Oracle® CODASYL DBMS for OpenVMS

Release Notes

Release 7.4.1.5 for OpenVMS Alpha and
OpenVMS Industry Standard 64 for Integrity Servers

April 2026

ORACLE®

Oracle CODASYL DBMS Release Notes, Release 7.4.1.5 for OpenVMS Alpha and OpenVMS Industry Standard 64 for Integrity Servers

Oracle Corporation - Worldwide Headquarters, 2300 Oracle Way, Austin, TX 78741, United States

Primary Author: DBMS Engineering and Documentation group

Contents

3 Problems Corrected

4 Known Problems, Restrictions and Additional Documentation

Tables

Preface

Purpose of This Manual

This manual contains release notes for Oracle CODASYL DBMS release 7.4.1.5. The notes describe changed and enhanced features, upgrade and compatibility information, new and existing software problems and restrictions, and additional documentation.

Intended Audience

This manual is intended for use by all Oracle CODASYL DBMS users. Read this manual before you install, upgrade, or use Oracle CODASYL DBMS release 7.4.1.5.

Document Structure

This manual consists of the following chapters:

Chapter 1	Describes how to install Oracle CODASYL DBMS release 7.4.1.5.
Chapter 2	Describes new and changed features in Oracle CODASYL DBMS release 7.4.1.5.
Chapter 3	Describes problems fixed in Oracle CODASYL DBMS release 7.4.1.5.
Chapter 4	Describes problems, restrictions, and workarounds known to exist in Oracle CODASYL DBMS release 7.4.1.5.

Conventions

Oracle CODASYL DBMS is often referred to as DBMS in this manual.

OpenVMS Industry Standard 64 for Integrity Servers is often referred to as OpenVMS IA-64.

OpenVMS refers to both OpenVMS Alpha and OpenVMS IA-64.

1
Installing Oracle CODASYL DBMS

This chapter discusses installing Oracle CODASYL DBMS release 7.4 on OpenVMS Alpha and OpenVMS Industry Standard 64 for Integrity Servers.

All Oracle CODASYL DBMS release 7.4 kits are full kits. There is no requirement to install any prior release of Oracle CODASYL DBMS prior to installing this release.

1.1 Software Notes

1.1.1 Cross Platform Support

Oracle CODASYL DBMS is available on the OpenVMS Industry Standard 64 for Integrity Servers platform and the OpenVMS Alpha platform. In general, the functionality for one platform is available on the other platform. However, certain differences between the platforms may result in minor capability and functionality differences which will be clearly explained in these release notes.

The database format for Oracle CODASYL DBMS release 7.4 is the same on both I64 and Alpha platforms and databases may be accessed simultaneously from both architectures in a cluster environment.

This release provides a full set of Oracle CODASYL DBMS functionality for both platforms, including local and remote database access, as well as native DML and DDL operations.

Since the Oracle CODASYL DBMS database format is the same across all supported platforms, you can, for example, back up an Oracle CODASYL DBMS database on an Alpha system, then restore it on an I64 system (the reverse is also true). DBO will implicitly forward convert those backup files to the currently running version if required.

With remote access, you can bind to an Oracle CODASYL DBMS database on an Alpha system from an I64 system, or vice versa, as long as the appropriate Oracle CODASYL DBMS software is available on both platforms.

Additionally, if your environment consists of Alpha and I64 systems in a mixed cluster environment, you can access an Oracle CODASYL DBMS release 7.4.1.5 database from either system, or both systems concurrently.

1.1.2 Features Not Available for OpenVMS IA-64

The following features, capabilities or components are not available to run OpenVMS IA-64 with this Oracle CODASYL DBMS release.

- Using Ada with the DML precompiler to specify /LANGUAGE=ADA.
- Using PL/I with the DML precompiler to specify /LANGUAGE=PLI.

1.1.3 Missing Details from HELP files

Please note that some DBO qualifiers and commands are missing from the various HELP files supplied with this kit. Omissions and corrections will be added in future releases.

Please refer to the revised *Oracle CODASYL DBMS: Database Administration Reference Manual* for complete details of the DBO utility.

1.2 Hardware Notes

1.2.1 Note on Alpha EV56 Requirement

Oracle CODASYL DBMS Release 7.4.1.5 has been optimized for the Alpha EV56 platform as a minimum. If running on an older platform, Oracle CODASYL DBMS will still execute properly, but will run some operations in emulation mode and performance will be significantly slower. Oracle supports running Oracle CODASYL DBMS in this emulation mode as long as any problems encountered can be reproduced on an EV56 or greater system.

1.2.2 Intel Itanium Processor 9500 "Poulson" Supported

For Oracle CODASYL DBMS on OpenVMS Industry Standard 64 for Integrity Servers, the Intel Itanium Processor 9500 series, code named "Poulson", is supported.

1.2.3 Intel Itanium Processor 9700 "Kittson" Supported

For Oracle CODASYL DBMS on OpenVMS Industry Standard 64 for Integrity Servers, the Intel Itanium Processor 9700 series, code named "Kittson", is the newest processor for which Oracle CODASYL DBMS is certified. Please note that OpenVMS V8.4-2L1 or later is required for this class of processors.

1.3 Using Databases from Releases Earlier Than V7.0

You cannot convert or restore databases from versions earlier than 7.0 directly. The DBO/CONVERT command for Oracle CODASYL DBMS 7.4 supports conversions from V7.0, V7.1, V7.2 and V7.3 only.

If you have a V3.3 through V6.1 database, you must convert it to at least V7.0 and then convert it to 7.4.1.5 in two steps. For example, if you have a V4.2 database, install the latest update to DBMS 7.1, convert the database to that version, install DBMS 7.4.1.5 then convert the V7.1 database to 7.4.1.5.

If you attempt to convert or restore a database version prior V7.0 directly to 7.4.1.5, Oracle DBO generates an error.

1.4 Requirements

The following conditions must be met in order to install this software:

- Oracle CODASYL DBMS is supported on these versions of OpenVMS:
  - OpenVMS Version V8.4-2L1 is the minimum supported version of OpenVMS Alpha. Oracle CODASYL DBMS has been certified on V8.4-2L2.
  - OpenVMS Version V8.4-2L1 is the minimum supported version of OpenVMS IA-64. Oracle CODASYL DBMS has been certified on V8.4-2L3.

  Oracle Corporation strongly recommends that all available OpenVMS patches are installed on all systems prior to installing Oracle CODASYL DBMS. Contact your VSI support representative for more information and assistance.
- If any Oracle CODASYL DBMS version 7.4 is running it must be shutdown before you install this kit. That is, the command file SYS$MANAGER:MONSTOP74.COM should be executed before proceeding with this installation. If you have an OpenVMS cluster, you must also shutdown the Oracle CODASYL DBMS version 7.4 monitor on all nodes in the cluster before proceeding.

  After executing MONSTOP74.COM, no process on any system in the cluster should have any current DBMSHR74.EXE image activated. See Section 1.4.1 for additional information.

1.4.1 Ensure No Processes Have DBMSHR Image Activated

The Oracle CODASYL DBMS installation procedure checks to make sure that the Oracle CODASYL DBMS Monitor (DBMMON) process is not running. However, it is also important to make sure that there are no processes on the cluster that share the system disk that have image activated a prior Oracle CODASYL DBMS 7.4 version DBMSHR image. Such processes may not be currently attached to a database but may do so in the future and could cause problems by using an older DBMSHR image with a later installation.

The following command procedure (referenced as FIND_DBMSHR74_PROC.COM below) can be used on each cluster node that shares the system device to determine if there are any processes that have activated the DBMSHR74.EXE image. This procedure should be executed by a privileged account after MONSTOP74 has been run. Any processes that have DBMSHR74.EXE activated at this point should be terminated prior to starting the installation procedure.

```
$ DBM$TMP = "SYS$SCRATCH:DBM$TMP" + F$GETJPI(0,"MASTER_PID") + ".TMP"
$ DEFINE /NOLOG /USER DBM$TMP 'DBM$TMP
$ ANALYZE /SYSTEM
    SET OUTPUT DBM$TMP
    SHOW PROCESS /CHANNELS ALL
    EXIT
$ SEARCH /OUTPUT='DBM$TMP' 'DBM$TMP';-1 DBMSHR74.EXE,"PID:"
$ SEARCH 'DBM$TMP' DBMSHR74.EXE /WINDOW=(1,0)
$ DELETE /NOLOG 'DBM$TMP';*
```

In the following example, the process 25E5A8E9 named "FOO$SERVER" has the image DBMSHR74.EXE activated even after MONSTOP74.COM has been executed and this process is terminated prior to starting the Oracle CODASYL DBMS installation procedure:

```
$ @SYS$MANAGER:MONSTOP74.COM
   .
   .
   .
$ @FIND_DBMSHR74_PROC.COM

OpenVMS system analyzer

Process index: 00E9   Name: FOO$SERVER  Extended PID: 25E5A8E9
 0290  7FEA0500 8A5A50C0 $1$DGA1:[VMS$COMMON.SYSLIB]DBMSHR74.EXE;6 (section file)

$ STOP/IDENTIFICATION=25E5A8E9
```

1.5 Installation of Oracle CODASYL DBMS Software

Please refer to the *Oracle CODASYL DBMS 7.4 Installation Guide* for detailed Oracle CODASYL DBMS installation instructions. Oracle strongly recommends that you read the installation guide before attempting an installation.

To extract either the Adobe Portable Document Format (PDF) or text (TXT) version of the installation guide from the kit, use one of the following commands:

For OpenVMS Alpha:

```
$ BACKUP <device>:DBM07415A074.A/SAVE/SEL=DBM074_INSTALL_GDE.PDF
$ BACKUP <device>:DBM07415A074.A/SAVE/SEL=DBM074_INSTALL_GDE.TXT
```

For OpenVMS IA-64:

```
$ BACKUP <device>:DBM07415I074.A/SAVE/SEL=DBM074_INSTALL_GDE.PDF
$ BACKUP <device>:DBM07415I074.A/SAVE/SEL=DBM074_INSTALL_GDE.TXT
```

The current installation guide is also available in PDF format from My Oracle Support and from the Oracle Rdb and Related products documentation page, see Section 1.7 for more details.

1.6 PCSI Support for DBMS Kit Installation and Deinstallation

Starting with Oracle CODASYL DBMS release 7.4.1, whenever DBMS is installed or deinstalled, Oracle CODASYL DBMS is registered in the PCSI software product database. This allows customers to use the PCSI PRODUCT SHOW HISTORY and PRODUCT SHOW PRODUCT commands to display information about releases of Oracle CODASYL DBMS that have been installed or deinstalled. This information is also helpful as input whenever a Service Request (SR) is submitted to Oracle Support.

The following lines are now displayed during the installation of Oracle CODASYL DBMS, showing that the installation has been registered in the PCSI database.

```
The following product has been selected:
    ORCL I64VMS DBMS74 V7.4-10             Transition (registration)

The following product will be registered:
    ORCL I64VMS DBMS74 V7.4-10             DISK$NODE84_2:[VMS$COMMON.]

File lookup pass starting ...

Portion done: 0%
...100%

File lookup pass completed search for all files listed in the product's PDF
Total files searched: 0   Files present: 0   Files absent: 0

The following product has been registered:
    ORCL I64VMS DBMS74 V7.4-10             Transition (registration)
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...
```

Registration in the PCSI software product database allows the use of commands, as shown below, to track which Oracle CODASYL DBMS releases are currently installed and the history of any past product installations and deinstallations.

```
$ PRODUCT SHOW HISTORY/SINCE
---------------------------------- ----------- ----------- --- -----------
PRODUCT                            KIT TYPE    OPERATION   VAL DATE
---------------------------------- ----------- ----------- --- -----------
ORCL I64VMS DBMS74 V7.4-10         Transition  Reg Product (U) 10-OCT-2021
---------------------------------- ----------- ----------- --- -----------

1 item found

$ PRODUCT SHOW HISTORY DBMS7*
---------------------------------- ----------- ----------- --- -----------
PRODUCT                            KIT TYPE    OPERATION   VAL DATE
---------------------------------- ----------- ----------- --- -----------
ORCL I64VMS DBMS74 V7.4-10         Transition  Reg Product (U) 10-OCT-2021
---------------------------------- ----------- ----------- --- -----------

1 item found

$ PRODUCT SHOW PRODUCT DBMS7*
---------------------------------- ----------- ---------
PRODUCT                            KIT TYPE    STATE
---------------------------------- ----------- ---------
ORCL I64VMS DBMS74 V7.4-10         Transition  Installed
---------------------------------- ----------- ---------

1 item found
```

The following lines are displayed during the deinstallation of Oracle CODASYL DBMS, showing that the removal of the release has been registered in the PCSI database.

```
The following product has been selected:
    ORCL I64VMS DBMS74 V7.4-10            Transition (registration)

The following product will be removed from destination:
    ORCL I64VMS DBMS74 V7.4-10            DISK$NODE84_2:[VMS$COMMON.]

Portion done: 0%...100%

The following product has been removed:
    ORCL I64VMS DBMS74 V7.4-10            Transition (registration)
```

The example below shows the additional information that is displayed by the PCSI PRODUCT commands as a result of the deinstallation of a release of Oracle CODASYL DBMS.

```
$ PRODUCT SHOW HISTORY/SINCE
---------------------------------- ----------- ----------- --- -----------
PRODUCT                            KIT TYPE    OPERATION   VAL DATE
---------------------------------- ----------- ----------- --- -----------
ORCL I64VMS DBMS74 V7.4-10         Transition  Remove       -  10-OCT-2021
ORCL I64VMS DBMS74 V7.4-10         Transition  Reg Product (U) 10-OCT-2021
---------------------------------- ----------- ----------- --- -----------
2 items found

$ PRODUCT SHOW HISTORY DBMS7*
---------------------------------- ----------- ----------- --- -----------
PRODUCT                            KIT TYPE    OPERATION   VAL DATE
---------------------------------- ----------- ----------- --- -----------
ORCL I64VMS DBMS74 V7.4-10         Transition  Remove       -  10-OCT-2021
ORCL I64VMS DBMS74 V7.4-10         Transition  Reg Product (U) 10-OCT-2021
---------------------------------- ----------- ----------- --- -----------
2 items found

$ PRODUCT SHOW PRODUCT DBMS7*
---------------------------------- ----------- ---------
PRODUCT                            KIT TYPE    STATE
---------------------------------- ----------- ---------
0 items found
```

1.7 Online Document Format

You can view the Adobe Portable Document Format (PDF) versions of the documentation using the Adobe Acrobat Reader, which allows anyone to view, navigate, and print documents in PDF. See http://www.adobe.com for information about obtaining a free copy of Acrobat Reader and for information on supported platforms.

The Oracle CODASYL DBMS documentation in PDF can be referenced from the Oracle Rdb main web page. At the bottom of the page are links to documentation for all of the Rdb and DBMS Releases. The following link is to the Rdb related products page; scroll down to locate the CODASYL DBMS section.

https://www.oracle.com/database/technologies/related/rdb-doc-rlp.html

1.8 Send Us Your Comments

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title, release date, chapter, section, and page number (if available).

Please direct all comments on, and corrections for this Oracle documentation to Oracle Support at https://support.oracle.com

If you have problems with the software described in this documentation, please contact your local Oracle Support for assistance. Please provide a clear description of the problem encountered as well as the product version and the operating system version.

2
Enhancements and Changed Features

This chapter describes features in Oracle CODASYL DBMS.

