Oracle® CODASYL DBMS for OpenVMS

Release Notes

Release 7.2.1.3 for OpenVMS Alpha and
HP OpenVMS Industry Standard 64 for Integrity Servers

JULY 2007

Oracle CODASYL DBMS Release Notes, Release 7.2.1.3 for OpenVMS Alpha and OpenVMS I64

Contents

4 Known Problems and Restrictions

5 New Features and Corrections in Previous Releases

Preface

Purpose of This Manual

This manual contains release notes for Oracle CODASYL DBMS release 7.2.1.3. 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 7.2.1.3.

Document Structure

This manual consists of the following chapters:

Chapter 1	Describes how to install Oracle CODASYL DBMS release 7.2.1.3.
Chapter 2	Describes new and changed features in Oracle CODASYL DBMS release 7.2.1.3.
Chapter 3	Describes problems fixed in Oracle CODASYL DBMS release 7.2.1.3.
Chapter 4	Describes problems, restrictions, and workarounds known to exist in Oracle CODASYL DBMS release 7.2.1.3.
Chapter 5	Describes new features and fixed problems in previous releases.

Conventions

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

HP 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

All Oracle CODASYL DBMS release 7.2.1.3 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 on OpenVMS I64

In addition to the HP OpenVMS Alpha platform, Oracle CODASYL DBMS is available on the HP OpenVMS Industry Standard 64 for Integrity Servers platform. In general, the Oracle CODASYL DBMS functionality is comparable between the two platforms.

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.

Because 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 7.2.1.3 database from either system, or both systems concurrently.

1.2 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 V7.2.1.0 supports conversions from V7.0 and V7.1 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 V7.2.1.0 in two steps. For example, if you have a V4.2 database, install the latest update to DBMS 7.0, convert the database to that version, install DBMS 7.2.1.3 then convert the v7.0 database to 7.2.1.3.

If you attempt to convert or restore a database version prior V7.0 directly to 7.2.1.3, Oracle DBO generates an error.

1.3 Requirements

This version of Oracle CODASYL DBMS supports OpenVMS Alpha and OpenVMS I64 version 8.3.

One of the following conditions must be met in order to install this software:

- OpenVMS Alpha version 8.2 or later
- OpenVMS I64 version 8.2-1 or later.

1.4 Installation of Oracle CODASYL DBMS Software

Please refer to the *CODASYSL DBMS V7.2 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) or text (TXT) version of the installation guide from the kit, use one of the following commands:

For OpenVMS Alpha:

```
$ BACKUP <device>:DBM07213A072.A/SAVE/SEL=DBM072_INSTALL_GDE.PS
$ BACKUP <device>:DBM07213A072.A/SAVE/SEL=DBM072_INSTALL_GDE.TXT
```

For OpenVMS I64:

```
$ BACKUP <device>:DBM07213I072.A/SAVE/SEL=DBM072_INSTALL_GDE.PS
$ BACKUP <device>:DBM07213I072.A/SAVE/SEL=DBM072_INSTALL_GDE.TXT
```

The release 7.2 installation guide is available on MetaLink and OTN in Adobe Acrobat PDF format.

1.5 Documentation in Adobe Acrobat Format

You can view the documentation in Adobe Acrobat format using the Acrobat Reader, which allows anyone to view, navigate, and print documents in the Adobe Portable Document Format (PDF). For information about obtaining a free copy of Acrobat Reader and for information on supported platforms, see the Adobe Web site at:

```
http://www.adobe.com
```

The Oracle CODASYL DBMS and Hot Standby documentation in Adobe Acrobat format is available on MetaLink and OTN.

2

Enhancements Provided in ORACLE CODASYL DBMS Release 7.2.1.3

This chapter describes new and changed features in Oracle CODASYL DBMS release 7.2.1.3.

2.1 DBO/BACKUP/MULTITHREAD /NORECORD New Qualifier

A new qualfier has been added which avoids 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 DBO/SHOW LOCKS Per Database New Feature

BUG 6004181

In previous Oracle CODASYL DBMS releases, using the DBO/SHOW LOCKS command could be difficult on systems with multiple open databases due to the amount of output and difficulty in determining what database a particular lock references.

This problem has been corrected. The DBO/SHOW LOCKS command now accepts a root file specification that can be used in some cases to additionally filter lock displays to a specific database.

Note that in some cases the DBO/SHOW LOCKS command may be unable to filter locks prior to display. When using the lock-partition tree feature, the DBO/SHOW LOCKS command with a root file specification will be unable to associate area, page, and record locks with the specified database because the database lock is not the lock tree root for these lock types.

2.3 COMPRESS Qualifier for After-Image Journal Backup Command

After-Image Journal backup files can now 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 is only supported for AIJ backup files using the NEW_TAPE format. Threfore all commands listed above must have /FORMAT=NEW_TAPE added to the command line.

