

Oracle® CDD/Repository for OpenVMS

Release Notes

Release 7.4.1.0.0

August 2024

This document contains the release notes for Oracle CDD/Repository for OpenVMS Industry Standard 64 Integrity Servers and OpenVMS Alpha operating systems.

ORACLE®

Oracle CDD/Repository Release Notes, release 7.4.1.0.0 for OpenVMS Alpha and OpenVMS Industry Standard 64 Integrity Servers.

Copyright © 1981, 2024 Oracle and/or its affiliates. All rights reserved.

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

Primary Author: Rdb Engineering and Documentation group

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited. The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing. If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle, Java, Oracle Rdb, Hot Standby, LogMiner for Rdb, Oracle SQL/Services, Oracle CODASYL DBMS, Oracle RMU, Oracle CDD/Repository, Oracle Trace, and Rdb7 are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

Contents

Preface	v
1 Release Notes for Oracle CDD/Repository Release 7.4.1.0.0	
1.1 Installing Oracle CDD/Repository Release 7.4.1.0.0	1-1
1.1.1 Requirements	1-1
1.2 DEC Distributed Transaction Manager Requirements	1-2
1.2.1 Enabling DECdtm Services for Oracle CDD/Repository	1-2
1.3 Invoking the VMSINSTAL Procedure	1-3
1.4 Enhancements and New Features	1-3
1.4.1 Improved Support for Oracle Rdb SQL Builtin and Aggregate Functions	1-3
1.4.2 Updated Error Message Documentation in SYSSHELP	1-4
1.5 Software Errors Fixed	1-5
1.5.1 Invalid CDD\$TEMPLATE Logical Causes DEFINE REPOSITORY to Fail	1-5
1.6 Enhancements and New Features in Prior Releases	1-5
1.6.1 Support for JOIN Clause in RSEs From Oracle Rdb	1-6
1.6.2 Support for Value List Expressions From Oracle Rdb	1-6
1.6.3 Explanations and User Actions Updated	1-6
1.6.4 Integrate Enhanced to Support New BLRSK_AGG_COUNT Syntax	1-6
1.7 Software Errors Fixed in Prior Releases	1-7
1.7.1 RDMS-F-INDTOOBIG, Requested Index is Too Big	1-7
1.7.2 Create Table From Fails When CDO Record Has Check Clause	1-7
1.7.3 DEFINE REPOSITORY Failed to Use Template With Multiversion Rdb	1-8
1.7.4 Problem Changing a Relation With Constraints	1-8
1.7.5 Unexpected CDD\$_PAGNOTFND, CDD\$_EMPTYCELL and CDD\$_CELNOTEMP Errors	1-9
1.7.6 RDB-F-SEGSTR_PARAM Error During Integrate	1-9
1.7.7 CDO Could Not be Invoked Using SYSMAN	1-10
1.7.8 CDO Show Version Only Shows Two Digits	1-10
1.7.9 %PASCAL-E-LNETOOLNG on Compile When Field Comments More Than 255 Characters	1-10
1.7.10 CDD Pathname Versions Truncated on SQL ALTER TABLE	1-11
1.7.11 Access Violation During Integrate Alter Files	1-12
1.7.12 Using SQL CREATE After DROP of Object With Same Name Caused ACCVIO	1-12
1.7.13 Alignment Faults Corrected	1-13
1.7.14 Database Prepared for OCI Services Fails to Integrate	1-13
1.7.15 Not All NOT NULL Constraints Are Integrated into Rdb	1-13
1.7.16 DEFINE REPOSITORY Command Creates Excessively Large Files	1-13

1.7.17	CDO Fails Displaying Signed Longword Initial Value	1-14
1.7.18	CDDL Did Not Properly Store Signed and Unsigned QUAD Initial Values	1-14
1.7.19	DMU Did Not Properly Display Signed and Unsigned QUAD Initial Values	1-14
1.7.20	DMU Performs an ACCVIO When Displaying an H_FLOAT Initial Value	1-14
1.7.21	CDO EXTRACT RECORD /LANGUAGE=CC Problem Handling Computed By Fields	1-15
1.7.22	CDO EXTRACT RECORD/LANGUAGE=CC Problem Handling Based On Fields	1-15
1.8	Known Restrictions	1-15
1.8.1	CDD Does Not Fully Support Index with Explicit Node Size or Percent Fill	1-16
1.8.2	Clarification on Object Name Syntax in CDO utility	1-16
1.8.3	Use of SQL DROP Statement Against CDD/Repository Definitions . . .	1-16

Preface

Intended Audience

This manual is intended for use by all Oracle CDD/Repository users. Read this manual before you install, upgrade, or use Oracle CDD/Repository release 7.4.1.0.0.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Reader Comments

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

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

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

Please direct all comments, and corrections to this email address:
infordb_us@oracle.com.

