Oracle® CODASYL DBMS for OpenVMS

Release Notes

Release 7.3.3.0 for OpenVMS Alpha and
OpenVMS Industry Standard 64 for Integrity Servers

August 2021

ORACLE®

Oracle CODASYL DBMS Release Notes, Release 7.3.3.0 for OpenVMS Alpha and OpenVMS I64

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

Contents

3 Problems Corrected

4 Known Problems and Restrictions

5 New Features and Corrections in Previous Releases

Tables

Preface

Purpose of This Manual

This manual contains release notes for Oracle CODASYL DBMS release 7.3.3.0. The notes describe changed and enhanced features, upgrade and compatibility information, new and existing software problems and restrictions, and software and documentation corrections.

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 7.3.3.0.

Document Structure

This manual consists of the following chapters:

Chapter 1	Describes how to install Oracle CODASYL DBMS release 7.3.3.0.
Chapter 2	Describes new and changed features in Oracle CODASYL DBMS release 7.3.3.0.
Chapter 3	Describes problems fixed in Oracle CODASYL DBMS release 7.3.3.0.
Chapter 4	Describes problems, restrictions, and workarounds known to exist in Oracle CODASYL DBMS release 7.3.3.0.
Chapter 5	Describes new features and fixed problems in previous releases.

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 I64.

OpenVMS refers to both OpenVMS Alpha and OpenVMS I64.

1
Installing Oracle CODASYL DBMS

This manual discusses installing Oracle CODASYL DBMS release 73 on OpenVMS for Alpha and Industry Standard 64 for Integrity Servers systems. This includes OpenVMS V8.4 from HPE and subsequent versions from VMS Software Inc. (VSI).

All Oracle CODASYL DBMS release 73 kits are full kits. There is no requirement to install any prior release of Oracle CODASYL DBMS prior to installing this release.

1.1 Oracle CODASYL DBMS for OpenVMS I64

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.

The database format for Oracle CODASYL DBMS release 73 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. This means that users running on OpenVMS I64 can create Oracle CODASYL DBMS databases, compile, link, and run their database applications natively.

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). If necessary, implicit forward conversions are performed to bring the database version to the currently installed level.

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 7.3.3.0 database from either system, or both systems concurrently.

1.2 Intel Itanium Processor 9500 "Poulson" Supported

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

1.3 Intel Itanium Processor 9700 "Kittson" Certified

For this release of 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.

1.4 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 73 supports conversions from V7.0, V7.1 and V7.2 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 7.3.3.0 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 7.3.3.0 then convert the V7.1 database to 7.3.3.0.

If you attempt to convert or restore a database version prior V7.0 directly to 7.3.3.0, Oracle DBO generates an error.

1.5 Requirements

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

- This Oracle CODASYL DBMS release requires the following OpenVMS environments:
  - OpenVMS Alpha version 8.4 or later.
  - OpenVMS Industry Standard 64 version 8.4 or later.
- Some hardware configurations require the installation of OpenVMS V8.4-1H1 or later from VMS Software Inc. (VSI).
- Oracle strongly recommends that all available OpenVMS patches are installed on all systems prior to installing Oracle CODASYL DBMS. Contact your HPE or VSI support representative for more information and assistance.
- Oracle CODASYL DBMS must be shutdown before you install this update kit. That is, the command file SYS$MANAGER:MONSTOP73.COM should be executed before proceeding with this installation. If you have an OpenVMS cluster, you must shutdown the Oracle CODASYL DBMS version 73 monitor on all nodes in the cluster before proceeding.
- After executing MONSTOP73.COM, no process on any system in the cluster should have any existing DBMSHR73.EXE image activated. See Section 1.5.1 for additional information.

1.5.1 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 73 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_DBMSHR73_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 DBMSHR73.EXE image. This procedure should be executed by a privileged account after MONSTOP73 has been run. Any processes that have DBMSHR73.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 DBMSHR73.EXE,"PID:"
$ SEARCH 'DBM$TMP' DBMSHR73.EXE /WINDOW=(1,0)
$ DELETE /NOLOG 'DBM$TMP';*
```

In the following example, the process 25E5A8E9 named "FOO$SERVER" has the image DBMSHR73.EXE activated even after MONSTOP73.COM has been executed and this process is terminated prior to starting the Oracle CODASYL DBMS installation procedure:

```
$ @SYS$MANAGER:MONSTOP73.COM
   .
   .
   .
$ @FIND_DBMSHR73_PROC.COM

OpenVMS system analyzer

Process index: 00E9   Name: FOO$SERVER  Extended PID: 25E5A8E9
 0290  7FEA0500 8A5A50C0 $1$DGA1:[VMS$COMMON.SYSLIB]DBMSHR73.EXE;6 (section file)

$ STOP/IDENTIFICATION=25E5A8E9
```

1.6 Installation of Oracle CODASYL DBMS Software

Please refer to the *Oracle CODASYL DBMS 7.3 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 PostScript (PS), 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>:DBM07330A073.A/SAVE/SEL=DBM073_INSTALL_GDE.PDF
$ BACKUP <device>:DBM07330A073.A/SAVE/SEL=DBM073_INSTALL_GDE.TXT
```

For OpenVMS I64:

```
$ BACKUP <device>:DBM07330I073.A/SAVE/SEL=DBM073_INSTALL_GDE.PDF
$ BACKUP <device>:DBM07330I073.A/SAVE/SEL=DBM073_INSTALL_GDE.TXT
```

The release 7.3 installation guide is available in Adobe Portable Document Format (PDF) format from My Oracle Support and from the Oracle Rdb and Related products documentation page, see Section 1.8 for more details.

1.7 Address and Phone Number Correction for Documentation

The contact information listed on the Send Us Your Comments page is no longer correct as shown in prior releases of the Oracle CODASYL DBMS documentation. Please direct all comments, and corrections to this email address: **infordb_us@oracle.com**.

1.8 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 Portable Document Format (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. Following is the link for the Oracle Rdb main page and locate the CODASYL DBMS section.

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

1.9 Hardware Note on Alpha EV56 Requirement

Oracle CODASYL DBMS Release 7.3.3.0 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 DBMS in this emulation mode as long as any problems encountered can be reproduced on an EV56 or greater system.

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 7.3.3.0.

2.1.1 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/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.

2.1.2 DBO/SHOW LOGICAL_NAME now supports /DESCRIPTION qualifier

Bug 3264793, 3682207, 5634563, 19545970

With this release of Oracle CODASYL DBMS 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" = 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

$
```

The definitions of all logical names are maintained in a HELP library called SYS$HELP:DBODISPLAY73.HLB. Users can also use the DCL HELP command to query this help library.

```
$ HELP/LIBR=SYS$HELP:DBODISPLAY73.HLB DBMS_Logical_names DBMS$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?
```

2.1.3 DBO-F-DBROOTFILE, -DBO-F-DBDATAFILE messages output with %DBO-F-BADAIJFILE

It is a common user error for an Oracle CODASYL DBMS database root file, area data file, or area snapshot file to be specified instead of a database after image journal (AIJ) file when executing the DBO/RECOVER command.

Now a new -DBO-F-DBROOTFILE error message will be output as a secondary message when the existing %DBO-F-BADAIJFILE fatal error message is output if a database root file (*.ROO) is specified in an DBO/RECOVER command where a database after image journal file should be specified.

```
%DBO-F-BADAIJFILE, this file is not a valid after image journal file,
please correct the file specification
-DBO-F-DBROOTFILE, specify a database after image journal file, this
is a database root file
```

If a database area data file (*.DBS) or area snapshot file (*.SNP) is specified in an DBO/RECOVER command where a database after image journal file should be specified, a new -DBO-F-DBDATAFILE error message will be output as a secondary message when the existing %DBO-F-BADAIJFILE fatal error message is output.

```
%DBO-F-BADAIJFILE, this file is not a valid after image journal file,
please correct the file specification
-DBO-F-DBDATAFILE, specify a database after image journal file, this
is a database data file
```

In the following example the secondary "-DBO-F-DBROOTFILE" message is output following the primary "%DBO-F-BADAIJFILE" fatal error message if a database root file, PARTS.ROO, is specified instead of a database after image journal file in an DBO/RECOVER command, and the secondary "-DBO-F-DBDATAFILE" message is output following the primary "%DBO-F-BADAIJFILE" fatal error message if a database data file (MARKET.DBS, MARKET.SNP) is specified instead of a database after image journal file in an DBO/RECOVER command. The last DBO/RECOVER command shows that if the after image journal file specified has an invalid format but is not a database root or data file only the "%DBO-F-BADAIJFILE" message is output.

```
$!
$! Database root file specified instead of an AIJ file
$!
$ dbo/recover device:[directory]parts.roo
%DBO-F-BADAIJFILE, this file is not a valid after image journal file,
please correct the file specification
-DBO-F-DBROOTFILE, specify a database after image journal file, this
is a database root file
$!
$! Database data files specified instead of an AIJ file
$!
$ dbo/recover device:[directory]market.dbs
%DBO-F-BADAIJFILE, this file is not a valid after image journal file,
please correct the file specification
-DBO-F-DBDATAFILE, specify a database after image journal file, this
is a database data file
$!
$ dbo/recover device:[directory]market.snp
%DBO-F-BADAIJFILE, this file is not a valid after image journal file,
please correct the file specification
-DBO-F-DBDATAFILE, specify a database after image journal file, this
is a database data file
$!
$! Invalid database AIJ file specified
$!
$ dbo/recover device:[directory]invalid.aij
%DBO-F-BADAIJFILE, this file is not a valid after image journal file,
please correct the file specification
```

2.1.4 DBO/RECOVER DBO-F-BACKUPNOAIJ, DBO-F-TSNNOSYNC, DBO-F-CANTSYNCTSNS Error Messages

For the recovery of Oracle CODASYL DBMS databases from After Image Journal (AIJ) files using the DBO/RECOVER command, Transaction Sequence Number (TSN) values are maintained in the database root file and in the open record and transaction records of each journal file. Each TSN number represents a database transaction which modified the database. The highest committed TSN number in the database root file determines where in the journal file or backed-up or optimized journal file DBO/RECOVER will start the roll forward operation. An AIJ file will only be applied to the database if the TSN number in the open record of the AIJ file is less than or equal to the highest committed TSN number in the database root file. Indvidual transactions contained in an AIJ file are ignored until the TSN of an individual transaction equals the highest committed TSN number in the database root file.

If the TSN number in the open record of an AIJ file is greater than the highest committed TSN number in the database root file, none of the transactions in the AIJ file will be recovered since there are missing transactions that need to be recovered before the transactions that are contained in the current AIJ file are recovered to prevent loss of data and database corruption. Previously, if the TSN number in the open record of an AIJ file was greater than the highest committed TSN number in the database root file, DBO/RECOVER would read through the entire journal file, ignoring all transactions because the TSN values of the individual transaction records in the AIJ file are all greater than the highest committed TSN number in the database root file. After ignoring all transactions, DBO/RECOVER would put out the warning message

```
%DBO-W-NOTRANAPP, no transactions in this journal were applied
```

Now, if at the start of the DBO/RECOVER operation the TSN number in the open record of the first AIJ file to be processed is greater than the highest committed TSN number in the database root file, the recovery operation will be immediately aborted to avoid reading through the entire AIJ file and any additional AIJ files to be processed, ignoring all transactions. The recovery operation is aborted based on the open record of the first AIJ file to be processed since all AIJ files processed after the first AIJ file should also have TSN numbers in their open records which are greater than the highest committed TSN number in the database root file because all transactions must be recovered in the correct original sequence to prevent loss of data and database corruption.

When the DBO/RECOVER operation is aborted because at the start of the recover operation the TSN number in the open record of the first AIJ file to be processed is greater than the highest committed TSN number in the database root file, one of the following two fatal message sequences will be output.

```
%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 96 in the after image journal
file exceeds last committed TSN 35 in the database root
```

This message sequence is output by DBO/RECOVER if the database is either backed up before AIJ journaling is enabled and any after image journal files are defined for the database, or if the database is backed up after AIJ journaling has been disabled and the after image journal state has not been reenabled and recovered by the database restore operation. A backup of the database should be made whenever changes are made to the database prior to the database recovery which cause the following message to be output.

```
%DBO-W-DOFULLBCK, full database backup should be done to ensure future
recovery
```

If the database was backed up subsequent to after image journaling being enabled and after image journal files being defined, restore the database from that backup file and retry the recovery operation.

```
%DBO-F-TSNNOSYNC, The transactions in this journal file are not
consistent with the transactions in this database root file
-DBO-F-CANTSYNCTSNS, Last committed TSN 448 in the after image journal
file exceeds last committed TSN 0 in the database root
```

This message sequence is output by DBO/RECOVER if unjournaled modifications were made to the database, or a copy of the database if the /ROOT qualifier was specified, that were not journaled. If the previous changes are contained in another after image journal file, that AIJ file should be applied before this AIJ file is applied. This message may also be output if the DBO/INITIALIZE/TSN command was executed to initialize the TSNs after the database was backed up. A full backup of the database should be made after any such operation is executed or any changes are made to the database which cause the following message to be output.