2.1 New and Changed Features

This section describes new and changed features in Oracle CODASYL DBMS release 7.4.1.5.

2.1.1 DBO/SHOW STATISTICS Now Adds Lock ID to STALL_INVOKE Subprocess Parameters List

Bug 37516950

The DBO/SHOW STATISTICS utility provides the ability to call a subprocess if a stall has exceeded the seconds threshold provided on the Alarm qualifier. If a process is waiting for a lock request, it would be helpful if the lock identifier could be passed to the subprocess so that the lock can be examined. For example, the lock could be examined and the user that is blocking other users could then be identified.

With this update, a lock identifier is an additional parameter given to the subprocess. The parameters are now:

- P1 = The database filename
- P2 = The current date and time
- P3 = The process identification (PID) for the stalled user
- P4 = The database stream identifier for the stalled user
- P5 = The value of the alarm threshold in seconds
- P6 = The stall message showing why the user is stalled
- P7 = If the user is waiting on a lock, the lock identifier for the lock, otherwise zero ("00000000")
- P8 = reserved for future use by Oracle

To implement this functionality, you need to do two things:

1. Create a configuration file for DBO/SHOW STATISTICS. The file will include an entry for STALL_INVOKE that specifies a command to execute when a long stall is detected.
2. Create a command file to be invoked by DBO/SHOW STATISTICS. The command file will be passed the arguments listed above in P1 - P7.

Please refer to the *Oracle CODASYL DBMS: Database Administration Reference Manual Appendix A: Configuration File Management and User Defined Events*.

A sample configuration file would look like:

```
ALARM = 60;
STALL_INVOKE = "@DISK:[USER]STALL.COM";
```

The command file should include something like the following line of DCL:

```
$ IF P7 .NES. "00000000" THEN DBO/SHOW LOCK/LOCK='P7 /OUTPUT=LOCK.TXT
```

The content of the output file, LOCK.TXT, can then be parsed to determine the PID of the user that is blocking other users.

The DBO Show Statistics command should include the Configure qualifier that specifies the configuration file specification and usually the NoInteractive qualifier so that the command can be run in the background to monitor the stalls.

2.1.2 Enhanced Integrity Checking Before Page Writes

Enhancement 37985413

Additional checks have been added to prevent updated data pages from being written to disk if the area number or page number in the page header have been corrupted. Similarly, the area and page number for the before and after journals (.ruj and .aij) are validated before the journal entries are written. If an error is encountered, then the process will bugcheck and exit.

Such circumstances should be rare, but are added to protect the database. Please report all bugcheck dumps to Oracle Support for further analysis.

2.2 Changes and Enhancements from Prior Releases

This section is provided as documentation for new features added in prior releases of Oracle CODASYL DBMS.

2.2.1 The /LOCK_TIMEOUT Qualifier Can Now be Used With All DBO/RECLAIM Qualifiers

Previously, the DBO/RECLAIM /LOCK_TIMEOUT qualifier could only be specified if the /FREE_PAGES qualifier was also specified. With this release of Oracle CODASYL DBMS the /LOCK_TIMEOUT qualifier can be specified for all DBO/RECLAIM command functions.

If another user is already accessing the storage area to be processed the DBO/RECLAIM command will wait until the other user releases the storage area, or a lock wait timeout occurs. If no default lock timeout has been defined for the database or no current lock timeout is in effect, or the user wants to specify a lock timeout value to be used just during the execution of this DBO/RECLAIM command, the /LOCK_TIMEOUT qualifier can be specified. If a lock wait timeout occurs, a **warning** message will be output that the storage area or logical area could not be processed because of a lock conflict with another user and a **warning** status will be returned by DBO. That storage area will not be processed, but the DBO/RECLAIM command will continue processing the next storage area in the specified list of storage areas.

To specify a temporary lock timeout value in seconds that will be in effect only during the execution of the DBO/RECLAIM command the syntax is

```
/LOCK_TIMEOUT=seconds
```

The value that must be specified is the maximum time in seconds during which the DBO/RECLAIM command will wait to acquire an EXCLUSIVE UPDATE lock on the storage area to be processed when accessing an on-line database with other users.

If the /LOCK_TIMEOUT qualifier is not specified the lock wait time value used is the value of the logical name DBM$BIND_LOCK_TIMEOUT_INTERVAL (if defined), or the /TIMEOUT=LOCK=integer qualifier specified by the DBO/CREATE or DBO/MODIFY commands. If neither value has been specified and the /LOCK_TIMEOUT qualifier is not specified the DBO/RECLAIM command will wait indefinitely to acquire an EXCLUSIVE UPDATE lock on the storage area to be processed.

If /LOCK_TIMEOUT=0 is specified, the DBO/RECLAIM command will ignore any lock timeout defaults that may be in effect and wait indefinitely to acquire an EXCLUSIVE UPDATE lock on the storage area to be processed.

In the following example, warning messages are output, even if Log is not specified, in the DBO/RECLAIM commands if there is a lock conflict with another user that cannot be resolved. The /LOCK_TIMEOUT qualifier is specified to allow the user to specify a temporary lock timeout value to be used for the DBO/RECLAIM command.

```
$ DBO/RECLAIM/NOLOG/AREA=AREA1/LOCK_TIMEOUT=600 SAMPLE.ROO
%DBO-W-RCLMARNOTPRC, Area AREA1 could not be processed due to a lock conflict
$
```

2.2.2 Updated Diagnostics for the DBO/MOVE_AREA Command

With this release of Oracle CODASYL DBMS the DBO/MOVE_AREA command has been updated to output improved messages when changes are made to the database.

- %DBO-W-FULLAIJBKUP, partially-journaled changes made; database may not be recoverable

  This message warns the user that the actual move of database files to another device or directory is not journaled. Therefore, the user should consider performing a full backup of the database once the **move** operation completes.

  The DBO/RESTORE of the full database backup will recover the device and directory changes that were made and subsequent DBO/RECOVER will update the moved areas as needed.

  The following example shows the %DBO-W-FULLAIJBKUP message being output for a move operation which moves the database root file of a database with After Image Journaling enabled.

  ```
  $ DBO/MOVE_AREA /ROOT=disk:[directory] /NOLOG PARTS_DB
  %DBO-W-FULLAIJBKUP, partially-journaled changes made; database may not
      be recoverable
  %DBO-I-AIJRSTAVL, 1 after-image journal available for use
  %DBO-I-AIJRSTMOD, 1 after-image journal marked as "modified"
  %DBO-I-AIJISON, after-image journaling has been enabled
  %DBO-W-DOFULLBCK, full database backup should be done to ensure future
      recovery
  ```

2.2.3 IEEE Floating Point Now Used For DBO/ANALYZE

The DBO/ANALYZE command now computes values using IEEE floating point data types. This may result in slightly different values being displayed because IEEE floating point has different precision and rounding rules from the previous implementation that used D-floating.

An example of such a difference is shown in this excerpt from a DBO/ANALYZE command.

```
================================================================================
Oracle CODASYL DBMS Analysis Report - dd-mmm-yyyy hh:mm:ss.xxxx       Page 27
    Database filename is DISK:[DIRECTORY]PARTS.ROO;
    Storage area PERSONNEL - Filename is DISK:[DIRECTORY]PERSONNEL.DBS;
================================================================================

------------------ Index summary for storage area PERSONNEL ------------------

Index..........................  Number.  Bytes  Average  Average  Average.  Average.
set............................  of.....  per..  nodes..  levels.  entries.  percent.
name...........................  indices  node.  per.idx  per.idx  per.node  fullness

CONSISTS_OF                           16   116      1.0      1.0       2.6      13.2
```

In previous versions the value displayed for the "Average percent fullness" field would be displayed as 13.3 because the calculated value was actually 13.25 and D-floating computations would round up, but with IEEE T-floating now being used such values are now rounded down.

The DBO/ANALYZE /BINARY_FILE qualifier still writes the data in D-floating format, any intermediate IEEE T-floating values are converted to D-floating before being written to the binary output file.

2.2.4 New DBO/RECOVER /PROGRESS_REPORT Qualifier and Ctrl/T Display

In this release a /PROGRESS_REPORT qualifier has been added to the DBO RECOVER command.

The /PROGRESS_REPORT=n qualifier, where n is the time interval in seconds, displays the performance and progress of the database DBO/RECOVER operation at timed intervals to SYS$OUTPUT. The same display can also be output whenever Ctrl/T is typed during the DBO/RECOVER operation. Ctrl/T must have been previously enabled using the DCL command SET CONTROL=T. Note that SET CONTROL=T also requires that SET TERMINAL/BROADCAST is enabled for the display terminal.

The DBO/RECOVER performance and progress display has the following format:

```
DEVICE:[DIRECTORY]TEST_JOURNAL.AIJ;1
   Read    21 MB (29%) at   21 MB/s, estimated completion time 11:50:29.80
```

- The first line is the file specification of the after image journal (AIJ) currently being recovered.
- The second line displays the number of megabytes that have been read and processed from the AIJ file during the current interval, the percentage of the AIJ file that has been processed, the number of megabytes that are currently being read per second, and the estimated completion time for the processing of this AIJ file.

The DBO/RECOVER performance and progress display will be different if the /FORMAT=NEW_TAPE qualifier has been specified with the DBO/RECOVER command. This is due to differences in the way AIJ data is processed.

```
DEVICE:[DIRECTORY]AIJ_WORKG8V0RS6M99D1GJKG4I80.AIJ;
   Read    36 MB at   36 MB/s
```

- The first line is the file specification of the current temporary AIJ work file being recovered.
- The second line displays the number of megabytes that have been read and processed from the AIJ work file during this interval, and the number of megabytes that are currently being read per second.

Syntax

The command line syntax for the DBO/RECOVER command /PROGRESS_REPORT qualifier is

```
/PROGRESS_REPORT[=seconds]
```

The default is no progress report. If /PROGRESS_REPORT is specified without a value it will default to 60 seconds. The minimum value for the /PROGRESS_REPORT display interval is 1 second.

Examples

The following example shows the recovery of one backed up AIJ file with a progress report interval of 1 second. A zero value indicates no data was read from the AIJ file during that interval. The completion time estimate is only an approximation.

```
$ DBO/RECOVER/PROGRESS_REPORT=1/LOG/ROOT=PARTS.ROO; DBORECTST.BAIJ;
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PARTS.ROO;2
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]DBORECTST.AIJ;1
 at 11-AUG-2021 16:04:39.19
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    10 MB ( 7%) at   10 MB/s, estimated completion time 16:04:52.75
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    19 MB (13%) at    9 MB/s, estimated completion time 16:04:55.39
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    52 MB (35%) at   32 MB/s, estimated completion time 16:04:45.16
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    74 MB (49%) at   21 MB/s, estimated completion time 16:04:46.57
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    74 MB (49%) at    0 KB/s, estimated completion time 16:04:44.19
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    74 MB (49%) at    0 KB/s, estimated completion time 16:04:45.19
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    74 MB (49%) at    0 KB/s, estimated completion time 16:04:46.19
%DBO-I-LOGRECSTAT, transaction with TSN 225 committed
```

```
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read    78 MB (52%) at    3 MB/s, estimated completion time 16:05:04.80
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read   100 MB (68%) at   22 MB/s, estimated completion time 16:04:50.26
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read   118 MB (79%) at   17 MB/s, estimated completion time 16:04:50.88
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read   141 MB (95%) at   23 MB/s, estimated completion time 16:04:50.45
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read   148 MB (99%) at    6 MB/s, estimated completion time 16:04:51.20
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read   148 MB (99%) at    0 KB/s, estimated completion time 16:04:52.19
DEVICE:[DIRECTORY]DBORECTST.BAIJ;1
   Read   148 MB (99%) at    0 KB/s, estimated completion time 16:04:53.19
%DBO-I-LOGRECSTAT, transaction with TSN 226 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%DBO-I-LOGRECOVR, 2 transactions committed
%DBO-I-LOGRECOVR, 0 transactions rolled back
%DBO-I-LOGRECOVR, 0 transactions ignored
%DBO-I-AIJNOACTIVE, there are no active transactions
%DBO-I-AIJSUCCES, database recovery completed successfully
%DBO-I-AIJNXTSEQ, to continue this AIJ file recovery, the sequence number
 needed will be 1
%DBO-I-AIJALLDONE, after-image journal roll-forward operations completed
%DBO-I-LOGSUMMARY, total 2 transactions committed
%DBO-I-LOGSUMMARY, total 0 transactions rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJSUCCES, database recovery completed successfully
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery, the sequence number
 needed will be 1
%DBO-I-AIJNOENABLED, after-image journaling has not yet been enabled
$ SHOW TIME
  11-AUG-2021 16:04:53
```

2.2.5 Oracle CODASYL DBMS Timer Queue Limit Increased

Bug 28451930

Oracle CODASYL DBMS maintains a fixed sized vector of timer queue entries that are shared by each BIND within a process. In some cases multiple timer queue entries are required for a single bind.

Note

This vector is not sized by the TQELM process quota. The internal Oracle CODASYL DBMS limit controls the sizing of a fixed sized vector built into the Oracle DBMS Kernel.

In prior releases of Oracle CODASYL DBMS the limit for internal timer queue entries was set at 100. In some cases database servers which attempted to exceed this limit would fail.

This internal limit has been increased to 128.

2.2.6 Changed Behavior for the NOEDIT_FILENAME Qualifier in DBO/BACKUP /AFTER_JOURNAL command

Bug 31711278

In prior releases of Oracle CODASYL DBMS the /NOEDIT_FILENAME qualifier on the DBO /BACKUP /AFTER_JOURNAL was ignored. With this release it takes on a new meaning as described below:

- /EDIT_FILENAME

  As with previous versions, this qualifier defines the editing to be performed for the output backup file name. This editing is performed on the provided backup filename, or if "" (an empty string) is specified the default backup filename defined in the database.

  This qualifier replaces any EDIT_FILENAME defined for the database.

- /NOEDIT_FILENAME

  This qualifier negates any prior usage of the /EDIT_FILENAME qualifier on the DBO command line and also instructs DBO to ignore the EDIT_ FILENAME defined in the database by the DBO /CREATE or DBO /MODIFY commands. This is a change of behavior from prior versions.

  No editing is performed on the provided backup filename, or if "" (an empty string) is specified the default backup filename defined in the database is used without changes.

- Neither /EDIT_FILENAME nor /NOEDIT_FILENAME was used.

  In this case DBO /BACKUP /AFTER_JOURNAL will use the default if defined in the database.

2.2.7 Desupport for DBO/BACKUP HUFFMAN or LZSS Compression Options

This release of Oracle CODASYL DBMS removes the compression options HUFFMAN and LZSS from the DBO /BACKUP and DBO /BACKUP /AFTER_ JOURNAL commands. Please use ZLIB compression instead.

These older compression algorithms are much slower than the default ZLIB compression; in some cases orders of magnitude more CPU is required. If you receive a diagnostic as shown in the following example then Oracle recommends accepting the default (using /COMPRESSION without a keyword), or changing the DCL command procedure, or DBO PLAN file to explicitly state ZLIB.

Additionally, ZLIB compression allows the database administrator to determine levels of compression efficiency, from less time to more effective compression. Refer to the *Oracle CODASYL DBMS: Database Administration Reference Manual* for more details.