If you have problems with the software, please contact your local Oracle Support Services.

Conventions

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

In this manual, OpenVMS means both the OpenVMS Alpha operating system and the OpenVMS IA-64 operating system.

Release Notes for Oracle CDD/Repository Release 7.4.1.0.0

This document provides release notes for Oracle CDD/Repository for OpenVMS Industry Standard 64 Integrity Servers and OpenVMS Alpha operating systems. The two systems are collectively referred to as OpenVMS. However, certain differences between the platforms may result in minor capability and functionality differences.

1.1 Installing Oracle CDD/Repository Release 7.4.1.0.0

This software update is installed using the standard OpenVMS Install Utility.

NOTE

All Oracle CDD/Repository Release 7.4 kits are full kits. There is no requirement to install any prior release of Oracle CDD/Repository when installing new Oracle CDD/Repository release 7.4 kits.

1.1.1 Requirements

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

- Oracle CDD/Repository requires the following OpenVMS environments:
 - OpenVMS Alpha Version 8.4 or later
 - OpenVMS IA-64 Version 8.4 or later
 - Oracle Rdb release 7.3 or later
- Oracle Rdb must be running before you install this kit.
- Oracle CDD/Repository requires DEC Distributed Transaction Manager (DECdtm) services for all transactions.

Before you proceed with this installation, you must ensure that the appropriate Oracle Rdb version is established. Use `SYS$LIBRARY:RDB$SHOVER.COM` to show the current version. If necessary execute a command similar to this:

```
$ @SYS$LIBRARY:RDB$SETVER 74
```

- The installation requires approximately 60,000 blocks for OpenVMS Alpha systems.
- The installation requires approximately 90,000 blocks for OpenVMS IA-64 systems.

1.2 DEC Distributed Transaction Manager Requirements

Oracle CDD/Repository requires DEC Distributed Transaction Manager (DECdtm) services for all transactions.

1.2.1 Enabling DECdtm Services for Oracle CDD/Repository

A DECdtm transaction log must be created for each node in your VMScluster environment, and you must define SYSSJOURNAL as a search list of all directories that contain DECdtm transaction logs.

If you do not have a DECdtm transaction log, you will receive the following error messages when you try to install Oracle CDD/Repository:

```
-CDO-E-ERRSTARTSESS, error starting an Oracle CDD/Repository session  
-CDD-F-STARTSESS, error starting session  
-SYSTEM-F-ABORT, abort
```

To create a transaction log, invoke the OpenVMS Log Manager Control Program (LMCP) utility. Then use the LMCP CREATE LOG command in the following format:

```
$ RUN SYS$SYSTEM:LMCP  
LMCP>CREATE LOG /SIZE=nnnn -  
_LMCP>device:[dirspec]SYSTEM$node.LM$JOURNAL/OWNER=SYSTEM  
LMCP>EXIT
```

In this example, nnnn is the size of the transaction log in blocks. By default, the size of the transaction log is 4000 blocks. The device:[dirspec] is the full specification of the directory in which you want to create the transaction log, and node is the name of the node.

For detailed information on creating transaction logs and managing DECdtm services, see the OpenVMS documentation for system managers.

Oracle CDD/Repository uses DECdtm to handle its two-phase commit actions. DECdtm startup is automatic as long as the logical name SYSSDECDTM_INHIBIT is not defined for your system. If DECdtm is not started, a "file spec cannot be parsed" or a %SYSTEM-F-ABORT error occurs.

To prevent these errors, perform the following steps:

1. Log in to any node in the VMScluster.
2. Enable OPER privilege.
3. Check that the system executive logical SYSSDECDTM_INHIBIT is not defined. If it is defined, regardless of its translation, deassign it using the OpenVMS System Management (SYSMAN) utility as shown in the following example:

```
$ SHOW LOGICAL SYSSDECDTM_INHIBIT  
"SYSSDECDTM_INHIBIT" = "YES" (LNM$SYSTEM_TABLE)  
$ RUN SYS$SYSTEM:SYSMAN  
SYSMAN> SET ENVIRONMENT/CLUSTER  
SYSMAN> SET PROFILE/PRIVILEGES=SYSNAM  
SYSMAN> DO DEASSIGN/SYSTEM/EXECUTIVE SYSSDECDTM_INHIBIT
```

4. Execute the DECdtm startup procedure, which defines the SYSSJOURNAL logical name.

```
SYSMAN>DO @SYSS$STARTUP:DECDTM$STARTUP.COM FULL  
SYSMAN>EXIT
```

5. Edit the SYSS\$STARTUP:SYLOGICALS.COM command procedure to delete the SYSS\$DECDTM_INHIBIT definition. This ensures that DECdtm services start automatically when you boot the system.