```
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
```

Make sure that all previous changes made to the database have been journaled and that all after image journal files containing those changes are specified in the DBO/RECOVER command in the correct sequence order and applied to the correct database or copy of the database.

2.1.5 AIJ Backup Data Compression Information is now in the ABS Process Logs

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 using data compression can be defined for a database by the DBO/MODIFY/AFTER_JOURNAL command. 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.

The new automatic 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 currently supported 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, which immediately follows the first 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
%RDMS-I-OPERNOTIFY, system operator notification: AIJ backup operation started
 4-OCT-2018 08:44:33.45 - No Hot Standby servers active
%RDMS-I-AIJBCKSEQ, backing up after-image journal sequence number 23
%RDMS-I-LOGBCKAIJ, backing up after-image journal J4 at 08:44:33.45
%RDMS-I-LOGCREBCK, created backup file DEVICE:[DIRECTORY]AIJBCKCOMP.ABF;24
%RDMS-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
$
```

2.1.6 DBO/SET/AFTER_JOURNAL/SWITCH_JOURNAL Now Can Create An Emergency AIJ

The Oracle CODASYL DBMS DBO/SET/AFTER_JOURNAL/SWITCH_JOURNAL 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/SET/AFTER_JOURNAL/SWITCH_JOURNAL 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.

If a switch over to the next AIJ file cannot complete because the next AIJ file is not available, because it has not been backed up by the Automatic Backup Server (ABS), or for any other reason, the database enters the "AIJ suspended" state to avoid the loss of database data because it cannot be later 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.

Currently, if a database recovery (DBR) process is active during the AIJ suspended state, or a Hot Standby database replication process is active during the AIJ suspended state, or the AIJ Log Server (ALS) process is active and the DBM$BIND_ALS_CREATE_AIJ system database bind logical is either

not defined or defined as 1 (true), a new permanent "emergency" AIJ file will automatically be created for the switch over to terminate the AIJ suspended state.

If for any reason an emergency journal cannot be created, because the maximum number of AIJ files defined for the database has already been reached or for any other reason, 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).

Fuctionality has been added to the DBO/SET/AFTER_JOURNAL/SWITCH_JOURNAL command to allow it also to automatically create a permanent emergency AIJ journal file. The only way to prevent DBO/SET/AFTER/SWITCH 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 already existing cases mentioned above, the emergency journals created by the DBO/SET/AFTER_JOURNAL/SWITCH_JOURNAL 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/SET AFTER_JOURNAL/SWITCH_JOURNAL command is used to switch from "JOURNAL1" to "JOURNAL2".
- Then, when the DBO/SET AFTER_JOURNAL/SWITCH_JOURNAL 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/SET AFTER_JOURNAL/SWITCH_JOURNAL command automatically creates an emergency AIJ journal with the unique generated name "EMERGENCY_00B03639309BF694" and switches over to this permanent new database AIJ journal. Note that 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/SET AFTER_JOURNAL/SWITCH_JOURNAL/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/SET AFTER_JOURNAL/SWITCH_JOURNAL/LOG TEST
%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_00B03639309BF694"
$
$ ! Insert data (next journal)
   .
   .
   .
```

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 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.

2.2.2 /NOSNAPSHOTS, /[NO]DATA_FILE added to DBO/MOVE_AREA

Currently DBO/MOVE_AREA moves or creates a new version of BOTH the storage area data (*.DBS) and snapshot (*.SNP) files. This new syntax allows moving ONLY the data area file or ONLY the snapshot area file for all or for named storage areas. /NODATA_FILE 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])]
```

NOSNAPSHOTS does not move the storage area snapshot file(s). It only moves the data storage area file(s). SNAPSHOTS is the default. Neither FILE=filespec nor ALLOCATION=n can be specified with NOSNAPSHOTS.

SNAPSHOTS[=([FILE=filespec][ALLOCATION=n])] is an existing qualifier but now it can be negated. SNAPSHOTS as a local qualifier can override NOSNAPSHOTS as a global qualifier. NOSNAPSHOTS as a local qualifier can override SNAPSHOTS as a global qualifier.

```
/[NO]DATA_FILE
```

NODATA_FILE does not move the storage area data file(s). It only moves the snapshot storage area file(s). DATA_FILE is the default. It does not accept any values. DATA_FILE as a local qualifier can override NODATA_FILE as a global qualifier. NODATA_FILE as a local qualifier can override DATA_FILE as a global qualifier.

If NODATA_FILE is specified the storage area data file is not moved or modified. A new version of the file will not be created. If NOSNAPSHOT is specified the storage area snapshot file is not moved or modified. A new version of the file will not be created. Any existing DBO/MOVE_AREA qualifiers that would require an update/change to the data area file are disallowed if /NODATA_FILE is specified

and any qualifiers that would require an update/change to the snapshot area file are disallowed if /NOSNAPSHOTS is specified.

Therefore the following existing /MOVE_AREA qualifiers cannot be specified with either /NODATA_FILE or /NOSNAPSHOTS:

- /ROOT
- /BLOCKS_PER_PAGE
- /NODES_MAX
- /USERS_MAX

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

- /FILE
- /SPAMS
- /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/nodata_file/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 /DATA_FILE did not need to be specified since it is the default.

```
$ DBO/move_area/nolog parts.ROO -
  EMPLOYEE /nodata_file -
           /snapshots=(file=DISK:[DIRECTORY]test_employee.snp), -
  MARKET  /data_file -
         /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 /DATA_FILE 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     /nodata_file -
           /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 /DATA_FILE needed to be specified in the options file to override the global specification of /NODATA_ FILE but /NODATA_FILE 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/nodata_file-
/option=testmove.opt  parts.ROO
  EMPLOYEE /data_file -
           /file=DISK:[DIRECTORY]test_employee.dbs -
           /nosnapshot
  MARKET     /nodata_file -
           /snapshot = (file=DISK:[DIRECTORY]test_market.snp)
  DEPARTMENTS -
              /data_file -
              /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
```

2.2.3 The /LOCK_TIMEOUT Qualifier Now Allows the Database Default

For the Oracle DBMS DBO commands DBO/BACKUP/MULTI/ONLINE, DBO/COPY/ONLINE, the /LOCK_TIMEOUT qualifier can be specified. Previously the /LOCK_TIMEOUT qualifier required a value, the maximum time in seconds to wait for acquiring the database QUIET POINT and other locks used for online database access. If "/LOCK_TIMEOUT = #" was not specified, DBO would 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 value specified for the database will be used. Specifically, the default lock timeout value used will 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, value used will be the maximum possible lock timeout value which can be specified for an Oracle DBMS database.

The new syntax for this qualifier is as follows -

```
/LOCK_TIMEOUT [ = # ]
```

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.

2.2.4 Compression of AIJ Backup Files for Automatic AIJ Backups

A new feature has been added to allow compression of AIJ files during automatic AIJ backups.

To set the compression and the ZLIB level the DBO SET command has been enhanced:

```
$ DBO /SET AFTER_JOURNAL /BACKUPS=(...,[[NO]COMPRESSION[=ZLIB[=n]]])
  default is NOCOMPRESSION
  n = ZLIB compression level,
      default is 6, minimum is 1, maximum is 9
```

2.2.5 Global Statistics Sections for Better Performance

On systems with many CPUs updating database statistics from many application processes causes memory cache invalidation and therefore prolongs the update of the statistics data.

With this change the DBMS Monitor creates 16 global statistic sections for systems with 16 or more CPUs. Application processes attach to a statistics section based on the modulo 16 of their process ID value.

This should reduce the coincidence that two or more processes use the same global section from different processors and thus causing memory cache invalidation when updating statistics data.

The default used for RAD (Resource Allocation Domain) systems still remains (see below).

The use of multiple global statistic sections can be overridden with the following system logical name:

```
$ DEFINE /SYSTEM DBM$BIND_MONITOR_GLOBAL_STATS_SECTIONS n
  n = 0 - always use statistics in the database's shared memory section
  n = 1..16 - use statistics in separate global sections
              with n the number of global sections being used
  If n is equal -1 or if the logical is not defined
  use the default (see below).
```

By default the statistics area in the database's shared memory section is used unless a system has more than one RAD with memory, or the system has 16 or more CPUs.

In the case of more than one RAD with memory one global statistics section is created per RAD with memory. In the case of 16 or more CPUs 16 global statistics sections are created. The more than one RAD with memory case has precedence over the 16 or more CPUs case.

2.2.6 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.

2.2.7 DBO/SHOW STATISTICS Global Sections Move to P2 Address Space

Bug 12921679

In releases prior to DBMS V7.3, the global sections created by DBO/SHOW/STATISTICS were mapped to P0 space. Those global sections are now automatically be mapped to P2 space.

This change should help alleviate any VASFULL errors that may have previously occurred.

2.2.8 EXQUOTA Caused Inaccessible AIJ

Bug 5120555

In prior versions of Oracle DBMS, if an After Image Journal (AIJ) write operation failed with a FILACCERR error and COMMIT TO JOURNAL was enabled, the AIJ file would be immediately marked as inaccessible, the database would be shutdown, and the offending process (database user or database server) would be terminated. Manual intervention would be required to reset journalling before the database could be re-opend.

Oracle DBMS has eased one of these restrictions. Now, if a user process gets a FILACCERR error due to "exceeded quota" (EXQUOTA), the AIJ will not be marked as inaccessible and the database will not be shutdown. The process will, however, still be terminated. This abnormal termination will cause a Database Recovery (DBR) process to be automatically started to recover that user. No manual intervention would be required and normal database processing would continue after the DBR completes.

Note that this change only will affect database user processes. The behavior of Database server processes (such as ALS, RCS, LCS, LRS) remains the same.

2.2.9 Changes to Default Behavior in Oracle DBMS

Oracle DBMS V7.3 changed the following default behavior.

DBO/CREATE and DBO/MODIFY

These new defaults will be used when creating a database.

* The default PAGE SIZE changes from 2 to 4 blocks
* The default BUFFER SIZE changes from 3 pages to 4 pages
* The default NUMBER OF BUFFERS changes from 20 to 250 buffers
* The default NUMBER OF RECOVERY BUFFERS changes from 20 to 250 buffers

These new defaults do not affect databases created in DBMS V7.2 (or older versions) and converted to CODASYL Oracle DBMS 7.3.3.0 using DBO/CONVERT or DBO/RESTORE/MULTI.

2.2.10 Minimum Number of Allowed Users Increased

In prior versions of Oracle CODASYL DBMS, the minimum number of allowed database users was one. In Oracle DBMS V7.3 this minimum has been increased to five to allow for various optional database servers (such as the ABS or RCS or ALS) to access the database.

2.2.11 Add 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
```

A new feature has been added in this release to correct this problem. 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
```

2.2.12 DBO/BACKUP/MULTI/COMPRESSION Qualifier

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

The /COMPRESSION qualifier accepts the following keyword:

- HUFFMAN - HUFFMAN encoding algorithm.
- LZSS - Lempel-Ziv algorithm.
- 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 FOO BCK
$ DBO/BACKUP/MULTI/COMPRESS=ZLIB:9 /LOG=FULL FOO 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.

2.2.13 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
```

Restrictions

Compression encoding method ZLIB works with both /FORMAT=NEW_TAPE and /FORMAT=OLD_FILE. However, neither HUFFMAN nor LZSS encoding are supported when using /FORMAT=OLD_FILE.

The /LOG qualfier reports the achieved compression at the end of the log output.

Example:

```
DBO /BACKUP /AFTER /FORMAT=NEW_TAPE  FOO.ROO FOO.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)
```

2.2.14 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 FOO BCK
```

2.2.15 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 FOO 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 FOO 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 ...
   .
   .
   .
```

2.2.16 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.
```

2.2.17 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 futher 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
```

2.2.18 Syntax for Prestart Transaction Timeout

The ability to define a timeout value for a prestarted transaction has been available on Oracle CODASYL DBMS since version 7.1, however the usage was not documented.

To define a timeout value for a prestarted transaction use this DBO command:

```
$ DBO/MODIFY/TRANSACTION=PRESTART=(TIMEOUT=n) <database name>
```

where, '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 attached 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.

2.2.19 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 exeception 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 a 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:FOO FAIJBCK1,FAIJBCK2,FAIJBCK4
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]FOO.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:FOO FAIJBCK1,FAIJBCK2,FAIJBCK4
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]FOO.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
```

2.2.20 DBO /SHOW STATISTICS /WRITE_REPORT_DELAY=n Feature

The /WRITE_REPORT_DELAY=n 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.

2.2.21 DRU Data Type Change Definitions Enhanced

Syntax has been added to the DRU 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

  DRU 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

  DRU 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

  DRU 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.

2.2.22 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"

2.2.23 Server Output File Names As Database Attributes

Previously, logical names could be used to control various server output or log file names and locations. In many cases, these logical names would have to be defined system-wide and thus could effect the servers of multiple databases.

This situation has been improved. The output or log file names for a number of database server processes are now also controlled by optional database attributes.

The DBO /SET SERVER /OUTPUT=filespec servertype 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.

Note

The DBO /SET SERVER commands should only be used on the Master database. Do not do any DBO /SET SERVER commands on the Standby database as that 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 corrected in Oracle CODASYL DBMS release 7.3.3.0.

3.1 Excessive Alignment Faults in DBO/VERIFY and DBO/ANALYZE

Bug 24691323, and 23525611

This release of Oracle CODASYL DBMS provide a number of alignment fault fixes to DBO/VERIFY and DBO/ANALYZE.