The following example shows the new diagnostic reported by DBO.

```
$ DBO/BACKUP-
    /MULTI-
    /COMPRESSION=LZSS-
    USER1:[DB]RNDDB.ROO-
    DB_COMPRESSED.DBF
%DBO-E-NOTSUPFORVER, The function COMPRESSION.LZSS is not supported
    for Oracle CODASYL DBMS V7.4-100
-DBO-I-COMPUSEZLIB, use the default, or specify ZLIB compression
$
```

2.2.8 Unexpected Change to After Image Journal BACKUP Behavior

Bug 31821266

In prior releases of Oracle CODASYL DBMS a command such as the following would unexpectly change the way an after image journal was backed up.

```
$ DBO/MODIFY/LOG SAMPLE/AIJ=(MODIFY,NAME=AIJ_1,BACK=(NONAME))
```

The BACKUP=NONAME clause, while removing the default after image journal (AIJ) name, would also erroneously and unexpectedly set the "same as journal" attribute for that after image journal.

The following example shows this unexpected behavior.

```
$! Try to reset/remove explicit AIJ backup and edit_filename
$ dbo/modify sample /aij=(modify,name=aij_1,back=(noname,noedit))
%DBO-I-LOGMODROO, modifying root file DBMS_USER:[TESTER]SAMPLE.ROO;1
%DBO-I-LOGMODVAL,     modified after-image journal backup file to SAME AS JOURNAL
%DBO-I-LOGMODSTR,     removed AIJ backup filename specification
%DBO-I-LOGMODSTR,     removed AIJ backup filename edits specification
$! dbo/modify sample /aij=(modify,name=aij_2,back=(noname,noedit))
$ dbo/dump/head=journal SAMPLE/out=sample.txt
$ search sample.txt "same as","Default backup filename","backup filename edits"
      - Default backup filename is "DBMS_USER:[TESTER]AIJ.BAIJ;"
      - Default backup filename edits are the following
          Backup filename is "same as journal"
$
```

This problem has been corrected in Oracle CODASYL DBMS version 7.4.1.5. To correct this problem the following changes have been made to the DBO/CREATE and DBO/MODIFY commands.

New options have been added to the BACKUP clause for the /JOURNAL_ OPTIONS and /AIJ_OPTIONS qualifiers.

- SAME_AS_JOURNAL

  This enables a fast form of AIJ backup. The database system creates a new empty AIJ with the same name and newer version number instead of locking the journal, backing it up to a separate file, and re-initializing the existing journal.

  The continuation of the backup may include renaming the old journal filename to that specified by the default naming scheme chosen by the database administrator. Otherwise, the name remains the same with a higher version number.

  When used with DBO/MODIFY /AIJ_OPTIONS qualifier this attribute will cause any NAME previously specified to be removed. When used with the /JOURNAL_OPTIONS qualifier it establishes a database wide default for subsequent /AIJ_OPTIONS usage.

- NOSAME_AS_JOURNAL

  This disables the BACKUP FILENAME IS SAME AS JOURNAL style backup for this after image journal (/AIJ_OPTIONS), or removes it as a database wide default (/JOURNAL_OPTIONS).

The BACKUP=NONAME clause now only removes any name associated with the BACKUP definition for this after image journal.

The clauses BACKUP=NAME and BACKUP=SAME_AS_JOURNAL are incompatible.

```
$ DBO/MODIFY/LOG sample-
    /AIJ=(MODIFY,NAME=AIJ_1,BACK=(SAME_AS_JOURNAL,NAME=SYS$DISK:[]XXX))
%DBO-F-CONFLSWIT, conflicting options BACKUP=NAME and BACKUP=SAME_AS_JOURNAL
```

2.2.9 DBO/MODIFY /JOURNAL_OPTIONS=ROTATE Now Creates An Emergency AIJ

Overview of Emergency AIJ

If a switch over to the next AIJ file cannot complete because the next AIJ file is unavailable, the database enters the "AIJ suspended" state. This state avoids the loss of database data because it cannot later be recovered from an AIJ file. During this state the database administrator can add new AIJ files or backup existing AIJ files to terminate the AIJ suspended state and allow suspended AIJ operations to continue.

If any of the following conditions are met and the DBM$BIND_ALS_CREATE_ AIJ system database logical is either not defined or defined as 1 (true) then a new permanent "emergency" AIJ file will automatically be created for the switch over to terminate the AIJ suspended state.

- a database recovery (DBR) process is active during the AIJ suspended state
- a Hot Standby database replication process is active during the AIJ suspended state
- the AIJ Log Server (ALS) process is active

If for any reason an emergency journal cannot be created, for example the maximum number of AIJ files defined for the database has already been reached, the AIJ suspended state will continue and the database administrator must resolve the situation or the database may be shut down (please see the Oracle CODASYL DBMS AIJ related documentation for the complete details).

Enhancement to DBO/MODIFY /JOURNAL_OPTIONS=ROTATE

The DBO/MODIFY /JOURNAL_OPTIONS=ROTATE command changes the currently active After Image Journal (AIJ) file to the next available AIJ file if a fixed size AIJ journaling configuration is defined for a database. Normally, it is not necessary to use this command because the switch to the next available journal occurs automatically when the currently active fixed size AIJ file is full.

However, the DBO/MODIFY /JOURNAL_OPTIONS=ROTATE command can be used in cases where it is necessary to force a switch to the next available AIJ file, such as when it is necessary to switch to the next AIJ file on another disk when the disk used by the currently active fixed size AIJ file requires maintenance.

This command will also automatically create a permanent emergency AIJ journal file if necessary.

Note

The only way to prevent DBO/MODIFY /JOURNAL_OPTIONS=ROTATE from creating an emergency journal is to explicitly define the

system DBM$BIND_ALS_CREATE_AIJ logical to be 0 (false) in the LNM$SYSTEM_TABLE.

As with emergency journals created in the other cases mentioned above, the emergency journals created by the DBO/MODIFY /JOURNAL_OPTIONS=ROTATE command are permanent AIJ journals defined for the database. By default they are created using the same device and directory as the currently active AIJ journal being switched from, unless the DBM$BIND_AIJ_EMERGENCY_DIR database bind logical is defined to specify a different device and directory. Emergency AIJ journals are created using the same allocation definitions as the currently active AIJ journal being switched from. The generated name of the emergency AIJ is "EMERGENCY_x", where x is a series of 16 characters generated to create a unique name.

The following example shows this new feature.

- The DBM$BIND_ALS_CREATE_AIJ logical has been defined as "1" in the LNM$SYSTEM_TABLE to allow emergency AIJ journals to be created. This is also the default if the DBM$BIND_ALS_CREATE_AIJ logical is not defined.

```
$ DEFINE/SYSTEM DBM$BIND_ALS_CREATE_AIJ 1
$ SHOW LOGICAL DBM$BIND_ALS_CREATE_AIJ
   "DBM$BIND_ALS_CREATE_AIJ" = "1" (LNM$SYSTEM_TABLE)
$
```

- The TEST database currently has two journals defined, "JOURNAL1" and "JOURNAL2", but additional AIJ slots are reserved in the database definition in case additional journals need to be created.
- The DBO/MODIFY /JOURNAL_OPTIONS=ROTATE command is used to switch from "JOURNAL1" to "JOURNAL2".
- Then, when the DBO/MODIFY /JOURNAL_OPTIONS=ROTATE command is used to switch from "JOURNAL2" back to "JOURNAL1", "JOURNAL1" is not available because it has not been backed up for some reason, perhaps because the Oracle CODASYL DBMS AIJ Automatic Backup Server (ABS) is not running.
- The DBO/MODIFY /JOURNAL_OPTIONS=ROTATE command automatically creates an emergency AIJ journal with the unique generated name "EMERGENCY_00B6A892F7BECC10" and switches over to this permanent new database AIJ journal. This is an exceptional case that only happens if no currently defined AIJ journal is available.

```
$ ! Insert data (first defined journal)
   .
   .
   .
$ ! Switch to the next defined journal.
$
$ DBO/MODIFY /JOURNAL_OPTIONS=ROTATE /LOG TEST
%DBO-I-OPERNOTIFY, system operator notification: After-image journal 0 switch-over in progress (to 1)
%DBO-I-OPERNOTIFY, system operator notification: Last unmodified AIJ journal has been selected
%DBO-I-OPERNOTIFY, system operator notification: After-image journal switch-over complete
%DBO-I-LOGMODSTR,     switching to after-image journal "JOURNAL2"
$
$ ! Put data in the first defined journal.
   .
   .
   .
$
$ ! Switch to the next journal.
$ ! An EMERGENCY journal with a generated
$ ! name such as "EMERGENCY_00B0333F5D37E224"
$ ! will be created since the existing
$ ! journals have not been backed up.
$
$ DBO/MODIFY/JOURNAL_OPTIONS=ROTATE PARTS
%DBO-I-LOGMODROO, modifying root file SERDB_USER1:[HOCHULI.V74.WORK.EMERGENCY_AIJ.REVISE]PARTS.ROO;1
%DBO-I-OPERNOTIFY, system operator notification: After-image journal 1 switch-over in progress (to 2)
%DBO-I-OPERNOTIFY, system operator notification: Last unmodified AIJ journal has been selected
%DBO-I-OPERNOTIFY, system operator notification: After-image journal switch-over complete
%DBO-I-LOGMODSTR,     switching to after-image journal "EMERGENCY_00B6A892F7BECC10"
$
$ ! Insert data (next journal)
   .
   .
   .
```

2.2.10 DBO/MOVE_AREA Adds /LIVE, /NOLIVE and /NOSNAPSHOTS Qualifiers

By default DBO/MOVE_AREA moves or creates a new version of both the storage area data (.dbs) and snapshot (.snp) files. This syntax allows moving *only* the data area file or *only* the snapshot area file for all or for named storage areas. The qualifiers /NOLIVE and /NOSNAPSHOTS are positional qualifiers that can be specified globally as a default and/or for one or more named storage areas. They can be specified on the command line or in an options file using the existing DBO/MOVE_AREA /OPTION=filespec qualifier.

The syntax for these qualifiers is as follows:

- /[NO]SNAPSHOTS[=([FILE=filespec][ALLOCATION=n])]

  The /NOSNAPSHOTS qualifier requests that DBO/MOVE_AREA exclude the snapshot file, and therefore only the data storage area file is moved. SNAPSHOTS is the default. Neither FILE=filespec nor ALLOCATION=n can be specified with NOSNAPSHOTS.

- /[NO]LIVE

  The /NOLIVE qualifier requests that DBO/MOVE_AREA exclude the live data file, and therefore only the snapshot area file is moved. LIVE is the default. It does not accept any values. LIVE or NOLOVE as a local qualifier will override NOLIVE or LIVE as a global qualifier.

The following /MOVE_AREA qualifiers cannot be specified with either /NOLIVE or /NOSNAPSHOTS:

- /ROOT
- /BLOCKS_PER_PAGE
- /CLUSTER_NODES
- /USERS

The following existing /MOVE_AREA qualifiers cannot be specified with /NOLIVE:

- /FILE
- /SPACE_MANAGEMENT
- /THRESHOLDS
- /EXTENSION

The following existing /MOVE_AREA qualifiers cannot be specified with /NOSNAPSHOTS:

- /SNAPSHOTS=(FILE=filespec)
- /SNAPSHOTS=(ALLOCATION=n)

In the following example only the storage area snapshot files are moved for all database storage areas.

```
$ DBO/MOVE_AREA/ALL/NOLIVE/NOLOG/DIR=[.MOVE] parts.ROO
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

In the following example only the storage area data files are moved for all database storage areas.

```
$ DBO/MOVE_AREA/ALL/NOSNAPSHOTS/NOLOG/DIR=[.MOVE] parts.ROO
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

In the following example only the snapshot storage area file is moved for the EMPLOYEE storage area and only the data storage area file is moved for the MARKET storage area. Note that for the MARKET storage area /LIVE did not need to be specified since it is the default.

```
$ DBO/MOVE_AREA/NOLOG parts.ROO -
  EMPLOYEE /nolive -
           /snapshots=(file=DISK:[DIRECTORY]test_employee.snp), -
  MARKET  /live -
          /file=DISK:[DIRECTORY]test_market -
          /nosnapshots
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

In the following example an options file is used to specify the storage areas to be moved. Only the data storage area file is moved for EMPLOYEE, only the snapshot storage area file is moved for MARKET, and both the snapshot and data storage area files are moved for DEPARTMENTS. Note that /LIVE and /SNAPSHOT are the defaults.

```
$ DBO/MOVE_AREA/NOLOG/DIRECTORY=DISK:[DIRECTORY]/OPTION=testmove.opt -
 parts.ROO
  EMPLOYEE -
           /file=DISK:[DIRECTORY]test_employee.dbs -
           /nosnapshot
  MARKET   /nolive -
           /snapshot = (file=DISK:[DIRECTORY]test_market.snp)
  DEPARTMENTS -
              /file=DISK:[DIRECTORY]test_departments -
              /snapshot = (file=DISK:[DIRECTORY]test_departments.snp)
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

In the following example the global default qualifiers designate that only the snapshot files should be moved for all storage areas. However, an options file is used to override the default for specific storage areas. Therefore, only the data storage area file is moved for EMPLOYEE, only the snapshot storage area file is moved for MARKET, and both the snapshot and data storage area files are moved for DEPARTMENTS. Note that in this case /LIVE needed to be specified in the options file to override the global specification of /NOLIVE but /NOLIVE did not have to be specified in the options file. Also /NOSNAPSHOT had to be specified in the options file to override the assumed global default of /SNAPSHOT.

```
$ DBO/MOVE_AREA/ALL/DIRECTORY=DISK:[DIRECTORY]/NOLOG/NOLIVE-
 /OPTION=testmove.opt  parts.ROO
  EMPLOYEE /live -
           /file=DISK:[DIRECTORY]test_employee.dbs -
           /nosnapshot
  MARKET   /nolive -
           /snapshot = (file=DISK:[DIRECTORY]test_market.snp)
  DEPARTMENTS -
              /live -
              /file=DISK:[DIRECTORY]test_departments -
              /snapshot = (file=DISK:[DIRECTORY]test_departments.snp)
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

2.2.11 DBO/SHOW LOGICAL_NAME Supports /DESCRIPTION Qualifier

Bug 3264793, 3682207, 5634563, 19545970

The DBO /SHOW LOGICAL_NAME command includes a /DESCRIPTION qualifier. This new qualifier retrieves a brief description of the logical name and displays it along with the current definition. If wildcards are used for the logical name, then any matching logical names will also include output of the description.

The following example shows the use of the /DESCRIPTION qualifier, a wildcard logical name specification and the use of the /UNDEFINED qualifier to include output - even for logical names not defined.

```
$ dbo/show logical/undefined/description dbm$*sort*
  "DBM$BIND_OPT_SORT_THRESHOLD" = Undefined

       This logical name is used by DBO/OPTIMIZE/AFTER_JOURNAL.

       This logical name may be defined to limit the number of records
       sorted at one time, which reduces the size of the sort workfiles
       when device space is limited. The efficiency of optimizing the
       journal is not as good as sorting all records in one pass, but
       the sort may be faster and device space for sort workfiles is
       greatly reduced. The optimized file will be a little larger if
       the sort is limited to a specified number of records.
```

```
            Default: no limit
            Minimum: 1 (if zero is specified the limit is assumed to be
            unlimited)
            Maximum: no limit

         See also the logical name DBM$BIND_OPT_TXN_THRESHOLD.

   "DBM$SORT_FILES" = "5" (LNM$PROCESS_TABLE)

         Use this logical name to specify the number of temporary work
         files that a database sort operation is to use if the work files
         are required. The default number of work files is 2 and the
         maximum number is 10. For example:

         $ DEFINE DBM$SORT_FILES   3

         To control the placement of the work files define the various
         SORTWORKn logical names. The SORTWORKn logicals identify the
         device and directory of the work files. These logical names
         must point to a device that has a directory (allowing WRITE
         protection) with the same name as the user's home directory.