1.3 Invoking the VMSINSTAL Procedure

To start the installation procedure, invoke the VMSINSTALL command procedure:

```
$ @SYS$UPDATE:VMSINSTAL saveset-name kit-source
```

For saveset-name, use either CDDV74100I074 for OpenVMS IA-64 systems, or CDDV74100A074 for OpenVMS Alpha systems.

For kit-source, specify a directory into which the kit has been downloaded. For example: DKA400:[CDD.KIT]

The installation guide *Installing Oracle CDD/Repository* is available on MyOracleSupport in Adobe Acrobat PDF format:

```
Top Tech Docs\Database\Rdb\CDD/Repository\Documentation Index\  
Installing Oracle CDD/Repository Release 7.4 for OpenVMS
```

and from the Oracle Rdb main web page. At the bottom of the page are links to documentation for all of the Rdb product family releases. Following is the link for the Rdb product family page:

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

1.4 Enhancements and New Features

This section describes new and changed functionality in Oracle CDD/Repository.

1.4.1 Improved Support for Oracle Rdb SQL Builtin and Aggregate Functions

This release of Oracle CDD/Repository includes improved support for Oracle Rdb SQL builtin and aggregate functions.

- SUBSTRING (Oracle Database NULL semantics), SUBSTR, and SUBSTRB
- TRIM (Oracle Database NULL semantics), BTRIM, LTRIM, RTRIM
- COUNT(expression)

This new support matches the changes made to Oracle Rdb V7.3.1.1 (see the Oracle Rdb release notes: *NULL Elimination Semantics Now Supported by COUNT Function*).

- MEDIAN
- STDDEV, STDDEV_POP, STDDEV_SAMP

- VARIANCE, VAR_POP, VAR_SAMP
- TRANSLATE (Oracle Database syntax variation)
- CURRENT_UID, SESSION_UIC, SYSTEM_UID
- Sequence references using the pseudo columns CURRVAL and NEXTVAL

Note

The CREATE SEQUENCE statement is not supported at this time.

- SYS_GET_DIAGNOSTIC

In prior versions INTEGRATE DATABASE ... CREATE PATHNAME may generate informational messages such as shown in this example when these functions were used.

```
integrate database
  filename CDD_INTEGRATE_DB
  create path cdd$default.CDD_INTEGRATE_DB;
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
```

Subsequent attempts to INTEGRATE DATABASE ... ALTER FILES would report fatal errors trying to integrate missing tables because that definition was incomplete.

```
integrate database
  pathname cdd$default.CDD_INTEGRATE_DB
  alter files;
%CDD-I-INT_ERROR, error during integrate caused by EMPS
%RDB-E-NO_META_UPDATE, metadata update failed
-RDMS-F-GFLDNOEX, there is not a global field named NAME1 in this database
%CDD-I-INT_CONT, integrate continuing with next object
```

This release of Oracle CDD/Repository will now process such references in default values, views and other table metadata. Additionally, CDO SHOW RECORD will attempt to format the function definition.

Also supported is the enhanced SQL ORDER BY clause introduced in Oracle Rdb release 7.4. The ORDER BY clause supports the NULLS FIRST and NULLS LAST clauses and these are now understood by the INTEGRATE statement.

1.4.2 Updated Error Message Documentation in SYSSHELP

In release 7.4.1.0.0 all Oracle CDD/Repository error message files have been updated to provide updated formatting. The following files, available in SYSSHELP, have been updated:

- DMU_MSG.DOC
- CDO_MSG.DOC
- CDDL_MSG.DOC
- CDDV_MSG.DOC
- CDDX_MSG.DOC
- CDD_MSG.DOC

- MCS_MSG.DOC

1.5 Software Errors Fixed

This kit contains all fixes made to previous versions of Oracle CDD/Repository and also addresses the problems described in the following sections.