3.2 DBO/DUMP Produces Strange Error When Version Is Set Incorrectly

Bug 29803696

In a prior release of Oracle CODASYL DBMS there was a problem when performing a DBO/DUMP command on a version 7.2 database where a ROOTMAJVER error is incorrectly displayed. The following example demonstrates the problem.

```
$ dbo/dump/header parts
%DBO-F-FILACCERR, error opening database root file DISK:[SNOOPY.WORK]PARTS.ROO;1
-DBO-F-ROOTMAJVER, database format 0.4763968 is not compatible with software version 0.0
```

This problem is now fixed.

3.3 Unexpected MONITOR BUGCHECK dump

Under certain circumstances, the Oracle CODASYL DBMS Monitor process would bugcheck following the use of the DBO/CLOSE/CLUSTER command.

The bugcheck had a footprint similar to:

```
SYSTEM-F-ACCVIO, access violation, virtual address=0000000000000170
Exception occurred at Image DBMMON + 801508C1
Called from Image DBMMON + 80119130
Called from Image DBMMON + 800D0010
Called from Image DBMMON + 8008E850
Running image DBMMON.EXE
```

This problem has now been corrected.

3.4 Excessive Alignment Faults

Bug 23228875, and 29415476

This release of Oracle CODASYL DBMS fixes a number of alignment fault problems that were previously reported for the DBMSHR73.EXE shareable image.

3.5 Fix Memory Leak with DBQ$INTERPRET

Bug 22695028

In a prior release of Oracle CODASYL DBMS there was a problem where calls to DBQ$INTERPRET and DBQ$INTERPRET_STREAM allocated memory that was never freed. Thus after multiple BIND operations an application could fail with this error:

```
%DBM-F-CANTBINDRT, error mapping database root file
-SYSTEM-F-VASFULL, virtual address space is full
```

This problem is now fixed.

3.6 Bugcheck at DIOFETCH$FETCH_SNAP_SEG

Bug 8881798

Starting in Oracle CODASYL DBMS version 7.2, it was possible for a batch retrieval (aka read-only) transaction to bugcheck with an exception at routine DIOFETCH$FETCH_SNAP_SEG. The actual offset within that routine would depend on the current platform and version number.

For database consistency, if a record is modified on a data page by a read-write transaction, but not committed prior to the start of a batch-retrieval transaction, those updates should not be visible to that batch-retrieval transaction. In such a case, the batch-retrieval transaction would need to find some prior version of the record that would be visible. This prior version is written by a read-write transaction to a page in a snapshot file. The data page maintains a pointer to this snapshot page. Transaction Sequence Numbers (TSNs) play an important role in determining which prior copy of a record is visible and when a snapshot page is considered obsolete and re-useable.

The bugcheck occurs when the batch-retrieval transaction needs to find a prior version of a data record, follows the pointer to one or more snapshot pages, but cannot find a version that is visible. This problem was caused by read-write transactions that erroneously re-used valid snapshot pages needed by these batch-retrieval transactions.

This problem does not cause any data inconsistencies or corruption. Simply restarting the batch-retrieval transaction would often suffice. Alternately, running the batch-retrieval transactions could be deferred until all currently existing read-write transactions have concluded. The worst-case scenario, depending on the number of active batch-retrieval and long-running read-write transactions, might require the database to be closed and re-opened.

This problem has now been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.7 DBMDBRBUG Bugcheck at RUJUTL$BIJBL_GET_FORWARD + 1E0

Bug 18506440

In rare circumstances, a Database Recovery Process (DBR) may fail an internal sanity check and bugcheck with the above exception (the actual offset would depend on the Oracle CODASYL DBMS version and OpenVMS platform). After the recovery fails, the database will shutdown all users. However, the next database attach will cause another DBR process to start. This follow-up DBR should be able to successfully complete the original recovery, allowing access to the database once more.

The problem was caused by mismanagement of a data buffer while reading the Run Unit Journal (RUJ) recovery file. As such, there is no user workaround. There is no database or RUJ file corruption associated with this failure.

This problem has been corrected.

3.8 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.

This problem has been corrected.

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.

3.9 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.

This problem has been corrected.

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.

3.10 Unexpected Bugchecks After Incremental Restore

Bug 31826798

After restoring a database and applying an incremental backup, attempts to use it might result in a bugcheck.

This problem would only occur when additional journals were added to the database after a full backup was performed and before an incremental backup was executed.

1. DBO/BACKUP
2. DBO/MODIFY/RESERVE=AFTER_JOURNAL=
3. DBO/BACKUP/INCREMENTAL

Various data structures are stored in an Oracle CODASYL DBMS database root (.roo) file, row cache structures, user structures, and the list of journal files. If any of those structures change in size then the contents of the root file must be shuffled to accommodate the new size. The offsets for each of those data structures are saved at the beginning of the root file.

In the problem situation, after new journals were added to the database, subsequent structures were moved. The new offset for those structures were backed up in the incremental backup. However, when the database incremental backup was applied to the restored database, DBO did not update the database root file with the new offsets for the various root file data structures. Subsequent attempts to access those structures could fail since some structures might not be at the expected location. Using the wrong offset would cause unrelated data to be read which could lead to various failures.

To avoid this problem, always issue a full backup when changing the following values.

- Number of journals
- Number of storage areas
- Number of row caches

This problem has been corrected in this release.

When DBO processes an incremental restore it now properly updates the data structure offsets in the database root file.

3.11 %DBO-F-FILACCERR Reported when DBO/RECOVER Did Not Close AIJ Files it Created

When journaling is enabled for a database the creation of a new journal file is recorded in the currently active journal. During database recovery the information to re-create that journal file is used, the journal file definition is added to the database root file, and the file recreated if it doesn't exist.

A problem existed in DBO /Recover where the newly recreated journal is not correctly closed and may result in an error similar to this example.

```
%DBO-I-AIJALLDONE, after-image journal roll-forward operation completed
%DBO-F-FILACCERR, error opening after-image journal file DEVICE:[DIRECTORY]JOUR_3.AIJ;1
-SYSTEM-W-ACCONFLICT, file access conflict
```

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.12 Possible Misleading Messages From DBO/RECOVER/JUST_CORRUPT and DBO/RECOVER/AREAS

The Oracle CODASYL DBMS DBO/RECOVER/JUST_CORRUPT qualifier specifies that only inconsistent pages in the corrupt page table (CPT) and areas marked as inconsistent should be recovered. The DBO/RECOVER/AREAS qualifier specifies that named or all inconsistent storage areas should be recovered including the inconsistent pages in those areas. During these by page and by area recoveries the following messages were displayed only if logging was enabled.

```
%DBO-I-AIJBADPAGE, inconsistent page n from storage area
DISK:[DIRECTORY]FILENAME.DBS;n needs AIJ sequence number n

%DBO-I-AIJBADAREA, inconsistent storage area
DISK:[DIRECTORY]FILENAME.DBS;n needs AIJ sequence number n

%DBO-I-AIJNXTSEQ, to continue this AIJ file recovery, the sequence
number needed will be n
```

When logging was not enabled these messages were not displayed. These messages will now be displayed for /JUST_CORRUPT by-page and /AREA by-area recoveries, if database corruption still exists when the recovery operation terminates even if logging is not enabled. This information is needed to inform the user of the next journal sequence number to use to apply additional journal files in an additional /JUST_CORRUPT by-page or /AREA by-area recovery to fully eliminate corrupt database pages or corrupt database storage areas.

The messages

```
%DBO-I-AIJGOODPAGE, page n from storage area
DISK:[DIRECTORY]FILENAME.DBS;n is now consistent

%DBO-I-AIJGOODAREA, storage area
DISK:[DIRECTORY]UNIFORM_AREA.DBS;n is now consistent
```

were also displayed only if logging was enabled for /JUST_CORRUPT by-page and /AREA by-area recoveries. Now these messages will be displayed for /JUST_CORRUPT by-page and /AREA by-area recoveries, even if logging is not enabled, to confirm the page corruption and area corruption that has been eliminated by the recovery.

The message

```
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery, the sequence
number needed will be n
```

which specifies the journal sequence number for the next full database recovery was displayed at the end of /JUST_CORRUPT by-page or /AREA by-area recoveries even if corrupt pages or areas still existed in the database. Now, if corruption still exists at the end of the /JUST_CORRUPT by-page or /AREA by-area recovery, the new message

```
%DBO-I-AIJCNTSEQ, The database is still inconsistent, to continue AIJ
file recovery the sequence number needed will be n
```

will be displayed instead of the %DBO-I-AIJFNLSEQ message. The new %DBO-I-AIJCNTSEQ message specifies the next journal sequence number to be used by a /JUST_CORRUPT by-page or /AREA by-area recovery to continue to eliminate the remaining database corruption, not the next journal sequence number for the next full database recovery. The next journal sequence number to be used by a

/JUST_CORRUPT by-page or /AREA by-area recovery to continue to eliminate the remaining database corruption is often different from the journal sequence number for the next full database recovery.

The following example shows this problem for a /JUST_CORRUPT by-page database recovery. The first DBO/RECOVER/JUST_CORRUPT command specifies /LOG so the %DBO-I-AIJNXTSEQ and %DBO-I-AIJBADPAGE messages are displayed. The second DBO/RECOVER/JUST_CORRUPT command specifies /NOLOG so the %DBO-I-AIJNXTSEQ and %DBO-I-AIJBADPAGE messages are not displayed. The %DBO-I-AIJFNLSEQ message is always displayed but it specifies the journal sequence number for the next full database recovery. Since corrupt pages still exist when the recovery terminates, it should specify the journal sequence number for the next /JUST_CORRUPT by-page recovery to continue to eliminate the database page corruption, which in this case is different from the journal sequence number for the next full database recovery.

```
$ dbo /recover /just_corrupt AIJB_0.AIJ /log
%DBO-I-AIJBADPAGE, inconsistent page 111 from storage area
 DISK:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 0
%DBO-I-AIJBADPAGE, inconsistent page 115 from storage area
 DISK:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 0
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PARTS.ROO;1
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]AIJB_0.AIJ;1 at
  6-NOV-2014 13:36:17.12
%DBO-I-LOGRECSTAT, transaction with TSN 833 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 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-AIJACTIVE, 1 active transaction not yet committed or aborted
%DBO-I-LOGRECSTAT, transaction with TSN 832 is active
%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 1 transaction committed
%DBO-I-LOGSUMMARY, total 1 transaction rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJBADPAGE, inconsistent page 111 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJBADPAGE, inconsistent page 115 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery,
 the sequence number needed will be 8
$
$ dbo /recover /just_corrupt AIJB_0.AIJ /nolog
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PARTS.ROO;1
%DBO-I-LOGRECSTAT, transaction with TSN 833 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%DBO-I-AIJALLDONE, after-image journal roll-forward operations completed
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery,
 the sequence number needed will be 8
```

The following example shows that this problem has been fixed for a /JUST_CORRUPT by-page database recovery. The first DBO/RECOVER/JUST_CORRUPT command specifies /LOG and the second DBO/RECOVER/JUST_CORRUPT command specifies /NOLOG. Now the messages %DBO-I-AIJNXTSEQ and %DBO-I-AIJBADPAGE are always displayed. Note that in the /LOG case %DBO-I-AIJBADPAGE messages are displayed both at the start and at the end of the recovery operation but in the /NOLOG case %DBO-I-AIJBADPAGE

messages are only displayed at the end of the recovery operation for page corruption that still exists when the recovery operation terminates. The new %DBO-I-AIJCNTSEQ message is displayed instead of the %DBO-I-AIJFNLSEQ message only if corrupt pages still exist when the recovery terminates to specify the journal sequence number for the next /JUST_CORRUPT by-page recovery to continue to eliminate the database page corruption, which in this case is different from the journal sequence number needed for the next full database recovery.

```
$ dbo /recover /just_corrupt AIJB_0.AIJ /log
%dbo-I-AIJBADPAGE, inconsistent page 111 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 0
%DBO-I-AIJBADPAGE, inconsistent page 115 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 0
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PARTS.ROO;1
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]AIJB_0.AIJ;1 at
  6-NOV-2014 13:14:09.15
%DBO-I-AIJONEDONE, AIJ file sequence 0 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-AIJACTIVE, 1 active transaction not yet committed or aborted
%DBO-I-LOGRECSTAT, transaction with TSN 832 is active
%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 1 transaction committed
%DBO-I-LOGSUMMARY, total 1 transaction rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJBADPAGE, inconsistent page 111 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 1
%DBO-I-AIJBADPAGE, inconsistent page 115 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 1
%DBO-I-AIJCNTSEQ, The database is still inconsistent,
 to continue AIJ file recovery the sequence number needed will be 1
$
$ dbo /recover /just_corrupt AIJB_0.AIJ /nolog
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]PARTS.ROO;1
%DBO-I-LOGRECSTAT, transaction with TSN 833 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%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-AIJBADPAGE, inconsistent page 111 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 1
%DBO-I-AIJBADPAGE, inconsistent page 115 from storage area
 DEVICE:[DIRECTORY]MYAREA.DBS;1 needs AIJ sequence number 1
%DBO-I-AIJCNTSEQ, The database is still inconsistent, to continue AIJ file recovery
 the sequence number needed will be 1
$
```

