  Allocation Time
-------------  -------------  -------------------------------
CUSTD1         CUSTD1         2021.287 20:20:09.432306 -07:00
CUSTD1         CUSTD1         2021.287 20:20:44.215129 -07:00
PCUST21        PCUST21A       2021.287 20:27:01.948396 -07:00
PCUST21        PCUST21A       2021.287 20:30:06.767713 -07:00
PCUST21        PCUST21A       2021.287 22:03:36.166819 -07:00
PCUST21        PCUST21A       2021.287 22:07:07.468367 -07:00
PCUST21        PCUST21M       2021.287 23:07:15.983751 -07:00
PCUST22        PCUST22A       2021.287 20:28:08.134886 -07:00
PCUST22        PCUST22A       2021.287 20:31:12.975831 -07:00
PCUST22        PCUST22A       2021.287 22:04:42.605166 -07:00
PCUST22        PCUST22A       2021.287 22:08:13.709894 -07:00
PCUST22        PCUST22A       2021.287 22:14:55.429617 -07:00
PCUST22        PCUST22A       2021.287 22:19:00.210476 -07:00
PCUST42        PCUST42A       2021.287 22:43:42.104709 -07:00
PCUST42        PCUST42M       2021.287 22:43:26.016135 -07:00

Record Type: Alloc                # Records Deleted:               25
***********************************************************************
-> Image Copy Cleanup Report - IC Records Deleted
***********************************************************************
Record Type: Image Copy           # Records Deleted:                0
***********************************************************************
-> DB Recovery Cleanup Report – RECOV Records Deleted
***********************************************************************
DBD/DEDB Name  DDN/Area Name  Recovery Run Time
-------------  -------------  -------------------------------
CUSTD1         CUSTD1         2021.287 23:16:54.200000 -07:00

Record Type: Recovery             # Records Deleted:                1
***********************************************************************
-> Reorganization Cleanup Report - REORG Records Deleted
***********************************************************************
DBD/DEDB Name  DDN/Area Name  Reorg Run Time
-------------  -------------  -------------------------------
PCUST21        PCUST21A       2021.287 22:50:54.843375 -07:00
PCUST21        PCUST21A       2021.287 23:37:27.514491 -07:00
PCUST21        PCUST21M       2021.287 22:50:54.843375 -07:00

Record Type: Reorg                # Records Deleted:                3
**********************************************************************
-> Disabled Partition Cleanup Report - Partition Records Deleted
**********************************************************************
Record Type: Disabled Part        # Records Deleted:                0
***********************************************************************
-> No Image Copy Cleanup Report - DBDSs with No Image Copy
***********************************************************************
DBD/DEDB Name    DDN/Area Name
-------------    -------------
PCUST21          PCUST21M
PCUST42          PCUST42M
Record Type: No image copy        # Records Listed:                 2
***********************************************************************
-> No IC After REORG Cleanup Report - DBDSs With NO IC
DBD/DEDB Name  DDN/Area Name  REORG Run Time
-------------  -------------  -------------------------------
PCUST22        PCUST22A       2021.269 00:58:38.900000 -07:00
PCUST42        PCUST42A       2021.269 00:58:39.000000 -07:00

Record Type: No IC after reorg    # Records Listed:                 2
```

Figure 10. Example of an Extended Functions RECON cleanup report (part 2 of 2)

The RCU report contains the following sections:

Invocation Parameters

This section of the report shows the options that RCU used for its processing.

TIME

This field indicates the cleanup time.

SOURCE
This field displays whether to mark primary or secondary log or image copy records. In the previous example report, the SOURCE keyword was not specified, so NONE was displayed (which means that neither primaries or secondaries are marked in error).

DELDISABLEPART
This field displays whether any disabled partitions are deleted. In the previous example report, all disabled partitions would have been deleted.

The next section of the report describes the following groups of information:

Log Record Cleanup Report - Log Records Deleted
This section of the report shows information about the specific log records that were deleted or modified. The details shown are the SSID name and the Log Start Time of each log record. In addition, a total number of log records that were deleted and log records that were modified are reported.

Online Log Data Set Cleanup Report - OLDS Records Deleted
This section of the report shows information about the specific log records that were deleted or modified. The details that are listed are the SSID name and the Log Start Time of each log record. In addition, a total number of log records that were deleted and log records that were modified are reported.

Subsystem Cleanup Report - SSYS Records Deleted
This section of the report shows information about the specific subsystem (SSYS) records that were deleted. The details that are listed are the Subsystem name, subsystem type (online or batch), and the subsystem status. In addition, a total number of subsystem records deleted are reported.

Backout Record Cleanup Report - Backout Records Deleted
This section of the report shows the total number of Backout records that were deleted.

Change Accum Execution Cleanup Report - CAs Deleted
This section of the report shows the total number of Change Accumulation (CA) records that were deleted.

Database/DEDB Cleanup Report - Database(s) Modified
This section of the report shows the DBD or DEDB name of each DB record that was modified. In addition, a total number of database records that were modified is reported.

Database Data Set Cleanup Report - DBDSs Modified
This section of the report shows the DBD or DEDB name and the DDN or area name of each DBDS record that was modified. In addition, a total number of database data set records that were modified is reported.

Alloc Record Cleanup Report - ALLOC Records Deleted
This section of the report shows the DBD or DEDB name, the DDN or area name, and the allocation time of all allocation (ALLOC) records that were deleted. In addition, a total number of allocation records that were deleted is reported.

Image Copy Cleanup Report - IC Records Deleted
This section of the report shows the total number of image copy (IC) records that were deleted.

DB Recovery Cleanup Report - RECOV Records Deleted
This section of the report shows the DBD or DEDB name, the DDN or area name, and the recovery time of all RECOV records that were deleted. In addition, a total number of RECOV records that were deleted is reported.

Reorganization Cleanup Report - REORG Records Deleted
This section of the report shows the DBD or DEDB name, the DDN or area name, and the reorganization time of all of the REORG records that were deleted. In addition, a total number of REORG records that were deleted is reported.

Disabled Partition Cleanup Report - Partition Records Deleted
This section of the report shows the total number of HALDB disabled partitions that were deleted.

No Image Copy Cleanup Report - DBDSs with No Image Copy

This section of the report shows the DBD/DEDB name and the DDN/Area Name of all databases that have no image copies recorded in the RECONs. The database is not recoverable without at least one image copy. A total number of DBDSs that do not have IC records is reported.

No IC After REORG Cleanup Report - DBDSs With No IC

This section of the report shows the DBD or DEDB name, the DDN or area name, and the REORG run time of all databases that have no image copies recorded after a REORG record. The database might or might not have had the ICNEEDED flag on, depending on whether RCU deleted the image copy record or not. A database must have an image copy taken after a REORG. A total number of DBDSs that do not have IC records after a REORG is reported.

RCU return and reason codes

The RCU function issues these return and reason codes for the IRO0102E message.

Module	Return code	Reason code	Description
IROERC10	4	C3100001	The warning is contained in the report. DBDS or area without an image copy. Time stamp recovery spans cleanup time.
IROERC70	8	C3700001	Non zero return code from a call to DBRC DELETE.BKO command.
IROERC10	8	C3100002	Interim log records exist and must be removed.
IROERC40	12	C3400001	DBRC had an error deleting an image copy record. A DSP*nnnn*I message is issued.
IROERC80	12	C3800001	DBRC had an error deleting a partition record. A DSP*nnnn*I message is issued.
IROERC10	16	C3100001	RECON open failed.
		C3100002	RECON I/O error on sequential locate of interim logs and disabled partitions.
		C3100003	RECON I/O error on reading BACKOUT records.
		C3100004	RECON I/O error on reading SUBSYS records.
		C3100005	RECON I/O error on reading a log record while processing an ALLOC record.
		C3100006	RECON I/O error on changing an image copy record.
		C3100007	RECON I/O error on locating an image copy after REORG.
		C3100008	RECON I/O error deleting a REORG record.
		C3100009	RECON I/O error deleting a RECOV record.
		C310000A	RECON I/O error updating a DB record in NewDBD.
IROERC50	16	C3500001	RECON I/O error on first locate.
		C3500002	RECON I/O error on sequential locate.
		C3500003	A call to DBRC returned a non zero return code.
IROERC60	16	C3600001	RECON I/O error on first OLDS locate.

Part 4. IMS Database Recovery Facility job generation (RPID)

You can use the Recovery Point Identification ISPF interface (RPID ISPF interface) either with or without the RPID report to automatically generate IMS Database Recovery Facility jobs.

Before you generate a job, you must first specify database components and recovery times.

You can then customize that job to fit the requirements of your environment.

Topics:

Chapter 11. Using Recovery Point Identification (RPID)

Before you generate IMS Database Recovery Facility Extended Functions jobs, you must specify the database components to be recovered and the times for recovery.

To specify database components and recovery times, start the RPID ISPF interface either with or without the Recovery Point ID (RPID) report.

Topics:

Selecting database components and recovery times

To automatically generate IMS Database Recovery Facility jobs, the Recovery Point Identification (RPID) ISPF interface provides two methods for selecting database components and recovery times.

- Use the RPID ISPF interface with the RPID report.
- Use the RPID ISPF interface without the RPID report.

Using the RPID report

When you use the RPID ISPF interface with the RPID report, you select database components and recovery times directly from the RPID report based on common recovery time spans. You do not have to independently verify database components and recovery times.

Using the RPID ISPF interface with the RPID report is recommended because the report determines ideal recovery times for you.

Not using the RPID report

Alternatively, you can use the RPID ISPF interface without the RPID report.

This method requires you to manually specify database components and recovery times. Specifying components and times without the report can save you time because you do not have to navigate through the RPID report to select database components and recovery times.

However, you are responsible for manually verifying that the database components and recovery times are valid from IMS Database Recovery Control.

Run RPID with the RPID report

When you use the Recovery Point ID (RPID) ISPF interface to generate IMS Database Recovery Facility jobs, you must specify the database components that you want to recover and the recovery time.

Before you begin

Using the RPID ISPF interface with the RPID report allows you to specify database components and recovery times directly from the RPID report.

Because the RPID report lists validated recovery time spans for one or more databases, you do not have to independently verify database components and recovery times.

Before you begin, complete the following prerequisite tasks:

- Install the Extended Functions ISPF interface and allocate the ISPF libraries required for the RPID ISPF interface.
- Run the RPID function to generate an RPID report.

  The report must be in an EDIT or VIEW session.
- To recover a group of database components to a common recovery point, make sure that a common time span exists for all of the database components.

  If the database components do not have a common time span, you can generate one by using the Recovery Point Create function.

About this task

To select the database components and recovery times by using the RPID ISPF interface with the RPID report:

Procedure

1. Start the RPID ISPF interface with the RPID report by issuing the IROTREXL command on the command line of an RPID report:

   ```
   Command ===> irotrexl
   ```

   After you start the RPID ISPF interface, the initial RPID ISPF interface panel is displayed, as shown in the following figure:

   ```
                           IMS DATABASE FACILITY EXTENDED FUNCTIONS
                                 Recovery Point ID Report
   Command ===> ----------------------------------------------------------------

     Select option and press ENTER to continue.  Press END to exit.
       _   1. Common Time Span processing
           2. Database component processing
           3. View selected recovery basis
           4. Generate IMS Database Recovery Facility JCL

    RPID report parameters
       Execution . . . : 12/08/2021 19:26          Time format. . . : UTC

       Time range. . . : 21.314 14:15:40.1+0000 -- 21.364 14:15:40.1+0000
                         Nov 10 2021               Dec 30 2021
   ```

   Figure 11. The initial RPID ISPF interface panel with the RPID report

   The RPID ISPF interface contains help panels that provide more detailed information about the interface and its options.

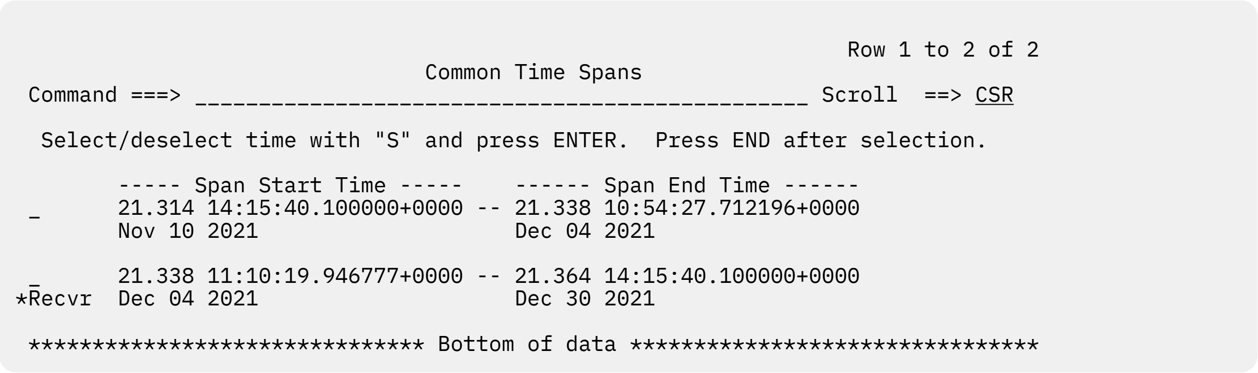
   You can access help panels by pressing the PF1 key at any time.
2. Select **Common Time Span processing**.
3. In the Common Time Spans panel, select a common time span to limit the recovery to a specific time span.

   A common time span helps you recover a group of database components to a common recovery point.

   If you do not select a common time span, generated control statements will not contain an RCVTIME parameter, and any subsequent option that relies on a common time span cannot be selected.

   For examples, the following figure shows that the 21.338 11:10:19.946777+0000 -- 21.364 14:15:40.100000+0000 common time span was selected after S was entered to the left of the selection.

   The selection is now indicated by `*Recvr`.



```
                                                                 Row 1 to 2 of 2
                                Common Time Spans
 Command ===> ________________________________________________ Scroll  ==> CSR

  Select/deselect time with "S" and press ENTER.  Press END after selection.