1.5.1 Invalid CDD\$TEMPLATE Logical Causes DEFINE REPOSITORY to Fail

If the logical CDD\$TEMPLATE points to an invalid directory or one where the file CDD\$DIRECTORY.CDD does not exist and the logical CDD\$TEMPLATEDB is valid and that directory contains file CDD\$DATABASE.RBF, the CDO DEFINE REPOSITORY command fails in releases prior to 7.4.1.0.0.

Example:

```
$ show logical cdd$template*
(LNM$SYSTEM_TABLE)
  "CDD$TEMPLATE" = "DISK1:[NOTTHERE]"
  "CDD$TEMPLATEDB" = "SYS$COMMON:[CDD$TEMPLATEDB73]"

$ dictionary operator
CDO> define repository DISK2:[CDD].
%RMU-W-PREVAACL, Restoring the root ACL over a pre-existing ACL.
  This is a normal condition if you are using the CDO utility.
%CDO-E-ERRDEFINE, error defining an object
-CDD-F-ERRCREDIC, error creating dictionary
-RDB-F-SYS_REQUEST, error from system services request
-RDMS-F-FILACCERR, error creating database file DISK2:[CDD]CDD$DATABASE.RDB;1
-RMS-E-FEX, file already exists, not superseded
-CDD-W-NOTEMPLATE, no template dictionary available
-CDD-E-CPYERRIN, error opening input file .....
-RMS-E-DNF, directory not found
-FOM-W-NOTRAN, no string translation performed
```

Workaround:

Redefine the logical CDD\$TEMPLATE to point to the correct directory.

This problem has been corrected in Oracle CDD/Repository Release 7.4.1.0.0.

The DEFINE REPOSITORY command will now execute as follows, if the CDD\$TEMPLATE logical points to an invalid directory or one where the file CDD\$DIRECTORY.CDD does not exist.

```
$ dictionary operator
CDO> define repo DISK2:[CDD].
%CDD-I-CRECONT, define repository continuing without template
CDO>
```

1.6 Enhancements and New Features in Prior Releases

1.6.1 Support for JOIN Clause in RSEs From Oracle Rdb

In releases prior to 7.3.0.0.0, the existence of a join clause in a relational selection expression would cause the integrate of a database or table into an Oracle CDD/Repository repository to issue the following informational message and the repository definition would be marked as incomplete, therefore not able to be integrated back into the database.

```
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
```

The JOIN clause is now supported in Oracle CDD/Repository release 7.3.0.0.0 for Rdb integrate functionality and the SQL INTEGRATE command will no longer result in the informational message.

1.6.2 Support for Value List Expressions From Oracle Rdb

In releases prior to 7.3.0.0.0, the existence of a value list expression would cause the integrate of a database or table into an Oracle CDD/Repository repository to issue the following informational message and the repository definition would be marked as incomplete, therefore not able to be integrated back into the database.

```
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
```

Value lists are now supported in Oracle CDD/Repository release 7.3.0.0.0 for Rdb integrate functionality and the SQL INTEGRATE command will no longer result in the informational message.

1.6.3 Explanations and User Actions Updated

Bug 12606677

In release 7.2.0.4.0, all Oracle CDD/Repository error message files have been modified to provide updated explanations and user actions and to match the formatting of Oracle Rdb message files. The following files, available in SYSSHELP, have been updated:

- DMU_MSG.DOC
- CDO_MSG.DOC
- CDDL_MSG.DOC
- CDDV_MSG.DOC
- CDDX_MSG.DOC
- CDD_MSG.DOC
- MCS_MSG.DOC

1.6.4 Integrate Enhanced to Support New BLR\$K_AGG_COUNT Syntax

Oracle CDD/Repository release 7.2.0.4.0 now supports the new Oracle Rdb BLR syntax for BLR\$K_AGG_COUNT.

1.7 Software Errors Fixed in Prior Releases

1.7.1 RDMS-F-INDTOOBIG, Requested Index is Too Big

Bug 3114793

Even though an Oracle Rdb index definition limited the size of an index key, Rdb returned an error when integrating a record with that index from Oracle CDD/Repository. Oracle CDD/Repository was losing the index information limiting the size of the field in the key during the integrate into CDD operation. The following example shows the scenario where the problem happened and the resulting errors.

```
$ dictionary operator
define field x datatype is text size is 255.
define record zzz.
    x.
end record.
exit
$
$ SQL$
attach 'path test';
create table from cdd$default:zzz;
create index x_idx on x (x size is 30);
commit;
exit;
$
$ dictionary operator
define field y datatype is text size is 30.
define record zzz.
    x.
    Y.
end record.
exit
$
$ SQL$
attach 'path cdd$default:test';
integrate table zzz alter files;
exit
%CDD-I-INT_ERROR, error during integrate caused by
%RDB-E-NO_META_UPDATE, metadata update failed
-RDMS-F-INDTOOBIG, requested index is too big
%CDD-I-INT_CONT, integrate continuing with next object
```

This problem has been corrected in Oracle CDD/Repository release 7.3.0.0.0.