The following example shows this problem for a /AREA by-area database recovery. The first DBO/RECOVER/AREA command specifies /LOG so the %DBO-I-AIJNXTSEQ and %DBO-I-AIJBADAREA messages are displayed. The second DBO/RECOVER/AREA command specifies /NOLOG so the %DBO-I-AIJNXTSEQ and %DBO-I-AIJBADAREA messages are not displayed. The message %DBO-I-AIJFNLSEQ is always displayed but it specifies the journal sequence number for the next full database recovery. Since corrupt areas still exist when the recovery terminates, it should specify the journal sequence number for the next /AREA by-area recovery to continue to eliminate the database area corruption, though in this case the journal sequence number happens to be the same for the next full recovery and the next /AREA by-area recovery.

```
$ dbo/recover/log/AREA testj_spool1.aij
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_1_BLK.DBS;1
 needs AIJ sequence number 0
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_2_BLK.DBS;1
 needs AIJ sequence number 0
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]TEST.ROO;1
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]TESTJ_SPOOL1.AIJ;1
 at  6-NOV-2014 15:33:16.52
%DBO-I-LOGRECSTAT, transaction with TSN 480 committed
%DBO-I-LOGRECSTAT, transaction with TSN 481 committed
%DBO-I-LOGRECSTAT, transaction with TSN 482 committed
%DBO-I-LOGRECSTAT, transaction with TSN 483 committed
%DBO-I-LOGRECSTAT, transaction with TSN 484 committed
%DBO-I-LOGRECSTAT, transaction with TSN 485 committed
%DBO-I-LOGRECSTAT, transaction with TSN 486 rolled back
%DBO-I-LOGRECSTAT, transaction with TSN 487 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%DBO-I-LOGRECOVR, 7 transactions committed
%DBO-I-LOGRECOVR, 1 transaction 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 7 transactions committed
%DBO-I-LOGSUMMARY, total 1 transaction rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_1_BLK.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_2_BLK.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery,
 the sequence number needed will be 1
$
$ dbo/recover/nolog/AREA testj_spool1.aij
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]TEST.ROO;1
%DBO-I-LOGRECSTAT, transaction with TSN 480 committed
%DBO-I-LOGRECSTAT, transaction with TSN 481 committed
%DBO-I-LOGRECSTAT, transaction with TSN 482 committed
%DBO-I-LOGRECSTAT, transaction with TSN 483 committed
%DBO-I-LOGRECSTAT, transaction with TSN 484 committed
%DBO-I-LOGRECSTAT, transaction with TSN 485 committed
%DBO-I-LOGRECSTAT, transaction with TSN 486 rolled back
%DBO-I-LOGRECSTAT, transaction with TSN 487 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%DBO-I-AIJALLDONE, after-image journal roll-forward operations completed
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery,
 the sequence number needed will be 1
$
```

The following example shows that this problem has been fixed for a /AREA by-area database recovery. The first DBO/RECOVER/AREA command specifies /LOG and the second DBO/RECOVER/AREA command specifies /NOLOG. Now the messages %DBO-I-AIJNXTSEQ and %DBO-I-AIJBADAREA messages are always displayed. Note that in the /LOG case %DBO-I-AIJBADAREA messages are displayed both at the start and at the end of the recovery operation but in the /NOLOG case %DBO-I-AIJBADAREA messages are only displayed at the end of the recovery operation for area corruption that still exists when the recovery operation terminates. The new %DBO-I-AIJCNTSEQ message is displayed instead of the %DBO-I-AIJFNLSEQ message for both the /LOG and /NOLOG case only if corrupt areas still exist when the recovery terminates to specify the journal sequence number for the next /AREA by-area recovery to continue to

eliminate the database area corruption, which in this case happens to be the same as the journal sequence number needed for the next full database recovery.

```
$ dbo/recover/log/AREA testj_spool1.aij
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_1_BLK.DBS;1
 needs AIJ sequence number 0
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_2_BLK.DBS;1
 needs AIJ sequence number 0
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]TEST.ROO;1
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]TESTJ_SPOOL1.AIJ;1
 at  6-NOV-2014 15:14:24.79
%DBO-I-LOGRECSTAT, transaction with TSN 480 committed
%DBO-I-LOGRECSTAT, transaction with TSN 481 committed
%DBO-I-LOGRECSTAT, transaction with TSN 482 committed
%DBO-I-LOGRECSTAT, transaction with TSN 483 committed
%DBO-I-LOGRECSTAT, transaction with TSN 484 committed
%DBO-I-LOGRECSTAT, transaction with TSN 485 committed
%DBO-I-LOGRECSTAT, transaction with TSN 486 rolled back
%DBO-I-LOGRECSTAT, transaction with TSN 487 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%DBO-I-LOGRECOVR, 7 transactions committed
%DBO-I-LOGRECOVR, 1 transaction 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 7 transactions committed
%DBO-I-LOGSUMMARY, total 1 transaction rolled back
%DBO-I-LOGSUMMARY, total 0 transactions ignored
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_1_BLK.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_2_BLK.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJCNTSEQ, The database is still inconsistent, to continue AIJ file recovery
 the sequence number needed will be 1
$
$ dbo/recover/nolog/AREA testj_spool1.aij
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]TEST.ROO;1
%DBO-I-LOGRECSTAT, transaction with TSN 480 committed
%DBO-I-LOGRECSTAT, transaction with TSN 481 committed
%DBO-I-LOGRECSTAT, transaction with TSN 482 committed
%DBO-I-LOGRECSTAT, transaction with TSN 483 committed
%DBO-I-LOGRECSTAT, transaction with TSN 484 committed
%DBO-I-LOGRECSTAT, transaction with TSN 485 committed
%DBO-I-LOGRECSTAT, transaction with TSN 486 rolled back
%DBO-I-LOGRECSTAT, transaction with TSN 487 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%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-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_1_BLK.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJBADAREA, inconsistent storage area DEVICE:[DIRECTORY]UNIF_2_BLK.DBS;1
 needs AIJ sequence number 1
%DBO-I-AIJCNTSEQ, The database is still inconsistent, to continue AIJ file recovery
 the sequence number needed will be 1
$
```

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.13 Lack of LRS Reply Status on HOT STANDBY Shutdown

In prior versions of Oracle CODASYL DBMS, during Hot Standby termination, an AIJ Log Catch-up Server (LCS) may send an INFO_NAK type message to the AIJ Log Roll-Forward Server (LRS). The LCS would then log the message without including any diagnostic information:

```
17-JAN-2021 09:50:56.00 - Shutdown reason: "received INFO_NAK"
```

This problem has been corrected starting with Oracle CODASYL DBMS version 7.3.3.0. Now the LRS will log the return status from the LCS indicating the reason for the termination:

```
4-JUN-2021 23:29:15.64 - Shutdown reason: "received user shutdown request from LCS"
4-JUN-2021 23:29:15.64 - Reply status returned from LRS: 00000001
```

3.14 Full Backup No Longer Required After Altering the Snapshot Area Page Allocation

Modifying the Oracle CODASYL DBMS database snapshot storage area page allocation previously required the next database backup to be a full database backup. If an incremental backup was executed without a preceding full backup following a modification of the snapshot storage area page allocation, the fatal DBO-F-NOFULLBCK error message was output and the incremental backup was aborted.

This restriction has now been removed since database corruption does not occur when an incremental database restore is not preceded by a full database restore following a modification of the database snapshot storage area page allocation. When the snapshot storage area page allocation is modified the incremental backup now succeeds when the incremental backup is executed without a preceding full backup.

The following example shows the previous incorrect behavior. If the next database backup after altering a database storage area snapshot page allocation was not a full backup the %DBO-F-NOFULLBCK message was output and the backup was aborted.

```
$ dbo/backup/incremental/nolog parts prts.dbb
%DBO-F-NOFULLBCK, no full backup of this database exists
$
```

This problem has been corrected in Oracle CODASYL DBMS 7.3.3.0.

3.15 DBO/RECOVER Bugcheck Dump Caused by OpenVMS SYSTEM-W-NONLOCAL error

The Oracle CODASYL DBMS DBO/RECOVER command could fail with the following fatal error when opening an After Image Journal file if the AIJ file specification included a node name.

```
%COSI-F-UNEXPERR, unexpected system error
-SYSTEM-W-NONLOCAL, device is not a local device
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-I-BUGCHKDMP, generating bugcheck dump file DEVICE:[DIRECTORY]DBOBUGCHK.DMP;
```

The DBOBUGCHK.DMP file which was created included the following exception address.

```
***** Exception at 00000000002EA05C : RMU731\COSI$IO_OPEN_FILE + 0000089C
%COSI-F-UNEXPERR, unexpected system error
-SYSTEM-W-NONLOCAL, device is not a local device
```

This problem happened because DBO/RECOVER did not properly handle the SYSTEM-W-NONLOCAL OpenVMS error returned because a device name included a node specification. This problem has been fixed and now the DBO/RECOVER operation will succeed if a node name is specified as part of the AIJ file specification and OpenVMS returns the SYSTEM-W-NONLOCAL error. Note however that the user must have been granted the necessary OpenVMS privileges to access the specified node.

The following example shows that this problem has been fixed. However, the first DBO/RECOVER command fails because the user did not specify the user name and password on the target node necessary in this case to access the AIJ file on the target node. Therefore, OpenVMS returns a privilege error. In the second case the user specifies the necesssary user name and password and the RECOVER operation completes successfully.

```
$ DBO/RECOVER/LOG/ROOT=DEVICE:[DIRECTORY]mf_personnel.roo -
  NODE::DEVICE:[DIRECTORY]rmu_recover_4.aij_1
%DBO-F-FILACCERR, error opening journal file
    NODE::DEVICE:[DIRECTORY]RMU_RECOVER_4.AIJ_1;1
-RMS-E-PRV, insufficient privilege or file protection violation
$
$ DBO /RECOVER/LOG/ROOT=DEVICE:[DIRECTORY]mf_personnel.roo -
  NODE"username password"::DEVICE:[DIRECTORY]rmu_recover_4.aij_1
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]MF_PERSONNEL.ROO;1
%DBO-I-LOGOPNAIJ, opened journal file
    NODE"username password"::DEVICE:[DIRECTORY]RMU_RECOVER_4.AIJ_1;1
    at 18-AUG-2015 17:02:31.66
%DBO-I-LOGRECSTAT, transaction with TSN 544 committed
%DBO-I-LOGRECSTAT, transaction with TSN 576 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-AIJVNOSYNC, AIJ file DEVICE:[DIRECTORY]RMU_RECOVER_4.AIJ_2;1
 synchronized with database
%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
```

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.16 DBO/RESTORE Now Always Displays the %DBO-I-AIJRECFUL Message

Bug 14375975

With this release of Oracle CODASYL DBMS the DBO message DBO-I-AIJRECFUL is always displayed by DBO /Restore when after image journaling was enabled for the database. Previously this message was only output if the DBO /Restore /LOG qualifier was specified.

The DBO /Recover command is often executed after a DBO /Restore operation to apply the contents of one or more after image journal (AIJ) files to update the database with any changes made since the database backup file was created. DBO /Restore displays, using the AIJRECFUL message, the AIJ file sequence number of the first AIJ file where the recovery should start. This is important information for the database administrator to determine where a database recovery should start and to make sure that AIJ files are applied to the database in the correct order.

Note

The DBO/Dump/After_Journal/ONLY=TYPE=OPEN command can be used to dump the AIJ sequence number contained in the Open records of AIJ files.

Examples

In the following example the database ABC, previously backed up by the DBO/BACKUP command with circular after image journaling enabled, is restored by the DBO/RESTORE/NOLOG command.

The DBO-I-AIJRECFUL message states that the next recovery of the database should start with the AIJ file which has the sequence number "0" specified in its Open record. A DBO/DUMP/AFTER_JOURNAL dump of the AIJABC1.AIJ file shows that this AIJ file has a "0" sequence number in its Open record and belongs to the ABC database. The DBO/RECOVER command is then used to bring the database up to date by applying journaled changes made to the database since the database backup file was created contained in the AIJABC1.AIJ after image journal file.

```
$ DBO/RESTORE/NOCDD/NORECOVER/NOLOG ABC_SAVE.DBB
%DBO-I-AIJRECFUL, Recovery of the entire database starts with AIJ file sequence 0
%DBO-I-AIJRSTAVL, 3 after-image journals 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
$!
$ DBO/DUMP/AFTER/ONLY=TYPE=OPEN AIJABC1.AIJ
*------------------------------------------------------------------------------
* Oracle CODASYL DBMS V7.3-30                          17-MAR-2021 15:27:49.16
*
* Dump of After Image Journal
*     Filename: DEVICE:[DIRECTORY]AIJABC1.AIJ;1
*
*------------------------------------------------------------------------------
```

```
1/1             TYPE=O, LENGTH=510, TAD=17-MAR-2021 15:27:48.47, CSM=00
    Database DEVICE:[DIRECTORY]ABC.ROO;1
    Database timestamp is 27-MAY-2019 13:47:27.03
    Facility is "DBMSAIJ ", Version is 721.0
    Database version is 73.0
    AIJ Sequence Number is 0
    Last Commit TSN is 96
    Synchronization TSN is 0
    Journal created on VMS platform
    Type is Normal (unoptimized)
    Open mode is Initial
    Backup type is Active
    I/O format is Record
    Commit-to-Journal optimization disabled
    AIJ journal activation ID is 00B1E299B4FE8A62
    LogMiner is disabled

$!
$ DBO/RECOVER/NOLOG AIJABC1.AIJ
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]ABC.ROO;1
%DBO-I-LOGRECSTAT, transaction with TSN 128 committed
%DBO-I-LOGRECSTAT, transaction with TSN 129 committed
%DBO-I-LOGRECSTAT, transaction with TSN 130 committed
%DBO-I-AIJONEDONE, AIJ file sequence 0 roll-forward operations completed
%DBO-I-AIJAUTOREC, starting automatic after-image journal recovery
%DBO-I-AIJONEDONE, AIJ file sequence 1 roll-forward operations completed
%DBO-W-NOTRANAPP, no transactions in this journal were applied
%DBO-I-AIJALLDONE, after-image journal roll-forward operations completed
%DBO-I-AIJVNOSYNC, AIJ file DEVICE:[DIRECTORY]AIJABC1.AIJ;1 synchronized with database
%DBO-I-AIJSUCCES, database recovery completed successfully
%DBO-I-AIJFNLSEQ, to start another AIJ file recovery, the sequence number needed will be 2
$
```

3.17 Possible Lost Database Updates when Using DBO Backup Incremental

Bug 26864598

In prior releases of Oracle CODASYL DBMS it was possible that DBO Backup Incremental would incorrectly skip updated pages when both ROW CACHE and INCREMENTAL BACKUP SCAN OPTIMIZATION were enabled on the database. Please be aware that the INCREMENTAL BACKUP SCAN OPTIMIZATION is the default setting for new databases.