        ----- Span Start Time -----    ------ Span End Time ------
 _      21.314 14:15:40.100000+0000 -- 21.338 10:54:27.712196+0000
        Nov 10 2021                    Dec 04 2021

 _      21.338 11:10:19.946777+0000 -- 21.364 14:15:40.100000+0000
*Recvr  Dec 04 2021                    Dec 30 2021

 ******************************* Bottom of data *******************************
```

Figure 12. Common Time Spans - Selected

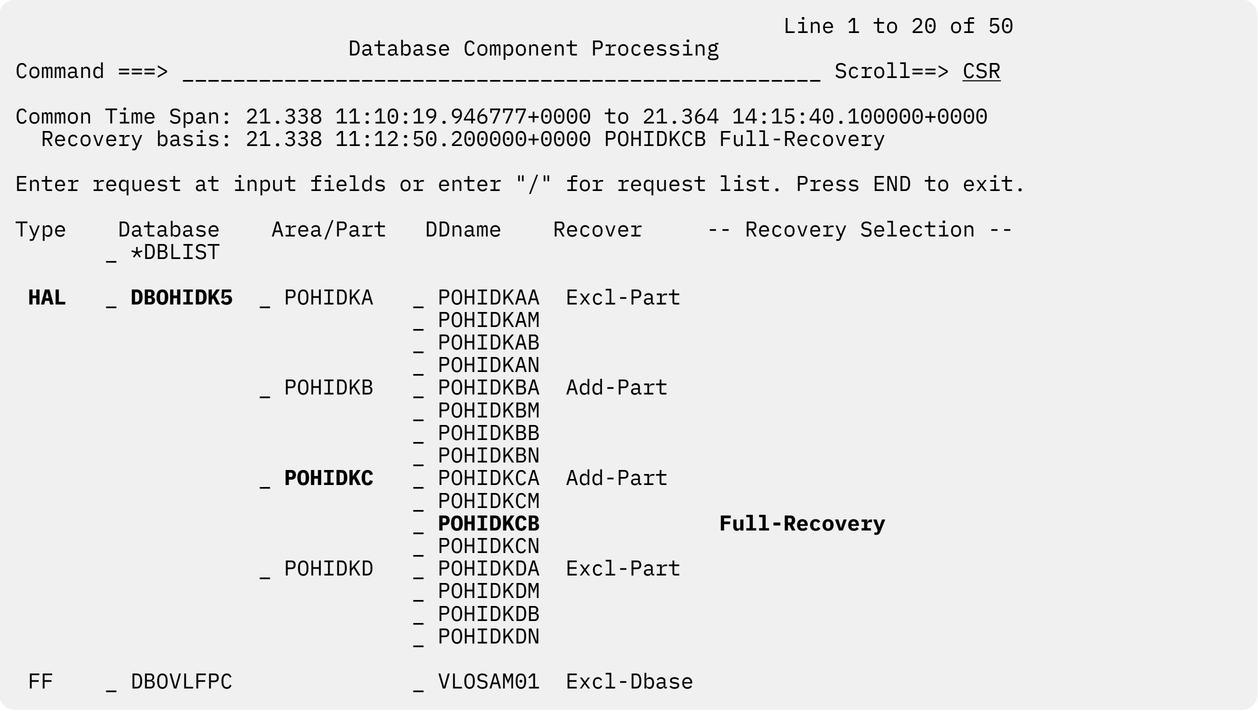
4. Exit the Common Time Spans panel and select **Database component processing** from the IMS Recovery Point ID Report panel.
5. In the Database Component Processing panel, enter a forward slash (/) to the left of the components that you want to recover.

   A list of processing options is displayed that allows you to specify the recovery options for the selected component.

   For example, you can specify the time to which components are to be recovered, exclude specific components for recovery, or view more information about each component.

   Attention: To avoid data integrity exposure, do not specify individual ddnames for recovery. If you want to specify individual ddnames, make sure that you are familiar with ddname recovery and its implications.

   The following figure shows that the partitions POHIDKB and POHIDKC are selected for recovery and that the full recovery time of ddname POHIDKCB is selected as the recovery time. All other components are excluded from recovery.



```
                                                            Line 1 to 20 of 50
                          Database Component Processing
Command ===> __________________________________________________ Scroll==> CSR

Common Time Span: 21.338 11:10:19.946777+0000 to 21.364 14:15:40.100000+0000
  Recovery basis: 21.338 11:12:50.200000+0000 POHIDKCB Full-Recovery

Enter request at input fields or enter "/" for request list. Press END to exit.

Type    Database    Area/Part   DDname    Recover     -- Recovery Selection --
       _ *DBLIST

 HAL   _ DBOHIDK5  _ POHIDKA   _ POHIDKAA  Excl-Part
                               _ POHIDKAM
                               _ POHIDKAB
                               _ POHIDKAN
                   _ POHIDKB   _ POHIDKBA  Add-Part
                               _ POHIDKBM
                               _ POHIDKBB
                               _ POHIDKBN
                   _ POHIDKC   _ POHIDKCA  Add-Part
                               _ POHIDKCM
                               _ POHIDKCB              Full-Recovery
                               _ POHIDKCN
                   _ POHIDKD   _ POHIDKDA  Excl-Part
                               _ POHIDKDM
                               _ POHIDKDB
                               _ POHIDKDN

 FF    _ DBOVLFPC              _ VLOSAM01  Excl-Dbase
```

Figure 13. Recovery time selected on the Database Component Processing panel

When you select a recovery time, the database component that you selected the recovery time from is highlighted. Any parents and children of that database component are also highlighted. The preceding figure uses bold to represent highlights.

If you have specified a recovery time, the time is shown in the **Recovery basis** field.

The Recover column contains recovery status indicators that define how JCL will be generated for each component.

6. After you have selected the components that you want to recover, exit the Database Component Processing panel.
7. Optional: To review a summary of your recovery selections, select **View selected recovery basis** from the IMS Recovery Point ID Report panel.
8. Select **Generate IMS Database Recovery Facility JCL** to complete the selection of database components and recovery times.

   The IMS Database Recovery Facility Job Generation panel is displayed, which allows you to customize parameters, customize control statements, and add additional JCL before generating the IMS Database Recovery Facility job.

RPID report initialization errors

If the RPID report fails validation when you attempt to start the RPID ISPF interface with the RPID report, an error message is issued and the RPID ISPF interface does not start.

The following sections describe initialization errors that might cause the RPID report validation to fail.

Physical file characteristics

Files with a data width of up to 256 bytes (LRECL=256 for FB, or LRECL=260 for VB) are supported. Files that exceed this size cause the RPID interface to stop and issue the following ISPF error message.

Short message: Unsupported LRECL
Long message: File being processed has Data_Width=*nnn*. Data_Width greater than 256 is not supported.

Make sure that the pre-allocated data set in which you save the RPID report is set to a data width of up to 256 bytes (LRECL=256 for FB, or LRECL=260 for VB), and then generate another RPID report. If the problem persists, contact IBM Software Support.

Title, section, and page headings

The RPID ISPF interface processes and presents composite information that is extracted from the RPID report sections with the following title and headers.

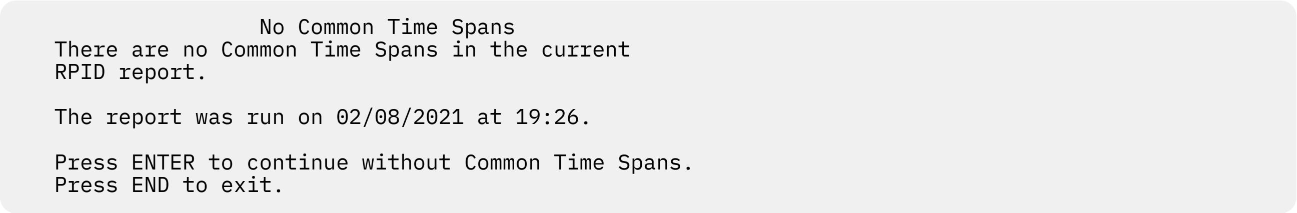
```
IMS DATABASE FACILITY EXTENDED FUNCTIONS RECOVERY POINT ID REPORT

-> Invocation Parameters:
--> Recovery Time Spans Common to All Entries in the DBLIST:
--> Recovery Time Spans for Individual Entries in the DBLIST:
--> Recovery Points (Utility End Times) for Individual Entries in the DBLIST:
```

If the title or headings cannot be located in the report that is being processed, the RPID interface stops, and an ISPF error message is displayed. To avoid this error, rerun the RPID function to generate another RPID report, and do not delete any titles or headers from the report.

Common time spans

The RPID ISPF interface examines the common time spans that are defined in the RPID report. If no common time spans are contained in the report, the following panel is displayed.



```
                No Common Time Spans
There are no Common Time Spans in the current
RPID report.

The report was run on 02/08/2021 at 19:26.

Press ENTER to continue without Common Time Spans.
Press END to exit.
```

Figure 14. No Common Time Spans panel

If you want to generate a common time span, use the Recovery Point Creation function.

Recovery status indicators

Recovery status indicators show which database components are selected for recovery.

The recovery status indicators are shown only in the Database Component Processing panel of the RPID ISPF interface with the RPID report.

The following table describes each recovery status indicator from the Recover column of the Database Component Processing panel.

Table 7. Recovery status indicators

Status indicator	**Description**
Add-Dbase	This database is selected for recovery. A DRF ADD statement is generated for the database's recovery.
Add-Area	This FastPath area is selected for recovery. A DRF ADD statement is generated for the area's recovery.
Add-Part	This HALDB partition is selected for recovery. A DRF ADD statement is generated for the partition's recovery.
Add-DD	This ddname is selected for recovery. A DRF ADD statement is generated for the ddname's recovery.
Add-Implied	This status occurs only for HALDB OLR partner ddnames when their primary ddnames have been selected for recovery and marked with Add-DD. DRF ADD control statements are not generated for ddnames that are marked Add-Implied. The recovery of these ddnames is implied by the recovery of their primary ddnames.
Excl-Dbase	This database is excluded from recovery. No ADD statements are generated for the database.
Excl-Area	This FastPath area is excluded from recovery. No ADD statement is generated for the area.
Excl-Part	This HALDB partition is excluded from recovery. No ADD statements are generated for the partition.
Excl-DD	This ddname is excluded from recovery. No ADD statements are generated for the ddname.

Table 7. Recovery status indicators (continued)

Status indicator	Description
blank	ADD statements are not generated for components for which the Recover column is blank. Recovery of these components is incorporated into the recovery or exclusion of their database, area, or partition. For example, if a partition is indicated as Add-Part, the ddnames within the partition are recovered by the recovery of the partition. The Recover column for these ddnames is blank. The Recover column for the *DBLIST entry is always blank.

Run RPID without the RPID report

When you use the Recovery Point ID (RPID) ISPF interface to generate IMS Database Recovery Facility jobs, you must specify the database components that you want to recover and the recovery time.

Before you begin

If you want to specify database components and recovery times that you have already verified, use the RPID ISPF interface without the RPID report.

An ISPF panel is displayed in which you can specify any number and combination of database components to be recovered and the recovery time of those components.

Because using the RPID ISPF interface without the RPID report will not use database components and time spans that have been previously verified by the RPID function, you are responsible for validating recovery time spans for all database components from IMS Database Recovery Control.

Before you can use the RPID ISPF interface, you must install the Extended Functions ISPF interface.

About this task

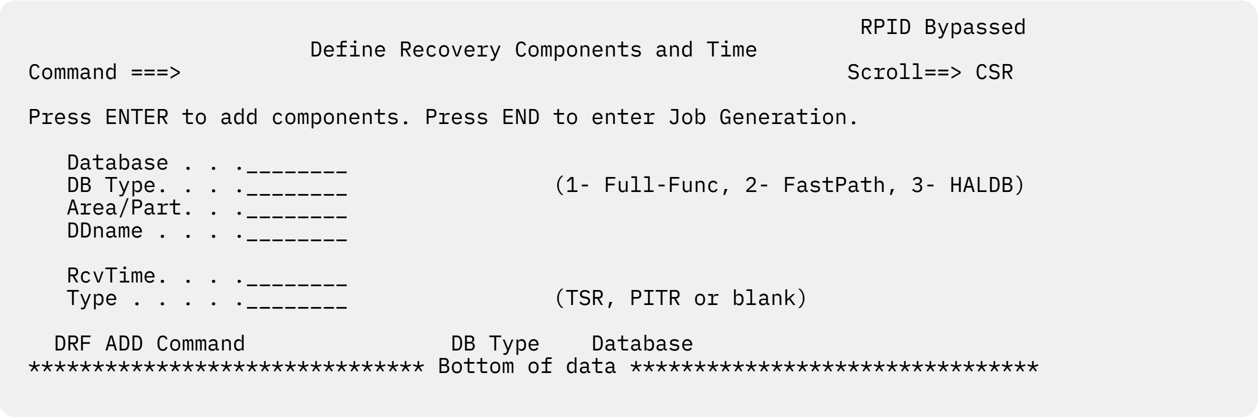
To specify database components and recovery times:

Procedure

1. Start the RPID ISPF interface without RPID report processing by issuing the TSO %IROTREXL command on the ISPF command line:

   ```
   Command ===> tso %irotrexl
   ```

   After starting the interface, the Define Recovery Components and Time panel is displayed, in which you can specify database components and recovery times.



```
                                                                 RPID Bypassed
                      Define Recovery Components and Time
Command ===>                                                    Scroll==> CSR

Press ENTER to add components. Press END to enter Job Generation.