1.7.2 Create Table From Fails When CDO Record Has Check Clause

Bug 25790905

If the default directory was set to a directory other than the one where a record would be defined, Oracle CDD/Repository would sometimes create a check constraint with an incorrect definition for the fully qualified field defined in the constraint for the record. The following is an example of a scenario where the problem would occur.

```

$ dictionary operator
CDO> set default cdd$default
CDO> define field cdo$fld.w datatype signed longword.
CDO> define field cdo$fld.v datatype date vms.
CDO> define record cdo$rec.t
constraint check_date check (v in t >= '17-nov-1858')
NOT DEFERRABLE.
cdo$fld.w.
cdo$fld.v.
end record.
CDO> sho rec/full cdo$rec.t
Definition of record T
| Contains field W
| | Datatype signed longword
| Contains field V
| | Datatype date
| Constraint CHECK_DATE (V IN CDO$REC.T GE "17-nov-1858") NOT DEFERRABLE
CDO> exit
$ SQL$
SQL> att 'path cdo$db:my_db';
SQL> create table from cdo$rec.t;
%RDB-E-NO META UPDATE, metadata update failed
-RDB-E-OBSOLETE_METADA, request references metadata objects that no longer
exist
-RDMS-F-RELNOEXI, relation CDO$REC.T does not exist in this database

```

This problem has been corrected in Oracle CDD/Repository release 7.3.0.0.0.

1.7.3 DEFINE REPOSITORY Failed to Use Template With Multiversion Rdb

The CDO DEFINE REPOSITORY command creates a repository from an existing template repository using RMU. It did not properly invoke the image, in an Rdb multiversion environment, causing the repository to be created correctly, but not by more efficiently doing so using the template. This problem is demonstrated in the following example.

```

$ dictionary operator
define repository disk1:[cddrepository].
%CDD-I-CRECONT, define repository continuing without template

```

This problem has been corrected in Oracle CDD/Repository release 7.3.0.0.0. Repositories will now be created correctly from a template after invoking the appropriate RMU image.

1.7.4 Problem Changing a Relation With Constraints

When changing a relation constraint, it is possible for an invalid constraint source to be generated by Oracle CDD/Repository on OpenVMS 8.4-1H1 systems. The following could be the result of the changed relation, where the source includes an erroneous series of commas and spaces. The incorrect constraint source is visible via the SQL SHOW TABLE command.

```

$ REPOSITORY OPERATOR
define field a datatype signed longword.
define field b datatype signed longword.
define relation r constraint r_unique unique a.
    a. b. end.
change relation r
    constraint r_unique unique a
end.
exit
$ SQL$
declare alias pathname testdb;
create table from r;
show table r
Information for table R
...
Table constraints for R:
R UNIQUE
    Unique constraint
        Null values are considered distinct
Table constraint for R
Evaluated on each VERB
Source:... unique A, , , , , ,

```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.6.0.

1.7.5 Unexpected CDD\$_PAGNOTFND, CDD\$_EMPTYCELL and CDD\$_CELNOTEMP Errors

Bug 20184106

In release 7.2.0.4.0, the DMU COPY and BACKUP commands failed as follows:

```

$ MCR DMU
DMU> backup aaa tst.bak
%CDD-F-PAGNOTFND, page not in hash table

```

Changes made to eliminate alignment faults in the processing of lists, such as lists of attributes and history lists, could sometimes access incorrect data, leading to incorrect CDD\$_PAGNOTFND, CDD\$_EMPTYCELL or CDD\$_CELNOTEMP errors. Another example where the CDD\$_CELNOTEMP error would occur is when the /AUDIT qualifier was used for the DMU CREATE command.

```

$ MCR DMU
DMU> create/audit new directory
%CDD-E-CELNOTEMP, cell is not empty

```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.5.0.

1.7.6 RDB-F-SEGSTR_PARAM Error During Integrate

Bug 2211614

When integrating a database into Oracle CDD/Repository, the integration failed with a RDB-F-SEGSTR_PARAM error. The error occurred when a NULL ("") character was the last character in a QUERY HEADER description.