Note

The DBO Dump Header=BACKUP command can be used to determine the current setting for a database.

```
$ pipe dbo/dump/header=backup parts.roo | -
       search sys$pipe "Fast incremental backup"
      - Fast incremental backup is enabled
$
```

The incremental backup scan optimization allows DBO Backup Incremental to skip over SPAM page ranges which have not been changed since the last FULL database backup. This optimization can save database I/O by quickly locating actively updated page ranges. In rare cases the RCS (Row Cache Server) might check point rows to the database and mark them with a TSN (transaction

sequence number) that was older (smaller value) than that recorded for the FULL database backup. This would indicate to DBO that they need not be backed up during an incremental backup.

For any customer using ROW CACHE, INCREMENTAL BACKUP SCAN OPTIMIZATION and DBO Backup Incremental Oracle recommends that this release be installed and that a new DBO Backup Incremental be performed immediately, to replace any that was previous performed.

A work around to this problem is to avoid the use of the incremental backup scan optimization. This can be done by disabling INCREMENTAL BACKUP SCAN OPTIMIZATION as shown below.

```
$ DBO/MODIFY/BACKUP_OPTIONS=NOSCAN_OPTIMIZATION DISK:[DIR]PARTS.ROO
```

Or, by specifying that this optimization not be used on the DBO Backup Incremental command using the /NoScan_Optimization qualifier.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.18 DBO/BACKUP/AFTER_IMAGE/FORMAT=NEW_TAPE AIJ Sequencing Problem

In prior versions of Oracle CODASYL DBMS the DBO Backup After_Image command would skip any empty fixed length journals. This only occurs when the /FORMAT=NEW_TAPE qualifier is used and results in an after image backup file which can not be used to recover the database. This problem is most likely to occur when either /COMPRESSION or /ENCRYPT are used as these modes require the use of the /FORMAT=NEW qualifier.

When multiple fixed length After Image Journal (AIJ) files were backed up by the DBO Backup After_Image command to the same AIJ backup file using the /FORMAT=NEW_TAPE qualifier and if any of the AIJ files contained valid Open records but did not contain any other AIJ records they were ignored and not backed up. This created a problem later for DBO Recover because eliminating these AIJ files from the backup file could lead to fatal sequencing errors caused by the missing sequence numbers of the eliminated AIJ files when using DBO Recover command.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0. DBO Backup After_Image now backs up all database fixed length AIJ files with valid Open records when the /FORMAT=NEW_TAPE qualifier is specified with the DBO Backup After_Image command.

Example showing DBO Recover problem

The following example shows the problem. The DBO/BACKUP/AFTER_ IMAGE/FORMAT=NEW_TAPE command does not back up an AIJ file containing only a valid Open record with the sequence number "1" to the AIJ backup file but instead outputs the %DBO-I-EMPTYAIJ informational message. Later, when the AIJ backup file is recovered by the DBO/RECOVER/FORMAT=NEW_ TAPE command the recover operation is aborted because the AIJ Open record of the AIJ file that was not backed up by the DBO/BACKUP/AFTER_ IMAGE/FORMAT=NEW_TAPE command contained the next sequence number "1" expected by the DBO/RECOVER/FORMAT=NEW_TAPE command.

```
$ dbo/backup/after_image/encrypt=name:hamlet/compress/log/format=new-
 abc abc_save63_aij2.baij
%DBO-I-AIJBCKBEG, beginning after-image journal backup operation
%DBO-I-OPERNOTIFY, system operator notification: AIJ backup operation started
%DBO-I-AIJBCKSEQ, backing up after-image journal sequence number 1
%DBO-I-LOGBCKAIJ, backing up after-image journal AIJ2 at 14:38:34.28
%DBO-I-EMPTYAIJ, after-image journal file is empty
%DBO-I-AIJBCKSEQ, backing up after-image journal sequence number 2
%DBO-I-LOGBCKAIJ, backing up after-image journal AIJ1 at 14:38:34.33
%DBO-I-QUIETPT, waiting for database quiet point at 30-OCT-2017 14:38:34.33
%DBO-I-QUIETPTREL, released database quiet point at 30-OCT-2017 14:38:34.40
%DBO-I-LOGAIJBCK, backed up 3 committed transactions at 14:38:34.47
%DBO-I-OPERNOTIFY, system operator notification: AIJ backup operation completed
%DBO-I-AIJBCKEND, after-image journal backup operation completed successfully
%DBO-I-LOGAIJJRN, backed up 2 after-image journals at 14:38:34.47
%DBO-I-LOGAIJBLK, backed up 148 after-image journal blocks at 14:38:34.47
%DBO-I-LOGAIJBCK, backed up 3 committed transactions at 14:38:34.47
%DBO-I-LOGCOMPR, data compressed by 81% (75427 Bytes in/14419 Bytes out)
%DBO-I-ENCRYPTUSED, Encryption key required when future restore performed.
$!
$! Later, when the database is recovered a fatal sequence error is returned
$!
$ dbo/recover/encrypt=name:hamlet/noautomatic/log/format=new -
    abc_save63_aij2.baij
%DBO-I-LOGRECDB, recovering database file DEVICE:[DIRECTORY]ABC.ROO;1
%DBO-F-AIJNORCVR, recovery must start with journal sequence 1
%DBO-F-FATALERR, fatal error on RECOVER
```

Example showing the Corrected Behavior

The following example shows that this problem has been fixed. The DBO/BACKUP/AFTER_IMAGE/FORMAT=NEW_TAPE command now backs up an AIJ file containing only a valid Open record with the sequence number "1" to the AIJ backup file and also outputs the %DBO-I-EMPTYAIJ informational message. Later, when the AIJ backup file is recovered by the DBO/RECOVER/FORMAT=NEW_TAPE command the recover operation succeeds because the AIJ Open record of the AIJ file that is now backed up by the DBO/BACKUP/AFTER_IMAGE/FORMAT=NEW_TAPE command contains the next sequence number "1" expected by the DBO/RECOVER/FORMAT=NEW_TAPE command.

```
$ dbo/backup/after_image/encrypt=name:hamlet/compress/log/format=new -
 abc abc_save63_aij2.baij
%DBO-I-AIJBCKBEG, beginning after-image journal backup operation
%DBO-I-OPERNOTIFY, system operator notification: AIJ backup operation started
%DBO-I-AIJBCKSEQ, backing up after-image journal sequence number 1
%DBO-I-LOGBCKAIJ, backing up after-image journal AIJ2 at 14:40:29.63
%DBO-I-EMPTYAIJ, after-image journal file is empty
%DBO-I-AIJBCKSEQ, backing up after-image journal sequence number 2
%DBO-I-LOGBCKAIJ, backing up after-image journal AIJ1 at 14:40:29.67
%DBO-I-QUIETPT, waiting for database quiet point at 30-OCT-2017 14:40:29.67
%DBO-I-QUIETPTREL, released database quiet point at 30-OCT-2017 14:40:29.73
%DBO-I-LOGAIJBCK, backed up 3 committed transactions at 14:40:29.79
%DBO-I-OPERNOTIFY, system operator notification: AIJ backup operation completed
%DBO-I-AIJBCKEND, after-image journal backup operation completed successfully
%DBO-I-LOGAIJJRN, backed up 2 after-image journals at 14:40:29.80
%DBO-I-LOGAIJBLK, backed up 150 after-image journal blocks at 14:40:29.80
%DBO-I-LOGAIJBCK, backed up 3 committed transactions at 14:40:29.80
%DBO-I-LOGCOMPR, data compressed by 81% (76451 Bytes in/14621 Bytes out)
%DBO-I-ENCRYPTUSED, Encryption key required when future restore performed.
$!
$! Later, when the database is recovered, the recovery is successful
$!
$ dbo/recover/encrypt=name:hamlet/noautomatic/log/format=new -
    abc_save63_aij2.baij
%DBO-I-LOGRECDB, recovering database file DEVICE:DIRECTORY]ABC.ROO;1
%DBO-I-LOGOPNAIJ, opened journal file DEVICE:[DIRECTORY]ABC_SAVE63_AIJ2.BAIJ
 at 30-OCT-2017 14:40:33.26
%DBO-I-AIJONEDONE, AIJ file sequence 1 roll-forward operations completed
%DBO-I-LOGRECSTAT, transaction with TSN 162 ignored
%DBO-I-LOGRECSTAT, transaction with TSN 224 committed
%DBO-I-LOGRECSTAT, transaction with TSN 225 committed
%DBO-I-LOGRECSTAT, transaction with TSN 256 committed
%DBO-I-AIJONEDONE, AIJ file sequence 2 roll-forward operations completed
%DBO-I-LOGRECOVR, 3 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 3
%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 3
```

3.19 Master ALS Restart Does Not Resume Updating Standby Database

Bug 20755641

When running Hot Standby with Oracle CODASYL DBMS, it is possible that an abnormal termination of the AIJ Log Server Process (ALS) process on the Master side will cause the AIJ Log Roll-Forward Server Process (LRS), on the Standby side, to stop writing updates to the Standby database. The LRS will continue to write updates to the Standby After-image Journal Files (AIJ) until all journals are full, which will cause the Hot Standby operation to terminate.

Because of the potential for data loss on the Standby side, Oracle recommends, in such a situation, that you resynch the Master and Standby manually, then restart Hot Standby.

This problem occurs if the Master database is opened only on a single node.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.20 MONITOR Bugchecks When Running DBO/CLOSE/STAT=EXPORT

Bug 21620184

In prior versions of Oracle CODASYL DBMS, the DBMS monitor would bugcheck and crash when both of the following conditions were met:

- A user issued a database close request specifying that the SHOW STATISTICS data collected during the session be saved; and,
- The current node had been configured for Resource Affinity Domains (RAD).

Additionally, the user issuing the close command would hang, waiting for a response from the monitor.

To determine if RADs have been enabled, you can use the following DCL function: `F$GETSYI ("RAD_MAX_RADS")`. A return value greater that "1" indicates that the node has been configured for RADs.

In the following example, the DBMS monitor would be abnormally terminated:

```
$ WRITE SYS$OUTPUT F$GETSYI ("RAD_MAX_RADS")
3
$ DBO/CLOSE/STATISTIC=EXPORT TEST.DB
$
```

and a bugcheck would be produced with a similar call stack:

```
***** Exception at 000000000019D3E2 : Image DBMMON73 + 0019D3E2
%SYSTEM-F-ACCVIO, access violation, reason mask=04, virtual
address=0000000000004A0E, PC=000000000019D3E2, PS=0000001B
Saved PC = 000000000011F040 : Image DBMMON73 + 0011F040
Saved PC = 0000000000125EA0 : Image DBMMON73 + 00125EA0
Saved PC = 0000000000165190 : Image DBMMON73 + 00165190
Saved PC = 0000000000112BB0 : Image DBMMON73 + 00112BB0
Saved PC = 0000000000110C20 : Image DBMMON73 + 00110C20
Saved PC = 00000000000CF070 : Image DBMMON73 + 000CF070
Saved PC = 00000000000CDB00 : Image DBMMON73 + 000CDB00
Saved PC = 0000000000110860 : Image DBMMON73 + 00110860
Saved PC = FFFFFFFF80542B90 : symbol not found
```

There is no workaround. The problem has been fixed in Oracle CODASYL DBMS version 7.3.3.0.

3.21 DBO/BACKUP Did Not Output the OpenVMS RMS STV Status for Errors Opening Storage Areas

The STV status field from the OpenVMS RMS FAB structure was not always output by the Oracle CODASYL DBMS DBO/BACKUP command when it contained an error status and errors occurred opening storage areas and some other database files. The STV status is often needed to diagnose the cause of an RMS file error.