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)
```

3
Problems Corrected

This chapter describes software errors corrected in Oracle CODASYL DBMS release 7.2.1.3.

3.1 Latch Hangs Possible From DBO /SHOW STATISTICS

Bugs 4397634 and 5842040

In prior release of Oracle CODASYL DBMS, it was possible in a very small timing window for processes running the DBO /SHOW STATISTICS command to become hung while manipulating "latches" during the database attach sequence. Depending on the exact timing and sequence of events, this process may block other users of the database.

This problem has been corrected.

3.2 LRS Shutdown Failure DBM-F-PARTDTXNERR/SYSTEM-F-NOSUCHID

Bug 5754461

A possible problem with the Oracle CODASYL DBMS Hot Standby feature has been identified. If the OpenVMS $GETGTI system service returns a status value of SS$_NOSUCHID, the LRS process could be unable shutdown cleanly. This could result in an inconsistent standby database.

This problem has been corrected. The LRS process now treats a returned SS$_NOSUCHID status the same as a SS$_NOSUCHTID status and will be handled normally and will not be cause the LRS to fail.

3.3 DBO/SHOW LOGICAL_NAMES Does Not Include DBM$MONITORnn

Bug 5847856

Previously, the list of logical names displayed by the DBO/SHOW LOGICAL_NAMES did not include the logical name “DBM$MONITORnn” though it did include the logical name “DBM$MONITOR”.

This problem has been corrected. The logical name “DBM$MONITORnn” (where “nn” refers to the multi-version DBMS installation version of DBO being executed) is now displayed.

3.4 DBO/SHOW STATISTICS Bugcheck In KUTDIS$LONG_TX_NOTIFY

In Oracle CODASYL DBMS Release 7.2.1, it was possible for the DBO /SHOW STATISTICS command to fail with a bugcheck dump when using the configuration option LONG_TX_SECONDS. The bugcheck dump footprint would be similar to the following:

```
SYSTEM-F-ACCVIO, access violation
Exception occurred at DBO72\KUTDIS$LONG_TX_NOTIFY + 00000424
Called from DBO72\KUTDIS$EVENT_NOTIFY +  00000054
Called from DBO72\KUTDIS$DISPLAY_ASTX + 00000584
Called from DBO72\KUT$DISPLAY + 0000199C
```

This problem has been corrected.

3.5 AIJ Backup Operation Aborts With NONAME-F-NOMSG Message Number 00000004

In rare cases, an after-image journal backup operation may fail with an unexpected incorrect status value. The actual value may vary, but at least one customer report of the problem indicated a value of 00000004. A bugcheck dump file “footprint” of this problem is:

```
***** Exception at 0054D94C : AIJBCK$GET_NEXT_JOURNAL + 00000CFC
Saved PC = 005452E8 : AIJBCK$FULL_BACKUP + 00000FF8
Saved PC = 00543C0C : AIJBCK$BACKUP + 0000113C
```

This problem has been corrected. The errant status value was the result of an uninitialized return status being passed back. The correct status is now returned.

3.6 DBO/SHOW STATISTICS AIJ ARB:I/O ratio, Blocks-per-I/O ratio Problems

There was a problem with detecting the warning thresholds set for the "Examine ARB:I/O ratio" and "Examine blocks-per-I/O ratio" options on the DBO/SHOW STATISTICS "AIJ Analysis" screen. This caused the "ARB:I/O ratio" warning:

```
#.# ARBs per I/O below #.# threshold
```

and the the "blocks-per-I/O ratio" warning:

```
#.# blocks written per I/O below #.# threshold
```

to not always be output when the threshold was crossed.

This problem was caused by treating these thresholds as percent values instead of count values. This problem has been fixed and the misleading percent signs have been removed from these warning messages.

3.7 Adding a large AIJ file to a DB fails with either an ACCVIO or OPCDEC error

Creating a large after image journal file for a database fails with either an ACCVIO or an OPCDEC error.

The reason was a bug in the code which prematurely cleared a synchronization flag. This allowed the request, created by an expired prestarted transaction timer, to execute before the current request had completed. The side effect of this was a stack corruption.

As a workaround use a larger prestarted transactions timer value or, disable the prestarted transactions timer completely.

This problem has been corrected.

3.8 Unable to Fully Disable Hot Standby Governor

Bug 5166721

Previously, when the Hot Standby Governor was explicitly disabled, it was still possible during periods of very high load for the Governor to be re-enabled when more than 75% of the LRS buffers on the standby system were in use.

This behaviour has been disabled by default. The Governor will not be re-enabled even when more than 75% of the LRS buffers on the standby system are in use.

If it is desired to revert back to the prior behaviour, the system-wide logical name DBM$BIND_LRS_ALLOW_AUTOMATIC_HOT_STANDBY_GOVERNOR may be defined to a value of "1" to allow the Governor to re-enable itself as in prior releases.

3.9 DBO/DUMP/BACKUP/MULTITHREAD fails with an ACCVIO or an overrun error

The command DBO/DUMP/BACKUP/MULTITHREAD for a backup file generated with the /COMPRESSION=ZLIB qualifier fails with:

```
%SYSTEM-F-ACCVIO, access violation, reason mask=04, virtual address=..., PC=..., PS=0000001B
%DBO-F-FATALOSI, Fatal error from the Operating System Interface.
```

The command DBO/DUMP/BACKUP/MULTITHREAD for a backup file generated with the /COMPRESSION=HUFF qualifier fails with:

```
Illegal output buffer overrun
%DBO-E-INVRECEXP, Error expanding compressed backup file record.
%DBO-F-FATALERR, fatal error on DUMP_BACKUP
```

The error occured in a /DUMP specific code section which did not switch to the correct compression context for an area.

Both problems have been corrected.

3.10 DBO Tape Support Added For SDLT600, LTO2, LTO3 Drives

Support for the VMS tape density and compaction values for the Super DLT600, Ultrium460 and Ultrium960 tape drives has been added to the DBO multithreaded (/BACKUP/MULTITHREAD and /RESTORE/MULTITHREAD) utilities.

This will allow the following new density values to be specified with the /DENSITY qualifier for those DBO commands that write to these drives:

```
/DENSITY = (SDLT600,[NO]COMPACTION) - Super DLT600
/DENSITY = (LTO2,[NO]COMPACTION) - Ultrium460
/DENSITY = (LTO3,[NO]COMPACTION) - Ultrium960
```

3.11 Incorrect DBO/SHOW STATISTICS Transaction Recovery Duration Estimates

In previous DBMS 7.2 releases, it was possible for the time values on the Transaction Recovery Duration Estimates display to be inaccurate. The values were generally vastly larger than they should have been.

This problem has been corrected. The estimates are now scaled to more realistic range values.

3.12 DBO/SHOW STATISTCS Stall Statistics Aggregate Duration Incorrectly Scaled Values

In previous DBMS 7.2 releases, stall duration values on the aggregate stall statistics display were not correctly scaled and displayed as exceptionally large values.

This problem has been corrected.

3.13 DBO /SHOW STATISTICS /ALARM=n Not Waiting n Seconds

In previous DBMS 7.2 releases, it was possible for stall notification alarms to fire inaccurately. Typically the stall notification was much faster than expected. For example, the following should alarm via OPCOM if a stall longer than 60 seconds was detected. Alarms could be generated much earlier than they should be:

```
$ DBO /SHOW STATISTICS PRODUCTION_DB -
    /NOINTERACTIVE -
    /BROADCAST -
    /TIME=60 -
    /ALARM=60 -
    /NOTIFY=(OPER12) -
    /UNTIL=TOMORROW
```

This problem has been corrected. Stall notification alarms are not generated unexpectedly early.

3.14 Hot Standby Node Failure Recovery When Using DBO/OPEN/CACHE=NOENABLE

BUG 5957364

In configurations using the Row Cache and Hot Standby features, row caching must be explicitly disabled on the standby database using the DBO/MODIFY /NOCACHE command prior to starting hot standby for the first time on the database. However, it is also possible (though not recommended) to use the DBO/OPEN/CACHE=NOENABLE command on the standby database in order to suppress row caching.

When using the DBO/OPEN/CACHE=NOENABLE command, if a system failure occurred, it was possible that the database recovery upon reopening the database would attempt to start with a very old last checkpoint location. This location was based on the row cache checkpoint from when the master database had been

originally backed up to create the standby database. In some cases, the required AIJ files would be no longer online and the recovery would fail.

This problem has been corrected. The DBR process now ignores the row cache oldest checkpoint location when not recovering from a node failure when the RCS process had been active.

3.15 State Value Truncated on Hot Standby Statistics Display

Bug 6044632

Previously, it was possible when using a Hot Standby TCP/IP port number greater than 9999 that the port number would be truncated on the DBO /SHOW STATISTICS Hot Standby Statistics dislpay.

For example, when using a TCP/IP port number of 12345, the state display could be shown as "State: TCP/IP:1234"

This problem has been corrected. The state display field now allows a 5-digit TCP/IP port number to be displayed.

3.16 Possible Shared Memory Corruption When Multiple Databases Attached

Starting with Oracle CODASYL DBMS V7.1.2 and Oracle CODASYL DBMS V7.2.0, it was possible for shared memory to become corrupt. The corruption often would appear as (or would be caused by) data from one DBMS root file being written into the shared memory for another database. Once this corruption has occurred, reliability and functionality of the database and database users can be compromised.

Conditions leading to this corruption include:

- Processes accessing multiple databases
- Multiple database users
- Databases accessed from multiple nodes in a cluster
- Databases configured with node count greater than 1.

The memory corruption was caused by incorrect IO buffer synchronization while refreshing root file information in to shared memory.

This problem has been resolved. Oracle strongly recommends that customers with applications or procedures that may attach or more than one database at a time upgrade to this or a later release to avoid this potential memory corruption problem.

3.17 Bugcheck at COSI$TIMER_GET_REQIDT With DBMS-F-NOREQIDT

Applications using the fast commit feature that periodically detach and reattach to a datbase within the same program run may eventually run out of DBMS timer blocks and bugcheck with a footprint similar to:

```
***** Exception at 01235994 : COSI$TIMER_GET_REQIDT + 00000294
%DBMS-F-NOREQIDT, reached internal maximum number of simultaneous
timer requests
Saved PC = 01235C38 : COSI_TIMER_SET + 00000288
Saved PC = 01235DA8 : COSI_TIMER_SLEEP + 00000078
Saved PC = 012A7548 : KOD$COMMIT + 00000508
```

This problem has been corrected. The problem was caused by an internal timer data structure being allocated but not being deallocated if the timer had expired. If the timer had not expired, the internal timer data structure was correctly deallocated. Thus, in some cases, the timer data structure was being "leaked" which could eventually lead to the bugcheck exception of "DBMS-F-NOREQIDT, reached internal maximum number of simultaneous timer requests".

3.18 Qualifier /[NO]CONFIRM For DBO /RECOVER Command

The /CONFIRM qualifier for the DBO /RECOVER command causes the operator to be queried when an incorrect sequence of AIJ files is detected.

In the following example, note that the backed up AIJ files are specified in the order B1, B3, B2, B4 representing sequence numbers 1, 3, 2, 4:

```
$ DBO/RECOVER/NOLOG B1,B3,B2,B4
%DBO-I-LOGRECDB, recovering database file $1$DGA203:[DB]FOO.ROO;1
%DBO-W-AIJSEQPRI, AIJ file sequence number 1 created prior to
expected sequence 2
%DBO-I-LOGRECSTAT, transaction with TSN 0:224 ignored
%DBO-I-AIJONEDONE, AIJ file sequence 1 roll-forward operations completed
%DBO-W-NOTRANAPP, no transactions in this journal were applied
%DBO-W-AIJSEQAFT, incorrect AIJ file sequence 3 when 2 was expected
do you wish to continue the roll-forward operation [N]:
```

DBO detects the improper journal order and displays the message "DBO-W-AIJSEQAFT, incorrect AIJ file sequence 3 when 2 was expected". DBO then asks the operator if the roll-forward operation using the incorrect AIJ file sequence 3 should be allowed to continue. If the operator specifies "Y", then the roll-forward operation on AIJ file sequence 3 will continue. Otherwise, DBO will move to the next journal (AIJ file sequence 2 in this example).

Note

Oracle recommends that, in general, an incorrect journal sequence not be applied as a corrupt database may result.

The /ORDER_AIJ_FILES qualifier can be used to help ensure that the specified journals are applied in the correct order.

The default setting for the /CONFIRM qualifier is /NOCONFIRM for batch processes and /CONFIRM otherwise.

3.19 /ORDER_AIJ_FILES Removes Some Unnecessary Files For DBO /RECOVER Command

The /ORDER_AIJ_FILES qualifier, in addition to ordering the specified input AIJ files by ascending sequence number, now also can eliminate some AIJ files from processing if they are known to be prior to the database recovery sequence starting point.

In the following example, note that the backed up AIJ files are specified in the order B1, B3, B2, B4 representing sequence numbers 1, 3, 2, 4. The /ORDER_ AIJ_FILES sorts the journals to be applied into ascending sequence order and then is able to remove B1 from processing because the database recovery starts with AIJ file sequence 2 as shown in the DBO/RESTORE output.

```
$ DBO/RESTORE/NEW/NOCDD/NOAFTER FOO
%DBO-I-RESTXT_00, Restored root file DUA0:[DB]FOO.ROO;16
    .
    .
    .
%DBO-I-AIJRECFUL, Recovery of the entire database starts with
AIJ file sequence 2
%DBO-I-COMPLETED, RESTORE operation completed at 24-MAY-2007 12:23:32.99
$!
$ DBO/RECOVER/LOG/ORDER_AIJ_FILES B1,B3,B2,B4
    .
    .
    .
%DBO-I-LOGOPNAIJ, opened journal file DUA0:[DB]B2.AIJ;24
%DBO-I-LOGRECSTAT, transaction with TSN 0:256 ignored
%DBO-I-LOGRECSTAT, transaction with TSN 0:257 ignored
%DBO-I-RESTART, restarted recovery after ignoring 2 committed transactions
%DBO-I-AIJONEDONE, AIJ file sequence 2 roll-forward operations completed
%DBO-I-LOGRECOVR, 0 transactions committed
%DBO-I-LOGRECOVR, 0 transactions rolled back
%DBO-I-LOGRECOVR, 2 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
    .
    .
    .
```

Note that due to the fact the AIJ backup files might have more than one journal sequence in them, it is not always possible for DBO to eliminate every journal file that might otherwise appear to be unneeded. But for those journals where DBO is able to know for certain that the journal could not be needed based on the database recovery restart information, journals can be avoided from having to be processed.

3.20 ACCVIO with DBO/BACKUP/MULTITHREAD/DISK=WRITER=m/THREADS=n.

Certain combinations of m and n for DBO/BACKUP/MULTITHREAD /DISK=WRITER=m/THREADS=n cause an ACCVIO.

For example:

```
$ DBO /BACKUP /MULTITHREAD /NOLOG TESTDB [.T1]TESTBCK,[.T2]/DISK=(WRITER=1)/THREADS=1
%DBO-I-BUGCHKDMP, generating bugcheck dump file ...
%SYSTEM-F-ACCVIO, access violation, reason mask=00,
            virtual address=0000000000000070, PC=...
```

This problem has been corrected.

3.21 DBO /BACKUP/MULTI to tape with /ENCRYPTION can cause a bugcheck.

A plain DBO/BACKUP/MULTI to tape using a non-zero XOR group and encryption can cause a bugcheck.

Example:

```
$ DBO/BACKUP/MULTI/ENCRYPTION=(VAL="mysecretkey") PARTS $1$MKA100:PARTS.DBF
...
%DBO-I-BUGCHKDMP, generating bugcheck dump file ...
%SYSTEM-F-ACCVIO, access violation, reason mask=04, virtual address=...
%DBO-F-FATALERR, fatal error on BACKUP
```

The problem was caused by the encryption code using a wrong buffer size which was modified before by the XOR code.

This problem has been corrected.

4

Known Problems and Restrictions

This chapter describes problems and restrictions relating to Oracle COADASYL DBMS release 7.2.1.3 and includes workarounds where appropriate.

4.1 Slight 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.

4.2 Patch Required When Using VMS V8.3 and Dedicated CPU Lock Manager

During qualification testing of Oracle CODASYL DBMS Release 7.2.1 on OpenVMS V8‘ systems, a problem with the use of Extended Lock Value Blocks and the OpenVMS Dedicated CPU Lock Manager feature was discovered.

To avoid this problem, Oracle strongly recommends that customers wishing to use DBMS and the OpenVMS Dedicated CPU Lock Manager feature with OpenVMS V8.3 install one of the following architecture-specific patch kit (or subsequent replacement if superseded) prior to using Oracle CODASYL DBMS Release 7.2.1, or later, on OpenVMS V8.3 systems:

- VMS83I_SYS-V0200 (I64)
- VMS83A_SYS-V0100 (Alpha)

4.3 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.

4.4 Features Not Yet Available for OpenVMS I64

The following features or capabilities or components are not currently 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

4.5 Oracle CODASYL DBMS and IEEE Floating Point Support

Currently, Oracle CODASYL DBMS does not support floating point IEEE formats for either OpenVMS Alpha or OpenVMS I64. Because of the default float point behavior on OpenVMS IA64, if your Oracle CODASYL DBMS metadata contains floating point data items, you must compile your OpenVMS I64 applications with the FLOAT=G_FLOAT compiler switch.

Note

This restriction should not impact the storing or fetching float point items with the DBQ utility.

Oracle will look into lifting or easing this restriction for a future release.

For more information about IEEE floating point and OpenVMS I64, please refer to Section 4.8 in this document.

4.6 Expect Additional Memory Consumption

Due to the increased sizes of image files (especially on Integrity servers) and more aggressive buffering and caching schemes and larer 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 7.2. In particular, when running on Integrity servers, a page file quota of perhaps three times larger may be required for some applications.

4.7 ILINK-E-INVOVRINI Error on I64

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

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

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

4.8 Default Floating Point Format

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

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

- On OpenVMS 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

For more information, Oracle recommends reviewing the white paper "OpenVMS floating-point arithmetic on the Intel Itanium architecture" available from HP.

4.9 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>
```

4.10 Oracle CODASYL DBMS and OpenVMS ODS-5 Volumes

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

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

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

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

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

4.11 Carryover Locks and NOWAIT Transaction Clarification

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

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

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

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

4.12 Both Application and Oracle CODASYL DBMS Using SYS$HIBER

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

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

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

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

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

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

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

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

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

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

4.13 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.

4.14 Exclusive Access Transactions May Deadlock with RCS Process

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

5.1 New Features for Release 7.2.1.0

This section contains new features and technical changes for Oracle CODASYL DBMS release 7.2.1.0.

5.1.1 DBO/BACKUP/MULTI/COMPRESSION

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

The /COMPRESSION qualifier accepts the following keywords:

- 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.

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

A new qualfier has been added to allow the user to better control the system load created by a multithreaded backup, copy or, move operation. The new qualifier allows the user to 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 susbsystem 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 ...
    .
    .
    .
```

5.1.3 Increased Date/Time String Display Precision

For several values where there is enough space on the display, the DBO SHOW STATISTICS utility now displays time/date stamps with precisions greater than 0.01 second units. In several cases (stall displays, for example), the screen display width must be 100 or more columns in order to display the full date/time with seven fractional digits.

For example, the “short” time and/or date format displays include only two fractional digits:

- 16:23:16.17
- 13-NOV-2006 16:23:16.17

While the “long” time and/or date format displays include seven fractional digits:

- 16:23:16.1776975
- 13-NOV-2006 16:23:16.1776975

5.2 Corrections in Release 7.2.1.0

This section describes software errors corrected in Oracle CODASYL DBMS release 7.2.1.0.

5.2.1 Problem with Remote Access and FETCH..USING

BUG 5685084

A problem has been uncovered with Oracle CODASYL DBMS when using remote database access with either DBQ or DML applications. If you attempt to FETCH a record via a USING clause, the fetch may fail with a DBM-F-END condition, even though the record does exist.

The problem will ONLY occur if one of the data items specified in the USING clause is the last data item defined in that record.

The error does not occur with local database access or with remote access when using the WHERE clause.

For example, given the following schema:

```
AREA NAME IS A1

RECORD NAME IS R1
    WITHIN A1
    ITEM NAME IS I1
        TYPE IS CHARACTER 5
    ITEM NAME IS I2
        TYPE IS CHARACTER 5
    ITEM NAME IS I3
        TYPE IS CHARACTER 5

SET NAME IS ALL_R1
    OWNER IS SYSTEM
    MEMBER IS R1
        INSERTION IS AUTOMATIC
            RETENTION IS FIXED
        ORDER IS SORTED BY
                    ASCENDING I3
```

and assuming that there is an R1 record with the following values:

```
I1 = 'AAAAA'
I2 = 'BBBBB'
I3 = 'CCCCC'
```

The following remote query attempting to fetch record R1 will fail:

```
dbq> bind dbmfetrmtdb
dbq> ready
dbq> set noprompt
dbq> move 'CCCCC' TO I3
dbq> fetch first within ALL_R1 using I3
%DBM-F-END, end of collection
```

whereas, the same logical query using a WHERE clause will succeed:

```
dbq> fetch first within ALL_R1 where I3 eq 'CCCCC'
I1 = AAAAA
I2 = BBBBB
I3 = CCCCC
```

This problem has now been fixed. No application programming changes are required.

5.2.2 Area File not Renamed after DBO/MODIFY/RESTRUCTURE

BUG 2260168

The Oracle CODASYL DBMS reload utility (DBO/MODIFY/RESTRUCTURE) moves database records from a specified target area to a new area.

In versions of DBMS prior to V7.0, the default behavior was to create the new storage area with the same filename (and in the same directory) as the target original area, with an incremented file version number. Note: only offline reload is available in pre-V70 versions.

Starting with DBMS v7.0, these defaults were modified to ensure that the new storage area filename was unique by attempting to append an "_A" (OR "_B", etc) to the storage area name. This was done as part of the work to support online reload (DBO/MODIFY/RESTRUCTURE/ONLINE), where the reload could be stopped and restarted in the middle of execution.

The idea was to make sure that there was no confusion between the original area and the new area, if the reload were stopped for any reason, and to make sure that certain file actions, such as a $PURGE, would not delete the original area prior to reload completion.

If you wish to retain the old behavior, include the /FILE= qualifier on the DBO /MODIFY/RESTRUCTURE command and specify the original storage area filename as the parameter. This qualifier should be included on the restructure operation that performs the EXECUTE phase for offline reload, or the PREPARE phase in the case of online reload.

For example, assume that you wished to reload the BUY area in the PARTS database.

In pre-V70 offline reloads, the default would be to create a storage area, BUY.DBS;2 (assuming that BUY.DBS;1 was the original area filename). In V70 and later, the default would be to create BUY_A.DBS;1.

To maintain the old behavior, issue DBO/MODIFY/RESTRUCTURE PARTS BUY/FILE=BUY. Note that you could also specify the /DIRECTORY qualifier to have the new storage area created in a new directory.

To modify the storage area file name of a previously reloaded area, you can rename the file, then use the DBO/ALTER utility and execute:

```
DBALTER> DEPOSIT FILE <area> SPECificaion <new-filename>
```

5.2.3 Mixing Cobol and Fortran or DML modules on Interity Servers

Starting with DBMS 7.2 on Integrity Servers, if you attempt to link together Cobol modules with modules compiled with the DBMS Fortran or DML precompilers, you would receive a linker error message similar to:

```
%ILINK-E-INVOVRINI, incompatible multiple initializations for overlaid section
        section: DBM$UWA_B
        module: DMLMIXUWA
        file: DISK:[DIRECTORY]DMLMIXUWA_COB1.OBJ;VERSION
        module: DMLMIXUWA_FOR
        file: DISK:[DIRECTORY]DMLMIXUWA_FOR1.OBJ;VERSION
```

This error message occurs because the Cobol compiler and CODASYL DBMS precompiler utilities generate a different value for one field in the User Work Area (UWA) data structure. Specifically, COBOL initialize the UWA message condition field (DB-CONDITION) with a '0', while DBMS precompilers initialize the same field with a '1'. UWA structures from modules with the same stream will be overlayed by the linker into one program section (PSECT).

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.

However, even with "-E-" message status, in most cases an executable image is produced and useable.

Starting with this release of Oracle CODASYL DBMS, the Fortran and DML precompilers will generate the same initialization values for the UWA as does the COBOL compiler.

As a side effect of this change, you could still continue to see the ILINK-E-INVOVRINI if you attempt to link a FORTRAN or DML module (compiled under this release) with other FORTRAN or DML modules compiled under previous releases of DBMS 7.2.

To work around this problem, two new logicals have been established for this release. The DBM$FDML_INIT_DBCOND_1 and DBM$DML_INIT_DBCOND_1 logicals will allow the DBMS precompilers to revert to the prior behavior and make newly compiled modules linker-compatible with previously compiled modules.

For example, if you define DBM$FDML_INIT_DBCOND_1 to any value, the DBMS FORTRAN precompiler will assign a value of 1 to the UWA condition field. If you define DBM$DML_INIT_DBCOND_1 to any value, the DBMS DML precompiler will assign a value of 1 to the UWA condition field.

These logicals only affect the compilation of Integrity Servers, as this linker error does not occur on Alpha systems.

5.2.4 Using OpenVMS Reserved Memory Registry With DBMS

For Oracle CODASYL DBMS memory-resident global sections (either row cache global sections or the database root global section), it is possible to utilize the OpenVMS Reserved Memory Registry feature to reserve physical memory. This reserved memory can be useful to allow the use of granularity hint (GH) regions which can further improve performance by using fewer processor translation buffer entries to map a large range of physical memory pages. Use of the reserved memory is optional and any performance gains are application specific.

In order to take advantage of the OpenVMS Reserved Memory Registry feature, global sections must be configured as "SHARED MEMORY IS PROCESS RESIDENT". This can be done with DBO statements "DBO/MODIFY/MEMORY_MAPPING=(PROCESS,RESIDENT)" and "DBO/CACHE/MODIFY/MEMORY_MAPPING=(PROCESS,RESIDENT)".

The name of the global section is required in order to register a global section in the OpenVMS shared memory registry. The "DBO/DUMP/HEADER" command can be used to display the global section names for the database root global section and the row cache global sections. This command also displays the size of the global sections in megabytes rounded up to the next whole megabyte.

For example, information about a row cache global section in the output from the DBO/DUMP/HEADER command might include the following:

```
Shared Memory...
  - Shared memory will be mapped resident
  - Global Section Name is "DBM72R$1$DGA2031064003D000000000005"
  - Shared memory section requirement is 77,070,336 bytes (74MB)
```

Information about the database global section in the output from the DBO/DUMP /HEADER command might include the following:

```
Derived Data...
  - Global section size
      With global buffers disabled is 2,047,042 bytes (2MB)
      With global buffers enabled is 33,860,114 bytes (33MB)
       .
       .
       .
  - Global Section Name is "DBM72N$1$DGA2031064003D000000000000"
```

From these examples, the row cache section size would be 74 megabytes and the database global section size (with global buffers enabled) would be 33 megabytes.

To reserve the memory, use the SYSMAN utility RESERVED_MEMORY ADD command and then run AUTOGEN as in the following examples:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> RESERVED_MEMORY ADD DBM72N$1$DGA2031064003D000000000000 -
        /ALLOCATE /SIZE=33
SYSMAN> RESERVED_MEMORY ADD DBM72R$1$DGA2031064003D000000000005 -
        /ALLOCATE /SIZE=74
SYSMAN> EXIT
$ @SYS$UPDATE:AUTOGEN ...
```

The OpenVMS system must be then shutdown and restarted for the memory reservations to be in effect.

After rebooting and reopening databases, the SHOW MEMORY /RESERVED command can be used to see that the reserved memory is in use. For example:

```
$ SHOW MEMORY/RESERVED
Memory Reservations (pages):       Group  Reserved   In Use       Type
 DBM72R$1$DGA408451A6A000000000002
                                  SYSGBL         2        2  Page Table
 DBM72R$1$DGA408451A6A000000000002
                                  SYSGBL      1536     1353   Allocated
 Total (12.01 MBytes reserved)                1538     1355
```

Database Root File Specific

Changes to the size of the database or row cache global sections will require that the memory reservation size be updated (either by removing and re adding or modifying the existing reservation). Further, because the device and file identification of the database root file are encoded in the global section names, any operation (such as restoring or moving) that changes either the file identification or the device identification of the root file will result in the global section names changing.

If the reserved memory is specified with a size smaller than the actual size of the global section, the section may fail to be created when the database is opened or accessed with a message similar to "SYSTEM-F-INSFLPGS, insufficient Fluid Pages available".

For further information, review the OpenVMS documentation set including "HP OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems", "HP OpenVMS Version 8.2-1 for Integrity Servers New Features and Release Notes", and "HP OpenVMS System Services Reference Manual".

5.2.5 DBO/CONVERT Bugchecks in PIO$LOCK_PAGE When Statistics Disabled

If statistics were disabled while executing an **DBO/CONVERT** command the DBO utility would bugcheck with a stack footprint similar to the following:

```
***** Exception at 0000000000719708 : DBO72\PIO$DEMOTE_PAGE + 000001A8
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual
address=0000000000000000, PC=0000000000719708, PS=0000001B
Saved PC = 0000000000722E14 : DBO72\PIOUTL$EMPTY_ONE_BUFFER + 000002B4
Saved PC = 000000000071D654 : DBO72\PIOFETCH$WITHIN_DB_HNDLR + 00000134
Saved PC = FFFFFFFF81104EC8 : Image LIBOTS + 00008EC8
Saved PC = FFFFFFFF800A693C : symbol not found
***** Exception at 000000000071C020 : DBO72\PIO$LOCK_PAGE + 00000320
Saved PC = 000000000071E114 : DBO72\PIOFETCH$WITHIN_DB + 00000924
Saved PC = 000000000071B444 : DBO72\PIOFETCH$FETCH + 000002E4
Saved PC = 000000000071A4E4 : DBO72\PIO$FETCH + 000008F4
```

The same problem may also occur when an implied conversion is done by restoring a backup that was made with a prior version of Oracle CODASYL DBMS.

This problem can be avoided by deassigning the DBM$BIND_STATS_ENABLED logical prior to executing the **DBO/CONVERT** command.

This problem has been corrected.

5.2.6 Hangs or Looping When Lots of Page Contention

Applications that had lots of page contention could sometimes hang due to page locks not being released by a process or they could enter a CPU loop. This problem was only in Oracle CODASYL DBMS Release 7.2.

This problem would occur when an internal queue used to managing blocking AST requests would become corrupt. In that situation blocking ASTs could be lost, or processing of the queue could result in an infinite loop.

There is no workaround for this problem.

This problem has been corrected.

5.2.7 DBO/SHOW STATISTICS Hot Standby Statistics State Display Field

Bug 5396571

Previously, when using the TCP/IP network transport with the Hot Standby feature, the DBO /SHOW STATISTICS "Hot Standby Statistics" display "State:" field could overwrite the "UserSync:" heading as in the following example:

```
Node: HSVMS  (1/1/16)  Oracle CODAYSL DBMS V7.1-24 Perf. Monitor 18-JUL-2006 06‘
Rate: 3.00 Seconds           Hot Standby Statistics         Elapsed: 00:07:28.63
Page: 1 of 1  $MYDISK:[MYDB]PARTS.ROO;1  Mode: Online
-------------------------------------------------------------------------------
State: TCP/IP:72     rSync: Cold Current.Msg: 1       Cl Mstr.AIJ: 1:2
LagTime: 00:00:00 AutoSync: Cold Stalled.Msg: none     1 Stby.AIJ: 1:2
Stby.DB: $MYDISK:[MYDB_STANDBY]PARTS.
```

The line starting with “State:” partly overwrites “UserSync:”.

This problem has been corrected.