For example:

```
SQL> CREATE DATABASE FILE TESTDB;
SQL> CREATE TABLE FOO (C1 CHAR(4) QUERY HEADER ' ABC ' / '');
SQL> COMMIT;
SQL> INTEGRATE DATABASE FILE TESTDB CREATE PATH DB2;
%CDD-F-INTFAIL, integration failed
-RDB-F-SEGSTR_PARAM, illegal parameter to segmented string routine
```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.4.0.

1.7.7 CDO Could Not be Invoked Using SYSMAN

Bug 898545

In releases of Oracle CDD/Repository prior to 7.2.0.4.0, if you attempted to invoke CDO using SYSMAN, it would go into a process loop.

For example:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> do dictionary operator
%SYSMAN-I-OUTPUT, command execution on node XYZ
...
```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.4.0. The SYSMAN utility can now be used to invoke the CDO utility.

1.7.8 CDO Show Version Only Shows Two Digits

Bug 11782875

In releases of Oracle CDD/Repository prior to 7.2.0.4.0, the CDO SHOW VERSION command only displayed the first 2 digits of the release number.

For example:

```
$ REPOSITORY OPERATOR
CDO> show version
Installed version of Oracle CDD/Repository is V7.2
```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.4.0. CDO will now display the complete release number as follows.

```
$ REPOSITORY OPERATOR
CDO> show version
Installed version of Oracle CDD/Repository is V7.2-040
```

1.7.9 %PASCAL-E-LNETOOLNG on Compile When Field Comments More Than 255 Characters

Bug 671947

With releases prior to Oracle CDD/Repository release 7.2.0.4.0, the SQL integrate command did not preserve multiline comments and instead stored them in CDD as a single line. This caused an error when compiling a Pascal program using a field with a multiline comment that was more than 255 characters long. This problem is shown by the following example.

```

$ sql
create database file dic_test;
create domain grunter_dom char(1);
    comment on domain grunter_dom is
        ''
        / 'This comment is at least 255 chars long .Standard AUTHORIZE '
        / 'access is allowed. Only the first 24 characters of this field '
        / 'are valid, with each character position representing one hour.'
        / 'The presence of a # in a character position indicates that '
        / 'access is allowed for that particular hour.';
create table thumper (grunter grunter_dom);
commit;
exit
$ type dic_test.pas
module cdd_test (input,output);
type    %dictionary 'dic_test.rdb$relations.thumper'
end.
$ sql
integrate database file dic_test create pathname dic_test;
commit;
exit
$ pascal /noobj/nowarn dic_test.pas
Error: %PASCAL-E-LNETOOLNG, Line too long, is truncated to 255 characters

```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.4.0.

1.7.10 CDD Pathname Versions Truncated on SQL ALTER TABLE

Bug 467011

The CDD Pathname displayed in SQL after an ALTER TABLE command was missing the last digit of the version number, using Oracle CDD/Repository releases prior to 7.2.0.4.0.

For example, when showing a table that used the second version of an Oracle CDD/Repository table, SQL would display the following, where the version of 2 was missing because the version was truncated by 1 digit.

```

$ SQL$
SQL> attach 'pathname my_db';
SQL> show table my_table;
...
CDD Pathname:  SYS$COMMON:[CDDPLUS]MY_TABLE;
...

```

When displaying a table that was using version 10 of the table in CDD, the following was displayed.

```

$ SQL$
SQL> attach 'pathname my_db';
SQL> show table my_table;
...
CDD Pathname:  SYS$COMMON:[CDDPLUS]MY_TABLE;1
...

```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.4.0.

1.7.11 Access Violation During Integrate Alter Files

Bug 2706622

If a user attached to an Oracle Rdb database via filename and dropped a table, then later created the same table while attached by pathname, a duplicate table was created in the repository. This caused an access violation during a subsequent integrate.

For example:

```
$ SQL$
SQL> attach 'filename my_db';
SQL> drop table abc;
SQL> commit;
SQL> exit
$ SQL$
SQL> attach 'pathname cdd$default.my_db';
SQL> create table abc (uvw char(5), xyz char(10));
SQL> commit;
SQL> exit
$ SQL$
SQL> integrate database path cdd$default.my_db alter files;
%CDD-E-INTFAIL, integration failed
-SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual address=000000000000
0018, PC=0000000000E1E3C4, PS=0000001B
```

The problem causing the access violation has been corrected in Oracle CDD/Repository release 7.2.0.4.0. Any attempt to create an object that already exists in a database definition in the repository will result in the following error.