This problem has been fixed and now when RMS is used to open a database file and the STV status field contains an error status, that error status will be output by DBO/BACKUP as an additional status in the DBO error message.

The following example shows this problem. The RMS STV status is not included in the %DBO-F-FILACCERR error message for the RMS-E-FND error. Therefore, the cause of the RMS-E-FND error is not output.

```
$ DBO/BACKUP/ONLINE/QUIET/NOLOG PARTS.ROO PRTS.DBB
%DBO-F-FILACCERR, error opening storage area file
 DEV:[DIR]XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.DBS;1
-RMS-E-FND, ACP file or directory lookup failed
```

The following example shows that this problem has been fixed. The RMS STV status is now included in the %RMS-F-FILACCERR error message for the RMS-E-FND error. Therefore the cause of the RMS-E-FND error is displayed as SYSTEM-W-BADFILENAME, bad file name syntax.

```
$ DBO/BACKUP/ONLINE/QUIET/NOLOG PARTS.ROO PRTS.DBB
%DBO-F-FILACCERR, error opening storage area file
 DEV:[DIR]XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.DBS;1
-RMS-E-FND, ACP file or directory lookup failed
-SYSTEM-W-BADFILENAME, bad file name syntax
```

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.22 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 7.3.3.0.

3.23 DBO/SHOW STATISTICS Playback Zeroed Final Transaction Duration Screen

Bug 14549459

When running Oracle CODASYL DBMS on an OpenVMS Integrity system and DBO/SHOW STATISTICS/INPUT was invoked to play back a recorded binary statistics file the TRANSACTION DURATION (TOTAL) screen statistics were all incorrectly set to zeroes at the end of the playback.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0. Now, after the DBO/SHOW STATISTICS playback reaches the end of file, the TRANSACTION DURATION (TOTAL) screen statistics displayed will have valid values.

3.24 DBO/SHOW STATISTICS User-defined Events were not activated

Bug 21884990

There was a problem in prior versions of Oracle CODASYL DBMS which caused correctly defined User-defined Events in the DBO/SHOW STATISTICS configuration file to be read but never activated and executed as expected.

The following example shows the problem.

- A DBO/SHOW STATISTICS User-defined Event is defined in the USER_EVENT.CFG (configuration) file which will keep track of database attaches by executing the EVENT_LOGGER.COM procedure whenever the PROCESS ATTACHES field in the RECOVERY STATISTICS screen is incremented.

```
$ create user_event.cfg
EVENT_DESCRIPTION="ENABLE 'process attaches' MAX_CUR_TOTAL \
 INITIAL 0 EVERY 1 LIMIT 0 SKIP 0 INVOKE SHOW_EVENT";
$
```

- The DCL procedure will create a new version of the EVENT_LOGGER.TMP file which contains the current values of event parameters passed by DBO/SHOW STATISTICS to the DCL procedure.

```
$ create event_logger.com
$DECK
$ open /write evt_logger sys$scratch:event_logger.tmp
$ write evt_logger " ''p1' ''p2' ''p3' ''p4'  ''p5' ''p6' (count is ''p7') "
$ close evt_logger
$EOD
$
```

- The SHOW_EVENT global symbol in the event definition specifies the procedure to be executed. However, because of this problem, the event was ignored and no EVENT_LOGGER.TMP files were created even though three users attached to the database

```
$ SHOW_EVENT :== @event_logger.com
$ show symbol SHOW_EVENT
  SHOW_EVENT == "@EVENT_LOGGER.COM"
```

```
$ dbo/show statistics/configure=USER_EVENT.CFG-
/nointeractive/until=17:00:00 mf_personnel
$!
$ type sys$scratch:event_logger.tmp
%TYPE-W-SEARCHFAIL, error searching for DEVICE:[DIRECTORY]EVENT_LOGGER.TMP;
-RMS-E-FNF, file not found
$
```

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0. User-defined Events will now be correctly activated and executed by the DBO/SHOW STATISTICS command.

3.25 DBO/SHOW STATISTICS Process IO Overview Screen Showed Wrong Values

Bug 21614166

The Oracle CODASYL DBMS DBO/SHOW STATISTICS "Read.Stall" and "WriteStall" column values for the "Process IO Overview" screen were not getting scaled to make sure they could fit in the allocated column space. This caused very large stall times to be output. Also, the read and write stall times only included the synchronous I/O stall times when each should have been the total of both the synchronous and asynchronous I/O stall times. Another problem was that sometimes the screen's column statistics for a process were not incremented but remained all zeroes. This happened most often at very frequent sampling rate intervals such as 0.01 seconds.

These problems have been fixed. The DBO/SHOW STATISTICS "Read.Stall" and "WriteStall" column values for the "Process IO Overview" screen will now be scaled down to thousandths of a second (X1000) to make sure they can fit in the allocated column space, the same scaling factor currently used for the "data read time" and "data write time" rows for the same read and write stall time statistics in the "IO Stall Time" screen. The problem where all the statistics columns for a process in the "Process IO Overview" screen could sometimes remain zeroes at very frequent sampling rate intervals has also been fixed.

Note that because of buffering or transaction commits or asynchronous I/O operations or frequent sampling rates, read and write I/O stall times may not be incremented on a DBO/SHOW STATISTICS screen immediately when a user command is issued. The database statistic being displayed is always incremented at the actual time the data is read from or written to the database.

These problems have been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.26 DBO/SHOW STATISTICS Checkpoint Information Screen CurTSN Column was Too Small

The space allocated for the Oracle CODASYL DBMS DBO/SHOW STATISTICS "CurTSN" column in the "Checkpoint Information" screen was too small. A Transaction Sequence Number (TSN) size greater than nine digits would exceed the start of the following "Tx.Start.Time" column value.

This problem has been fixed. Now the space allocated for the "CurTSN" column in the "Checkpoint Information" screen has been increased to allow for the maximum possible TSN size of 15 digits.

Because of this change, if the display terminal width is set to 80 columns only the first 5 digits of the final "TSN:" column which displays the numeric value of the oldest known TSN on the current database node will be displayed following "TSN:". Since this value may be up to 15 digits long a terminal display width of at least 90 columns should be used for this screen.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.27 DBO/SHOW STATISTICS "Locking (stall time x1000)" Screen was Not Scaled

The Oracle CODASYL DBMS DBO/SHOW STATISTICS "Locking (stall time x1000)" screen lock stall time values were not getting scaled down to make it more likely that they could fit in the allocated row column space even though the screen header specified that all the lock stall time statistic values for this screen were scaled down by thousandths of a second (X1000). This caused very large stall time statistic values to be output and asterisks to be output in frequent cases where unscaled lock stall time statistic values became too large to fit in the allocated column space.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.28 DBO/SHOW STATISTICS Lock Timeout History Screen Data Overwrite Problem

There was a problem where the DBO/SHOW STATISTICS "Lock Timeout History" screen "#Timeouts" column data was not shifted to the right when the terminal display width was increased from a value below 100 characters to a value equal to or greater than 100 characters, causing part of the preceding "Lock.timeout.reason" column data to be overwritten by the "#Timeouts" column data. In the wide screen mode for the "Lock Timeout History" screen the "Occured" column time field is expanded to include the date as well as the time and the number of digits for the time data is increased for a more precise time specification. Therefore, all data columns following the "Occured" column have to be shifted to the right. Because of this problem the column header for the "#Timeouts" column was correctly shifted to the right in wide screen mode but the data for the "#Timeouts" column was not shifted to the right, causing the overwrite of data in the previous "Lock.timeout.reason" column.

The data for the DBO/SHOW STATISTICS "#Timeouts" column in the "Lock Timeout History" screen is correctly displayed so fields no longer overlap.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.29 Possible DBO/SHOW STATISTICS "Device Information" Screen Incorrect Values

The Oracle CODASYL DBMS DBO/SHOW STATISTICS "Device Information" screen would sometimes display incorrect values for the "FreeBlocks", "Max#Blocks" and "%Full" columns. This happened for large disks when disk device block count values exceeded the maximum positive value that can be contained in a signed longword as specified by the OpenVMS operating system

documentation. When this happened non numeric character values were usually displayed instead of valid numeric values.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

3.30 DBO/SHOW STATISTICS Displayed Timed Out Pre-Started Transactions as Active

Bug 14403168

Pre-started Read Write transactions are enabled by default for an Oracle CODASYL DBMS database to reduce transaction overhead by having the commit of the preceding Read Write transaction do most of the work for starting the next Read Write transaction. Pre-started transactions can also be enabled with a timeout, so that after the specified time an unused Pre-started Read Write transaction will be rolled back. This prevents snapshot files from getting too large because of long running idle Read Write transactions preventing freeing up of space in snapshot files.

There was a problem where the DBO/SHOW STATISTICS command screens that monitor processes, such as the "Process Accounting" screen and the "Checkpoint Information" screen, would incorrectly display "Read/write transaction in progress" for a process with Pre-started Read Write transactions that had been rolled back because they had timed out. This happened when "Z" was selected to "zoom" in on a particular process to display transaction and other specific information about that process using one of the "Process.ID" field Process Identification values displayed on the current DBO/SHOW STATISTICS process monitoring screen.

This problem has been fixed in Oracle CODASYL DBMS version 7.3.3.0. The DBO/SHOW STATISTICS command screens that monitor processes, such as the "Process Accounting" screen and the "Checkpoint Information" screen, will now correctly display "No transaction in progress" in the "zoom" information box for a particular process for Pre-started Read Write transactions that have been rolled back because they have timed out.

3.31 Incorrect DBO/SHOW STATISTICS Transaction Duration Histogram Transaction Rate

Bug 6967530

The Oracle CODASYL DBMS DBO/SHOW STATISTICS command histogram graph "Transaction Duration (Read/Write)", "Transaction Duration (Read-Only)", "Active Tx (Read/Write)" and "Active Tx (Read-Only)" screens display statistics on database Read-Write or Read-Only transaction duration lengths and contain a "Transaction rate" display line in the following format for the number of transactions being performed per second or per minute.

```
Transaction rate (per second):   current =     0      average = 0.0
Transaction rate (per minute):   current =     0      average = 0.0
```

There was a problem where the DBO/SHOW STATISTICS command histogram graph "Transaction Duration (Read/Write)", "Transaction Duration (Read-Only)", "Active Tx (Read/Write)" and "Active Tx (Read-Only)" screen display "Transaction rate" values always incorrectly included both the Read-Write and Read-Only transactions being performed when the "Transaction rate" values displayed on the Read-Write screens should have only been for Read-Write transactions and the "Transaction rate" values displayed on the Read-Only screens should have only been for Read-only transactions.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0.

4
Known Problems and Restrictions

This chapter describes problems, restrictions and documentation omiision relating to Oracle CODASYL DBMS version 7.3.3.0 and includes workarounds where appropriate.

4.1 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

(continued on next page)

Table 4–1 (Cont.) Details of Message FULLAIJBKUP

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.

4.2 Undocumented Hot Standby Logical Names

Table 4–2 Hot Standby Logical Names

Logical Name Description	Default Value	Minimum Value	Maximum Value
DBM$BIND_ALS_LOG_REOPEN_SECS Defines the number of seconds after which the ALS output file will automatically be reopened.	0 seconds (will not be reopened automatically)	0 seconds	31449600 (1 year)
DBM$BIND_ALS_LOG_REOPEN_SIZE Defines the number of blocks after which the ALS output file will automatically be reopened.	0 blocks (will not be reopened automatically)	0 blocks	Infinite
DBM$BIND_HOT_ABS_SUSPEND_SHUTDOWN Defines whether or not the AIJ backup server (ABS) should be automatically suspended on graceful shutdown.	0	0	1
DBM$BIND_HOT_CHECKPOINT Specifies the number of messages per server checkpoint interval. If specified, the first threshold to be exceeded (message count or elapsed time) will cause the checkpoint.	100	1	50000
DBM$BIND_HOT_CHECKPOINT_INTERVAL Specifies a checkpoint interval, in minutes, to be used in addition to the /CHECKPOINT qualifier specified at Hot Standby startup. If specified, the first threshold to be exceeded (message count or elapsed time) will cause the LRS checkpoint.	0 minutes (don't use elapsed time)	0 minutes	10080 (7 days)
DBM$BIND_HOT_IGNORE_NET_TIMEOUT Specifies whether or not to ignore network timeout parameters if the LRS process is still active.	0	0	1

(continued on next page)

Table 4–2 (Cont.) Hot Standby Logical Names

Logical Name Description	Default Value	Minimum Value	Maximum Value
DBM$BIND_HOT_LOG_REOPEN_SECS Defines the number of seconds after which the AIJSERVER output file will automatically be reopened.	0 seconds (will not be reopened automatically)	0 seconds	604800 (1 week)
DBM$BIND_HOT_LOG_REOPEN_SIZE Defines the number of blocks after which the AIJSERVER output file will automatically be reopened.	0 blocks (will not be reopened automatically)	0	Infinite
DBM$BIND_HOT_NETWORK_ALT_NODE Defines the secondary network nodename to be used in the event of primary nodename network failure. This logical name allows you to specify an alternate routing pathway to the same standby database.	None		
DBM$BIND_HOT_NETWORK_RETRY Specifies a network retry timeout interval.	120 seconds	0	1800 (30 minutes)
DBM$BIND_LCS_AIJ_SCAN_IO_COUNT Defines the number of asynchronous I/O operations to be performed simultaneously during LCS catch-up.	64	1	128
DBM$BIND_LCS_LOG_REOPEN_SECS Defines the number of seconds after which the LCS output file will automatically be reopened.	0 seconds (will not be reopened automatically)	0 seconds	31449600 (1 year)
DBM$BIND_LCS_LOG_REOPEN_SIZE Defines the number of blocks after which the LCS output file will automatically be reopened.	0 blocks (will not be reopened automatically)	0 blocks	Infinite
DBM$BIND_LCS_QUIET_TIMEOUT Defines the number of seconds to wait for the LCS process to obtain the standby database quiet-point.	600 seconds	0 seconds (wait indefinitely)	Infinite
DBM$BIND_LCS_SYNC_COMMIT_MAX Defines the number of catch-up messages to synchronize with the standby database. A message may contain multiple transactions.	128 messages	32 messages	10000 messages
DBM$BIND_LRS_LOG_REOPEN_SECS Defines the number of seconds after which the LRS output file will automatically be reopened.	0 seconds (will not be reopened automatically)	0 seconds	31449600 (1 year)
DBM$BIND_LRS_LOG_REOPEN_SIZE Defines the number of blocks after which the LRS output file will automatically be reopened.	0 blocks (will not be reopened automatically)	0 blocks	Infinite
DBM$BIND_LRS_QUIET_TIMEOUT Defines the number of seconds to wait for the LRS process to obtain the standby database quiet-point.	600	0 seconds (wait indefinitely)	Infinite
DBM$BIND_STAREA_EMERGENCY_DIR Defines an alternate device and directory for the creation of storage areas on the standby database. The logical must be defined in the LNM$SYSTEM_TABLE table and it is shared by all standby databases on that node.			