   Database . . .________
   DB Type. . . .________                (1- Full-Func, 2- FastPath, 3- HALDB)
   Area/Part. . .________
   DDname . . . .________

   RcvTime. . . .________
   Type . . . . .________                (TSR, PITR or blank)

  DRF ADD Command                DB Type    Database
******************************* Bottom of data ********************************
```

Figure 15. Initial RPID ISPF interface without the RPID report

2. In the **Database** field, enter the database name to be recovered.
3. In the **DB Type** field, select the correct database type.

   The selection of data types are shown next to the **DB Type** field.
4. In the **Area/Part** field, enter the component of the database that you want to recover.
5. In the **DDname** field, specify the individual ddname for recovery.

   ⚠ **Attention:** To avoid data integrity exposure, do not specify individual ddnames for recovery. If you want to specify individual ddnames, make sure that you are familiar with ddname recovery and its implications.
6. Optional: In the **RcvTime** field, you can specify the recovery time of the components that you want recovered.

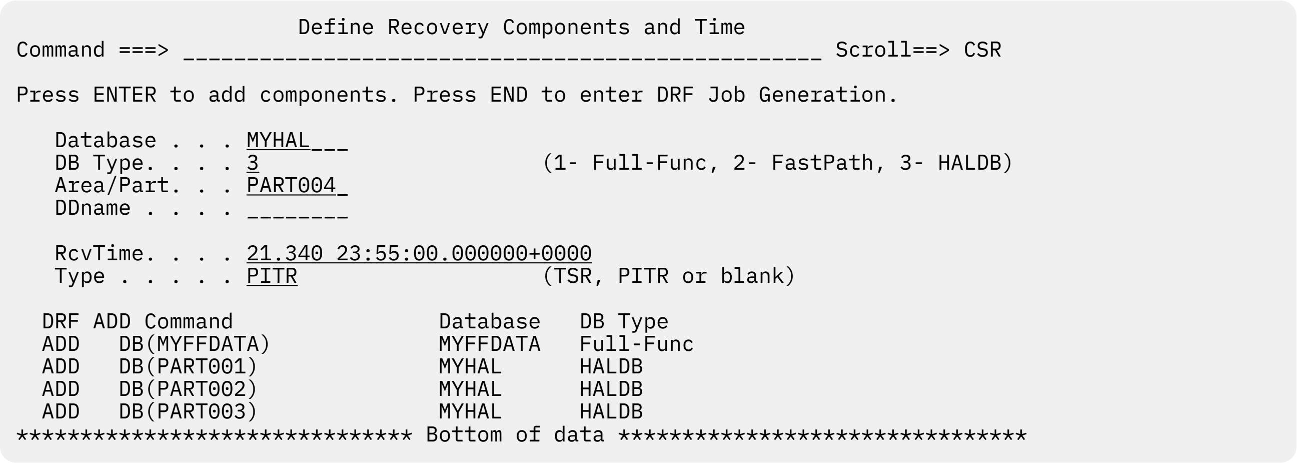
   You can specify any time format that is acceptable in IMS Database Recovery Facility.

   If you specify a time, you must also identify the time as either TSR or PITR in the **Type** field. If you do not specify a time, the RPID ISPF interface generates a full recovery of the components.
7. Press **Enter** to complete your entry.
8. Optional: Enter more components to be recovered.
9. After you have specified all of the entries, press **Exit** to exit the Define Recovery Components and Time panel.

   When you exit the Define Recovery Components and Time panel, the IMS Database Recovery Facility Job Generation panel is displayed. The job generation panel allows you to customize parameters, customize control statements, and add additional JCL before generating a job.

Example

The following figure shows a full-function database (MYFFDATA) and three partitions of a HALDB database (MYHAL) that have all been entered for IMS Database Recovery Facility job generation. A fourth MYHAL partition will be added when **Enter** is pressed.



```
                      Define Recovery Components and Time
Command ===> __________________________________________________ Scroll==> CSR

Press ENTER to add components. Press END to enter DRF Job Generation.

   Database . . . MYHAL___
   DB Type. . . . 3                      (1- Full-Func, 2- FastPath, 3- HALDB)
   Area/Part. . . PART004_
   DDname . . . . ________

   RcvTime. . . . 21.340 23:55:00.000000+0000
   Type . . . . . PITR                   (TSR, PITR or blank)

  DRF ADD Command                Database   DB Type
  ADD   DB(MYFFDATA)             MYFFDATA   Full-Func
  ADD   DB(PART001)              MYHAL      HALDB
  ADD   DB(PART002)              MYHAL      HALDB
  ADD   DB(PART003)              MYHAL      HALDB
******************************* Bottom of data ********************************
```

Figure 16. Define Recovery Components and Time panel

Chapter 12. Job customization and generation

After you have selected database components for recovery, you can customize your job by maintaining parameter sets, adding additional JCL to your job, or by specifying other options in IMS Database Recovery Facility Extended Functions.

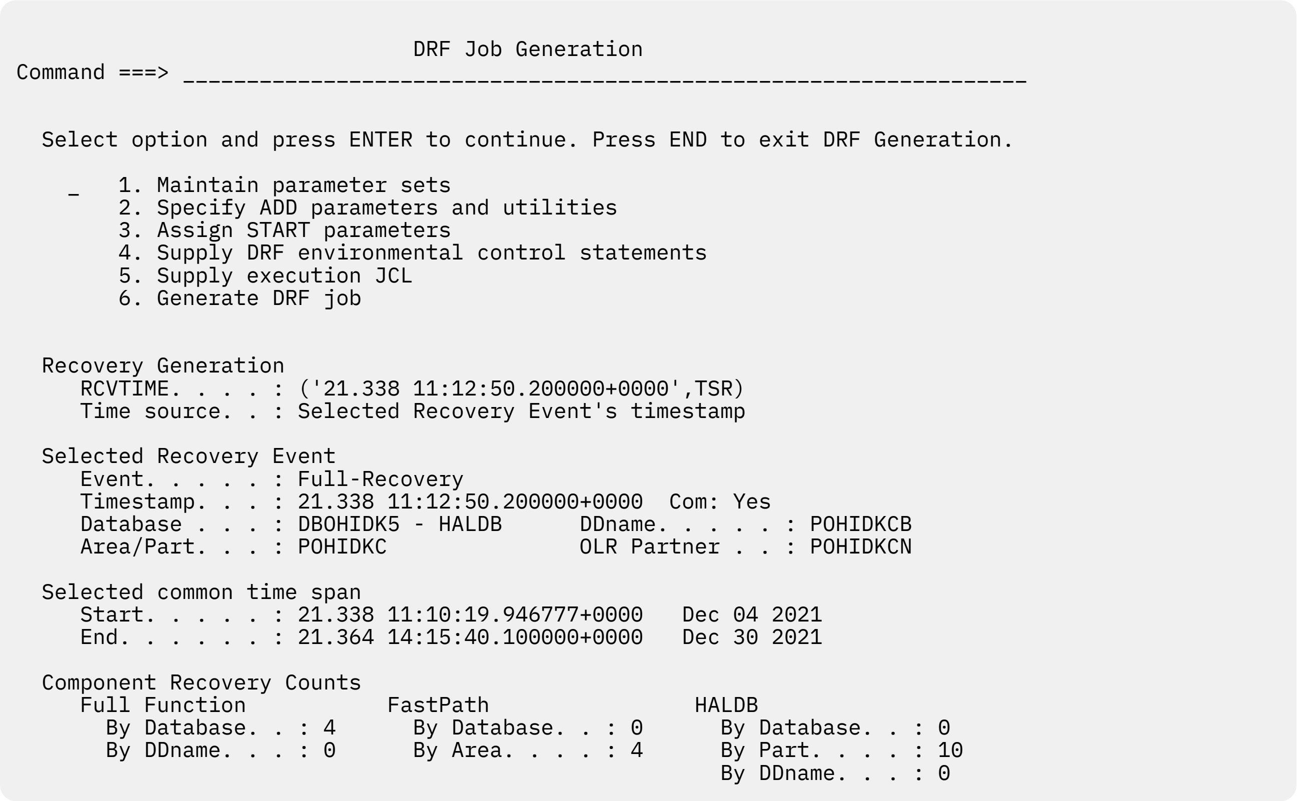
The job generation panel is displayed after you have chosen database components and recovery times from the RPID ISPF interface with either the RPID report or without the RPID report.

Job customization helps you generate jobs that are specific to your environment.

If you do not want to customize your job, select **Generate DRF job** to generate a generic job.

Important: You must supply a data set that contains the JCL that is required to execute the recovery utility. The job generation process creates only the ADD statements that are needed for the input to IMS Database Recovery Facility. These input control statements that are generated by the job generation process are appended to the JCL that you supply.

The following figure shows an example of the IMS Database Recovery Facility Job Generation panel. Your recovery selections are shown after the list of options.



```
                               DRF Job Generation
Command ===> _________________________________________________________________


  Select option and press ENTER to continue. Press END to exit DRF Generation.

    _   1. Maintain parameter sets
        2. Specify ADD parameters and utilities
        3. Assign START parameters
        4. Supply DRF environmental control statements
        5. Supply execution JCL
        6. Generate DRF job


  Recovery Generation
     RCVTIME. . . . : ('21.338 11:12:50.200000+0000',TSR)
     Time source. . : Selected Recovery Event's timestamp

  Selected Recovery Event
     Event. . . . . : Full-Recovery
     Timestamp. . . : 21.338 11:12:50.200000+0000  Com: Yes
     Database . . . : DBOHIDK5 - HALDB      DDname. . . . . : POHIDKCB
     Area/Part. . . : POHIDKC               OLR Partner . . : POHIDKCN

  Selected common time span
     Start. . . . . : 21.338 11:10:19.946777+0000   Dec 04 2021
     End. . . . . . : 21.364 14:15:40.100000+0000   Dec 30 2021

  Component Recovery Counts
     Full Function           FastPath                HALDB
       By Database. . : 4      By Database. . : 0      By Database. . : 0
       By DDname. . . : 0      By Area. . . . : 4      By Part. . . . : 10
                                                       By DDname. . . : 0
```

Figure 17. Main IMS Database Recovery Facility job generation option display menu

The ISPF interface contains help panels that provide more detailed information about the interface and its options.

You can access help panels by pressing the PF1 key at any time.

IMS Database Recovery Facility job generation options

The DRF Job Generation panel provides the following six options that you can use to customize and generate IMS Database Recovery Facility jobs.

1. Use the Maintain Parameter Sets option to maintain groups of parameters that might be associated with IMS Database Recovery Facility ADD and START control statement.

   You can define separate parameter sets for Index Builder, Image Copy, Pointer Checker, and START.
2. Use the Specify ADD Parameters and Utilities option to customize each generated IMS Database Recovery Facility ADD control statement.

   For example, you can set individual ADD parameters, include utilities, and propagate parameter settings among ADD statements.
3. Use the Assign START Parameters option to customize the generated IMS Database Recovery Facility START control statement by specifying START parameters.
4. Use the Supply DRF Environmental Control Statements option to specify or override parameters that are contained in the FRXDRF*xx* control member.
5. Use the Supply Execution JCL option to include additional JCL that is required for executing the IMS Database Recovery Facility job.
6. Use the Generate DRF job option to produce output that is based on your recovery selections and customization specifications.

Database component and recovery time re-selection

You can exit the IMS Database Recovery Facility job generation process, re-select the set of database components to be recovered, and then reenter the IMS Database Recovery Facility job generation process. This ability to exit and reenter the job generation process allows you to modify the set of ADD control statements without starting over.

All parameters, except those that are related to ADD statements, are maintained when you exit and reenter the IMS Database Recovery Facility job generation process.

Because exiting and reentering the IMS Database Recovery Facility job generation process can invalidate ADD statement parameter associations, all ADD statement parameters are discarded when you exit IMS Database Recovery Facility job generation.

Generated jobs

Generated jobs incorporate your database component selections, time selection, and parameter specifications.

After you generate a job, the RPID ISPF interface initiates an ISPF/PDF edit session with a temporary data set that contains the generated statements. The interface does not save these statements to a permanent data set or submit the generated job. You use the editor's facilities to perform these functions.

An example of generated output is shown in the following figure. When multiple database components have the same attributes and have been assigned the same parameters, they are grouped into single ADD control statement.

If IMS Database Recovery Facility environment control statements were specified, they are added to the generated IMS Database Recovery Facility control statement.

You must specify the IMS Database Recovery Facility JCL statements to form a complete job. Otherwise, the input control statements will be generated with no execution JCL.

```
//EXAMPLE  JOB (&SYSUID),'&SYSUID',
//         CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
//         REGION=6M,NOTIFY=&SYSUID,USER=&SYSUID
//*
//*-------------------------------------------------------------------*
//*  EXECUTE FRXJCLDF PROC FOR DATABASE RECOVERY FACILITY (DRF)       *
//*-------------------------------------------------------------------*
//*
// JCLLIB ORDER=(IMSTESTG.PROCLIB)
//*
//        EXEC FRXJCLDF,RGN=0M,
//             SOUT='*',
//             VCAT=VCATQAV,
//             DELTA01='IMSTESTG.DELTALIB',
//             DELTA02='IMSTESTG.DELTALIB',
//             DELTA03='IMSTESTG.DELTALIB',
//             RESLIB='IMSBLD.P21RTS16.DRF.SFRXRESL',
//             NODE=I91RTS16,
//             DRFMBR=01,
//             BPECFG=BPEDRFCG,
//             DRFPROC=DRFS,
//             GSGNAME=,
//             PLEXNAME=
//*
//*
//*------------------------------------------------------------------*/
//*                                                                  */
//*  Selected Recovery Time:                                         */
//*     Event......... Full-Recovery                                 */
//*     Time stamp.... 21.338 11:12:50.200000+0000  Com: Yes         */
//*     Database...... DBOHIDK5                                      */
//*     Area/Part..... POHIDKC                                       */
//*     DDname........ POHIDKCB                                      */
//*                                                                  */
//*------------------------------------------------------------------*/
//SYSIN    DD  *
  DRFIAX(DRFL)
  REPORT(RPTTYPE=SEP,DRFUNIT=SYSDA,DRFHLQ=DRBX10)
  UTILGBL(COMP=Y,DSN(&ICHLQ..&DBD..&DDN.),UNIT(3390),-
          ICBUF(15),VOLSER(333333),COMPRTN(FABJCMP3))
  OUTPUT(PRO)
  DBDSL101(DISP(NEW))

    /*---------------------------------------------------------------*/
    /*  Fast-Path  Recover Area(s)                                   */
    /*             IC:SOME PARMS                                     */
    /*---------------------------------------------------------------*/
ADD AREA(DEDBDD01 DD01AR0,                                       -
         DEDBJN24 DB24A000,                                       -
         DEDBJN24 DB24A001,                                       -
         DEDBJN24 DB24A239)                                       -
         DBATRB(DBDSL(001),DBDSN(003))                            -
      IC(COMP(Y),                                                 -
         COMPRTN(FABJCMP3),                                       -
         DSNTYPE(BASIC),                                          -
         EXPDT(99000),                                            -
         ICBUF(15),                                               -
         ICCAT(Y),                                                -
         ICHLQ(TEST.IC.PRIMARY,                                   -
               TEST.IC.SECOND),                                   -
         ICNMRULE(N),                                             -
         MGMTCLAS(STANDARD),                                      -
         SPACE(CYL,50,75,RLSE),                                   -
         STORCLAS(STORE#IC))
```

Figure 18. Sample Generated Output (Part 1 of 2)