            Table: PROCESS
            Values: value between 2 (default) and 10

$
```

The definitions of all logical names are maintained in a HELP library called SYS$HELP:DBODISPLAY74.HLB. OpenVMS users can also use the DCL HELP command to query this help library.

```
$ HELP/LIBR=SYS$HELP:DBODISPLAY74.HLB DBMS_Logical_names DBMS$RUJ

DBMS_LOGICAL_NAMES

  DBMS$RUJ

       By default the Recovery Unit Journal file is located on the
       device on which the attached users default directory is located
       and in a special directory [DBM$RUJ]. This placement is used to
       avoid accidental deletion of this important file.

       However, you can use this logical name to locate the .ruj file on
       a different device and directory from the default location. This
       can help to reduce contention on the default device.

Topic?
```

2.2.12 AIJ Backup Data Compression Information now in the ABS Process Log

New messages have been added to Oracle CODASYL DBMS After Image Journal Automatic Backup Server process logs which will be output if data compression is enabled for automatic database AIJ backups.

- Automatic database AIJ backups can be enabled using the DBO/MODIFY /AFTER_JOURNAL /JOURNAL_OPTION=SPOOLER.
- Automatic database AIJ backups using data compression can be defined for a database by the DBO/MODIFY/AFTER_JOURNAL command (see Section 2.2.14 for more details).
- A log file to be output by each automatic AIJ backup server process can be enabled by the DBO/SET SERVER ABS command for a database or by defining the system DBM$BIND_ABS_LOG_FILE logical for a cluster node (see Section 2.2.28 for more details).

The new AIJ backup server process log messages output if data compression is enabled for database automatic AIJ backup files are the following.

```
Compression ZLIB level 6
```

This first message shows the data compression algorithm "ZLIB" and the ZLIB level used, which will be an integer between 1 and 9. The higher the level number the greater the compression but also the greater amount of CPU time spent doing the compression. The default level of 6 is a good trade off between the necessary CPU time and the amount of data compression.

```
Data compressed by 39% (3956 KB in/2428 KB out)
```

This second compression message shows the amount of compression as a percent value based on the total number of input uncompressed bytes compared to the total number of output compressed bytes, followed by the total number of uncompressed bytes and the total number of compressed bytes expressed in scaling units which will vary depending on the amount of data compressed. The scaling units will be one of "Bytes", "KB" for kilobytes, "MB" for megabytes, "GB" for gigabytes or "TB" for Terabytes.

The following example shows the last portion of a log file named ABS_23CF6459.OUT created by an automatic backup server process with a process id of 23CF6459. The log messages show that this process has created the After Image Journal backup file AIJBCKCOMP.ABF;24 and that the data in the backup file was compressed using the ZLIB compression level 6 algorithm. The compressed output data was 38% smaller than the uncompressed input data. The number of uncompressed input kilobytes is 3932 and the number of compressed output kilobytes in the AIJBCKCOMP.ABF;24 output AIJ backup file is 2445.

```
$ TYPE DEVICE:[DIRECTORY]ABS_23CF6459.OUT

 4-OCT-2018 08:44:33.45 - AIJ Backup Server (ABS) activated
 4-OCT-2018 08:44:33.45 - Database is DEVICE:[DIRECTORY]AIJBCKCOMP.ROO;1
 4-OCT-2018 08:44:33.45 - Backing up AIJ 23
 4-OCT-2018 08:44:33.45 - By-sequence AIJ backup for sequence 23 to 23
%DBMS-I-OPERNOTIFY, system operator notification: AIJ backup operation started
 4-OCT-2018 08:44:33.45 - No Hot Standby servers active
%DBMS-I-AIJBCKSEQ, backing up after-image journal sequence number 23
%DBMS-I-LOGBCKAIJ, backing up after-image journal J4 at 08:44:33.45
%DBMS-I-LOGCREBCK, created backup file DEVICE:[DIRECTORY]AIJBCKCOMP.ABF;24
%DBMS-I-OPERNOTIFY, system operator notification: AIJ backup operation completed
 4-OCT-2018 08:44:34.34 - AIJ backup complete
 4-OCT-2018 08:44:34.34 - Compression ZLIB level 6
 4-OCT-2018 08:44:34.34 - Data compressed by 38% (3932 KB in/2445 KB out)
 4-OCT-2018 08:44:34.34 -  ELAPSED:  0 00:00:00.91  CPU: 0:00:00.79
                           BUFIO: 20  DIRIO: 356  FAULTS: 242
$
```

2.2.13 /LOCK_TIMEOUT Qualifier Assumes the Database Default

The DBO commands DBO/BACKUP/MULTI/ONLINE and DBO/COPY/ONLINE accept a /LOCK_TIMEOUT qualifier. This qualifier specifies a value which is the maximum time (in seconds) to wait for acquiring the database QUIET POINT and other locks used for online database access. If /LOCK_TIMEOUT is not specified, DBO will wait indefinitely to acquire the database lock it needed.

Now the /LOCK_TIMEOUT qualifier can be specified without a value. In this case the default lock timeout specified for the database will be used. Specifically, the default lock timeout may be the value of the logical name DBM$BIND_LOCK_TIMEOUT_INTERVAL if it has been specified. If DBM$BIND_LOCK_TIMEOUT_INTERVAL has not been specified then the timeout will be the maximum possible lock timeout value which can be specified for an Oracle CODASYL DBMS database.

The syntax for this qualifier is as follows:

```
/LOCK_TIMEOUT [= n]
```

Note that /LOCK_TIMEOUT is not the default and must be specified. The default if /LOCK_TIMEOUT is not specified continues to be to wait indefinitely to acquire the QUIET POINT or other database locks requested by DBO.

2.2.14 Compression of Backup Files Supported for Automatic AIJ Backups

This feature allows the database administrator to save the COMPRESSION settings for after image backups. This allows the Automatic AIJ Backups (ABS) process to apply compression to the output backup files, and also allows the defaulting of compression for manual backups.

- To set the compression and the ZLIB level in the database use this DBO/MODIFY command:

  ```
  $ DBO/MODIFY/AFTER_JOURNAL /BACKUPS=(...,COMPRESSION[=ZLIB[=n]])
  ```

  The default setting for a created database is NOCOMPRESSION.

  If /BACKUPS=COMPRESSION is specified without modifiers it will default to ZLIB level 6. This is good a balance between low CPU usage (level 1) and low space usage (level 9).

  To override the setting in the database root the manual DBO/BACKUP can specify /COMPRESSION or NOCOMPRESSION qualifiers.

- To clear this setting in the database use:

  ```
  $ DBO/MODIFY/AFTER_JOURNAL /BACKUPS=(...,NOCOMPRESSION)
  ```

The DBO/DUMP/HEADER command will report the database setting.

The compression setting is recorded in the backup file and therefore DBO/RECOVER can automatically decompress the backup prior to applying the changes to the database.

2.2.15 Maximum Buffer Size Increase

Previously, the maximum allowed database buffer size was 128 blocks. This limit has been increased. The current maximum allowed database buffer size is 256 disk blocks.

Be aware that using larger database buffer sizes will require additional virtual memory.

2.2.16 Minimum Number of Allowed Users Increased

In prior versions of Oracle CODASYL DBMS, the minimum number of allowed database users was one (1). Starting Oracle DBMS V7.3 this minimum was been increased to five (5) to allow for various optional database servers, such as the ABS (Automatic Backup Server), RCS (Row Cache Server) or ALS (AIJ Log Server) to access the database.

2.2.17 Added Option to Pass Values to /CONFIRM during RESTORE Operation

In prior releases of Oracle CODASYL DBMS, if problems occur during tape restore operations and the /CONFIRM qualifier is specified, DBO will wait for user input on the terminal.

```
$  DBO/RESTORE/MULTI/NOCDD/REWIND LMA1001:VOL002.DBF PARTS /DIRECTORY=[] /LOG
%DBO-I-WRNGLBL, Tape on LMA1001 was incorrectly labeled. Expected VOL002 - Found MF_PER
%DBO-I-TAPEDISPR, Specify tape disposition for LMA1001 (QUIT,OVERRIDE,RETRY,UNLOAD)
DBO> QUIT (User has to enter the RESPONSE.)
%DBO-F-ABORT, operator requested abort on fatal error
%DBO-F-FATALERR, fatal error on RESTORE
%DBO-F-FTL_RSTR, Fatal error for RESTORE operation at 13-JUL-2009 11:22:32.90

$  DBO/RESTORE/MULTI/NOCDD/REWIND LMA1001:VOL002.DBF PARTS /DIRECTORY=[] /LOG
%DBO-I-WRNGLBL, Tape on LMA1001 was incorrectly labeled. Expected VOL002 - Found MF_PER
%DBO-I-TAPEDISPR, Specify tape disposition for LMA1001 (QUIT,OVERRIDE,RETRY,UNLOAD)
DBO> RETRY (User has to enter the RESPONSE.)
%MOUNT-I-MOUNTED, MF_PER mounted on LMA1001:
%DBO-I-WRNGLBL, Tape on LMA1001 was incorrectly labeled. Expected VOL002 - Found MF_PER
%DBO-I-TAPEDISPR, Specify tape disposition for LMA1001 (QUIT,OVERRIDE,RETRY,UNLOAD)
DBO> QUIT (User has to enter the RESPONSE.)
%DBO-F-ABORT, operator requested abort on fatal error
%DBO-F-FATALERR, fatal error on RESTORE
%DBO-F-FTL_RSTR, Fatal error for RESTORE operation at 13-JUL-2009 11:22:55.86

$  DBO/RESTORE/MULTI/NOCDD/REWIND LMA1001:VOL002.DBF PARTS /DIRECTORY=[] /LOG
%DBO-I-WRNGLBL, Tape on LMA1001 was incorrectly labeled. Expected VOL002 - Found MF_PER
%DBO-I-TAPEDISPR, Specify tape disposition for LMA1001 (QUIT,OVERRIDE,RETRY,UNLOAD)
DBO> OVERRIDE (User has to enter the RESPONSE.)
%DBO-F-FILACCERR, error opening input file LMA1001:[000000]VOL002.DBF;
-SYSTEM-W-NOSUCHFILE, no such file
%DBO-F-FATALERR, fatal error on RESTORE
%DBO-F-FTL_RSTR, Fatal error for RESTORE operation at 13-JUL-2009 11:23:05.59
```

The user has the option of selecting values for /CONFIRM during a RESTORE from tape operation. The new syntax and valid values are:

```
DBO/RESTORE/MULTI...   /CONFIRM[=QUIT|RETRY=x|OVERRIDE|UNLOAD]
```

```
$  DBO/RESTORE/MULTI/NOCDD/REWIND LMA1001:VOL002.DBF PARTS /DIRECTORY=[] /LOG /CONFIRM=QUIT
%MOUNT-I-MOUNTED, MF_PER mounted on LMA1001:
%DBO-I-TAPEDEF, Terminating restore operation as requested by user
%DBO-F-ABORT, operator requested abort on fatal error
%DBO-F-FATALERR, fatal error on RESTORE
%DBO-F-FTL_RSTR, Fatal error for RESTORE operation at 13-JUL-2009 11:43:31.35
```

```
$  DBO/RESTORE/MULTI/NOCDD/REWIND LMA1001:VOL002.DBF PARTS /DIRECTORY=[] /LOG /CONFIRM=RETRY=2
%DBO-I-TAPEDEF, Retrying tape operation as requested by user
%MOUNT-I-MOUNTED, MF_PER mounted on LMA1001:
%DBO-I-TAPEDEF, Retrying tape operation as requested by user
%MOUNT-I-MOUNTED, MF_PER mounted on LMA1001:
%DBO-F-ABORT, operator requested abort on fatal error
%DBO-F-FATALERR, fatal error on RESTORE
%DBO-F-FTL_RSTR, Fatal error for RESTORE operation at 13-JUL-2009 11:43:42.97

$  DBO/RESTORE/MULTI/NOCDD/REWIND LMA1001:VOL002.DBF PARTS /DIRECTORY=[]V/LOG /CONFIRM=OVERRIDE
%DBO-I-TAPEDEF, Overriding tape label as requested by user
%DBO-F-FILACCERR, error opening input file LMA1001:[000000]VOL002.DBF;
-SYSTEM-W-NOSUCHFILE, no such file %DBO-F-FATALERR, fatal error on RESTORE
%DBO-F-FTL_RSTR, Fatal error for RESTORE operation at 13-JUL-2009 11:43:58.38
```

2.2.18 DBO/BACKUP /MULTITHREADED /COMPRESSION Qualifier

The DBO/BACKUP/MULTITHREADED utility supports data compression via the /COMPRESSION qualifier.

The /COMPRESSION qualifier accepts the keyword ZLIB.

- ZLIB=level - ZLIB algorithm.

  The "level" value is an integer between 1 and 9 specifying the relative compression level with one being the least amount of compression and nine being the greatest amount of compression. Higher levels of the compression use increased CPU time while generally providing better compression. The default compression level of 6 is a balance between compression effectiveness and CPU consumption.

  The ZLIB algorithm and software was developed by Jean-loup Gailly and Mark Adler. This implementation generally uses the same or less CPU time and is generally more effective (compresses better) than either of the HUFFMAN or LZSS algorithms.

If you specify the /COMPRESSION qualifier without a value, the default is /COMPRESSION=ZLIB=6.

Examples using the /COMPRESS qualifier. Note that if "/LOG=FULL" is specified, data compression statistics information is displayed.