4.3 Undocumented DBM$BIND_STAREA_EMERGENCY_DIR Logical Name

Bug 19545970, 3682207

The logical name DBM$BIND_STAREA_EMERGENCY_DIR can be used by Hot Standby when replicating the creation of a new storage area from a master database to the its standby database.

This logical name provides an alternate device and/or directory specification for the standby environment that can replace all or part of the master's file specification. Without the logical, the device and directory of the new storage area issued from the master must exist and match exactly on the standby.

For example, if on the master database the new storage area was named 1DGA11:[ADMIN_MASTER.AREAS]A1.DBS and if the standby environment did not have a device called 1DGA11:, the replication would fail and the AIJ Log Roll-Forward Server (LRS) logfile would log the failure...

```
3-SEP-2020 16:22:26.94 - Replicating master FILID 19
3-SEP-2020 16:22:26.94 - Attempting to create starea "$1$DGA11:[ADMIN_MASTER.AREAS]A1.DBS;1" ALQ=2808
3-SEP-2020 16:22:26.95 - Unable to create storage area. STATUS: 00DDA89C
3-SEP-2020 16:22:26.95 - No emergency directory defined
3-SEP-2020 16:22:26.95 - Failure reason: LRSSRV$CREATE_AREA_CALLBACK - Could not create storage area
```

Suppose the target disk in the standby environment was instead 1DGA109 then this definition could be used to redirect the new storage area to that device.

```
$ define/system DBM$BIND_STAREA_EMERGENCY_DIR "$1$DGA109:"
$ create/directory $1$DGA109:[ADMIN_MASTER.AREAS]
```

The replication operation would then succeed and the LRS logfile would show:

```
3-SEP-2020 15:42:45.65 - Attempting to create starea "$1$DGA11:[ADMIN_MASTER.AREAS]A1.DBS;1" ALQ=2808
3-SEP-2020 15:42:45.67 - Unable to create storage area. STATUS: 00DDA89C
3-SEP-2020 15:42:45.67 - Using emergency area "$1$DGA109:[ADMIN_MASTER.AREAS]A1.DBS"
3-SEP-2020 15:42:45.67 - Attempting to create starea "$1$DGA109:[ADMIN_MASTER.AREAS]A1.DBS" ALQ=2808
3-SEP-2020 15:42:45.68 - Starea creation successful
3-SEP-2020 15:42:45.70 - Attempting to create starea "$1$DGA11:[ADMIN_MASTER.AREAS]A1.SNP;1" ALQ=404
3-SEP-2020 15:42:45.70 - Unable to create storage area. STATUS: 00DDA89C
3-SEP-2020 15:42:45.70 - Using emergency area "$1$DGA109:[ADMIN_MASTER.AREAS]A1.SNP"
3-SEP-2020 15:42:45.70 - Attempting to create starea "$1$DGA109:[ADMIN_MASTER.AREAS]A1.SNP" ALQ=404
3-SEP-2020 15:42:45.71 - Starea creation successful
```

The DBM$BIND_STAREA_EMERGENCY_DIR logical must:

- exist on the standby system prior to the create storage area operation on the master database;
- be defined in the LNM$SYSTEM_TABLE table (/SYSTEM).
- be a valid file specification - device and/or directory.

All standby databases on the node where the logical is defined share its use.

4.4 Undocumented DBM$BIND_HOT_NETWORK_OBJECT Logical Name

Bug 2534163

The following logical name was not previously documented for Oracle CODASYL DBMS.

- DBM$BIND_HOT_NETWORK_OBJECT

  The name of the network object for the AIJSERVER process on the remote standby database. This logical can be used with the following transports: DECnet IV, DECnetOSI, and TCP/IP Services for OpenVMS.

  Specify DBMAIJ<version-number> for Oracle CODASYL DBMS databases (example: DBMAIJ73).

Please note that the DBO/SHOW LOGICAL/UNDEFINED/DESCRIPTION command will display the description for this logical name.

4.5 Undocumented DBM$BIND_AIJBCK_CHECKPOINT_TIMEOUT Logical Name

Bug 24614439

In prior versions of Oracle CODASYL DBMS the logical name DBM$BIND_ AIJBCK_CHECKPOINT_TIMEOUT was not clearly documented. Further, if this logical name was used, the value was interpreted incorrectly as 30 second units and not minutes as expected.

This problem has been corrected in Oracle CODASYL DBMS version 7.3.3.0. This logical name is now documented by the DBO/SHOW LOGICAL /DESCRIPTION command and the units adjusted so that it expects minutes. Please adjust procedures to account for this change.

This system logical can be configured to control the checkpoint stall duration independent of the after image journal (AIJ) shutdown parameter. This logical works for both the after image journal backup and Automatic Backup Server (ABS) utilities.

Default: half of the after image journal shutdown time
Minimum: 1 (minute)
Maximum: 5760 (4 days)

The default, when the logical name is not defined, is based on the currently defined after image journal SHUTDOWN TIME (defined in minutes) as specified by the DBO/MODIFY/JOURNAL=SHUTDOWN=minutes command.

4.6 Syntax for Prestart Transaction Timeout

The ability to define a timeout value for a prestarted transaction has been available on Oracle CODASYL DBMS since version 7.1, however the usage was never documented.

To define a timeout value for a prestarted transaction, enter the syntax:

```
$ DBO/MODIFY/TRANSACTION=PRESTART=(TIMEOUT=nnn) <database name>
```

where, 'nnn' is a value in seconds (range 0 to 3600). 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 attached to the database with long periods of inactivity.

Additionally, DBMS will force a process 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 provides the ability for processes that constantly reuse TSNs to periodically obtain a new TSN, thus preventing excessive snapshot growth.

4.7 Unexpected RCS termination

It has been observed in internal testing of Oracle CODASYL DBMS V7.2.2.0 that if the Record Cache Server (the 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 Row Cache feature either avoid defining the logical name DBM$BIND_RCS_VALIDATE_ SECS, or if this logical name needs to be defined, makes sure that DBM$BIND_ RCS_LOG_FILE is correctly defined (i.e. 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.

4.8 Relaxation of VMS$MEM_RESIDENT_USER Requirement

Bug 5859487

Previously, the VMS$MEM_RESIDENT_USER identifier was required to open a database that had any row cache configured for resident memory even if no caches were enabled for the database.

This restriction has been relaxed. If the database is not enabled for row caches, the VMS$MEM_RESIDENT_USER identifier is not required even if caches are defined for resident memory.

4.9 VMS$MEM_RESIDENT_USER Rights Identifier Required

Oracle CODASYL DBMS release 7.1 introduced additional privilege enforcement for the database or row 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.

4.10 Features Not Available for OpenVMS I64

The following features or capabilities or components are not available to run or are known to not run reliably on OpenVMS I64 with this Oracle CODASYL DBMS release.

- Oracle CODASYL DBMS ADA precompiler
- PL/I compiler and Oracle CODASYL DBMS PL/I precompiler

4.11 Expect Additional Memory Consumption

Due to the increased sizes of image files (especially on Integrity servers) and more aggressive buffering and caching schemes and larger I/O size defaults, you should expect to allocate additional page file quota, working set sizes and buffered I/O byte limit quota when using Oracle CODASYL DBMS release 7.3. In particular, when running on Integrity servers, a page file quota of perhaps three times larger may be required for some applications.

4.12 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 and VAX 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 VAX or 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
```

4.13 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 VAX and 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

4.14 SYSTEM-F-INSFMEM 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 row 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 row 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 row 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 to command;

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

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

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

4.15 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.

4.16 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.

4.17 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.

4.18 Row Cache Not Allowed While Hot Standby Replication is Active

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

A new command qualifier, /CACHE=NOENABLED, 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.

4.19 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 row 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 row 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.

5

New Features and Corrections in Previous Releases

This chapter describes software errors corrected in Oracle CODASYL DBMS in prior versions.

5.1 DBO/BACKUP[/MULTITHREAD]/ONLINE Bugcheck at DIOBCK$GET_PAGE

Bug 12741822

Starting with Oracle CODASYL DBMS version 7.2.4.2, it was possible for an online database backup to return an error with a bugcheck exception at DIOBCK$GET_PAGE.

The error indicates that there was insufficient free space on the data page being synthesized for backup to hold the line and TSN indices for that page. This was caused by incorrectly including the size of freed indices in the calculation of required bytes.

The problem only occurs when using the /ONLINE qualifier on the backup command. Although it is difficult to predict if a backup failure will occur, those data pages with little free space and a higher rate of deleted records may be more susceptible to this error. Restarting the backup may be successful or may fail again on the same or different data page.

5.2 DBO/SHOW STATISTICS Configuration File Issues

12710800 and 12710931

Starting in release V7.2-50, attempting to use a Configuration file on the DBO/SHOW STATISTICS command would generate an access violation. The workaround was to remove that Configuration file specification and use the equivalent command line qualifiers where possible.

5.3 DBO/SHOW STATISTICS Access Violation on Hot Row Information Screen

Staring in release V7.2-50, attempting to view "the Hot Row Information" screen will cause an access violation. This problem does not affect other DBO/SHOW STATISTICS functions. There is no workaround for this problem.

5.4 Problems If a Full DBO/BACKUP Was Not Done After DBO/MOVE_AREA

Bug 13070493

Oracle recommends taking a full database backup after a DBO/MOVE operation. Failure to do so may cause data corruption.

For example, suppose a full database backup was taken prior to the DBO/MOVE and an incremental database backup was taken after the DBO/MOVE. If the full backup was retored, followed by the incremental restore, changes made to the root file by DBO/MOVE would be lost and the database could be left in a corrupt state.

If the full backup had been taken after the DBO/MOVE operation, the problem would not have occurred.

This problem has now been fixed in Oracle CODASYL DBMS version 7.3.3.0. The DBO/MOVE command will now warn you to take a full backup at completion. Also, attempting an incremental backup after the DBO/MOVE, but before taking a full backup will not be allowed.

```
$ DBO/MOVE_AREA/DIRECTORY=[.move_db]/NOLOG
    device:[directory]PARTS.ROO PARTS
%DBO-W-DOFULLBCK, full database backup should be done to ensure future recovery
%DBO-I-COMPLETED, MOVE_AREA operation completed at 11-OCT-2011 12:09:15.40
$
$ DBO/backup/multi/online/nolog/incremental PARTS PARTS_INC
%DBO-F-NOFULLBCK, no full backup of this database exists
%DBO-F-FTL_BCK, Fatal error for BACKUP operation at 11-OCT-2011 12:30:18.08
```

5.5 DBO/SHOW Statistics Sometimes Bugchecks When Using Process Monitoring

Bug 13070947

The Oracle CODASYL DBMS DBO/SHOW Statistics command "Process monitoring" option would sometimes bugcheck. The following sequence of commands while in an DBO/SHOW STATISTICS window would sometimes result in a bugcheck.