```
%CDD-E-EXSTINDB, dictionary database already contains ABC
```

1.7.12 Using SQL CREATE After DROP of Object With Same Name Caused ACCVIO

Bug 415799

If attached to an Oracle Rdb database via pathname and a table was dropped without also committing the transaction, a subsequent create of another table with the same name caused an access violation.

For example:

```
$ SQL$
SQL> attach 'path cdd$default.foo.bar.db2';
SQL> create table from cdd$default.foo.my_record;
SQL> drop table my_record;
SQL> create table from cdd$default.foo.bar.my_record;
%CDD-F-NO_DESCRIPTION, error returning description of dictionary entity
-CDD-F-NO_DESCRIPTION, error returning description of dictionary entity
-SYSTEM-F-ACCVIO, access violation, reason mask=00, virtual address=00000113, PC
=00D40110, PS=0000001B
```

The problem causing the access violation has been corrected in Oracle CDD/Repository release 7.2.0.4.0. If a commit is not done prior to the create command, a CDD\$_INTCOMROLL error will now occur, informing the user that a commit or rollback is required.

1.7.13 Alignment Faults Corrected

Bug 4926614

In releases of Oracle CDD/Repository prior to release 7.2.0.4.0, alignment faults occurred when invoking CDD/Repository images. Many of these alignment faults have been resolved in Oracle CDD/Repository release 7.2.0.4.0.

1.7.14 Database Prepared for OCI Services Fails to Integrate

Bug 8637337

A database that had been prepared for use with OCI Services for Oracle Rdb could not be integrated into an Oracle CDD/Repository dictionary, using the SQL INTEGRATE command. It would fail with the CDD\$_VALDEFFAIL error, as shown by the following example.

```
SQL> integrate database filename foo create path cdd$default.foo;
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
%CDD-I-BLRSYNINFO, unsupported entity - marked Incomplete
%CDD-I-MBLRSYNINFO, unsupported entity - marked Incomplete at mblr offset 33
%CDD-I-MBLRSYNINFO, unsupported entity - marked Incomplete at mblr offset 21
%CDD-E-VALDEFFAIL, entity !AS definition failed validation !AS
```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.3.0. Oracle Rdb databases prepared for OCI Services for Oracle Rdb release 7.3.0.1 and later can now be integrated into Oracle CDD/Repository dictionaries.

1.7.15 Not All NOT NULL Constraints Are Integrated into Rdb

Bug 6964861

If a record contained several unnamed NOT NULL constraints, the generated constraint names were sometimes not unique. Constraints with duplicate names were not integrated into Rdb.

This problem has been corrected in Oracle CDD/Repository release 7.2.0.2.0. Duplicate constraint names will no longer be generated.

1.7.16 DEFINE REPOSITORY Command Creates Excessively Large Files

In previous versions of Oracle CDD/Repository Release 7.2, it was possible for the DEFINE REPOSITORY command to errantly create a very large CDD\$DIRECTORY.CDD file. This file could potentially entirely fill a disk volume and if the disk volume had “file high-water marking” enabled, it was possible for the volume lock to be held during the creation and initialization of the CDD\$DIRECTORY.CDD file.

This problem has been corrected in Oracle CDD/Repository release 7.2.0.2.0.

1.7.17 CDO Fails Displaying Signed Longword Initial Value

Bug 5075301

When a CDO field was defined with datatype signed longword and the initial value was a very large negative number, CDO got an access violation. For example:

```
CDO> define field test
cont> datatype is signed longword
cont> initial_value is -2147483648.
CDO> show field test
Definition of field TEST
| Datatype          signed longword
%SYSTEM-F-ACCVIO, access violation, reason mask=04,
  virtual address=0000000003130100, PC=FFFFFFFF84260D51, PS=0000001B
```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.1.0.

1.7.18 CDDL Did Not Properly Store Signed and Unsigned QUAD Initial Values

Bug 4904683

In release 7.2 of Oracle CDD/Repository, CDDL did not correctly convert QUAD initial values from text to internal binary type, and the values were stored incorrectly. A record displayed with the DMU EXTRACT/RECORD command would display the incorrectly stored QUAD value as zero.

This problem has been corrected in Oracle CDD/Repository release 7.2.0.1.0.

1.7.19 DMU Did Not Properly Display Signed and Unsigned QUAD Initial Values

Bug 4904683

In release 7.2 of Oracle CDD/Repository, the DMU EXTRACT /RECORD command of the DMU utility did not properly display the initial values of its UNSIGNED QUAD and SIGNED QUAD data types. Incorrect numeric values were displayed.

This problem has been corrected in Oracle CDD/Repository release 7.2.0.1.0.

1.7.20 DMU Performs an ACCVIO When Displaying an H_FLOAT Initial Value

Bug 4904683

In release 7.2 of Oracle CDD/Repository, DMU would get an ACCVIO error if you used the DMU EXTRACT/RECORD command to display an H_FLOAT value. The H_FLOAT value was stored correctly but not displayed correctly.

This problem has been corrected in Oracle CDD/Repository release 7.2.0.1.0.

1.7.21 CDO EXTRACT RECORD /LANGUAGE=CC Problem Handling Computed By Fields

The CDO EXTRACT RECORD /LANGUAGE=CC command did not properly handle computed by fields. It would extract the fields, rather than ignore them, as illustrated below:

```
CDO> extract record RETURN_STATUS /language=cc

struct return_status
{
  char return_code; /* Text */
  struct {char Unspecified1; } successful; /* Text */
  struct {char Unspecified1; } failed; /* Text */
  signed long return_value; /* Signed Longword */
  char status_parameters[100]; /* Text */
};
```

This problem has been fixed in Oracle CDD/Repository release 7.2. It will now extract this record as follows:

```
CDO> extract record RETURN_STATUS /language=cc

struct return_status
{
  char return_code; /* Text */
/* virtual field ignored */
/* virtual field ignored */
  signed long return_value; /* Signed Longword */
  char status_parameters[100]; /* Text */
};
```

1.7.22 CDO EXTRACT RECORD/LANGUAGE=CC Problem Handling Based On Fields

The CDO EXTRACT RECORD /LANGUAGE=CC command did not properly handle the length attribute for fields with a based on clause. If a length was specified for the field and the field it was based on, the two lengths would be added together in the displayed length, for example:

```
CDO> define field f1 datatype text 3.
CDO> define record r1.
cont> field1 datatype text size is 3 characters based on f1.
cont> end.
CDO> extract record r1 /lang=cc
struct r1
{
  char field1[6]; /* Text */
};
```

This problem has been fixed in Oracle CDD/Repository release 7.2. The length in the previous example is now displayed as 3, rather than 6.

1.8 Known Restrictions

The following are known restrictions in Oracle CDD/Repository.

1.8.1 CDD Does Not Fully Support Index with Explicit Node Size or Percent Fill

If an index is defined in an Oracle Rdb database using an explicitly specified node size or percent fill clause and SQL INTEGRATE is used to define the index in a CDD repository, the index definition created in the repository will not include those attributes of the index. If the index is later integrated from the repository to the database files, those attributes will be lost.

1.8.2 Clarification on Object Name Syntax in CDO utility

Bug 10040830

The CDD/Repository CDO utility commands support field, record, database and other object names specified as either an identifier or a quoted string. Object names can contain the letters A to Z, digits 0 to 9, \$ and _, with a maximum of 31 characters.

Identifiers require that the first character be a letter. If you would like to use a name that starts with a digit, \$ or _, you must use a quoted string to specify the name. If you do not use a quoted string, you will get a syntax error. For example:

```
$ SQL$
SQL> CREATE DATABASE FILENAME 1TEST;
SQL> INTEGRATE DATABASE FILENAME 1TEST CREATE PATHNAME CDD$TOP.1TEST;
SQL> EXIT
$ DICTIONARY OPERATOR
CDO> DIR 1TEST
DIR 1TEST
  ^
%CDO-E-KWSYNTAX, syntax error in command line at or near 1
CDO> DIR "1TEST"
Directory SYS$COMMON: [CDDPLUS]

1TEST                                CDD$DATABASE
```

Also note that the DMU utility does not support names specified as a quoted string. Therefore, you will not be able to directly reference any objects that must be specified using a quoted string.

1.8.3 Use of SQL DROP Statement Against CDD/Repository Definitions

If attached to an Oracle Rdb database via pathname and a table or domain is dropped, a COMMIT must be executed before recreating another table or domain with the same name. For example:

```
$ SQL$
SQL> attach 'path cdd$default.foo.bar.db2';
SQL> create table from cdd$default.foo.my_record;
SQL> drop table my_record;
SQL> commit;
SQL> create table from cdd$default.foo.bar.my_record;
```

If the commit is not done, a CDD\$_INTCOMROLL error will occur when executing the create command, informing the user that a commit or rollback is required.