```
$ DBO/BACKUP/MULTI/COMPRESS/NOLOG prod_db BCK
$ DBO/BACKUP/MULTI/COMPRESS=ZLIB:9 /LOG=FULL prod_db BCK
   .
   .
   .
BACKUP summary statistics:
    Data compressed by 53% (9791 KB in/4650 KB out)
```

Compression Effectiveness Varies

The actual amount of compression for any algorithm is strongly dependent on the actual data being compressed. Some database content may compress quite well and other content may compress not at all and may actually result in expansion of the output.

When using the /ENCRYPT and /COMPRESS features together, data is first compressed and then encrypted. This provides effective compression as well as effective encryption.

2.2.19 COMPRESS Qualifier for After-Image Journal Backup Command

After-Image Journal backup files can be compressed the same way database backup files can be compressed. Compression for AIJ backup files can be combined with encryption.

The following commands have been modified to work with compressed AIJ backup files:

```
DBO /BACKUP /AFTER_JOURNAL /COMPRESSION
DBO /DUMP   /AFTER_JOURNAL
DBO /RECOVER
```

Compression encoding method ZLIB works with both /FORMAT=NEW_TAPE and /FORMAT=OLD_FILE. The /LOG qualfier reports the achieved compression at the end of the log output.

Example:

```
DBO /BACKUP /AFTER /FORMAT=NEW_TAPE  PROD_DB.ROO PROD_DB.BAIJ /COMPRESS=ZLIB /LOG
%DBO-I-AIJBCKBEG, beginning after-image journal backup operation
...
%DBO-I-LOGCOMPR, data compressed by 55% (27152 KB in/12471 KB out)
```

2.2.20 DBO/BACKUP/MULTITHREAD /NORECORD Qualifier

This qualfier can be used to avoid the modification of the database with recent backup information. Hence the database appears like it had not been backed up at this time.

The main purpose of this qualifier is to allow a backup of a hot standby database without modifying the database files.

Examples using the /NORECORD qualifier:

```
$ DBO /BACKUP /MULTITHREAD /NORECORD prod_db BCK
```

2.2.21 DBO /{BACKUP/MULTI|COPY|MOVE} /THREADS=n Qualifier

The /THREADS qualifier allows the user to better control the system load created by a multithreaded backup, copy, or move operation. The user can specify the number of threads to use by DBO.

DBO creates so called internal 'threads' of execution to read data from one specific storage area. Threads run quasi parallel within the process executing the DBO image. Each thread generates its own I/O load and consumes resources like virtual address space and process quotas (e.g. FILLM, BYTLM). The more threads the more I/Os can be generated at one point in time and the more resources are needed to accomplish the same task.

Performance increases with more threads due to parallel activities which keeps disk drives more busy. However, at a certain number of threads performance suffers because the disk I/O subsystem is saturated and I/O queues build up for the disk drives. Also the extra CPU time for additional thread scheduling

overhead reduces the overall performance. Typically 2-5 threads per input disk drive are sufficient to drive the disk I/O subsystem at its optimum. However, some controllers may be able to handle the I/O load of more threads, e.g. disk controllers with RAID sets and extra cache memory.

In a copy or move operation one thread moves the data of one storage area at-a-time. If there are more storage areas to be moved than there are threads then the next idle thread takes on the next storage area. Storage areas are moved in order of the area size - largest areas first. This optimizes the overall elapsed time by allowing other threads to move smaller areas while an earlier thread is still working on a large area. If no threads qualifier is specified then 10 threads are created by default. The minimum is 1 thread and the maximum is the number of storage areas to be copied or moved. If the user specifies a value larger than the number of storage areas then DBO silently limits the number of threads to the number of storage areas.

In a multithreaded backup operation one writer thread is created per output stream. An output stream can be either a tape drive, a disk file, or a media library manager stream. In addition DBO creates a number of reader threads and their number can be specified. DBO assigns a subset of reader threads to writer threads. DBO calculates the assignment so that roughly the same amount of data is assigned to each output stream. By default five reader threads are created for each writer thread. If the user has specified the number of threads then this number is used to create the reader thread pool. DBO always limits the number of reader threads to the number of storage areas. A threads number of 0 causes DBO to create one thread per storage area which start to run all in parallel immediately. Even though this may sound like a good idea to improve performance this approach suffers for databases with a larger number (>10) of storage areas. For a very large number of storage areas (>800) this fails due to hard limitations in system resources like virtual address space.

The old READER_THREAD_RATIO qualifier has been deprecated but is still accepted and works exactly the same as in previous versions.

Examples using the /THREADS qualifier:

- Copying one storage area at a time:

```
$ DBO /COPY /THREADS=1 /LOG prod_db BCK
%DBO-I-MOVTXT_04, Starting move of storage area ...
%DBO-I-MOVTXT_01, Completed move of storage area ...
%DBO-I-MOVTXT_05, Moved snapshot area file ...
%DBO-I-MOVTXT_04, Starting move of storage area ...
%DBO-I-MOVTXT_01, Completed move of storage area ...
%DBO-I-MOVTXT_05, Moved snapshot area file ...
   .
   .
   .
```

- Copying three storage areas in parallel:

```
$ DBO /COPY /THREADS=3 /LOG prod_db BCK
%DBO-I-MOVTXT_04, Starting move of storage area ...
%DBO-I-MOVTXT_04, Starting move of storage area ...
%DBO-I-MOVTXT_04, Starting move of storage area ...
%DBO-I-MOVTXT_01, Completed move of storage area ...
%DBO-I-MOVTXT_05, Moved snapshot area file ...
%DBO-I-MOVTXT_04, Starting move of storage area ...
%DBO-I-MOVTXT_01, Completed move of storage area ...
%DBO-I-MOVTXT_05, Moved snapshot area file ...
   .
   .
   .
```

2.2.22 DBO/MODIFY /PAGE_PAD_LEN Qualifier

Both DBO/CREATE and DBO/MODIFY allow the modification of a storage area's PAGE PADDING LENGTH (PPL) using the /PAGE_PAD_LEN qualifier.

The PPL is an area reserved for Oracle CODASYL DBMS page management and would not typically be used to store user data. In prior versions, the size of this area was a fixed at 10% of the database page size for that storage area.

In certain database designs, the fixed PPL resulted in much unused space on the data page and limited DBA's ability to set threshholds properly. With this qualifier, this restriction has been lifted.

Note

Oracle CODASYL DBMS suggests that the default for this parameter not be changed, unless warranted after a thorough design review.

If you specify the /PAGE_PAG_LEN qualifier, you must provide a value (between 0 and the storage area page size) which will be the number of bytes to be set aside for the PPL in lieu of the default. The qualifier is positional; it can be used either globally or locally on selected areas.

```
$ DBO/MOD/PAGE_PAD_LEN=25 PARTS/LOG
%DBO-I-LOGMODROO, modifying root file D:PARTS.ROO;2
%DBO-I-LOGMODSTO, modifying storage area MAKE
%DBO-I-LOGMODVAL,     modified page pad len to 25
%DBO-I-LOGMODSTO, modifying storage area BUY
%DBO-I-LOGMODVAL,     modified page pad len to 25
%DBO-I-LOGMODSTO, modifying storage area MARKET
%DBO-I-LOGMODVAL,     modified page pad len to 25
%DBO-I-LOGMODSTO, modifying storage area PERSONNEL
%DBO-I-LOGMODVAL,     modified page pad len to 25
$
$ DBO/MOD PARTS MAKE/PAGE_PAG_LEN=45/LOG
%DBO-I-LOGMODROO, modifying root file D:PARTS.ROO;2
%DBO-I-LOGMODSTO, modifying storage area MAKE
%DBO-I-LOGMODVAL,     modified page pad len to 45
$
$ PIPE DBO/DUMP/HEADER/OPT=DEBUG PARTS -
    | SEARCH SYS$PIPE: PAG_PAD_LEN, PAG_DBID
PAG_DBID = 1.  PAG_LEN = 1024.  PAG_PAD_LEN = 45.
PAG_DBID = 2.  PAG_LEN = 1024.  PAG_PAD_LEN = 25.
PAG_DBID = 3.  PAG_LEN = 1024.  PAG_PAD_LEN = 25.
PAG_DBID = 4.  PAG_LEN = 1024.  PAG_PAD_LEN = 25.
```

2.2.23 Monitor Memory Management Enhancements

Previously, the Oracle CODASYL DBMS Monitor (DBMMON) process would map each database global (TROOT) section into P0 virtual address space. This could, in some cases, consume a significant portion of the 1GB available space and could also result in the virtual address space becoming sufficiently fragmented such that the monitor would be unable to open a database.

As a possible workaround the monitor process can be restarted.

With current releases the impact of this virtual memory fragmentation has been somewhat reduced. The DBMMON process now maps database global sections that use PROCESS or RESIDENT SHARED MEMORY=PROCESS into 64-bit P2 virtual address space. In addition, on OpenVMS Integrity Server systems, the executable code of the DBMMON process is mapped into 64-bit P2 virtual address space further reducing the amount of P0 virtual address space consumed.

To specify PROCESS or RESIDENT shared memory, use the DBO/MODIFY command:

```
$ DBO/MODIFY/MEMORY_MAPPING=(PROCESS,[RESIDENT]) db-name.roo
```

2.2.24 Syntax for Prestart Transaction Timeout

The ability to enable and disable prestarted transactions and define a timeout value for a prestarted transaction has been available on Oracle CODASYL DBMS since version 7.1 but was not documented.

For example:

```
$ dbo/modify/transaction=prestart=(enabled,timeout=120) SAMPLE
%DBO-I-LOGMODROO, modifying root file USER2:[DB_TEST]SAMPLE.ROO;1
%DBO-I-LOGMODFLG,     enabled prestarted transactions
%DBO-I-LOGMODFLG,     modified prestarted transaction timeout to 120 seconds
```

The PRESTART option to TRANSACTION accepts one or more of the following options:

- ENABLED

  Sets a database attribute to ensure that prestarted transactions are used by default.

  ```
  $ DBO/MODIFY/TRANSACTION=PRESTART:ENABLED dbroot
  ```

  Prestarted transactions are enabled by default when a database is created.

- NOENABLED

  Sets a database attribute to ensure that prestarted transactions are not used by default.

- TIMEOUT=n

  When prestarted transactions are enabled this timeout will automatically abort the transaction.

  ```
  $ DBO/MODIFY/TRANSACTION=PRESTART=(TIMEOUT=n) dbroot
  ```

  Here 'n' is a value (range 0:3600) in seconds. This value represents the number of seconds to wait before aborting the prestarted transaction. Timing out the prestarted transaction may prevent snapshot file growth in environments where servers stay bound to the database with long periods of inactivity.

Additionally, a process will be forced to obtain a new transaction sequence number (TSN) if the same TSN has been reused throughout the duration of the prestarted transaction timeout interval. This permits processes that constantly reuse TSNs to periodically obtain a new TSN, thus preventing excessive snapshot growth.

2.2.25 Default Behavior Change, New Syntax for DBO/RECOVER /CONFIRM

The default behavior for confirming DBO/RECOVER operations that require user's intervention has been changed.

Typically, intervention is required if a missing or out-of-sequence after-image journal (AIJ) file is encountered. The default could be changed by specifying the /NOCONFIRM qualifier on the command line, in which case processing would continue.

The old behavior for batch mode operations was /NOCONFIRM and an exception would be raised if you attempt to specify /CONFIRM. This assumed that the user did not want to terminate the recovery and would continue applying journals, even if a missing AIJ sequence was detected. Skipping the recovery of an AIJ file could result in loss of data or invalid indexes. It was possible that these errors could go undetected until a full verify operation was performed.

The default qualifiers for interactive recoveries continues to be /CONFIRM, which prompts the user to see if he wants to continue, and /NOCONFIRM for batch mode. However, the behavior of the /NOCONFIRM qualifier has changed. Now, DBO/RECOVER /NOCONFIRM, in either interactive or batch mode, will cause the recovery to terminate at the point where an out-of-sequence AIJ file is detected.

Two keywords are available on /CONFIRM to provide compatibility with the behavior of prior releases.

- /CONFIRM=CONTINUE

  Do not prompt the user; if a sequence gap is detected on the next AIJ file to be rolled forward, ignore the missing AIJ file and continue rolling forward.

- /CONFIRM=ABORT

  Do not prompt the user; if a sequence gap is detected on the next AIJ roll forward, terminate the recovery at this point.

  Effectively, /NOCONFIRM has the same meaning as /CONFIRM=ABORT.

The following example shows the /CONFIRM=CONTINUE syntax used to allow DBO/RECOVER to continue rolling forward if a sequence gap is detected:

```
$ DBO/RECOVER /CONFIRM=CONTINUE/LOG/ROOT=USER$TEST:prod_db FAIJBCK1,FAIJBCK2,FAIJBCK4
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PROD_DB.ROO;1
...
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]FAIJBCK4.AIJ;1
    at 25-FEB-2019 17:26:04.00
%DBO-W-AIJSEQAFT, incorrect AIJ file sequence 8 when 7 was expected
%DBO-I-AIJONEDONE, AIJ file sequence 8 roll-forward operations completed
%DBO-I-LOGRECOVR, 1 transaction committed
%DBO-I-LOGRECOVR, 0 transactions rolled back
%DBO-I-LOGRECOVR, 0 transactions ignored
%DBO-I-AIJNOACTIVE, there are no active transactions
%DBO-I-AIJSUCCES, database recovery completed successfully
%DBO-I-AIJNXTSEQ, to continue this AIJ file recovery, the sequence number
    needed will be 9
%DBO-I-AIJALLDONE, after-image journal roll-forward operations completed
%DBO-I-LOGSUMMARY, total 3 transactions committed
%DBO-I-LOGSUMMARY, total 0 transactions rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJSUCCES, database recovery completed successfully
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery, the sequence number
    needed will be 9
%DBO-I-AIJNOENABLED, after-image journaling has not yet been enabled
```

The following example shows the /CONFIRM=ABORT syntax used so that DBO/RECOVER will not continue rolling forward if a sequence gap is detected.

```
DBO/RECOVER /CONFIRM=ABORT/LOG/ROOT=USER$TEST:prod_db FAIJBCK1,FAIJBCK2,FAIJBCK4
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PROD_DB.ROO;1
...
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]FAIJBCK4.AIJ;1
    at 25-FEB-2019 17:27:42.29
%DBO-W-AIJSEQAFT, incorrect AIJ file sequence 8 when 7 was expected
%DBO-E-AIJRECESQ, AIJ roll-forward operations terminated due to sequence error
%DBO-I-AIJALLDONE, after-image journal roll-forward operations completed
%DBO-I-LOGSUMMARY, total 2 transactions committed
%DBO-I-LOGSUMMARY, total 0 transactions rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery, the sequence number
    needed will be 7
%DBO-I-AIJNOENABLED, after-image journaling has not yet been enabled
```

2.2.26 DBO/MODIFY/RESTRUCTURE Data Type Change Definitions Enhanced

Syntax has been added to the DBO/MODIFY/RESTRUCTURE data type change definition. In addition to INITIALIZE ALWAYS and INITIALIZE ON ERROR, the syntax now allows TRUNCATE ALWAYS and CONTINUE ON ERROR as follows:

```
DEFINE change-name MODIFY record-name
    ITEM item-name TYPE new-data-type
        INITIALIZE ALWAYS
        INITIALIZE ON ERROR
        CONTINUE ON ERROR
        TRUNCATE ALWAYS
```

The behavior of each clause is as follows:

* INITIALIZE ALWAYS

  The item occurrences are always initialized. The actual data may be loaded later.

  This clause is useful when you expect that the item cannot be converted, and do not want to waste time attempting it.

* INITIALIZE ON ERROR

  DBO/MODIFY/RESTRUCTURE attempts to convert the item occurrences, and if the conversion fails, the item is initialized. An error message is displayed to show the location of failed record conversions.

  This clause is useful when most, but not all, of the item occurrences are expected to be converted. Only the failed item occurrences may be loaded later.

* CONTINUE ON ERROR

  DBO/MODIFY/RESTRUCTURE attempts to convert the item occurrences, and if the conversion fails, the item is left as it is, which means the value is unpredictable. An error message shows the location of the failed record.

  This clause is useful when most of the item occurrences are expected to be converted successfully. If the loading of the failed items is planned for later, then you can expedite the change execution now by not having to initialize the items.

* TRUNCATE ALWAYS

  DBO/MODIFY/RESTRUCTURE truncates the character string without bothering to check if the characters being cut off are spaces or tabs.

  This clause is useful when you are shortening a character type and are sure that the characters being truncated may be just discarded.

2.2.27 Hot Standby Status Symbols From DBO /SHOW AFTER_JOURNAL /BACKUP_CONTEXT

Additional DCL symbols indicating the Hot Standby replication state are now created by the DBO /SHOW AFTER_JOURNAL /BACKUP_CONTEXT command.

The symbol names are listed below:

- DBM$HOT_STANDBY_STATE - Contains the current replication state. Possible state strings and the description of each state are listed below:
  - "Inactive" - Inactive
  - "DB_Bind" - Binding to database
  - "Net_Bind" - Binding to network
  - "Restart" - Replication restart activity
  - "Connecting" - Waiting for LCS to connect
  - "DB_Synch" - Database synchronization
  - "Activating" - LSS server activation
  - "SyncCmpltn" - LRS synchronization redo completion
  - "Active" - Database replication
  - "Completion" - Replication completion
  - "Shutdown" - Replication cleanup
  - "Net_Unbind" - Unbinding from network
  - "Recovery" - Unbinding from database
  - "Unknown" - Unknown state or unable to determine state

- DBM$HOT_STANDBY_SYNC_MODE - Contains the current replication synchronization mode when replication is active. Possible synchronization mode strings are listed below:
  - "Cold"
  - "Warm"
  - "Hot"
  - "Commit"
  - "Unknown"

2.2.28 Server Output File Names As Database Attributes

As an alternative to using logical names to control server log file names and locations these names can also be controlled by database attributes.

The DBO /SET SERVER /OUTPUT=filespec command can be used to specify the default output file specification for several of the database server processes. Existing logical names are still valid and supported and will override the database attribute if defined. If the output file specification is empty, the entry is disabled. Use DBO/DUMP /HEADER to see the server log files defined as database attributes.

Note

In the case of Hot Standby configurations the DBO /SET SERVER commands should not be used on the STANDBY database, only the MASTER. Using DBO /SET SERVER commands on the STANDBY database updates the root and Hot Standby will no longer start.

Valid values for the "servertype" parameter and the matching logical name are:

Table 2–1 Server Types and Logical Names

Server	Servertype	Logical Name
AIJ Backup Server	ABS	DBM$BIND_ABS_LOG_FILE
AIJ Log Server	ALS	DBM$BIND_ALS_OUTPUT_FILE
AIJ Log Roll-Forward Server	LRS	DBM$BIND_LRS_OUTPUT_FILE
AIJ Log Catch-Up Server	LCS	DBM$BIND_LCS_OUTPUT_FILE
Database Recovery Server	DBR	DBM$BIND_DBR_LOG_FILE
Row Cache Server	RCS	DBM$BIND_RCS_LOG_FILE

The /LOG qualifier can be used to display a log message at the completion of the DBO /SET operation.

Examples of using the "DBO /SET SERVER /OUTPUT=filespec servertype" command follow.