```
    /*---------------------------------------------------------------*/
    /*  Full-Func  Recover Database(s)                               */
    /*             IB:TEST         IC:SOME PARMS   PC:FEW REPORTS    */
    /*---------------------------------------------------------------*/
ADD   DB(DBOVLFPC,                                               -
         DHVNTZ02,                                               -
         DIVNTZ02,                                               -
         DXVNTZ02)                                               -
      IB(ICNDX(YES),                                             -
         NDXIOBUF(DATA(128),INDEX(128)),                         -
         SORTE35(YES),                                           -
         SORTID(E),                                              -
         SORTOUT(COPY),                                          -
         SORTSTAT(YES))                                          -
      IC(COMP(Y),                                                -
         COMPRTN(FABJCMP3),                                      -
         DSNTYPE(BASIC),                                         -
         EXPDT(99000),                                           -
         ICBUF(15),                                              -
         ICCAT(Y),                                               -
         ICHLQ(TEST.IC.PRIMARY,                                  -
               TEST.IC.SECOND),                                  -
         ICNMRULE(N),                                            -
         MGMTCLAS(STANDARD),                                     -
         SPACE(CYL,50,75,RLSE),                                  -
         STORCLAS(STORE#IC))                                     -
      PC(BITMAP(NO),                                             -
         DIAG(YES),                                              -
         DUMPFORM(FORMAT),                                       -
         FABASNAP(NO),                                           -
         FSEMAP(YES),                                            -
         INTERVAL(BLOCK(05)),                                    -
         INTFS(YES),                                             -
         INTST(YES),                                             -
         MAXFSD(NO),                                             -
         PRINTDATA(YES),                                         -
         RUNTM(NO),                                              -
         T2CHK(0,7))

    /*---------------------------------------------------------------*/
    /*  HALDB      Recover Partition(s)                              */
    /*---------------------------------------------------------------*/
ADD   DB(POHIDKA,                  /* MASTER DB=DBOHIDK5 */      -
         POHIDKB,                  /* MASTER DB=DBOHIDK5 */      -
         POHIDKC,                  /* MASTER DB=DBOHIDK5 */      -
         POHIDKD,                  /* MASTER DB=DBOHIDK5 */      -
         PVHDJ5A,                  /* MASTER DB=DBVHDJ05 */      -
         PVHDJ5B,                  /* MASTER DB=DBVHDJ05 */      -
         PVHDJ5C,                  /* MASTER DB=DBVHDJ05 */      -
         PVHDJ5D,                  /* MASTER DB=DBVHDJ05 */      -
         P2XHDJA,                  /* MASTER DB=D2XHDJ05 */      -
         P2XIDKA)                  /* MASTER DB=D2XHIDK5 */

      /*-------------------------------------------------------------*/
START RCVTIME('21.338 11:12:50.200000+0000',CHECK)   /*TSR*/     -
         VERIFY(LIST)                                            -
         ERROR(STOP)
/*
```

Figure 19. Sample Generated Output (Part 2 of 2)

Part 5. Troubleshooting

Use these topics to diagnose and correct problems that you experience with IMS Database Recovery Facility Extended Functions.

Topics:

Chapter 13. Runtime messages (IRO)

This reference section provides detailed information about IMS Database Recovery Facility Extended Functions messages.

Message format

The explanations and user responses provided in this reference can help you diagnose, troubleshoot, and solve IMS Database Recovery Facility Extended Functions problems.

IMS Database Recovery Facility Extended Functions messages adhere to the following format:

```
IROnnnnx
```

where:

IRO
Indicates that the message was issued by IMS Database Recovery Facility Extended Functions

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.

IRO0002I *tcbname* TCB INITIALIZATION COMPLETE

Explanation

This message indicates that one of the IMS Database Recovery Facility Extended Functions TCBs has successfully initialized. *tcbname* identifies the specific TCB as one of the following values:

REM
Main TCB

TDI
DSPAPI TCB

TOI
IMS Tools Online System Interface TCB

CSV
Common Services TCB

DRC
DBRC TCB

PRC
Function Processing TCB

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMICSV, IROMIDRC, IROMIPRC, IROMIREM, IROMITDI, IROMITOI

IRO0003I ***tcbname* TCB TERMINATION COMPLETE**

Explanation

This message indicates that one of the IMS Database Recovery Facility Extended Functions TCBs has successfully terminated. *tcbname* identifies the specific TCB as one of the following values:

REM
Main TCB

TDI
DSPAPI TCB

TOI
IMS Tools Online System Interface TCB

CSV
Common Services TCB

DRC
DBRC TCB

PRC
Function Processing TCB

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMTCSV, IROMTDRC, IROMTPRC, IROMTREM, IROMTTDI, IROMTTOI

IRO0004E **RECONID LOCATE PROCESSING FAILURE, REASON: *reason***

Explanation

This message indicates that an attempt to locate the RECONID member from IMS Tools Base for z/OS IMS Tools Knowledge Base (IMS Tools Knowledge Base) has failed. The *reason* variable identifies the cause of the failure as one of the following values:

UNABLE TO LOCATE RECONID OR RECON1 DSN
The RECONID() keyword was not used to specify the RECONID and the RECON1 data set name could not be located. Either specify a valid RECONID() value or ensure that the RECON1 data set is allocated in the JCL or, the RECON1 dynamic allocation member can be located in the IMSDALIB DD or the JOBLIB/STEPLIB.

RECONID INTERFACE INIT FAILURE
A failure occurred attempting to initialize the RECONID interface. Ensure that IMS Tools Knowledge Base has been initialized and is available.

RECONID RECORD NOT FOUND
A RECONID record matching either the RECONID() value or the RECON1 data set name could not be located in IMS Tools Knowledge Base. Ensure that the correct value is specified and that the RECONID member exists in IMS Tools Knowledge Base.

RECONID INTERFACE READ FAILURE
A failure occurred while attempting to read the RECONID record. Ensure that the ITKBSRVR that is used in the IMS Database Recovery Facility Extended Functions job has been brought up before starting IMS Database Recovery Facility Extended Functions. Ensure that the IMS Tools Knowledge Base server has a defined IMS Database Recovery Facility Extended Functions product ID (PRODUCTID=DE) and that the IMS Tools Knowledge Base server repositories have at least one defined IMS Database Recovery Facility Extended Functions RECONID. Refer to your installation jobs provided to install IMS Tools Knowledge Base for IMS Database Recovery Facility Extended Functions.

RECONID DSN MISMATCH
The RECON data set names that are specified in the JCL do not match the RECON data set names that are specified in the RECONID record. Messages IRO0013E and IRO0014E are also issued indicating the data set names that are in error.

For more information on the reason codes, see the *IBM Tools Base for z/OS: IMS Tools Knowledge Base User's Guide and Reference*.

System action:
Processing terminates.

System programmer response:
Determine the reason for the failure and ensure the correct RECONID record can be located. See "RECONID locate processing" on page 23 for details on how the RECONID record is located.

Module:

IROMDRV0

IRO0005E LOAD FAILED FOR MODULE *module*

Explanation:
This message indicates that an attempt to load module *module* failed.

System action:
Processing terminates.

System programmer response:
Ensure the correct load libraries are available through JOBLIB, STEPLIB, or LPALIB.

Module:
IROMINI0, IROMSDRC

IRO0006E MISSING IMSREIN DD

Explanation:
This message indicates that the IMSREIN DD statement, which is used to specify runtime parameters, could not be located.

System action:
Processing terminates.

System programmer response:
Ensure the IMSREIN DD is specified in the JCL.

Module:
IROMPARS

IRO0008E *process* PROCESS FAILED

Explanation

This message indicates that an error occurred during processing for the specified *process*. Valid values for *process* are:

LOCRECON
An attempt to locate the RECON data sets has failed. Make sure the RECON1, RECON2 and RECON3 data set names are available either through JCL or dynamic allocation.

RECONCOPY
The RECONCOPY process has failed. Review the messages printed to the IREDIAG DD statement for errors relating to this process.

ALLOCRECON
An error occurred attempting to dynamically allocate the RECON data sets.

INITDRC
Initialization for the DRC TCB has failed.

INITTDI
Initialization for the TDI TCB has failed.

INITTOI
Initialization for the TOI TCB has failed.

TERMDRC
Termination for the DRC TCB has failed.

TERMTDI
Termination for the TDI TCB has failed.

TERMTOI
Termination for the TOI TCB has failed.

System action:
Processing terminates.

System programmer response:
Check the joblog for additional messages relating to the error. Contact IBM if necessary.

Module:
IROMDRV0

IRO0009E IMS RE API INIT ERROR WHEN USESLBIC(Y) IS SELECTED RC= *nnnnnnnn* RSN= *nnnnnnnn*

Explanation:
The USESLBIC(YES) configuration parameter was set, but the initialization of the IMS Recovery Expert application programming interface encountered an error.

System action:
Processing terminates.

System programmer response:
If you cannot determine and correct an environmental cause for the problem, save the SYSLOG information (including the BSYTRACE SYSOUT data set) and send it to IBM Software Support.

Module:
IROMDRV0, IROSENSR

IRO0010E MISSING OR INVALID KEYWORD PARAMETER SPECIFIED - *keyword*

Explanation:
This message indicates that the *keyword* specified in the message is either invalid or is required and was not specified.

System action:
Processing terminates.

System programmer response:
Review the input control cards to ensure they are correctly specified.

Module:
IROMINI0

IRO0012E DYNAMIC ALLOCATION FOR DD *ddname* FAILED

Explanation:
This message indicates that an attempt to dynamically allocate the specified *ddname* has failed.

System action:
Processing terminates.

System programmer response:

Inspect the joblog for additional messages indicating the reason for the failure. If the specified *ddname* is allocated in the JCL, ensure that the attributes are correct.

Module:
IROMINI0, IROMDRV0

IRO0013E **RECON DATA SET NAME MISMATCH FOR RECON DD *ddname***

Explanation:
This message indicates that there is a mismatch between a RECON data set name specified in the JCL and in the RECONID member or between the JCL and the active IMS system. *ddname* indicates either RECON1, RECON2 or RECON3. Either IRO0014E or IRO0015E will follow.

System action:
Processing terminates.

System programmer response:
Ensure that the correct RECON data set names are specified in the JCL and that they match the RECON data set names specified in the RECONID member and the active IMS system.

Module:
IROMDRV0, IROCVRC0

IRO0014E **JCL=*dsn1* RECONID=*dsn2***

Explanation:
This message follows message IRO0013E when there is a mismatch between a RECON data set name specified in the JCL and in the RECONID member. *dsn1* specifies the data set name specified in the JCL, and *dsn2* specifies the data set name specified in the RECONID member.

System action:
Processing terminates.

System programmer response:
Ensure that the correct RECON data set names are specified in the JCL and that they match the RECON data set names specified in the RECONID member and the active IMS system

Module:
IROMDRV0

IRO0015E **IMSID=*imsid*, JCL=*dsn1*, IMS=*dsn2***

Explanation:
This message follows message IRO0013E when there is a mismatch between a RECON data set name specified in the JCL and in the active IMS system. *imsid* specifies the IMSID on which the mismatch occurred. *dsn1* specifies the data set name specified in the JCL, and *dsn2* specifies the data set name allocated to the IMS system.

System action:
Processing terminates.

System programmer response:
Make sure the correct RECON data set names are specified in the JCL and that they match the RECON data set names used by the online IMS system.

Module:
IROCVRC0

IRO0016E **UNABLE TO VERIFY RECON DATA SETS, REASON: *reason***

Explanation:
This message is issued whenever IMS Database Recovery Facility Extended Functions is unable to verify that the RECON data sets allocated to the IMS Database Recovery Facility Extended Functions job are the same as those allocated to the online IMS systems. IMS Database Recovery Facility Extended Functions issues an /RML DBRC='RECON STATUS' command to all active IMS systems in the TOSI XCF group. This output is then parsed to obtain the RECON data set names. This message is received whenever IMS Database Recovery Facility Extended Functions is unable to locate the RECON data set names in the command output. For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Make sure the correct RECON data set names are specified in the JCL and that they match the RECON data set names used by the online IMS system. If this problem persists, specify VERIFYRCN(NO) in your IMSREIN input and contact IBM support.

Module:
IROCVRC0

IRO0017E **IMS RE API MODULE LOAD ERROR WHEN USESLBIC SELECTED RC=*nnnnnnnn* RSN=*nnnnnnnn***

Explanation:
The USESLBIC(YES) configuration parameter was set, but the IMS Recovery Expert application programming interface module BSY#API0 could not be loaded.

System action:
Processing terminates.

System programmer response:
If you cannot determine and correct an environmental cause for the problem, save the SYSLOG information (including the BSYTRACE SYSOUT data set) and send it to IBM Software Support.

Module:
IROMINI0, IROSENSR

IRO0020I ********************************

Explanation:
This is a separator line for displaying parameter output to the joblog.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0021I *** IMS RE CONFIGURATION OPTIONS: ***

Explanation:
This is a title line for displaying configuration options to the joblog.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0022I **** keyword=value ****

Explanation:
This message is used to display a configuration *keyword* and *value* to the joblog.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0023I *** IMS RE RECONID CONFIGURATION: ***

Explanation:
This is a title line for displaying the RECONID member configuration to the joblog.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0024I **** keyword=value ****

Explanation:
This message is used to display a RECONID configuration *keyword* and *value* to the joblog.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0030E **INVALID AWE FUNCTION**

Explanation:
This message indicates that an invalid application work element (AWE) was detected by one of the servers.

System action:
The request represented by the AWE is rejected and processing continues.

System programmer response:
This is an internal error. Contact IBM support.

Module:
IROMSCSV, IROMSPRC, IROMSREM, IROMSDRC, IROMSTDI, IROMSTOI

IRO0031E **UNABLE TO ACQUIRE STORAGE, SERVICE: *service***

Explanation:
This message indicates that an attempt to acquire storage using the specified *service* has failed.

System action:
Processing terminates.

System programmer response:
Increase the storage region size for the job and resubmit the job.

Module:
IROMSREM, IROMDRV0, IROMSTOI

IRO0032E **SERVICE FAILURE, SERVICE: *service*, RC=*return code*, RSN=*reason code***

Explanation:
This message indicates that the specified *service* has failed. The return and reason code associated with this failure are provided. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
This is an internal error. Contact IBM Software Support.

Module:

IROMDRV0, IROMIDRC, IROMINI0, IROMIPRC, IROMSTOI, IROMSREM

IRO0050E **RECONCOPY SERVICE FAILURE RC=*return code*, RSN=*reason code*, FLAG=*flag***

Explanation:
This message indicates that an error was encountered while processing the RECONCOPY service. The *return code*, *reason code*, and *flag* (diagnostic flag) are returned in this message. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Contact IBM with the information from the message.

Module:
IROMDRV0

IRO0051I **RECONCOPY COMPLETE COPY1=*dsn***

Explanation:
This message indicates that the RECONCOPY service has successfully created COPY1 of the RECON data set name specified by *dsn*.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0052I **RECONCOPY COMPLETE COPY2=*dsn***

Explanation:
This message indicates that the RECONCOPY service has successfully created COPY2 of the RECON data set name specified by *dsn*.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0053I **RECONCOPY COMPLETE COPY3=*dsn***

Explanation:
This message indicates that the RECONCOPY service has successfully created COPY3 of the RECON data set name specified by *dsn*.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0054E **RECONCOPY FAILED, REASON: *reason***

Explanation

This message indicates that the RECON copy service has failed due to the indicated *reason*. The possible reasons for the failure are:

PARAMETER ADDRESS IS ZERO
This is an internal error. Contact IBM support.

INVALID PARAMETER BLOCK
This is an internal error. Contact IBM support.

INVALID MACRO VERSION
This is an internal error. Contact IBM support.

RECON COPY DELETE FAILED
The delete for the existing RECON copies failed. Inspect the information in the IREDIAG output DD for further details.

OUTPUT DD SPECIFIED BUT NOT FOUND
This is an internal error. Contact IBM support.

INPUT RECON1 DSN MISSING OR INVALID
The RECON1 data set name is either missing or invalid. Inspect the information in the IREDIAG output DD for further details.

INPUT RECON2 DSN MISSING OR INVALID
The RECON2 data set name is either missing or invalid. Inspect the information in the IREDIAG output DD for further details.

INPUT RECON3 DSN MISSING OR INVALID
The RECON3 data set name is either missing or invalid. Inspect the information in the IREDIAG output DD for further details.

OUTPUT COPY DSN INVALID
The output copy data set name is invalid. Inspect the information in the IREDIAG output DD for further details.

DSNPREF GREATER THAN MAX LENGTH
The length specified for the DSNPREF() value exceeds the maximum allowable length of 38 characters. Reduce the number of characters in the data set prefix.

INVALID DSNPREF VARIABLE
The resolved data set prefix contains invalid characters. Correct the DSNPREF() value and resubmit the job.

COPY EXISTS BUT REUSE(NO) SPECIFIED
The RECON copy data set already exists and REUSE(NO) was specified. Either delete the existing RECON copy data set, specify a different DSNPREF() value, or specify REUSE(YES) and resubmit the job.

DELETE FAILED FOR RECON COPY
The delete for the existing RECON copies failed. Inspect the information in the IREDIAG output DD for further details.

ALLOCATE FAILED FOR RECON COPY
The allocate for the RECON copy failed. Inspect the information in the IREDIAG output DD for further details.

IDCAMS REPRO FAILED
The IDCAMS REPRO of the RECON data set into the RECON copy failed. Inspect the information in the IREDIAG output DD for further details.

DYNALLOC FOR RECON1 FAILED
Dynamic allocation for the RECON1 data set has failed. Inspect the information in the IREDIAG output DD for further details.

DYNALLOC FOR RECON2 FAILED
Dynamic allocation for the RECON2 data set has failed. Inspect the information in the IREDIAG output DD for further details.

DYNALLOC FOR RECON3 FAILED
Dynamic allocation for the RECON3 data set has failed. Inspect the information in the IREDIAG output DD for further details.

DYNALLOC FOR BACKUP1 FAILED
Dynamic allocation for the BACKUP1 data set has failed. Inspect the information in the IREDIAG output DD for further details.

DYNALLOC FOR SYSPRINT FAILED
Dynamic allocation for the SYSPRINT output DD has failed. Inspect the information in the IREDIAG output DD for further details.

DYNALLOC FOR SYSIN FAILED
Dynamic allocation for the SYSIN input DD has failed. Inspect the information in the IREDIAG output DD for further details.

DSPURXRT BACKUP.RECON FAILED
The BACKUP.RECON command issued using DSPURXRT has failed. Inspect the information in the IREDIAG output DD for further details.

System action:
Processing terminates.

System programmer response:
Take the appropriate action based on the reason and resubmit the job.

Module:
IROMDRV0

IRO0054I RECONCOPY DELETE(YES) CHANGED TO DELETE(NO) AS REQUIRED FOR RCU

Explanation:
This message indicates that RECONCOPY specified DELETE(YES) which will delete the RECON copy data sets at job step termination. However, RCU requires that the RECON copy data sets are not deleted so, DELETE(YES) was changed to DELETE(NO).

System action:
Processing continues.

System programmer response:
When running RCU, ensure that the DELETE() parameter on RECONCOPY specifies NO.

Module:
IROMDRV0

IRO0060E RECON LOCATE FAILED, REASON: *reason* RC=*return code*

Explanation

This message indicates that an attempt to locate the RECON data set names specified in the JCL or through dynamic allocation failed due to *reason*. The *return code* is also specified. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

LOCATE JFCB FAILED - SWAREQ
The SWAREQ macro failed to locate the JFCB. This is an internal error.

LOAD FOR RECON1 FROM IMSDALIB FAILED
The RECON1 dynamic allocation member could not be loaded. Ensure that the RECON1 member is in this library.

System action:
Processing terminates.

System programmer response:
If the failure is due to a load failure for RECON1 from IMSDALIB, ensure that the RECON1 member is present in the library. Otherwise, contact IBM support.

Module:
IROMDRV0

IRO0061I ALLOC|DEALLOC for *ddname* *dsname* SUCCESSFUL

Explanation:
This message indicates that a RECON data set, identified by *ddname* and *dsname*, was successfully allocated or deallocated.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0062E ALLOC|DEALLOC for *ddname dsname* FAILED, RC=*return code*, RSN=*reason code*

Explanation:
This message indicates that an attempt to allocate or deallocate the specified RECON data set failed with *return code* and *reason code*. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Ensure that the specified RECON data set exists and is available.

Module:
IROMDRV0

IRO0063I RECON1 DFSMDA MEMBER FOUND IN IMSDALIB|STEPLIB, DSN=*dsn*