```
 ! (to get to the Select Tool)
 R. <<more>>
 J. Process monitoring
 B. Activate specific eligible process
 Enter process ID to "activate" (or "?"): (enter id here)

 ! (to get to the Select Tool)
 J. Process monitoring
 A. Select activated process to monitor
 Process detached; Continue reviewing? ("Yes" to continue, "RETURN" to cancel)
```

The bugcheck looked like the following:

```
***** Exception at FFFFFFFF84230400 : Image LIBOTS + 00002400
%SYSTEM-F-ACCVIO, access violation, reason mask=00,
    virtual address=0000000003A64000, PC=FFFFFFFF84230400, PS=0000001B
Saved PC = 000000008089AB10 : DBO72\KUTDIS$TOOLS_CONTROL + 0000EE80
Saved PC = 00000000809855A0 : DBO72\KUTDIS$TOOLS_MENU + 0000DBC0
```

DBO/SHOW STATISTICS no longer bugchecks under these circumstances.

5.6 Incorrect DBO/BACKUP/AFTER Truncate AIJ File Error Handling

When the DBO/BACKUP/AFTER command is finished backing up a single extensible After Image Journal (AIJ) file, the file is truncated to the beginning of the file. However, transaction data can continue to be added to the journal before and after the truncation operation.

If a non-DBMS process accesses the extensible AIJ file while DBO/BACKUP/AFTER is in the process of truncation, a file access conflict can occur which will cause the AIJ backup to be aborted. If the DBO/BACKUP/AFTER command was subsequently repeated, DBO would immediately detect that the prior AIJ truncation operation failed, and redo the truncation. This second truncation would cause the loss of the data written to the journal during the first truncation. A later recovery of that journal could cause the database to become corrupt.

This problem has been fixed and no loss of data will occur in the above scenario.

5.7 DBO/SHOW Statistics Sometimes Bugchecks on Row Cache Information Screen

Bug 13558012

The Oracle CODASYL DBMS DBO/SHOW Statistics command "Hot Row Information" option would sometimes bugcheck. The following sequence of commands while in an DBO/SHOW STATISTICS window would sometimes result in a bugcheck when monitoring row cache.

```
M (Menu)
X (Row Cache Information)
B (Hot Row Information)
Select one cache
```

The bugcheck looked like the following:

```
SYSTEM-F-ACCVIO, access violation, virtual address=FFFFFFFF817B80B0
Exception occurred at DBO72\KUTDIS$SETUP_RCD_ACTRTN + 00000460
```

5.8 Problem Writing Large TSN Values to Data and Snap Pages

Bug 12741822

A problem has been discovered that could affect how Transaction Sequence Numbers (TSNs) are stored on data and snapshot pages. This problem only occurs when TSNs have a value greater than 2,147,483,647.

A TSN is a sequentially increasing number associated with a DBMS database transaction. A unique TSN is assigned at the start of an update transaction and represents when the transaction started relative to other transactions.

TSNs are used throughout DBMS memory data structures. Additionally, DBMS writes the TSN to on-disk structures such as data files, snapshot files, and after-image journal (AIJ) files to reflect which transaction modified a particular row. Read-only transactions use the TSNs stored on data and snapshot pages to determine which version of a modified row can be seen by the reader.

The problem occurs due to a change to DBMS version 7.0 in how TSNs are stored on data and snap pages. Prior to that version, TSNs were 32-bit integers with a maximum value of 4,294,967,295. With 7.0, the size of the TSN field was changed to be a 64-bit integer. So as not to require a database unload and reload to support the larger field size, an algorithm was devised to break up those TSNs that would no longer fit into a 32-bit field into 2 pieces when the page is updated. The original value would be regenerated prior to reading from the page.

Recently, it was discovered that this algorithm could cause such large TSNs to possibly be stored incorrectly. Occurrences would be more prevalent during the window where some active transaction would have TSN values greater than the 2,147,483,647 value, while others had values less than that maximum.

The problem may manifest itself with errors during an DBO /VERIFY /ONLINE, or with incorrect data returned from read-only transactions. Additionally, a DBO /BACKUP [/MULTITHREAD] /ONLINE may produce a backup file that will restore a corrupted database (although an DBO /RECOVER of all spanning after-journal files will usually fix the corruption).

If you believe that your database exhibits these symptoms, you can use the DBO /INIT /TSN command to reset the TSNs. As always, Oracle recommends taking a full off-line database backup prior to any rootfile modifications.

Note

A Support article about this issue and how to resolve it has been written also. Please reference document number 1440100.1 on My Oracle Support to read this article. Support provides a command procedure that can patch the TSNBLK and CSNBLK numbers to get around the problem.

5.9 DBO/RECOVER/ORDER_AIJ_FILES ERROR HANDLING PROBLEM WHEN ORDERING FILES

There was an error handling problem if an error occurred when DBO/RECOVER/ORDER_AIJ_FILES was sorting the input AIJ files in ascending order by sequence number. An access violation occurred in the error handler followed by a loop producing multiple access violations which produced multiple DBOBUGCHK.DMP files. A "CTRL/Y" had to be executed to terminate the loop. This problem has now been fixed.

The following example shows the problem. An error exception occurs when DBO/RECOVER is sorting the input AIJ files in ascending order by sequence number at the start of the recovery operation because one of the AIJ files is locked by another user. The error handler for this error exception gets into a loop which produces multiple "%SYSTEM-F-ACCVIO" errors and multiple DBOBUGCHK.DMP files. This loop can only be terminated by a "CTRL/Y".

```
$ DBO/RECOVER AIJB* /ORDER_AIJ_FILES/NOLOG
%DBO-F-FILACCERR, error reading journal file
DEVICE:[DIRECTORY]AIJB_4.AIJ;1
-RMS-E-FLK, file currently locked by another user
%DBO-F-FTL_RCV, Fatal error for RECOVER operation at 13-FEB-2012
13:37:29.91
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
address=000000000000018C, PC=000000008078F4A1, PS=0000001B
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-I-BUGCHKDMP, generating bugcheck dump file
DEVICE:[DIRECTORY]DBOBUGCHK.DMP;
%DBO-F-FTL_RCV, Fatal error for RECOVER operation at 13-FEB-2012
13:37:30.03
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
address=000000000000018C, PC=000000008078F4A1, PS=0000001B
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-I-BUGCHKDMP, generating bugcheck dump file
DEVICE:[DIRECTORY]DBOBUGCHK.DMP;
%DBO-F-FTL_RCV, Fatal error for RECOVER operation at 13-FEB-2012
13:37:30.11
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
address=000000000000018C, PC=000000008078F4A1, PS=0000001B
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-I-BUGCHKDMP, generating bugcheck dump file
DEVICE:[DIRECTORY]DBOBUGCHK.DMP;
%DBO-F-FTL_RCV, Fatal error for RECOVER operation at 13-FEB-2012
13:37:30.19
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
address=000000000000018C, PC=000000008078F4A1, PS=0000001B
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-I-BUGCHKDMP, generating bugcheck dump file
DEVICE:[DIRECTORY]DBOBUGCHK.DMP_MINI;
%DBO-F-FTL_RCV, Fatal error for RECOVER operation at 13-FEB-2012
13:37:30.25
```

The following example shows that this problem has been fixed. Now when the AIJ file to be sorted is locked by another user the error exception is handled correctly. The fatal error is output and the recovery is aborted.

```
$ DBO/RECOVER AIJB* /ORDER_AIJ_FILES/NOLOG
%DBO-F-FILACCERR, error reading journal file
DEVICE:[DIRECTORY]AIJB_4.AIJ;1
-RMS-E-FLK, file currently locked by another user
%DBO-F-FTL_RCV, Fatal error for RECOVER operation at 13-FEB-2012
13:40:40.26
```

5.10 DBO/SHOW STATISTICS/CLUSTER Did Not Show Transaction Type

Bug 10009143

In releases prior to Oracle CODASYL DBMS version 7.3.3.0, there was a problem in the DBO/SHOW STATISTICS/CLUSTER command where the Transaction Type was not being displayed on the Checkpoint Information screen, when the transaction was executed on a node other than the current node. This problem only occurred when the DBO command was executed on an Itanium system.

5.11 DBO/SHOW Statistics not using the full screen size for details

Bug 13892668, 13359835

In prior versions of Oracle CODASYL DBMS, the DBO/SHOW Statistics display did not fully use the size of the display, and in some cases limited the output to 17 lines of details.

DBO/SHOW Statistics now uses more screen space when available. The screen size is determined by the terminal page size (SET TERMINAL/PAGE) or when using the /ROW qualifier on DBO/SHOW Statistics.

5.12 Node Failure May Cause DBRs to Hang

Bug 14582053

Under normal conditions, if Oracle CODASYL DBMS is running in a cluster environment and one of the members fails, database recovery processes (DBR) will automatically be created on one of the other cluster members, where the database is currently open, to recover the database users running on that failed node.

In rare circumstances, it was possible for one or more of those DBR's to become blocked waiting for a database page lock held by an active user. That user would be blocked waiting for cluster membership (PR). This scenario would be more likely occur when the systems were under a heavy load, such that the recovering node is resource constrained and creating recovery processes in such an environment takes longer than usual.

The workaround would be to DELPRC or STOP/ID all those blocker processes, or stop all database activity on the remaining nodes using DBO/CLOSE/CLUSTER /ABORT=DELPRC <database-name>, then reopen the database. The next attach would cause DBRs to be created and recovery should complete successfully.

The blocker transaction will now release the contested resource, allowing the recovery to continue.

5.13 DBO/SHOW STATISTICS (Transaction Duration (Total)) Playback Generates a Bugcheck

Bug 14549269

In prior versions of Oracle CODASYL DBMS the DBO/SHOW Statistics (Transaction Duration (Total)) screen on Integity systems generates a bugcheck with an arithmetic trap.

```
%SYSTEM-F-INTDIV, arithmetic trap, integer divide by zero at
PC=FFFFFFFF80A91E51, PS=0000001B
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-I-BUGCHKDMP, generating bugcheck dump file DISK1:[TESTING]DBOBUGCHK.DMP;
%DBO-F-FTL_SHOW, Fatal error for SHOW operation at 24-AUG-2012 06:29:45.45
```

DBO/SHOW Statistics no longer generates a fatal error for this screen.

5.14 %SMG-F-INVROW in DBO/SHOW STATISTICS Using Option WRITE REPORT GRAPH or BOTH

Bug 14807019

When using DBO/SHOW STATISTICS on a database (e.g. PARTS) and doing M (Menu), then a letter that gives a screen in which you can choose O (Option) to write a Report with Graphs or Numbers or Both, and you choose A (graph) or C (Both) the report is written partially and then DBO/SHOW STATISTICS crashes with the error message:

```
%SMG-F-INVROW, invalid row
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
%DBO-F-FTL_SHOW, Fatal error for SHOW operation <<DATE & TIME >>
```

DBO/SHOW Statistics no longer generates an invalid row fatal error for this screen.

5.15 Unable to Start DBO/SHOW STATISTICS During Cluster State Transition

Bug 14582053

In prior versions of Oracle CODASYL DBMS, an attempt to run DBO/SHOW STATISTICS during a cluster state transition would cause the process to hang until the transition was complete.

Under normal circumstances, if an DBMS database, opened on multiple cluster members, was abnormally closed on one node with active users, a Database Recover process (DBR) would be initiated on another node to recover those failed users. The abnormal closure could be caused by a number of situations including: node failure, DBMS monitor failure, or an DBO/CLOSE/ABORT=DELPRC command.

During this recovery period, all database processes are frozen and new attaches (including DBO/SHOW STATISTICS) are stalled. When the recovery completes, the database has normalized and activity can continue on the remaining available nodes (cluster state transition).

In very rare circumstances, it may be possible for the DBR process to become stalled, typically on an undetected deadlock. If the DBO/SHOW STATISTICS utility had been able to start, the "Stall Messages" and "DBR Activity" screens could help determine the nature of the stall, and the Tools facility could be used terminate the processes blocking the DBR.

This problem has now been fixed. DBO/SHOW STATISTICS is no longer subject to the freeze protocol and thus can start and collect information even when there is a stalled recovery process.

5.16 DBO/BACKUP/ONLINE/NOQUIET did not back up the root log file entries

The LOGFIL structure in the Oracle CODASYL DBMS database root file contains entries for different DBMS server log files. These entries can be initialized by the DBO/SET SERVER COMMAND. Due to a problem with an DBO/BACKUP[/MULTITHREAD]/ONLINE/NOQUIET backup, the log file entries in the root were not backed up to the *.DBF backup file. When the database was restored the root log file entries were not corrupt but were missing and had to be redefined. This only happened if the /NOQUIET qualifier was specified for the backup.

This problem has been fixed in Oracle CODASYL DBMS version 7.3.3.0. Now DBO/BACKUP/ONLINE/NOQUIET correctly backs up the LOGFIL log file entries in the database root so that they are all restored correctly.

5.17 Unexpected bugcheck dump from DBO/SHOW STATISTICS/HOT_STANDBY_LOG

Bug 15883785

In prior versions of Oracle CODASYL DBMS, the DBO/SHOW Statistics command with /Hot_Standby_Log qualifier may generate an access violation (ACCVIO) and produce a bugcheck dump. The bugcheck summary would be similar to this example:

```
Itanium OpenVMS 8.4
Oracle CODASYL DBMS Server 7.2.5.2.0
Got a DBOBUGCHK.DMP
SYSTEM-F-ACCVIO, access violation, virtual address=0000000050435400
Exception occurred at symbol not found
Called from DBO72\KUTDIS$HS_NOTIFY + 00000840
Called from DBO72\KUTDIS$EVENT_NOTIFY + 00000220
Called from DBO72\KUTDIS$DISPLAY_ASTX + 00000D00
Running image DBO72.EXE
Command line was: DBO/SHOW STATI/TIME=5/NOINTERACTIVE-
/UNTIL=14-NOV-2012 16:59:35.42/HOT_STANDBY=SYS$LOGIN:GET_
HS.LOG TEST_DATABASE
```