```
$ DBO /SET SERVER RCS /OUTPUT=RCS_PID.LOG /LOG DUA0:[DB]MYDB.ROO
$ DBO /SET SERVER ALS /OUTPUT=ALS$LOGS:ALS_DB1.LOG DUA0:[DB1]MFP.ROO
$ DBO /SET SERVER LRS /OUTPUT="" DUA0:[ZDB]ZDB.ROO
$ DBO /SET SERVER DBR /OUTPUT=DBR$LOGS:DBR.LOG DUA0:[ADB]ADB.ROO
```

3
Problems Corrected

This chapter describes software errors that have been corrected in Oracle CODASYL DBMS 7.4 versions.

3.1 Problems Corrected in Oracle CODASYL DBMS Release 7.4.1.5

3.1.1 DBO Recover Returns DBO-F-AIJNORCVR Even When Correct Journal Specified

Bug 18028302

When journals were backed up using the /NOQUIET_POINT qualifier, it was possible for the DBO Recover operation to request an older journal than necessary when a recovery operation was initiated.

In the following example, the DBO Restore operation requests journal sequence number 3 as the first journal needed for the DBO Recover command. However, DBO Recover gives an error when that journal is specified and requests an older journal, sequence number 2.

```
$ DBO/RESTORE/NOLOG/NOCDD/NORECOVER TEST
%DBO-I-AIJRECFUL, Recovery of the entire database starts with AIJ file sequence 3
...
$ DBO/RECOVER/LOG TEST_BACKUP_3
%DBO-I-LOGRECDB, recovering database file DISK:[DIRECTORY]TEST.ROO;1
%DBO-F-AIJNORCVR, recovery must start with journal sequence 2
```

Retrying the operation with journal sequence number 2 succeeds, although no transactions from that journal are actually applied to the restored database.

DBO/RECOVER was using incorrect criteria when determining which journal was needed to begin recovery. The journal requested by DBO/RESTORE was the correct journal. DBO/RECOVER now uses the journal specified by DBO/RESTORE when determining the journal needed to begin recovery of the database.

3.1.2 DBO/SHOW STATISTICS Process Monitor Counters Incorrect on RAD Systems

When monitoring database statistics using the DBO/SHOW STATISTICS utility the displayed counters could be incorrect when the Process Monitor option was selected from the tools "!" menu. The numbers were only incorrect if the system CPU supports the RAD/NUMA memory feature. When process monitoring was selected, the counters displayed would represent all users, not just the selected user, as expected.

DBO/SHOW STATISTICS has been corrected to only show counters unique to the selected process. There is no way to work around this problem.

3.1.3 Unexpected BUGCHECK from DBO/SHOW STATISTICS

DBO/SHOW STATISTICS could bugcheck when displaying some Row Cache statistics. This problem only occurs in Oracle CODASYL DBMS release V7.4.1.3.

The screens affected are:

- Row Cache Computations
- Row Cache Overview

3.1.4 Unexpected Bugcheck When Using DBO/SHOW STATISTICS /CONFIGURE Qualifier

Bug 32562913

In prior releases of Oracle CODASYL DBMS DBO/SHOW STATISTICS could bugcheck when accessing a database with ROW CACHE enabled.

The following example shows the reported problem:

```
$ DBO/SHOW STATISTICS PARTS.ROO -
    /CONFIGURE=REPORT.CFG -
    /NOINTERACTIVE -
    /WRITE_REPORT_DELAY=5 -
    /REPORT_SCREEN=(INCLUDE="checkpoint*")
%COSI-F-BUGCHECK, internal consistency failure
%DBO-I-BUGCHKDMP, generating bugcheck dump file USERS2:[BUGCHECK]DBOBUGCHK.DMP;
```

The bugcheck dump would have a footprint similar to the following:

```
***** Exception at 808A1530 : DBO74\DIOCCH$FREE_RCCB + 00000040
%SYSTEM-F-ACCVIO, access violation, reason mask=00,
    virtual address=0000000000000080, PC=FFFFFFFF808A1530, PS=0000001B
Saved PC = 80A8E020 : DBO74\KUTDIS$WRITE_REPORT + 000012D0
Saved PC = 80A24500 : DBO74\KUTDIS$NEW_NODE_ASTX + 00000C20
Saved PC = 8091D360 : DBO74\KUTDIS$DISPATCH + 00000160
Saved PC = 8090C1E0 : DBO74\KUT$DISPLAY + 000045D0
   .
   .
   .
Saved PC = 7AD97480 : Image DCL + 0007D480
```

3.1.5 DBO/SHOW STATISTICS Did Not Recognize Options Keywords Longer Than Four Characters

Bug 37482605

The DBO/SHOW STATISTICS utility would ignore any option specified in the /OPTIONS qualifier if the option required more than four characters to be considered unique. For example, even though /OPTIONS=LOG_STALL_ALARM is specified in the following command to prevent short duration stalls from being logged, they were logged anyway.

```
$ DBO/SHOW STATISTICS -
    /STALL_LOG=STALLS.LOG -
    /ALARM=300 -
    /OPTION=LOG_STALL_ALARM -
    MY_DATABASE
```

The following options would be ignored by DBO/SHOW STATISTICS.

- LOG_STALL_LOCK
- NOLOG_STALL_LOCK
- LOG_STALL_ALARM
- NOLOG_STALL_ALARM

Any option that requires more than four characters is now properly parsed.

3.1.6 OPCOM Messages Now Always Entered in Oracle CODASYL DBMS Monitor Log File

In prior releases, any events that may have been broadcast by the OpenVMS OPCOM facility would be lost if the database was not configured with /ALERT_OPERATOR=ENABLED option. With this update, any message that would have been sent to OPCOM is now entered into the Oracle CODASYL DBMS Monitor log file, regardless of the /ALERT_OPERATOR setting.

In some system environments, this may lead to a much higher growth rate in the size of the monitor log file, especially if features such as database security auditing are enabled. Oracle advises customers to monitor the size of the monitor log file, and if it becomes excessive, implement a procedure to periodically re-open the log and purge older logs. That is, issue the DBO/MONITOR REOPEN_LOG command.

3.2 Problems Corrected in Oracle CODASYL DBMS Release 7.4.1.3

3.2.1 DBO/RECOVER Fails With DBO-F-CANTSYNCTSNS Using Correct Journal

Bug 33906596

After restoring a database, DBO/RECOVER would sometimes fail with the following error, even though the correct journal was specified in the command.

```
%DBO-F-BACKUPNOAIJ, After Image Journaling was enabled after
    the database was backed up or has since been disabled and reinitialized
-DBO-F-CANTSYNCTSNS, Last committed TSN 610 in the after image
    journal file exceeds last committed TSN 609 in the database root
```

When database prestarted transactions were enabled (the default), there was a small timing window where the database backup could be assigned a transaction sequence number (TSN) that is less than the TSN stored in the active after-image journal. Attempts to use that journal to recover the restored database would fail with DBO-F-CANTSYNCTSNS.

The following scenario could lead to this problem.

1. There is an active user in the database that has committed an update and is sitting idle. By default, that user has a prestarted transaction active, even if no explicit READY was executed. The prestarted transaction is assigned a TSN at the time that the previous transaction commits.
2. An after-image backup is executed. The newly activated journal is assigned the next database TSN. Executing a DBO/DUMP /AFTER_JOURNAL command on the journal open record displays this value as "Last Commit TSN is".
3. The idle user executes another transaction and commits. Its older TSN becomes the last committed TSN for the database.
4. An online database backup is performed. The backup will record the last committed database TSN. This is displayed as "Latest full backup transaction sequence number is" in the DBO/DUMP/HEADER output.
5. A journal backup is performed for the journal that was active at the time of the database backup.
6. The database is restored.
7. A DBO/RECOVER command is executed, attempting to apply the backup of the journal that was active at the time the database full backup was performed.

When a DBO/RECOVER command is started, DBO verifies that the specified journal is applicable to the database. DBO compares the last commit TSN in the journal with the latest backup TSN in the restored database. If the database TSN is greater than or equal to the journal's TSN then the journal is applied. Otherwise, a DBO-F-CANTSYNCTSNS error is issued. In the scenario described above, due to an old, prestarted transaction sequence number, this test would fail, even though the journal should have been applicable.

To workaround this problem, provide the prior AIJ backup as well as the expected journal to the DBO/RECOVER command. In the following example, the expected journal is AIJ_BACKUP_1. For the command to succeed, the prior journal, AIJ_ BACKUP_0 is specified along with AIJ_BACKUP_1.

```
$ DBO/RECOVER/LOG AIJ_BACKUP_0, AIJ_BACKUP_1
```

This problem will not occur if prestarted transactions are disabled.

```
$ DBO/MODIFY/TRANSACTION=PRESTART=(NOENABLED) dbroot
```

Note that disabling prestarted transactions will likely have a noticeable impact on performance in high transaction rate applications.

This problem has been corrected in Oracle CODASYL DBMS. Now, when a quiet-point after-image journal backup is executed, all idle processes that have a prestarted transaction active are forced to obtain a new TSN when the next transaction is explicitly started.

3.2.2 Error From Disk Write Results in Hang

Bug 34156672

An I/O intensive application would sometimes hang in Oracle CODASYL DBMS in LEF state if an error was encountered when writing data page buffers to disk. Examination of the process stack revealed that an exception occurred, and DBMS was waiting for a SYS$SYNCH system call to return.

DBMS writes one or more modified pages to disk concurrently in a single batch. In the unusual situation where no write requests succeeded, DBMS would still wait for the write requests to complete, even though none were started.

This problem has been corrected in this release of Oracle CODASYL DBMS. If no disk writes were successfully initiated DBMS will no longer attempt to wait for the writes to complete.

3.2.3 Unexpected COSI-F-EXQUOTA Error Using Buffer Objects

Bug 34156672

It was possible for a *COSI-F-EXQUOTA* error to be signaled when the BUFFER OBJECT feature was enabled and many data page updates were being written to disk concurrently. An error message like the following may be returned, or a hang might occur.

```
%DBMS-F-CANTWRITEDBS, error writing pages 1:2-3
-COSI-F-EXQUOTA, exceeded quota
```

DBMS was previously reserving enough BUFFER OBJECT resources to have an asynchronous write active on each data page buffer in the buffer pool. For example, if the database user was allocated 20 buffers, then a maximum of 20 asynchronous buffer object writes were allowed. However, there are scenarios where more than 20 concurrent page buffer writes are possible.

To avoid this problem, the database administrator can disable data page buffer objects. This is controlled using a DBO command or by changing the controlling logical.

```
$ DBO/SET BUFFER_OBJECT /DISABLE=(PAGE) SHIPPING
$ DEASSIGN DBM$BIND_PAGE_BUFOBJ_ENABLED
```

or

```
$ DEFINE DBM$BIND_PAGE_BUFOBJ_ENABLED 0
```

This problem has been corrected in this release of DBMS. DBMS now allocates sufficient resources to allow the maximum possible concurrent database page buffer object writes.

3.2.4 Invalid DBO Show Statistics "File IO Overview" Screen PgDis Field Values

When the DBO/SHOW STATISTICS command was executed to display the "File IO Overview" screen, and the "PgDis" - pages discarded - field numeric values were too large to fit in this fixed length 4 digit field, the value "****k" was displayed, where the asterisks followed by the "k" indicate that the numeric value could not fit in this field, even though the number was scaled by 1000.

This scaling problem has been fixed. The value will be scaled to fit on the screen. If the value exceeds 10000 it will be scaled by 1000 with a "k" suffix, and so on. An "m" suffix indicates scaling by 1000^2, a "g" suffix indicates scaling by 1000^3, and a "t" suffix indicates scaling by 1000^4.

This problem has been corrected in Oracle CODASYL DBMS release V7.4.1.3. Note that if "****k" was displayed then that by itself indicates excessive pages discarded and investigation is encouraged.

3.2.5 Stale Entries Written to Statistics Lock Timeout Log

Bug 34957220

If the DBO/SHOW STATISTICS command is used to create a lock timeout logfile (when the /LOCK_TIMEOUT_LOG qualifier is specified) the logfile could contain entries from events that occurred before the utility was invoked.

In the following output, note the timestamp for the log creation "4-JAN-2023 00:10:04.34" is newer than the entry in the log with a timestamp of "2-JAN-2023 21:25:04.50".

```
 Oracle DBMS V7.4-10 Performance Monitor Lock Timeout Log
 Database DEV:[DIR]TEST.ROO;1
 Lock Timeout Log created  4-JAN-2023 00:10:04.34
00078E7B:1   4-JAN-2023 16:25:04.5290650 - waiting for record 520:2370514:0 (EX)
00079A6A:1   2-JAN-2023 21:25:04.5008225 - waiting for record 520:2354672:0 (EX)
```

This problem was caused by incorrect initialization of the lock timeout data structure used by DBO/SHOW STATISTICS.

This problem has been corrected in Oracle CODASYL DBMS Release 7.4.1.3. The logging data structure is now properly initialized. Only timeouts that occur after the DBO/SHOW STATISTICS command is invoked will be logged.

3.2.6 System Space Buffers not Freed Up After a DBO/CLOSE

Bug 13590034

In prior releases of Oracle CODASYL DBMS, a database that has SYSTEM SPACE BUFFERS buffers enabled (SHARED_MEMORY/TYPE=SYSTEM) never freed the memory that was allocated when the database was closed. For example, after some sequence of database open and close the following error would be reported.