Explanation:
This message indicates that the RECON1 dynamic allocation member was found in either STEPLIB or IMSDALIB. The RECON1 data set name is specified with *dsn*.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0064I RECON1|RECON2|RECON3 ALLOCATED IN JCL, DSN=*dsn*

Explanation:
This message indicates that specified RECON data set was specified through JCL. The RECON data set name specified by *dsn*.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMDRV0

IRO0095W DBLIST() WARNING, *reason*

Explanation:
This message indicates a warning condition in the specification For the DBLIST() keyword. The *reason* indicates the problem.

System action:
Processing continues.

System programmer response:
Correct the error in the DBLIST().

Module:
IROMDRV0

IRO0096W NO FUNCTIONS SPECIFIED IN IMSREIN DD

Explanation:
This message is issued when no valid IMS Database Recovery Facility Extended Functions functions were specified in the IMSREIN DD input.

System action:
Processing continues.

System programmer response:
Correct any function control card errors.

Module:
IROMDRV0

IRO0100I INITIALIZING FUNCTION *function*

Explanation

This message indicates that processing for the specified *function* is beginning. Valid values for *function* are:

HCHECK
Health Check function

IMSCMD
IMS Command function

RCU
RECON Cleanup function

RPCR
Recovery Point Create function

RPID
Recovery Point Identification function

VERIFY
Verify Recovery Assets function

System action:
Processing continues.

System programmer response:
After IMS systems are up, ensure that the proper TOSI initialization complete messages are displayed with the correct XCFGROUP name in all IMS control regions that share the same RECONs as the IMS Database Recovery Facility Extended Functions job. Also, ensure that the IMS Tools Base for z/OS IMS Tools Knowledge Base RECONID (TOIXCF field)

matches the XCFGROUP=*name* referenced in the FOIssidP members for all IMS subsystems sharing the same RECONs.

Module:
IROMDRV0

IRO0102E **FUNCTION *function* PROCESS COMPLETED WITH RC=*return code*, RSN=*reason code***

Explanation

This message indicates that the specified *function* completed the specified process with *return code* and *reason code*. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

Valid values for *function* are:

HCHECK
Health Check function

IMSCMD
IMS Command function

RCU
RECON Cleanup function

RPCR
Recovery Point Create function

RPID
Recovery Point Identify function

VERIFY
Verify Recovery Assets function

Valid values for *process* are:

INIT
Function initialization

EXEC
Function execution

TERM
Function termination

Valid values for the *rc* are:

0
Successful

4
Warning encountered

8
Error, function failed

16
Severe error encountered

Valid values for *rc*, by *function* are:

HCHECK

01 – INVALID RANGE() SPECIFICATION

02 – INVALID RANGE() BEGIN TIME

03 – INVALID RANGE() END TIME

04 – END TIME LESS THAN BEGIN TIME

05 – MULTIPLE TIME PARAMETERS SPECIFIED

06 – HOURS() GREATER THAN MAXIMUM

07 – DAYS() GREATER THAN MAXIMUM

08 – INVALID DAYS() SPECIFIED

09 – HOURS() SPECIFIED

IMSCMD

01 – INVALID IMS COMMAND

02 – USER NOT AUTHORIZED FOR COMMAND

04 – NO ACTIVE IMS SYSTEMS

05 – ONE OR MORE SYSTEMS NOT ACTIVE

08 – QUERY ACTIVE SYSTEMS FAILED

12 – IMS COMMAND FAILED

16 – IMS COMMAND FAILED

RCU

01 – INVALID TIME SPECIFIED

08 - INTERIM LOG RECORDS EXIST

RPCR

01 – DBLIST() REQUIRED BUT NOT SPECIFIED

02 – IROPAPI INIT FAILED

03 – INVALID WAITTIME() SPECIFIED

04 – IROPAPI CONNECT FAILED

05 – IROPAPI CREATE FAILED

06 – IROPAPI DISCONNECT FAILED

RPID

01 – INVALID RANGE() SPECIFICATION

02 – INVALID RANGE() BEGIN TIME

03 – INVALID RANGE() END TIME

04 – END TIME LESS THAN BEGIN TIME

05 – DBLIST() REQUIRED BUT NOT SPECIFIED

VERIFY

01 – TIME FORMAT INVALID

02 – INVALID TIME SPECIFIED

03 – DBLIST() REQUIRED BUT NOT SPECIFIED

System action:
Processing continues.

System programmer response:
If the *return code* is non-zero, check the joblog for additional messages that indicate the error. After IMS systems are up, ensure that the proper TOSI initialization complete messages are displayed with the correct XCFGROUP name in all IMS control regions that share the same RECONs as the IMS Database Recovery Facility Extended Functions job. Also, ensure that the IMS Tools Knowledge Base RECONID (TOIXCF

field) matches the XCFGROUP=*name* referenced in the FOIssidP members for all IMS subsystems sharing the same RECONs.

Module:
IROMDRV0, IROBDIST, IROERCUT, IROHHCKT, IROIICMT, IROPRPCT, IRORRPIT, IROVVFYT

IRO0103E **ERROR IN FUNCTION *function* REASON: *reason***

Explanation

This message indicates that an error occurred during the processing of the specified *function*. Valid values for *function* are:

HCHECK
Health Check function

IMSCMD
IMS Command function

RCU
RECON Cleanup function

RPCR
Recovery Point Create function

RPID
Recovery Point Identity function

VERIFY
Verify Recovery Assets function

Valid values for *reason* are:

DBLIST() IS REQUIRED
The specified function requires a list of databases to be processed using the DBLIST() keyword. Ensure that the DBLIST command (and keywords) is specified before the RPID, VERIFY, HCHECK, or RPCR command.

For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing for the specified function terminates. Other functions may continue to process.

System programmer response:
Specify the list of databases to be processed using the DBLIST() keyword.

Module:
IROMDRV0

IRO0104E **INVALID COMMAND INPUT: *reason***

Explanation

This message indicates that an error occurred while pre-processing command input. Valid values for *reason* are:

RCU SPECIFIED MULTIPLE TIMES
The RCU function was specified more than once in the IMSREIN control cards. Only a single RCU function can be processed for each job execution.

RECONCOPY() NOT ALLOWED WITH RPCR()
The RPCR function was specified along with the RECONCOPY service. RPCR is mutually exclusive with RECONCOPY.

RECONCOPY() REQUIRED FOR RCU()
The RCU function was specified without specifying the RECONCOPY service. RCU requires that RECONCOPY be specified. Ensure that the RECONCOPY command (and keywords) is specified before the RCU command.

ALTRECONS() NOT ALLOWED WITH RECONCOPY()
Choose either the RECONCOPY service or use ALTRECONS to identify the RECONS data sets.

ALTRECONS() NOT ALLOWED WITH RPCR()
The RPCR function was specified along with the ALTRECONS service. RPCR is mutually exclusive with ALTRECONS.

ALTRECONS() CANNOT EQUAL ALLOCATED RECONS
ALTRECONS must not identify the currently allocated RECONS; otherwise, unwanted input/output (I/O) to the live RECONS would result.

System action:
Processing terminates.

System programmer response:
Correct the error and resubmit the job.

Module:
IROMDRV0

IRO0201E **TOI INIT|TERM SERVICE FAILURE RC=*return code*, RSN=*reason code***

Explanation:
This message indicates that The IMS Tools Online System Interface (TOSI) initialization or termination service has failed with the specified *return code* and *reason code*. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Ensure that the TOSI is correctly installed.

Module:
IROMSTOI

IRO0202I **TOSI INTERFACE INITIALIZATION COMPLETE**

Explanation:

This message indicates that The IMS Tools Online System Interface (TOSI) initialization process has successfully completed.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMSTOI

IRO0203I **TOSI INTERFACE TERMINATION COMPLETE**

Explanation:
This message indicates that The IMS Tools Online System Interface (TOSI) termination process has successfully completed.

System action:
Termination process continues.

System programmer response:
None. This message is informational.

Module:
IROMSTOI

IRO0251E **INIT|TERM SERVICE FAILURE RC=*return code*, RSN=*reason code***

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DBRC Interface (DRC) initialization or termination service has failed with the specified *return code* and *reason code*. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Ensure that the IMS Tools Online System Interface is correctly installed.

Module:
IROMSDRC

IRO0252I **DRC INTERFACE INITIALIZATION COMPLETE**

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DBRC Interface (DRC) initialization process has successfully completed.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMSDRC

IRO0253I **DRC INTERFACE TERMINATION COMPLETE**

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DBRC Interface (DRC) termination process has successfully completed.

System action:
Termination process continues.

System programmer response:
None. This message is informational.

Module:
IROMSDRC

IRO0301E **TDI INIT|TERM SERVICE FAILURE RC=*return code*, RSN=*reason code***

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DSPAPI Interface (TDI) initialization or termination service has failed with the specified *return code* and *reason code*. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Ensure that the IMS Tools Online System Interface is correctly installed.

Module:
IROMSTDI, IROMDBL0

IRO0302I **TDI INTERFACE INITIALIZATION COMPLETE**

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DSPAPI Interface (TDI) initialization process has successfully completed.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROMSTDI, IROMDBL0

IRO0303I **TDI INTERFACE TERMINATION COMPLETE**

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DSPAPI Interface (TDI) termination process has successfully completed.

System action:
Termination process continues.

System programmer response:
None. This message is informational.

Module:
IROMSTDI, IROMDBL0

IRO0304E TDI ERROR AT *location reason*

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions DSPAPI Interface (TDI) encountered an error during processing.

System action:
Termination process continues.

System programmer response:
Contact IBM software support.

Module:
IROMSTDI, IROMDBL0

IRO0400E REPORT WRITER ERROR, REASON: *reason*

Explanation:
This message indicates that an error occurred during report writer processing. The reason for the error is specified in the message. These are internal errors.

System action:
Processing terminates.

System programmer response:
Contact IBM software support.

Module:
IROCRPW0

IRO0401E ITKB ERROR, FUNCTION=*function*, RC=*return code*, RSN=*reason code*

Explanation:
This message indicates that an error occurred during report writer processing while accessing IMS Tools Base for z/OS IMS Tools Knowledge Base. The *function*, *return code*, and *reason code* are specified in the message. These are internal errors. For more information on the return and reason codes, see the *IBM Tools Base for z/OS: IMS Tools Knowledge Base User's Guide and Reference*.

System action:
Processing terminates.

System programmer response:
Contact IBM software support.

Module:
IROCRPW0

IRO0405E RECONID INTERFACE ERROR, FUNCTION: *function* RC=*return code*, RSN=*reason code*

Explanation:
This message indicates that an error occurred during the RECONID interface processing. The *function*, *return code*, and *reason code* are displayed. These are internal errors. For more information on the return and reason codes, see the *IBM Tools Base for z/OS: IMS Tools Knowledge Base User's Guide and Reference*.

System action:
Processing terminates.

System programmer response:
Contact IBM software support.

Module:
IROCRID0

IRO0406E RECONID INTERFACE ERROR, REASON: *reason*

Explanation

This message indicates that an error has occurred attempting to access the RECONID record. This interface uses the IMS Tools Base for z/OS IMS Tools Knowledge Base services. The possible reasons for the error are:

ITKB SERVER IS NOT ACTIVE
The IMS Tools Base for z/OS IMS Tools Knowledge Base server, specified by the ITKBSERV() parameter, is not active. Ensure that the ITKBSRVR used in the IMS Database Recovery Facility Extended Functions job has been brought up before executing IMS Database Recovery Facility Extended Functions.

For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Correct the error based on the reason and resubmit the job.

Module:
IROCRID0

IRO0426E ERROR IN KEYWORD: RANGE| TIME REASON: *reason*

Explanation:
This message indicates that an invalid date or timestamp was detected in the RANGE or TIME keyword. For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
Correct the keyword and resubmit the job.

Module:

IROCRNG0

IRO0427I RANGE|TIME *time*

Explanation:
This message indicates the time value specified by the RANGE or TIME keyword.

System action:
Processing continues.

System programmer response:
None. This message is informational.

Module:
IROCRNG0

IRO0900I SHUTDOWN SCHEDULED

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions is beginning termination processing.

System action:
Termination process continues.

System programmer response:
None. This message is informational.

Module:
IROMSREM

IRO1401E VERIFY PARAMETER ERROR IN KEYWORD: *keyword* REASON: *reason*

Explanation:
This message indicates that an invalid *keyword* was detected during the Verify function processing. The *reason* for the error is displayed. For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing for the Verify function terminates.

System programmer response:
Correct the error and rerun the job.

Module:
IROVVFYI

IRO1402E VERIFY COMMAND ERROR, REASON: *reason*

Explanation:
This message indicates that an error was detected in the VERIFY command processing. The *reason* for the error is displayed.

System action:
Processing for the Verify function terminates.

System programmer response:
Correct the error and resubmit the job.

Module:
IROVVFYI

IRO1403E LOAD FAILED FOR MODULE *module*, RC=*return code*

Explanation:
This message indicates that an attempt to load module *module* failed with return code *return code*. For more information on the return codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates.

System programmer response:
For an explanation of the return and reason code, refer to 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. Ensure that the listed module can be loaded from the JOBLIB.

Module:
IROVVFYP

IRO1407E VERIFY: IMS RE API ERROR IN MODULE *module* FUNCTION *function*. RC= *nnnnnnnn* RSN= *nnnnnnnn*

Explanation:
The indicated Verify function module encountered an error when calling the indicated function of the IMS Recovery Expert application programming interface.

System action:
Processing terminates.

System programmer response:
Save the SYSLOG information and send it to IBM Software Support.

Module:
IROVVFYP

IRO1450I DYNAMIC ALLOCATION FAILED FOR IC/CA/LOG, RC=*return code*, RSN=*reason code*, DSN=*data set name*

Explanation

The VERIFY function of IMS Database Recovery Facility Extended Functions failed to allocate an IC, a log, or a change accumulation data set indicated by data set name.

Return Code:
The return code set by Dynamic Allocation (SVC 99).

Reason Code:
The Reason Code consists of two parts. The first four characters are the contents of error reason

code (S99ERROR). The second four characters are the information reason code set (S99INFO).

Data set name:
The name of the data set that is to be allocated.

For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing continues for the remaining members in the IMS Database Recovery Facility Extended Functions verify list.

System programmer response:
For an explanation of the return and reason code, refer to 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 code and reason code.

Module:
IROVALL0, IROVIDY0

IRO1451I DYNAMIC DEALLOCATION FAILED FOR IC/CA/LOG, RC=*return code*, RSN=*reason code*, DSN=*data set name*

Explanation

This message indicates that the VERIFY function of IMS Database Recovery Facility Extended Functions failed to deallocate an IC, a log, or a change accumulation data set indicated by data set name.

Return Code:
The return code set by Dynamic Allocation (SVC 99).

Reason Code:
The Reason Code consists of two parts. The first four characters are the contents of error reason code (S99ERROR). The second four characters are the information reason code set (S99INFO).

Data set name:
The name of the data set that is to be deallocated.

For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing continues for the remaining members in the IMS Database Recovery Facility Extended Functions verify list.

System programmer response:
For an explanation of the return and reason code, refer to 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:
IROVALL0, IROVIDY0

IRO1452I REPORT PROCESSING FAILED: *reason*

Explanation

This message indicates that generation of the VERIFY reports failed for the indicated internal reasons.

- Invalid report input control block.
- Null dataset input control block.
- Report initialization failed.
- Time conversion routine failed.
- Load of data conversion module failed.

System action:
Processing continues. However, the generated report might be incomplete.

System programmer response:
All of the error situations are not user correctable. Contact the IBM Service Center and provide the return code information, if applicable.

Module:
IROVRPT0

IRO1453I NULL INPUT CONTROL BLOCK ENTERED FOR ALLOCATION

Explanation:
This message indicates that the VERIFY function is not able to allocate the data set because the address of control block that needs to be allocated is zero.

System action:
Processing continues for the remaining members in the IMS Database Recovery Facility Extended Functions Verify report list. However, the generated report might be incomplete.

System programmer response:
Check the input control block for allocation.

Module:
IROVRPT0

IRO1454I UNABLE TO GET *storage type* FOR OPEN, RC=*return code*

Explanation:
This message indicates that a request to get storage could not be satisfied. In the message, *storage type* identifies what the storage request was for. For more information on the return codes, see the *IMS System Programming API Reference*.

System action:
Deallocates the data set.

System programmer response:

Check the storage size and correct the reason for the failure.

Module:
IROVALL0

IRO1455I **OPEN FAILED FOR IC/CA/LOG, RC=*return code*, DSN=*data set name***

Explanation:
This message indicates that the VERIFY function of IMS Database Recovery Facility Extended Functions failed to open an IC, a log, or a change accumulation data set indicated by *data set name*. For more information on the return codes, see the *IMS System Programming API Reference*.

System action:
Processing continues for the remaining members in the Verify list.