5.2.8 File-System Caching Avoided For Various IO Operations

In order to reduce CPU consumption and XFC spinlock contention and to help avoid “thrashing” the file system cache and to streamline file read and write operations, caching by the operating system is disabled for various files and operations including:

- DBO /COPY
- DBO /MOVE
- DBO /BACKUP /MULTITHREAD
- DBO /RESTORE /MULTITHREAD
- Most Database Root File IO
- Most Database RUJ File IO
- Most Row-Cache Backing Store File IO
- Most Recovery Work File IO

Testing on various configurations indicates that, in general, avoiding the operating system’s XFC cache for these database file IO operations results in better over-all performance as balanced between CPU and IO costs.

5.2.9 Processes Don’t Always Terminate After Monitor Terminates

BUG 5361981

When the Oracle CODASYL DBMS monitor process terminates abnormally all user processes that are attached to databases on that node should immediately terminate. However, there were cases where that didn’t happen, and those user processes would continue to access Oracle CODASYL DBMS resources after the monitor failed. Consider the following example.

User 1, node 1:

```
DBQ> BIND PARTS
```

User 2, node 2:

```
DBQ> BIND PARTS
```

User 3, node 1:

```
$ STOP/ID={pid of monitor process on node 1}
```

In the above sequence of events, the user process on node 1 should have terminated as soon as the monitor process was killed, but it remained active.

This problem can be avoided by using the **DBO/OPEN** command and manually opening databases on all nodes that will have users accessing the database.

This problem has been corrected.

5.2.10 DBO/RECOVER of Journaled Row Cache Changes Corrupts Database

BUG 5469750

If a database had row cache parameters changed, and the database was restored and recovered, the resulting database would be corrupt. Sometimes the **DBO /RECOVER** process would fail as well, and occasionally the Oracle CODASYL DBMS monitor process would fail.

Depending on what row cache parameters were changed, various failures may occur in the **DBO/RECOVER** operation or in the database monitor. In the reported problem **DBO/RECOVER** would fail with the following exception:

```
***** Exception at 007E35BC : PIO$FETCH + 000003EC
%SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual address=0000000000000000, PC=0000000000
```

Also, the database monitor failed with the following exception:

```
**** Exception at hhhhhhhh : MON$DELETE_UNREFERENCED_GBL + 00000DAC
%SYSTEM-F-ACCVIO, access violation, virtual address=0000000000414000
```

To avoid this problem do a full database and journal backup after altering any row cache parameters. If this problem is encountered it is possible to recover the restored database up until the point in the journal that contains the row cache changes. That is, using the /UNTIL qualifier, recover the journals up to the point in time that the row cache changes were made.

This problem has been corrected.

5.2.11 DBO/BACKUP/AFTER Ignores /EDIT_FILENAME When Backup Filespec Omitted

BUG 5464971

When a **DBO/BACKUP/AFTER** command was issued, if the **/EDIT_FILENAME** qualifier was included but no output filename was given, the default journal filename would be used and the contents of the **/EDIT_FILENAME** qualifier were ignored. For example:

```
$ DBO/BACKUP/AFTER/LOG -
  /EDIT=("_",VNO,"_",YEAR,MONTH,DAY_OF_MONTH,"_QP") -
  DBM$DATABASE ""
.
.
.
%DBO-I-LOGCREBCK, created backup file DEV:[DIR]JOURNAL_1.AIJ;1
```

In the above example, no output filename was specified, that is, "" was given as the output file. The journal that was being backed up had the filename "JOURNAL_1". The backup filespec constructed by DBO should have been "JOURNAL_1_0_20060829.AIJ", but the contents of the **/EDIT_FILENAME** qualifier were not incorporated in the output filename.

This problem can be avoided by explicitly providing the backup output filename in the backup command.

This problem has been corrected.

5.2.12 Concealed Logical Names Defined in LNM$SYSCLUSTER_TABLE Table Allowed

Previously, many uses of concealed logical device names were required to be defined in the LNM$SYSTEM_TABLE logical name table. This requirement is in place to ensure that various components of the database system running in separate process contexts would all have access to the same logical name definitions. Uses of concealed logical device names that were not defined in the LNM$SYSTEM_TABLE could result in a `COSI-F-NOTSYSCONCEAL "non-system concealed device name in filename" status`.

This restriction has been somewhat relaxed. While all processes using a database still require access to the same logical name definitions, this can now be accomplished by using LNM$SYSTEM_TABLE logical name table or the LNM$SYSCLUSTER_TABLE logical name table (which represents a cluster-wide resource). Note, however, that it is strongly recommended that concealed logical device names are not defined in both tables at the same time on any cluster node as this can lead to unpredictable results possibly leading to database corruption or instability.

5.2.13 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"

5.2.14 DBO /SHOW STATISTICS Defined Logicals List Incomplete

BUG 5600122

Previously, it was likely that the DBO /SHOW STATISTICS Defined Logicals display did not properly list all logicals when the display was set to “Full” mode. This problem was caused by an incorrect calculation of the number of logical names possible.

This problem has been corrected. The full list of logical names is correctly displayed.

5.2.15 Incorrect Backup Checksum And Crc Values On I64

In some cases, the checksum or crc values within a .DBF backup (DBO/BACKUP /MULTITHREADED) file on I64 systems starting with CODASYL DBMS Release V7.2.0.2 may be incorrect. This difference could result in checksum errors during restore operations when using /CRC=CHECKSUM.

As a workaround, Oracle recommends using the default CRC algorithm of /CRC=AUTODIN_II rather than /CRC=CHECKSUM.

This problem has been corrected. The checksum value calculated by Oracle CODASYL is now the same on all platforms and versions.