```
$ DBO/OPEN USERS1:[TESTER.DATABASE]TEST.ROO
%DBO-F-CANTOPENDB, database could not be opened as requested
-DBMS-F-CANTCREGBL, error creating and mapping database global section
-SYSTEM-F-INSFSPTS, insufficient SPTEs available
%DBO-W-CANTOCDB, Error encountered while opening or closing database file
    USERS1:[TESTER.DATABASE]TEST.ROO;1
```

In some rare cases a system crash could occur.

The following example shows how to enable SYSTEM SPACE BUFFERS.

```
$ DBO/SET SHARED_MEMORY/TYPE=SYSTEM TEST
%DBO-I-MODIFIED, Shared Memory state modified
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
$
```

Once system space buffers are enabled, a dump of the header will indicate that the database will be mapped into system space. Below is an abbreviated section of the Database Parameters that indicate that the database will be mapped in system space. For example:

```
$ DBO/DUMP/HEADER TEST
   .
   .
   .
Database Parameters:
    Root filename is "USERS1:[TESTER.DATABASE]TEST.ROO;1"
    Created at 19-SEP-2023 09:07:46.42
    Physical database created at 19-SEP-2023 09:07:46.42
    Oracle CODASYL DBMS structure level is 74.0
    Maximum user count is 50
    Maximum node count is 16
    Database open mode is AUTOMATIC
    Database close mode is AUTOMATIC
    Database will be mapped in system space
   .
   .
   .
$
```

Conversely, SYSTEM SPACE BUFFERS can be toggled back to using process space by the qualifier on the DBO Set command. For example:

```
$ DBO/SET SHARED_MEMORY/TYPE=PROCESS TEST.ROO
%DBO-I-MODIFIED, Shared Memory state modified
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
$
```

If you are experiencing this type of error, the work around would be to revert the database back to being mapped into process space, followed by a system reboot to regain usage of all of the available system page table entries.

This problem has been fixed in Oracle CODASYL DBMS V7.4.1.3. Any system space buffers that were left allocated by a previous version of Oracle CODASYL DBMS will still require a system reboot for reclamation. However, with this release open and close of the database will correctly return system memory to a free memory pool for future use.

3.3 Problems Corrected in Oracle CODASYL DBMS Release 7.4.1.0

3.3.1 DBO/BACKUP /AFTER /NOEDIT_FILENAME Still Applying Edit String

Since Oracle CODASYL DBMS version 7.3.3, if the /NOEDIT_FILENAME qualifier is specified in the DBO/BACKUP/AFTER command, then no edit strings should be applied to the After Image Journal backup file name created by this command.

However, in some cases, DBO would erroneously use the default AIJ backup file edit string when /NOEDIT_FILENAME qualifier was used. If any of the qualifiers /FORMAT=NEW_TAPE, /LIBRARIAN, or /ENCRYPT were specified in the DBO/BACKUP/AFTER command would work as expected.

3.3.2 Unexpected "Improperly handled condition" Reported by DBO /RESTRUCTURE Command

Bug 18908281

In prior releases of Oracle CODASYL DBMS it was possible that DBO /MODIFY /RESTRUCTURE would erroneously fail with an "Improperly handled condition" error and register dump.

The following example shows this problem.

```
$ DBO/MODIFY/RESTRUCTURE PARTS MAKE -
    /ALLOC=250 -
    /THRESHOLD=(72,72,72)
%DBO-I-DBBOUND, bound to database "DISK:[DIRECTORY]PARTS.ROO;1"
SET NOCONFIRM
CREATE CHGMAKEAREA
%DBO-I-LOGFILACC, created change file DISK:[DIRECTORY]CHGMAKEAREA.RCF;1
%DBO-I-LOGFILACC, opened change file DISK:[DIRECTORY]CHGMAKEAREA.RCF;1
DEFINE CHG1 RELOAD AREA MAKE -
SEQUENCE -
  loop -
    one class using set all_class -
    LOOP -
      ONE PART USING SET ALL_CLASS CLASS_PART -
      ALL COMPONENT    USING SET PART_USES -
    ENDLOOP -
  endloop -
ENDSEQUENCE
%DBO-F-NOCALCSET, CALC set ALL_CLASS cannot be used in USING clause

  Improperly handled condition, image exit forced by last chance handler.
    Signal arguments:   Number = 0000000000000005
                        Name   = 0000000000098B8AC
                                 0000000000000001
                                 000000000127D0B0
                                 00000000003E2770
                                 000000000000001B

    Register dump:
   .
   .
   .
```

DBO now correctly traps this error condition.

3.4 Problems Corrected in Oracle CODASYL DBMS Release 7.3.3

3.4.1 Fix Problems with EDIT_FILENAME Qualifier

In prior versions of Oracle CODASYL DBMS, if DBO/BACKUP /AFTER_ JOURNAL /FORMAT=NEW was specified, a path was taken which ignored using the default edit string and default backup name in the database root and journal structures. The default value in the journal definition, when specified, is for each individual journal file (.aij) and takes precedence over the general default value in the database root (.roo).

Additionally, the DBM$AIJ_LAST_OUTPUT_FILE symbol that the DBO/BACKUP /AFTER_JOURNAL command creates, was not being defined and if a partial backup file specification was used in the command the symbol was not expanded to a full file specification.

Finally, there was a general problem with the DBO/BACKUP /AFTER_JOURNAL /EDIT_FILENAME qualifier that prevented the default EDIT_FILENAME stored in the database root from being ignored when the /NOEDIT_FILENAME qualifier was specified for the DBO/BACKUP /AFTER_JOURNAL command.

These problems have been fixed in Oracle CODASYL DBMS version 7.4.1.5.

3.4.2 Long Running Delete Does Not Advance After-Image Journal Checkpoint

Bug 6323701

If a database bind executes a data manipulation operation, such as ERASE in a loop, that takes a very long time to complete, it might not move its checkpoint location. In extreme circumstances this could lead to all after-image journals filling without the possibility of a backup. This kind of problem is often referred to as a "long verb" issue.

To avoid this problem, try to break up large data manipulation operations into smaller ones.

Oracle CODASYL DBMS now periodically checks to see if it has items on its work queue waiting to be executed. If there is a waiting request it will be processed and then the current operation will continue. This should allow checkpoint locations to be advanced as the journals fill.

3.4.3 Long Running Query Blocks Other Queries

Bug 12972389

If a database user executes a long running query, and it is holding a lock needed by another user, it might not release that lock for a long period of time, causing other users to wait. This kind of problem is often referred to as a "long verb" issue.

Oracle CODASYL DBMS now periodically checks to see if it has items on its work queue waiting to be executed. If there is a waiting request it will be processed and then the current operation will continue. This should prevent most long running queries from blocking other users.

4

Known Problems, Restrictions and Additional Documentation

This chapter describes problems, restrictions and additional documentation relating to Oracle CODASYL DBMS version 7.4.1.5.

Note

Documentation for all Oracle CODASYL DBMS logical names is available using the DBO/SHOW LOGICAL_NAMES /DESCRIPTION command. The command accepts wildcard logical-names (OpenVMS style * and % characters). Omit the logical-name parameter to see a complete list of relevant logical names, and their current values. For example,

```
$ DBO/SHOW LOGICAL_NAMES /UNDEFINED
  "DBM$BIND_ABS_GLOBAL_STATISTICS" = Undefined
  "DBM$BIND_ABS_LOG_FILE" = "ABS_PID.OUT" (LNM$SYSTEM_TABLE)
  "DBM$BIND_ABS_LOG_HEADER" = Undefined
   .
   .
   .
  "DBMS$RUJ" = Undefined
  "DBM$SORT_FILES" = Undefined
$
```

The /DESCRIPTION qualifier can then be used to view more details for the selected logical names.

```
$ DBO/SHOW LOGICAL_NAMES *sort_files /DESCRIPTION
  "DBM$SORT_FILES" = Undefined

        Use this logical name to specify the number of temporary work
        files that a database sort operation is to use if the work files
        are required. The default number of work files is 2 and the
        maximum number is 10. For example:

        $ DEFINE DBM$SORT_FILES   3

        To control the placement of the work files define the various
        SORTWORKn logical names. The SORTWORKn logicals identify the
        device and directory of the work files. These logical names
        must point to a device that has a directory (allowing WRITE
        protection) with the same name as the user's home directory.

           Table: PROCESS
           Values: value between 2 (default) and 10

$
```

4.1 DBO/SHOW STATISTICS Undocumented Qualifiers

The following qualifiers to DBO/SHOW STATISTICS were previously documented in prior release notes and are collected here for reference. Please refer to the *Oracle CODASYL DBMS: Database Administration Reference Manual* for release 7.4.1.5 for complete descriptions of these qualifiers.

- /CLUSTER

  Specifies the list of remote nodes from which statistics collection and presentation are to be performed. The collected statistics are merged with the information for the current node and displayed using the usual statistics screens.

- /COLUMNS

  The dimensions of the DBO/SHOW STATISTICS window default to the user's display size or, in the case of non-interactive mode, 132 columns and 66 rows.

  The DBO/SHOW STATISTICS utility allows you to specify the horizontal dimension using the /COLUMNS=n qualifier. Use this in conjunction with the /ROWS=n qualifier.

- /CONFIGURE

  Specifies the name of a human-readable configuration file to be processed by the DBO/SHOW STATISTICS utility. The configuration file can be created using any editor, or it can be automatically generated from the DBO/SHOW STATISTICS utility using the current run-time configuration settings. The default configuration file type is .cfg.

  If you specify the /CONFIGURE=file-spec qualifier, the configuration file is processed by the DBO/SHOW STATISTICS utility prior to opening the database or the binary input file. If you do not specify this qualifier, all of the variables are the defaults based on command-line qualifiers and logical names.

- /DEADLOCK_LOG

  Records the last deadlock for the processes. There is no method to record each lock deadlock as it occurs.

  The file-spec in the qualifier is the name of the file to which you want all lock deadlock messages to be logged. The lock deadlock messages are written in human-readable format similar to the Lock Timeout History and Lock Deadlock History screens.

- /HOT_STANDBY_LOG

  Specifies the name of the Hot Standby log file. The "Start hot standby logging" option of the Tools menu (enter !) can be used to specify the name of the Hot Standby log file at runtime.

- /LOCK_TIMEOUT_LOG

  Records the last lock timeout message for the processes. There is no method to record each lock timeout as it occurs. The lock timeout messages are written in human-readable format.

- /MULTIPAGE_MAXIMUM

By default the DBO/SHOW STATISTICS utility will write all pages for each display when writing to a report file. For some databases, for example, those with a large number of record cache slots, this output may be quite lengthy. To assist in reducing the size of the output report file, the /MULTIPAGE_MAXIMUM=n qualifier can be used.

- /OPCOM_LOG

  Specifies the name of the file where OPCOM messages broadcast by attached database processes will be sent.

- /OPTIONS

  Allows you to select the collection of by-file or by-area statistics information. This qualifier is only applicable when used in conjunction with the /OUTPUT qualifier.

- /PROMPT_TIMEOUT

  Allows you to specify the user prompt timeout interval, in seconds. The default value is 60 seconds.

- /REPORT_SCREEN

  Allows individual screens to be included or excluded from the STATISTICS.RPT file by specifying a full screen name or parts of a screen name using the REPORT_SCREEN qualifier. The specification may include wild card character "*" to match multiple characters and "%" to match a single character.

- /ROWS

  The dimensions of the DBO/SHOW STATISTICS window default to the user's display size or, in the case of non-interactive mode, 132 columns and 66 rows.

  The DBO/SHOW STATISTICS utility allows you to specify the vertical dimension using the /ROWS=n qualifier. Use this in conjunction with the /COLUMNS=n qualifier.

- /WRITE_REPORT_DELAY

  This qualifier specifies that statistics are to be collected for "n" seconds (default of 60 seconds) and then a report file written and then the DBO/SHOW STATISTICS utility will exit. /WRITE_REPORT_DELAY implies /NOINTERACTIVE.

4.2 Oracle CODASYL DBMS Position on NFS Devices

This release note describes the supported usage of the NFS (Network File System) mounted devices by the Oracle CODASYL DBMS product. NFS devices appear in most regards as local mounted file systems but do not allow the same level of sharing as provided by local OpenVMS devices. In addition these files reside on a non-OpenVMS system (for instance a Linux or Windows system) and are therefore outside any scheme used by DBMS to lock buffers and pages of the database.

Active System Files

When Oracle CODASYL DBMS is actively using database files, these files require specific sharing and locking to guarantee database integrity and recovery. Therefore, because of the limitations of the NFS mounted devices active files such as the database root (.roo), storage areas (.dbs), snapshot files (.snp), row cache work file (.dbc), after image journal files (.aij), before image recovery journal (.ruj) must not reside on an NFS mounted device.

Archived Data Files

Files that are not part of the active system may be stored on NFS mounted devices. For example, DBO /BACKUP /AFTER_JOURNAL can be used to archive an after image journal to a target on an NFS device. Similarly, DBO /BACKUP can perform a full or incremental backup to a DBMS backup file (.dbb) on an NFS device and DBO /RESTORE can use that NFS mounted source for database recovery, along with archived after image files from an NFS device processed by DBO /RECOVER.

Other Miscellaneous Files

Other files that might be used by an Oracle CODASYL DBMS installation include options files, application procedures and sources, backup journals, load format language (.lfl) files, load sequence language (.lsl) files, unload sequence language (.usl) files, data file (.dat), log files, and so on. These sequential files may be stored on and referenced by DBO commands from an NFS mounted device.

Setting Up NFS

Complete instructions for setting up an NFS mounted device is beyond the scope of this release note and customers are directed to use system specific documentation for the server platform and for OpenVMS systems. However, during testing with Oracle CODASYL DBMS we noted the need for the following qualifiers for the TCPIP MOUNT command.

- Use /ADF=CREATE. This ensures that attributes (such as block size and record length) are preserved on the server.
- Use /STRUCTURE=5. This will emulate an ODS-5 device and therefore allow the most complete OpenVMS Files-11 On-Disk Structure emulation.
- Use /TRANSPORT=UDP.

  For example,

```
$ tcpip mount dnfs1:/host="test.company.com"/path="/scratch"
        /stru=5/serve=unix/adf=create/vers=2/tran=udp
```

Read Performance Issues

In prior versions of Oracle CODASYL DBMS a significant performance issue exists when reading sequential files from NFS mounted devices. DBMS uses the RMS read-ahead (RAH) attribute to improve sequential reads but this has an adverse effect when referencing an NFS device. The latest release of DBMS works around this issue by disabling the use of read-ahead when referencing an NFS device and would be the preferred version when using NFS devices.

Disclaimer

This information is provided to answer customer questions and should not be read as an endorsement or guarantee for NFS systems. Oracle expects configuration, functional testing, performance testing, security and integrity of the NFS data to be performed by our customers.

4.3 Limited IEEE Floating Support in Data Definition Language

Starting in release 7.3.1 Oracle CODASYL DBMS provided support for IEEE single and double precision floating point data types within the Data Definition Language (DDL).

Two new keywords have been added to the DDL TYPE clause:

- TYPE IS S_FLOATING ! single-precision floating point
- TYPE IS T_FLOATING ! double-precision floating point

Example:

```
RECORD IS FOO0677
     CHECK IS FOO0677_A EQ FOO0677_B
     ITEM FOO0677_A
        TYPE IS SIGNED BYTE
     ITEM FOO0677_B
        TYPE IS S_FLOATING