System programmer response:
For additional information on the return code, see the *z/OS MVS Programming: Authorized Assembler Services Guide*.

Module:
IROVALL0, IROVVFYP

IRO1456I **BAD DCB BLOCK SIZE FOR OPEN, RC=*return code***

Explanation:
This message indicates that IMS Database Recovery Facility Extended Functions is unable to set values for obtaining fixed storage. Register 14 contains the return code from the set storage service. For more information on the return codes, see the *IMS System Programming API Reference*.

System action:
Processing continues for the remaining members in the Verify list.

System programmer response:
Check block size.

Module:
IROVALL0

IRO1501E **HCHECK PARAMETER ERROR IN KEYWORD: *keyword* REASON: *reason***

Explanation

This message indicates that a required parameter was not coded or a duplicate *keyword* was found during HCHECK control card syntax validation. The *reason* variable indicates one of the following values:

HOURS(), DAYS(), RANGE() parameter required.

NUMBER() parameter required.

Duplicate instance of *keyword* found.

For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing terminates with a return code 16.

System programmer response:
Correct the job control statements and resubmit the job.

Module:
IROHHCKV

IRO1502E **HCHECK PARAMETER ERROR: *parameter***

Explanation:
This message indicates that an unsupported or invalid parameter was found during HCHECK control card validation. The *parameter* indicates the keyword or parameter in error.

System action:
Processing terminates with a return code 16.

System programmer response:
Correct the control card syntax and resubmit the job.

Module:
IROHHCKV

IRO1504E **DBRC *function* FAILED, RC=*return code*, RSN=*reason code***

Explanation:
This message indicates that a DSPAPI call to DBRC failed for the function specified in the message. The *return code* and *reason code* are documented in the "DBRC Application Programming Interfaces" section in the *IMS Database Recovery Control (DBRC) Guide and Reference*.

System action:
Processing terminates with a return code 16.

System programmer response:
This is an internal error. Contact IBM Software Support.

Module:
IROHHCKD

IRO1505E **LOAD FAILED FOR MODULE *module***

Explanation:
This message indicates that the requested *module* could not be loaded.

System action:
Processing terminates with a return code 16.

System programmer response:
Correct the JOBLIB/STEPLIB concatenation and resubmit the job.

Module:
IROHHCKP

IRO1506E DSPAPI Unexpected Data Returned, Found=*found*, Expected=*expected*

Explanation:
During a QUERY call to DSPAPI, and unexpected data buffer was returned. The *expected* and *returned* data fields are printed for diagnosis.

System action:
Processing terminates with a return code 16.

System programmer response:
This is an internal error. Contact IBM Software Support.

Module:
IROHHCAG, IROHHDBL, IROHHLOG

IRO1507E HCHECK: IMS RE API ERROR IN MODULE *module* FUNCTION *function*. RC= *nnnnnnnn* RSN= *nnnnnnnn*

Explanation:
The indicated Health Check function module encountered an error when calling the indicated function of the IMS Recovery Expert application programming interface.

System action:
Processing terminates.

System programmer response:
Save the SYSLOG information and send it to IBM Software Support.

Module:
IROHHCKD, IROHHDBL, IROHHDBV

IRO2000E RCU PARAMETER ERROR IN KEYWORD: *keyword* REASON: *reason*

Explanation:
This message indicates that an invalid keyword was detected during the RECON Cleanup function processing. The reason for the error is displayed. For more information on the return, see the *IMS System Programming API Reference*.

System action:
Processing for the RCU function terminates.

System programmer response:
Correct the error and resubmit the job.

Module:
IROERCUI

IRO2001E LOAD FAILED FOR MODULE *modname* RC=*nnnn*

Explanation:
This message indicates that RCU encountered an error loading the module *modname*. The z/OS LOAD macro completed with return code *nnnn*.

System action:
RCU terminates and clean up is not performed.

System programmer response:
Ensure the IMS reslib is included in the STEPLIB.

Module:
IROERCUP

IRO2002E RCU CAN NOT COMPLETE - INTERIM LOG RECORDS EXIST

Explanation:
This message indicates that RCU has determined that there are interim log records in the RECON to be cleaned up that will remain after clean up completes. Interim log records are created when running the log recovery utility and are removed when the utility completes. Because RCU may remove information necessary to complete log recovery, clean up is not performed. The interim log records are reported in the RCU Clean Up report.

System action:
RCU completes with return code 8, reason code C3100002. Clean up is not performed.

System programmer response:
Determine if these interim log records are needed. Either complete the log recovery utility or remove the interim log records using the DBRC DELETE.LOG command.

Module:
IROERC00

IRO2004E RECORD POINTS TO NON-EXISTENT LOG RECORD FOR DB=*dbname* DD=*ddname*

Explanation:
The RECON cleanup (RCU) function has found an allocation record for DB *dbname*, DD *ddname* that points to a nonexistent log record.

System action:
The RCU function terminates.

User response:
Review the job output and look for the associated IRO2005E messages that indicate the allocation and start time of the nonexistent log record. Remove the nonexistent log record and rerun the RCU function.

IRO2005E *type* TIME=*time*

Explanation

This message indicates the allocation record or log start time of the nonexistent log record that is shown in message IRO2004E. The *type* indicates:

- The *time* shown is the allocation record time.
- The *time* shown is the log start time.

System action:
The RECON cleanup (RCU) function terminates.

User response:
Review the job output and look for the associated IRO2004E and IRO2005E messages which indicate the allocation and start time of the nonexistent log record. Remove the nonexistent log record and rerun the RCU function.

IRO3001E IMS COMMAND ERROR: *reason*

Explanation:
An invalid IMS command was specified.

System action:
Processing for the command terminates.

System programmer response:
Correct the command and resubmit the job.

Module:
IROIICMI

IRO3002E IMS COMMAND FAILED WITH RC=*return code*, RSN=*reason code*, ERROR=*error*

Explanation:
This message indicates that an error occurred attempting to process the IMS command. The *return code* and *reason code* from the IMS Tools Online System Interface component as well as the error location are displayed. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

System action:
Processing for the command terminates.

System programmer response:
Determine the reason for the error by looking up the return and reason codes in the IMS Tools Online System Interface messages and codes and correct the error. After IMS systems are up, ensure that the proper TOSI initialization complete messages are displayed with the correct XCFGROUP name in all IMS control regions that share the same RECONs as the IMS Database Recovery Facility Extended Functions job. Also, ensure that the IMS Tools Base for z/OS IMS Tools Knowledge Base RECONID (TOIXCF field) matches the XCFGROUP=*name* referenced in the FOIssidP members for all IMS subsystems sharing the same RECONs.

Module:
IROIICMP

IRO3003I IMS COMMAND NOT ISSUED, IMS SYSTEM *imsid* IS NOT ACTIVE

Explanation:
This message indicates that a request to run an IMS command against IMS system *imsid* failed because the system was either not active or not a part of the IMS Tools Online System Interface XCF group.

System action:
The command is not issued.

System programmer response:
Ensure the correct IMS *imsid* was specified. If so, ensure that the correct XCF group was specified for issuing IMS commands.

Module:
IROIICMP

IRO3005W SAF AUTH SERVICE FAILURE SAF-RC=*saf-rc*, RACF-RC=*racf-rc*, RACF-RSN=*racf-rsn*

Explanation:
This message indicates that authorization for an IMS command failed with the specified SAF and RACF return and reason codes.

System action:
The IMS command is rejected.

System programmer response:
The return and reason codes shown can be found in the "RACROUTE REQUEST=AUTH" section in the *z/OS MVS Programming: Authorized Assembler Services Guide*.

Module:
IROIICMI

IRO3006W IMS COMMAND SECURITY FAILED, USERID=*userid*, COMMAND=*command*

Explanation:
This message indicates that the specified *userid* is not authorized to issue the specified IMS *command*.

System action:
The IMS command is rejected.

System programmer response:
Contact the security administrator to provide access for the userid to the IMS command.

Module:
IROIICMI

IRO3007E IMS COMMAND NOT SUPPORTED, COMMAND=*command*

Explanation:

This message indicates that the specified IMS command is not supported by the IMS Database Recovery Facility Extended Functions IMS Command interface.

System action:
The IMS command is rejected.

System programmer response:
None.

Module:
IROIICMI

IRO4006E **IMS RE API ERROR IN MODULE IROSENSR FUNCTION *function*.**

Explanation:
Module IROSENSR encountered an error when calling the indicated function of the IMS Recovery Expert application programming interface.

System action:
Processing terminates.

System programmer response:
Save the SYSLOG information (including the BSYTRACE SYSOUT data set) and send it to IBM Software Support.

Module:
IROSENSR

IRO7001E **RPID PARAMETER ERROR IN KEYWORD: *keyword* REASON: *reason***

Explanation

This message indicates that An error was found in a *keyword* of the RPID control statement.

keyword identifies the specific keyword in error:

RANGE
Keyword RANGE

reason identifies the reason for the error in keyword:

REQUIRED PARAMETER
RANGE is a required keyword parameter.

INVALID FORMAT
RANGE keyword has invalid format.

For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:
Processing continues with return code 8 and reason code 0.

System programmer response:
None.

Module:
IRORRPII

IRO7002E **RPID COMMAND ERROR, REASON: *reason***

Explanation

This message indicates that an error was found in the RPID command control statement. For more information on the reason codes, see the *IMS System Programming API Reference*.

reason identifies the reason for the error in the command:

(NODBLST)
There was no database list to process based on the DBLIST command.

System action:
Processing continues with return code 8 and reason code 0.

System programmer response:
None.

Module:
IRORRPII

IRO7003E **RPID CALL TO DSPAPI FAILED, FUNCTION: *function*, RC= *return code*, RSN= *reason code***

Explanation

This message indicates that Recovery Point Identification has made a call to the DBRC API DSPAPI and that call has resulted in a failing *return code*. For more information on the return and reason codes, see the *IMS System Programming API Reference*.

function identifies the specific DSPAPI function call values:

STARTDBRC
START the DSPAPI API

QUERYDB
Database Query

System action:
Processing continues with return code 8 and reason code 0.

System programmer response:
None.

Module:
IRORRPIP

IRO7004I **RPID FOUND NO IN COMMON RECOVERY TIME SPANS FOR THE DBLIST**

Explanation:
This message indicates that Recovery Point Identification found no Recovery Time Spans in

common for all the databases specified on the DBLIST control card read from IMSREIN.

System action:
Processing continues and ends normally with a return code 4 and reason 0.

System programmer response:
None. This message is informational.

Module:
IRORRPIP

IRO7005E **RPID QUERY LOG FAILED FOR DB=*dbname* DD=*ddname***

Explanation:
The Recovery Point ID (RPID) function found an allocation record for the database *dbname*, DD *ddname* that points to a nonexistent log record.

System action:
The RPID function terminates.

User response:
Review the job output and look for the associated IRO7006E messages that indicate the allocation and start time of the nonexistent log record. Remove the nonexistent log record and rerun the RPID function.

IRO7006E ***type* TIME=*time***

Explanation

This message indicates the allocation record or log start time of the nonexistent log record that is shown in message IRO7005E. The *type* indicates:

- The *time* shown is the allocation record time.
- The *time* shown is the log start time.

System action:
The Recovery Point ID (RPID) function terminates.

User response:
Review the job output and look for the associated IRO7005E and IRO7006E messages which indicate the allocation and start time of the nonexistent log record. Remove the nonexistent log record and rerun the RPID function.

IRO3008E **IMS COMMAND SECURITY NOT SET, IMS COMMAND INTERFACE IS NOT ACTIVE**

Explanation:
This message indicates that the IMS Database Recovery Facility Extended Functions IMS command interface is not active because the security information has not been specified in the RECONID record.

System action:
All IMS commands are rejected.

System programmer response:
Update the RECONID record and specify the desired IMS command security settings.

Module:
IROIICMI

IRO7301E **RPCR *phase* FAILED, ERROR: *error***

Explanation

This message indicates that an error was detected in the Recovery Point Create command processing during the phase listed. The reason for the error is displayed. Valid values for error are:

DBLIST() IS MISSING OR INVALID
The specified function requires a list of databases to be processed using the DBLIST() keyword.

System action:
Processing for the RPCR function terminates.

System programmer response:
Correct the error and resubmit the job.

Module:
IROPRPCI

IRO7302E **RPCR *phase status*, RC=*return code*, RSN=*reason code*, ERROR: *error***

Explanation

This message indicates that an error was detected in the Recovery Point Create command processing during the phase listed. The *phase status*, *return code*, *reason code* and *error* are displayed in the message. For more information on the return and reason codes, see the *IMS System Programming API Reference*. Valid values for *status* are:

WARNING
RPCR encountered a warning condition. Some RPCR processes were successful while others may have failed. This is specified when RC=4.

FAILED
RPCR encountered an error which has prevented RPCR from completing successfully. This is specified when RC=8.

ERROR
RPCR encountered a severe error which has prevented RPCR from completing successfully. This is specified when RC=10.

TRACE
RPCR encountered an error situation and has created trace entries in the TRACE report which can be used to diagnose the problem. This is specified when RC=14.

When the return code is 04, 08 or 10, the reason for the error will be shown in the error field. The following table shows the reason code and the associated error value. When the return code is 14, then an error was encountered and internal trace entries were created to assist in determining the cause for the error. In this case, additional messages will also be issued indicating the error that occurred.

The following are valid values for *error*:

Table 8. Valid values for error

Reason code	Error	Description
00000032	INVALID TOKEN FOR FUNCTION	Invalid token passed to RPCR. Contact IBM Software Support.
00000033	TOKEN NOT FOR SAME TCB	Invalid token passed to RPCR. Contact IBM Software Support.
00000034	UNABLE TO JOIN XCF GROUP	An attempt to join the XCF group for TOSI has failed. Ensure that the correct XCF group is specified in the RECONID record.
00000064	TARGET SYSTEM NOT AVAILABLE	An attempt to issue an IMS command against one or more target IMS systems has failed. Ensure that the IMS system is still active.
00000065	UNABLE TO OBTAIN RECON INFO	RPCR was unable to access the RECON data sets to obtain RECON information. Review the joblog for additional messages indicating the specific error.
00000066	UNABLE TO GET ONLINE DB STATUS	RPCR was unable to access the RECON data sets to obtain database information. Review the joblog for additional messages indicating the specific error.
00000067	UNABLE TO SWITCH LOG	RPCR was unable to complete the /SWI OLDS command processing. Review the joblog and the IMS messages for additional messages indicating the specific error.
00000068	UNABLE TO DBR A DB	RPCR was unable to DBR one or more databases. Review the joblog and the IMS messages for additional messages indicating the specific error.
00000069	UNABLE TO DBD A DB	RPCR was unable to DBD one or more databases. Review the joblog and the IMS messages for additional messages indicating the specific error.
0000006A	UNABLE TO START DB	RPCR was unable to start one or more databases. Review the joblog and the IMS messages for additional messages indicating the specific error.
0000006B	OPTIONS CONFLICT WITH DB TYPE	The RPCR options specified conflict with database type. Contact IBM software support.
0000006C	GETMAIN REQUEST FAILED	Unable to acquire storage. Ensure that the region size for the Extended Functions job is adequate.
0000006D	NAME TOKEN SERVICES FAILED	An error was encountered using MVS name token services. Contact IBM Software Support.
0000006E	UNABLE TO LOAD PROGRAM	RPCR was unable to load a required program. Ensure that Extended Functions has been correctly installed and all required load modules are include in the JOBLIB or STEPLIB.
0000006F	DYNAMIC ALLOCATION FAILED	Dynamic allocation failed for one or more database data sets. Review the joblog for details on the specific data set. This normally indicates that some job or user outside of IMS has the data set in use.
00000070	UNABLE TO LOCATE SUBTASK SVCS	Unable to locate subtask services. Contact IBM Software Support.
00000071	SVCS NOT AVAILABLE FOR SUBTASK	Subtask services are not available. Contact IBM Software Support.
00000072	ERROR STARTING SERVICE SUBTASK	Unable to start the subtask services. Contact IBM Software Support.

Table 8. Valid values for error (continued)

Reason code	Error	Description
00000073	IMS ENVIRONMENTAL ERROR	RPCR encountered an error within the IMS environment. Contact IBM Software Support.
00000074	NOT APF AUTHORIZED	The RPCR program is not running in an APF authorized environment. Ensure that the load libraries in the JOBLIB or STEPLIB are APF authorized.
00000075	MISSING IMS DD	The specific RPCR options specified require the IMS DD to be specified. Add the appropriate IMS DD concatenation.
00000076	INCORRECT DSNAME FOR RECON	The data set name associated with the RECON being accessed does not match the data set name passed to RPCR. Ensure that the same RECON data sets are accessed by the IMS Database Recovery Facility Extended Functions job and the online IMS system.
00000077	DB IN USE FOR TIMEOUT DURATION	One or more databases were in use for the timeout duration and RPCR failed. If PARTIAL(Y) was specified, some databases may have succeeded. If PARTIAL(N) was specified, then RPCR failed.
00000078	DB NOT PROCESSED, PARTIAL(N)	One or more databases were in use for the timeout duration and PARTIAL(N) was specified. All other databases in the list have been skipped.
00000096	DB IS NOT REGISTERED	One or more databases passed to RPCR were not registered to DBRC. Ensure all databases are registered.
000000C8	DB TYPE IS UNSUPPORTED	The database type is either unsupported by RPCR or cannot be determined.
000000C9	UNABLE TO LOCATE DBD	RPCR could not locate the DBD in the IMS DD concatenation. Ensure the correct blocks are located in the IMS DD concatenation.
000000CA	UNABLE TO LOCATE PART INFO	RPCR could not locate information for a HALDB partition. Ensure the correct blocks are located in the IMS DD concatenation.
000000CB	DUPLICATE DBD/PART IN DBLIST	A duplicate database or partition was passed to RPCR in the DBLIST. Contact IBM.
000000CC	ALLOCATED AFTER DBR	RPCR has determined that one or more databases processed was allocated after it was DBR'd. The RPCR process must fail because it cannot ensure that a correct recovery point was created.
000000CD	CMDTYPE(DBD) NOT VALID FOR DB	This is an internal error. Contact IBM Software Support.
000000CE	DB TYPE NOT PROVIDED	This is an internal error. Contact IBM Software Support.
000000CF	DB DSLIST NOT PROVIDED	This is an internal error. Contact IBM Software Support.
000000D0	DB IS CURRENTLY BEING USED	One or more databases were in use for the timeout duration and failed RPCR.

System action:
If the status is WARNING, then RPCR processing may have worked for some databases and failed for others. Otherwise, RPCR processing terminates.

System programmer response:
Review the RPTRPCR report and other messages in the joblog to determine the outcome of RPCR.

Module:
IROPRPCI, IROPRPCP

IRO7304E **RPCR PARAMETER ERROR IN KEYWORD: *keyword*, REASON: *reason***

Explanation

This message indicates that an error was detected in the specified keyword on the RPCR command. Currently, valid combinations of *keyword* and *reason* include:

KEYWORD: WAITTIME, REASON: VALUE EXCEEDS MAXIMIMUM ALLOWED

The value specified for WAITTIME exceeds the maximum allowed value. Refer to the documentation on the RPCR function for valid values.

For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:

The function is rejected.

System programmer response:

Correct the value for the specified keyword and resubmit the job.

Module:

IROPRPCI

IRO7305E RPCR FAILED FOR ALL DATABASES, REASON: *reason*

Explanation

This message indicates that the RPCR function has failed to create a recovery point for all databases specified in the DBLIST(). The reasons for this failure include:

DB IN USE FOR TIMEOUT DURATION

Either all databases specified in the DBLIST() were in use for the entire WAITTIME() duration or PARTIAL(NO) was specified and one or more of the databases were in use.

For more information on the reason codes, see the *IMS System Programming API Reference*.

System action:

The function is rejected.

System programmer response:

Correct the problem associated with the indicated reason and resubmit the job.

Module:

IROPRPCP

IRO7320E APPLICATION ABEND *code*

Explanation:

This message is issued by the RPCR BMP pause interface when the application program being paused ends abnormally. Normally, the abend code will be a U3303 associated with the pseudo-abend when the database is unavailable.

System action:

If the abend code is 3303, the application program waits for the database to become available.

System programmer response:

None.

Module:

IROPRCNN

Chapter 14. Abend codes

This reference section provides detailed information about IMS Database Recovery Facility Extended Functions abend codes.

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.

0200

Explanation

An error occurred in the use of a BPE service. The abend subcode describes the nature of the error:

X'01'
AWE get failed

X'05'
AWE enqueue failed

X'0A'
BPETIMER initialization failed

X'0F'
BPETIMER cancel failed

X'14'
BPEATTCH failed

X'19'
BPELTCB failed

X'1E'
BPELOADC failed

X'23'
BPEPOST failed

X'64'
BPETERM failed

X'69'
BPELAGET failed

X'6E'
BPELAREL failed

X'73'
BPESPRNT failed

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Retain the diagnostic information and contact IBM Software Support.

0030

Explanation

An error occurred during IMS Database Recovery Facility Extended Functions address space initialization. The abend subcode describes the nature of the error:

X'01'
Parser failed

X'02'
Configuration parameter error

X'03'
Load failed for a module

X'04'
RECONID locate failed

X'05'
RECONCOPY service failed

X'06'
Locate RECON process failed

X'07'
Allocate RECON process failed

X'09'
IMS Tools Knowledge Base failed

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Additional error messages are issued before the abend. Review these messages and take the appropriate action. If the problems persists, retain the diagnostic information and contact IBM Software Support.

0050

Explanation

An error occurred during the use of an internal IMS Database Recovery Facility Extended Functions service. The abend subcode describes the nature of the error:

X'01'
Internal logic error

X'02'
Open for DD failed

X'03'
CPOOL failed

X'04'
TCB initialization failed

X'05'
TCB termination failed

X'06'
Dynamic allocation failed

System action:
IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Additional error messages are issued before the abend. Review these messages and take the appropriate action. If the problems persists, retain the diagnostic information and contact IBM Software Support.

0060

Explanation

An error occurred during an attempt to acquire storage. The abend subcode describes the nature of the error:

X'01'
Error acquiring SABL control block

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Increase the REGION= size on the job and run the job again. If the problem persists, retain the diagnostic information and contact IBM Software Support.

0070

Explanation

An error occurred during IMS Tools Online System Interface processing. The abend subcode describes the nature of the error:

X'01'
FUNC=INIT failed

X'02'
FUNC=CONNECT failed

X'03'
FUNC=REQUEST failed

X'04'
FUNC=RESPONSE failed

X'05'
FUNC=TERM failed

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Additional error messages are issued before the abend. Review these messages and take the appropriate action. If the problem persists, retain the diagnostic information and contact IBM Software Support.

0080

Explanation

An error occurred during DBRC Interface processing. The abend subcode describes the nature of the error:

X'01'
FUNC=INIT failed

X'02'
FUNC=TERM failed

X'65'
DSPAPI FUNC=INIT failed

X'66'
DSPAPI FUNC=TERM failed

X'67'
DSPAPI FUNC=QUERY failed

X'68'
DSPAPI logic error

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Additional error messages are issued before the abend. Review these messages and take the appropriate action. If the problem persists, retain the diagnostic information and contact IBM Software Support.

0090

Explanation

An error occurred during report writer processing. The abend subcode describes the nature of the error:

X'01'
RWBLOCK is missing or invalid

X'02'
FUNC=INIT failed

X'03'
FUNC=CLOSE failed

X'04'
FUNC=OPEN failed

X'05'
FUNC=TERM failed

X'06'
FUNC=WRITE failed

X'07'
CPOOL failed

X'08'
FUNC=NEWPAGE failed

X'09'
FUNC=PRTHEAD failed

X'0A'
FUNC=REPSEP failed

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response:
Additional error messages are issued before the abend. Review these messages and take the appropriate action. If the problem persists, retain the diagnostic information and contact IBM Software Support.

0800

Explanation

An error occurred during IMS Recovery Expert API processing for System Level Backups. The abend subcode describes the nature of the error:

X'01'
FUNC=INIT failed

X'02'
FUNC=CONNECT failed

X'03'
FUNC=QUERY failed

X'04'
FUNC=TEST failed

X'05'
FUNC=PREPARE failed

X'06'
FUNC=DISCONNECT failed

X'07'
FUNC=TERMINATE failed

System action:
The IMS Database Recovery Facility Extended Functions address space is abnormally terminated.

User response

Additional error messages are issued before the abend. Review these messages and take the appropriate action.

If the problem persists, retain the diagnostic information and contact IBM Software Support.

Chapter 15. Gathering diagnostic information

Before you report a problem with IMS Database Recovery Facility Extended Functions to IBM Software Support, you need to gather the appropriate diagnostic information.

Procedure

Provide the following information for all IMS Database Recovery Facility Extended Functions 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 6. Reference: BPE

The topics in this section provide you with technical references for the Base Primitive Environment (BPE).

Topics:

Chapter 16. 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 151
- "BPE TRACETABLE commands" on page 151

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:

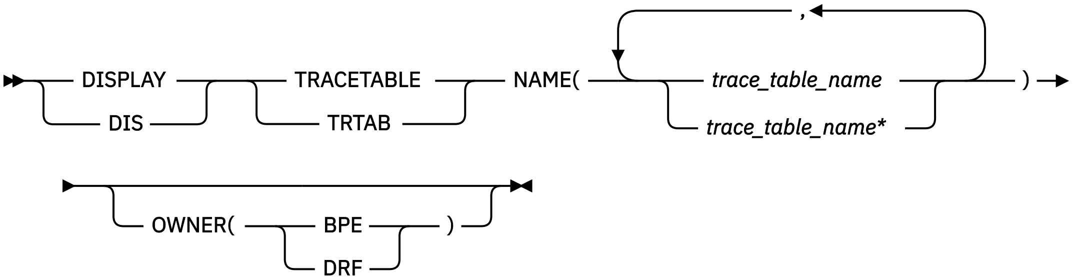
DISPLAY
Display trace level and number of trace table pages of specified trace tables.

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:



The following list describes the parameters of DISPLAY TRACETABLE:

DISPLAY | DIS

A required parameter that specifies that the action against the specified resource is to display attributes of the resource.

TRACETABLE | TRTAB

A required parameter that specifies that the resource type being acted upon is a BPE-managed trace table.

NAME(*trace_table_name*)

A required parameter that specifies the name of the trace table type or types about which you want attributes displayed. You can specify a single trace table name or a list of trace table names separated by commas. Trace table names can contain the wildcard characters * and ?. For example, `NAME(c*)` is equivalent to `NAME(CMD,CBS)`. 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

The following IMS Database Recovery Facility-defined trace table types are available:

INIT

IMS Database Recovery Facility initialization

DRF

IMS Database Recovery Facility

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.

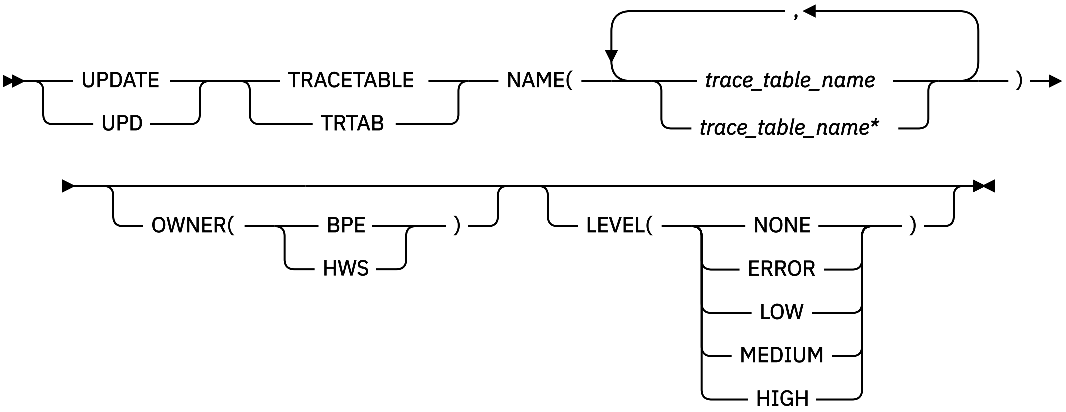
#PAGES

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 17. 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

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.

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

FRXAWI0$, FRXCBI0$, FRXCMDI$, FRXDSI0$, FRXHTI0$, FRXINIT$, FRXPCFG$, FRXRV1I$, FRXSTI0$, FRXTRI0$, FRXUXI0$, FRXXMOD$, FRXXSYS$

BPE0002E **ERROR READING *ddname* MEMBER *member* *details***

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$

BPE0003E **AN ERROR OCCURRED PARSING *description* AT LINE *line*, CHARACTER *char* FAILING TEXT: *text* *details***

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 *cdddd-eeeeeeee*, THD = *ffff retrystatus***
MODULE ID = *gggggggggggggggg*
EP = *hhhhhhhh*
PSW = *iiiiiiii iiiiiiii* OFFSET = *jjjjjjjj*
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 ended abnormally (BPE or FRX).

bbbb
The 4-character TCB type of the TCB that ended abnormally.

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.

eeeeeeee
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.

iiiiiiii iiiiiiii
: The PSW contents at the time of abend.

jjjjjjjj
: 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 ends abnormally 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 ends abnormally 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$

BPE0017I **MULTIPLE *type* STATEMENTS IN *member* PROCLIB MEMBER - LAST WILL BE USED**

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

FRXTRI0$, FRXUXRF$

BPE0018I **DUPLICATE EXIT *module* SPECIFIED FOR *component type* EXIT - IGNORED**

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$

BPE0019E ***component type* USER EXIT MODULE *module* ABEND *code***

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 ended abnormally.

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 ended abnormally.

module
The load module name of the exit that ended abnormally.

code
The abend code. For system abends, the format of code is S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal. For user abends, the format of code is U*dddd*, 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 ends abnormally, 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 ends abnormally 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$

BPE0020I **ABEND LIMIT REACHED FOR *component type* USER EXIT MODULE *module***

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 ended abnormally

module
The load module name of the exit that ended abnormally

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$

BPE0021E **ABEND *code* IN BPE SVC INIT MODULE FRXSVCI$, PSW=*psw1 psw2***

Explanation

An abend occurred while module FRXSVCI$ was in control. Module FRXSVCI$ is the module that initializes the IMS Database Recovery Facility BPE SVC routine (an internal SVC used by BPE and IMS Database Recovery Facility). FRXSVCI$ 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 FRXSVCI$ 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 S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal. For user abends, the format of code is U*dddd*, 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 FRXSVCI$. This routine attempts to release any global resources that FRXSVCI$ obtained as a part of its processing. The routine returns to the FRXSVCI$'s caller, which can choose to continue processing or to end abnormally.

The first time that FRXSVCI$ ends abnormally, 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 FRDVCI$ ends abnormally 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

FRXSVCI$

BPE0022E **ABEND *code* IN BPE SVC PROCESSING, PSW=*psw1 psw2***

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 S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal. For user abends, the format of code is U*dddd*, 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$ ends abnormally, 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$ ends abnormally 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$

BPE0023I ***command* COMMAND REJECTED**

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$, FRXCMD1$

BPE0024E ***command*** **COMMAND FAILED**

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:

- **BPESVC**: 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 S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal. For user abends, the format of code is U*dddd*, 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$ ends abnormally, 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$ ends abnormally 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$

BPE0028I **SDUMP FAILED FOR *abend* ABEND, RC=*rc*, RSN=*rsn***

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 S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal. For user abends, the format of code is U*dddd*, 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$, FRXRSM0$, FRXSVCI$, FRXSVC0$, FRXSYES$

BPE0029I **DAE SUPPRESSED DUMP FOR *abend* ABEND**

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 S*xxx*, where *xxx* is the 3-digit abend code in hexadecimal. For user abends, the format of code is U*dddd*, 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$, FRXRSM0$, FRXSVCI$, FRXSVC0$, 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 *resourcetype* 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:

resourcetype
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$, FRXCURF$

BPE0038E **BPE STACK MANAGER INTERNAL ERROR**

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$

BPE0039E **BPE STACK MANAGER MULTIPLE INTERNAL ERRORS**

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$

BPE0041E **UNABLE TO ALLOCATE REQUESTED STORAGE**

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.

BPERDPDS

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 STIMERM 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)

FRXSVCI$

Routing code

Initialize the BPE SVC module.

Return Code
Explanation

X'00000004'
Unable to load init routine FRXSVCI$ to initialize the BPE SVC.

X'00000008'
ESTAE create failed for ESTAE for FRXSVCI$. 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 the *IMS Messages and Codes* manuals.

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 end abnormally with a code other than 4095 in order to determine the cause of the subsequent 4095 abends.

Part 7. Reference

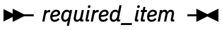
Reference information supports the tasks that you must complete to install, customize, and use IMS Database Recovery Facility Extended Functions.

Topics:

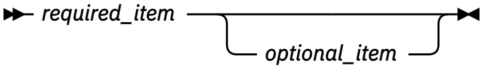
Chapter 18. How to read syntax diagrams

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

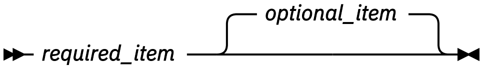
- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >>--- symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >--- symbol 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).



- Optional items appear below the main path.

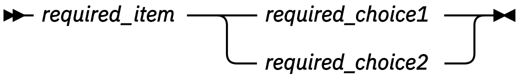


If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

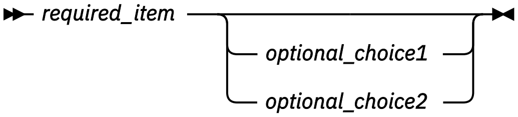


- If you can choose from two or more items, they appear vertically, in a stack.

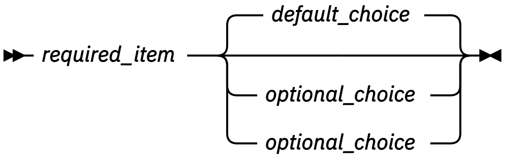
If you *must* choose one of the items, one item of the stack appears on the main path.



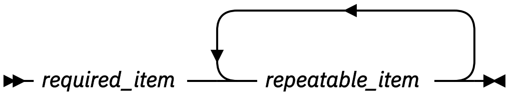
If choosing one of the items is optional, the entire stack appears below the main path.



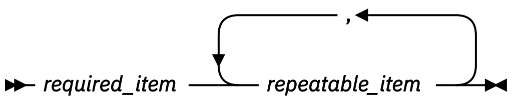
If one of the items is the default, it appears above the main path, and the remaining choices are shown below.



- An arrow returning to the left, above the main line, indicates an item that can be repeated.

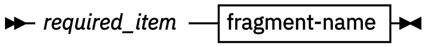


If the repeat arrow contains a comma, you must separate repeated items with a comma.

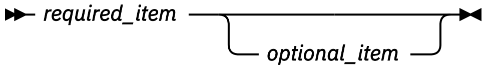


A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.



fragment-name



- A b symbol indicates one blank position.
- 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® 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

Product Number: 5655-ISR

SC27-8442-02