RECORD IS FOO0705
     CHECK IS FOO0705_A EQ FOO0705_B
     ITEM FOO0705_A
        TYPE IS PACKED DECIMAL 10 -2
     ITEM FOO0705_B
        TYPE IS T_FLOATING
```

The new data type can be used in SCHEMA and STORAGE_SCHEMA declarations. They can also be used within SUBSCHEMA declarations if the target host language supports the datatype. If an IEEE data item is specified in the SCHEMA or STORAGE_SCHEMA, but is not supported by the host language, you will need to specify a supported comparable data type in the SUBSCHEMA, allowing DBMS to convert internally between the two.

4.4 DBMS-F-FULLAIJBKUP, Partially Journaled Changes Made

Bug 7669735

The *Oracle Rdb and Oracle CODASYL DBMS Guide to Hot Standby Databases* states: "You can stop replication operations by explicitly entering the Replicate After_Journal Stop command on either the standby or master database nodes. Stopping replication on either database terminates replication on both databases."

Although the DBO/REPLICATE AFTER_JOURNAL STOP command may be issued against either Master or Standby to shut down replication, we have determined that there is at least one scenario where the choice is important relating to restarting replication in the future.

If you:

1. stop replication on the Standby,

2. set the old standby to be the new Master,

3. set the old Master to be the new Standby,

4. attempt to restart replication

the operation will fail with a 'FULLAIJBKUP' error when starting the Master.

This is expected behavior. If the Standby is stopped prior to the Master, Oracle CODASYL DBMS cannot determine if there has been any network traffic from the Master between the time that the Standby and Master shut down. Since any such information would be lost and may lead to data inconsistencies, replication will not start.

The workaround for this scenario would be to stop replication on the Master, not the Standby. Table 4–1 has further details.

Table 4–1 Details of Message FULLAIJBKUP

FULLAIJBKUP	partially-journaled changes made; database may not be recoverable
Explanation:	Partially journaled changes have been made to the database. This may result in the database being unrecoverable in the event of database failure; that is, it may be impossible to roll-forward the after-image journals, due to a transaction mis-match or attempts to modify objects that were not journaled. This condition typically occurs as a result of replicating database changes using the Hot Standby feature.
User Action:	IMMEDIATELY perform a full (not by-sequence) quiet-point AIJ backup to clear the AIJ journals, followed immediately by a full (no-quiet-point allowed) database backup.

4.5 Unexpected RCS (Record Cache Server) Termination

It has been observed in internal testing that if the Record Cache Server (RCS) terminates in an uncontrolled fashion this may, under some conditions, cause corruption of the database and/or the After Image Journal file.

When the RCS terminates the database is shut down and a message like the following is written to the monitor log:

```
6-DEC-2007 15:04:17.02 - Received Record Cache Server image termination from 22ED5144:1
  - database name "device:[directory]database.ROO;1" [device] (1200,487,0)
  - abnormal Record Cache Server termination detected
  - starting delete-process shutdown of database:
    - %DBM-F-RCSABORTED, record cache server process terminated abnormally
  - sending process deletion to process 22ED10F9
  - sending process deletion to process 22ECED59
  - sending process deletion to process 22EC0158
  - sending process deletion to process 22EB9543 (AIJ Log server)
  - database shutdown waiting for active users to terminate
```

A future attempt to roll forward the AIJ following a restore of a database backup might fail with a bugcheck dump if this problem has happened.

The only currently known situation where this problem has been observed is if the logical name DBM$BIND_RCS_VALIDATE_SECS is defined to some value and the logical name DBM$BIND_RCS_LOG_FILE at the same time is undefined or defined incorrectly.

To prevent this problem, Oracle recommends any customer using the Record Cache feature either avoid defining the logical name DBM$BIND_RCS_ VALIDATE_SECS, or if this logical name needs to be defined, ensure that DBM$BIND_RCS_LOG_FILE is correctly defined; defined with the /SYSTEM and /EXECUTIVE qualifiers and pointing to a valid file name in an existing directory on a cluster accessible device with sufficient free space.

This recommendation applies to all versions of Oracle CODASYL DBMS.

4.6 VMS$MEM_RESIDENT_USER Rights Identifier Required

Oracle CODASYL DBMS release 7.1 introduced additional privilege enforcement for the database or record cache qualifiers MEMORY_MAPPING=SYSTEM and LARGE_MEMORY. If a database utilizes any of these features then the user account that opens the database must be granted the VMS$MEM_RESIDENT_USER rights identifier. Also, any process attempting to change these attributes, to convert, or restore a database with these attributes enabled must also hold the right.

Oracle recommends that the DBO/OPEN command be used when utilizing these features.

Note

If one or more record cache are defined for the database but row caching is not enabled then the VMS$MEM_RESIDENT_USER identifier is not required even if caches are defined for resident memory.

4.7 ILINK-E-INVOVRINI Error on I64

When linking an application with multiple modules, the following error message may be returned:

```
%ILINK-E-INVOVRINI, incompatible multiple initializations for overlaid section
        section: DBM$UWA_B
        module: M1
        file: DKA0:[BLD]M1.OBJ;1
        module: M2
        file: DKA0:[BLD]SYS.OLB;1
```

On I64 systems, you cannot have a program section that attempts to be initialized a subsequent time where the non-zero portions of the initializations do not match. This is a difference from OpenVMS Alpha systems where the linker permitted such initializations.

This can be seen when linking multiple FORTRAN DML modules, where some modules use the default (non-stream) UWA, and another uses a "naked" invoke, which only contributes an abbreviated contribution to the DBM$UWA_B psect.

For example, A.FOR contains:

```
      PROGRAM AFOR
      INVOKE (SUBSCHEMA = FORTRAN_SUBSCHEMA,
     1       SCHEMA = PARTS,
     2       DATABASE = PARTS)
      CALL BSUB()
      END
```

B.FOR contains:

```
      SUBROUTINE BSUB
      INVOKE
      RETURN
      END
```

On ALPHA, the above code will link and run correctly, However, on I64, the linker will generate the following:

```
%ILINK-E-INVOVRINI, incompatible multiple initializations for overlaid section
        section: DBM$UWA_B
        module: AFOR
        file: A.OBJ
        module: BSUB
        file: B.OBJ
```

4.8 Default Floating Point Format

The Itanium architecture has a 64-bit model and basic system functions similar to the Alpha chip. However, there are some implementation differences between the two platforms that might affect user-written applications.

One of the differences is the availability of hardware-supported floating-point formats. The Itanium architecture implements floating-point arithmetic in hardware using the IEEE floating-point formats, including IEEE single and IEEE double. The Alpha architecture supports both IEEE and VAX floating-point formats in hardware, and OpenVMS compilers generate code using the VAX formats by default, with options (on Alpha) to use IEEE formats. Irrespective of whether it was originally written for VAX or Alpha, an OpenVMS application that uses the default VAX floating-point formats needs to produce equivalent behavior on the Itanium architecture using IEEE formats at the lowest level.

- On OpenVMS Alpha, VAX float is the default. VAX format data is assumed and VAX floating instructions are used.
- On OpenVMS Alpha, you can specify the compiler option /FLOAT=IEEE. In this case, IEEE format data is assumed and IEEE floating instructions are used.
- On OpenVMS I64, IEEE float is the default. IEEE format data is assumed and IEEE floating instructions are used.
- On OpenVMS I64, you can specify the compiler option /FLOAT=D_FLOAT or /FLOAT=G_FLOAT.

When you compile an OpenVMS application that specifies an option to use VAX floating-point on the Itanium architecture, the compiler automatically generates code for converting floating-point formats. Whenever the application performs a sequence of arithmetic operations, this code does the following:

1. Converts VAX floating-point formats to either IEEE single or IEEE double floating-point formats.
2. Performs arithmetic operations in IEEE floating-point arithmetic.
3. Converts the resulting data from IEEE formats back to VAX formats.

Note that where no arithmetic operations are performed (VAX float fetches followed by stores), conversions will not occur. The code handles such situations as moves. VAX floating-point formats have the same number of bits and precision as their equivalent IEEE floating-point formats. For most applications, the conversion process will be transparent. In a few cases, arithmetic calculations might have different results because of the following differences between VAX and IEEE formats:

- Values of numbers represented
- Rounding rules
- Exception behavior

4.9 Fatal Error With SHARED SYSTEM MEMORY or LARGE MEMORY Enabled in Galaxy Environment

When GALAXY support is enabled in an OpenVMS Galaxy environment, a `%SYSTEM-F-INSFMEM, insufficient dynamic memory error` message may be returned when mapping record caches or opening the database. One source of this problem specific to a Galaxy configuration is running out of Galaxy Shared Memory regions. For Galaxy systems, GLX_SHM_REG is the number of shared memory region structures configured into the Galaxy Management Database (GMDB).

While the default value of 64 regions (for OpenVMS versions through at least V7.3-1) might be adequate for some installations, sites using a larger number of databases or record caches when the SHARED MEMORY IS SYSTEM or LARGE MEMORY IS ENABLED features are enabled may find the default insufficient.

If a `%SYSTEM-F-INSFMEM, insufficient dynamic memory` error is returned when mapping record caches or opening databases, Oracle Corporation recommends that you increase the GLX_SHM_REG parameter by two times the sum of the number of record caches and number of databases that might be accessed in the Galaxy at one time. As the Galaxy shared memory region structures are not very large, setting this parameter to a higher than required value does not consume a significant amount of physical memory. It also may avoid a later reboot of the Galaxy environment. This parameter must be set on all nodes in the Galaxy.

Galaxy Reboot Required

Changing the GLX_SHM_REG system parameter requires that the OpenVMS Galaxy environment be booted from scratch. That is, all nodes in the Galaxy must be shut down and then the Galaxy reformed by starting each instance.

To enable Galaxy support, issue the command:

```
$ DBO/SET GALAXY/ENABLED <db>
```

To enable SYSTEM SHARED MEMORY, issue the command:

```
$ DBO/MODIFY/MEMORY_MAPPING=SYSTEM <db>
```

To enable LARGE MEMORY for record cache, issue the command:

```
$ DBO/CACHE/MODIFY/LARGE_MEMORY <db> <cache>
```

4.10 Oracle CODASYL DBMS and OpenVMS ODS-5 Volumes

The OpenVMS Version 7.2 release introduced an Extended File Specifications feature, which consists of two major components:

- A new, optional, volume structure, ODS-5, which provides support for file names that are longer and have a greater range of legal characters than in previous versions of OpenVMS.
- Support for "deep" directory trees.

ODS-5 was introduced primarily to provide enhanced file sharing capabilities for users of Advanced Server for OpenVMS 7.2 (formerly known as PATHWORKS for OpenVMS), as well as DCOM and JAVA applications.

In some cases, Oracle CODASYL DBMS performs its own file and directory name parsing and explicitly requires ODS-2 (the traditional OpenVMS volume structure) file and directory name conventions to be followed. Because of this knowledge, Oracle does not support any Oracle CODASYL DBMS database file components (including root files, storage area files, after-image journal files, record cache backing store files, database backup files, after-image journal backup files, and so forth) that utilize any non-ODS-2 file naming features. For this reason, Oracle recommends that Oracle CODASYL DBMS database components not be located on ODS-5 volumes.

Oracle CODASYL DBMS does support database file components on ODS-5 volumes provided that all of these files and directories strictly follow the ODS-2 file and directory name conventions. In particular, all file names must be specified entirely in uppercase and special characters in file or directory names are forbidden.

4.11 Carryover Locks and NOWAIT Transaction Clarification

In NOWAIT transactions, the BLAST (Blocking AST) mechanism cannot be used. For the blocking user to receive the BLAST signal, the requesting user must request the locked resource with WAIT (which a NOWAIT transaction does not do).

Oracle CODASYL DBMS defines a resource called NOWAIT, which is used to indicate that a NOWAIT transaction has been started. When a NOWAIT transaction starts, the user requests the NOWAIT resource. All other database users hold a lock on the NOWAIT resource so that when the NOWAIT transaction starts, all other users are notified with a NOWAIT BLAST.

The BLAST causes blocking users to release any carryover locks. There can be a delay before the transactions with carryover locks detect the presence of the NOWAIT transaction and release their carryover locks. You can detect this condition by examining the stall messages. If the "Waiting for NOWAIT signal (CW)" stall message appears frequently, the application is probably experiencing

a decrease in performance, and you should consider disabling the carryover lock behavior.

4.12 Both Application and Oracle CODASYL DBMS Using SYS$HIBER

In application processes that use Oracle CODASYL DBMS and the $HIBER system service (possibly by RTL routines such as LIB$WAIT), it is important that the application ensures that the event being waited for has actually occurred. Oracle CODASYL DBMS uses $HIBER/$WAKE sequences for interprocess communications particularly when the ALS (AIJ log server) feature is enabled.

The Oracle CODASYL DBMS use of the $WAKE system service can interfere with other users of $HIBER (such as the routine LIB$WAIT) that do not check for event completion, possibly causing a $HIBER to be unexpectedly resumed without waiting at all.

To avoid these situations, consider altering the application to use a code sequence that avoids continuing without a check for the operation (such as a delay or a timer firing) being complete.

The following pseudo-code shows one example of how a flag can be used to indicate that a timed-wait has completed correctly. The wait does not complete until the timer has actually fired and set TIMER_FLAG to TRUE. This code relies on ASTs being enabled.

```
ROUTINE TIMER_WAIT:
BEGIN
! Clear the timer flag
TIMER_FLAG = FALSE

  ! Schedule an AST for sometime in the future
  STAT = SYS$SETIMR (TIMADR = DELTATIME, ASTRTN = TIMER_AST)
  IF STAT <> SS$_NORMAL
  THEN BEGIN
       LIB$SIGNAL (STAT)
       END

  ! Hibernate.  When the $HIBER completes, check to make
  ! sure that TIMER_FLAG is set indicating that the wait
  ! has finished.
  WHILE TIMER_FLAG = FALSE
  DO  BEGIN
      SYS$HIBER()
      END
  END

ROUTINE TIMER_AST:
  BEGIN
  ! Set the flag indicating that the timer has expired
  TIMER_FLAG = TRUE

  ! Wake the main-line code
  STAT = SYS$WAKE ()
  IF STAT <> SS$_NORMAL
  THEN BEGIN
       LIB$SIGNAL (STAT)
       END
  END
```

In OpenVMS V7.2, the LIB$WAIT routine has been enhanced through the FLAGS argument (with the LIB$K_NOWAKE flag set) to allow an alternate wait scheme (using the $SYNCH system service) that can avoid potential problems with multiple code sequences using the $HIBER system service.

4.13 Record cache Not Allowed While Hot Standby Replication is Active

The Record Cache feature may not be enabled on a Hot Standby database while replication is active. The Hot Standby feature will not start if record cache is enabled.

A new command /CACHE=NOENABLED qualifier has been added to the DBO/OPEN command. To open the Hot Standby database prior to starting replication, use the /CACHE=NOENABLED qualifier on the DBO/OPEN command.

4.14 Exclusive Access Transactions May Deadlock with RCS Process

If a record is frequently accessed by long running transactions that request read/write access, reserving the record for exclusive update, and if the record has one or more indexes, you may experience deadlocks between the user process and the Record Cache Server (RCS) process.

There are at least three suggested workarounds to this problem:

1. Reserve the record for CONCURRENT UPDATE.
2. Close the database and disable record cache for the duration of the exclusive transaction
3. Change the checkpoint interval for the RCS process to a time longer than the time required to complete the batch job and then trigger a checkpoint just before the batch job starts. Set the interval back to a smaller interval after the checkpoint completes.