Oracle® CDD/Repository for OpenVMS

Table of Contents

Oracle® CDD/Repository for OpenVMS

Release Notes

Release 7.2.0.2

November 2007

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

party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

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 7.2.0.2.

Conventions

HP OpenVMS Industry Standard 64 Integrity Servers is often referred to as OpenVMS I64.

In this manual, OpenVMS means both the OpenVMS Alpha operating system and the OpenVMS I64 operating system.

Chapter 1
Release Notes for Oracle CDD/Repository Release 7.2.0.2

This document provides release notes for Oracle CDD/Repository for HP 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 7.2.0.2

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

NOTE

All Oracle CDD/Repository Release 7.2 kits are full kits. There is no requirement to install any prior release of Oracle CDD/Repository when installing new CDD/Repository release 7.2 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 8.2 or later
    - ♦ OpenVMS I64 Version 8.2–1 or later
    - ♦ Oracle Rdb release 7.0 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 execute the following command:

```
$ @ SYS$LIBRARY:RDB$SETVER 72
```

- The installation requires approximately 60,000 blocks for OpenVMS Alpha systems.
- The installation requires approximately 90,000 blocks for OpenVMS I64 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 SYS$JOURNAL 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 SYS$DECDTM_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 SYS$DECDTM_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 SYS$DECDTM_INHIBIT
  "SYS$DECDTM_INHIBIT" = "YES" (LNM$SYSTEM_TABLE)
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER
SYSMAN> SET PROFILE/PRIVILEGES=SYSNAM
SYSMAN> DO DEASSIGN/SYSTEM/EXECUTIVE SYS$DECDTM_INHIBIT
```

4. Execute the DECdtm startup procedure, which defines the SYS$JOURNAL logical name.

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

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

1.2.2 Invoking the VMSINSTAL Procedure

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

```
@SYS$UPDATE:VMSINSTAL saveset-name device-name
```

For *saveset−name*, use CDDV72020I072 for OpenVMS I64 systems, and CDDV72020A072 for OpenVMS Alpha systems.

For *device−name*, use the name of the device on which the media is mounted. If the device is a disk drive, you also need to specify a directory, for example: *DKA400:[RDB.KIT]*

The full Oracle CDD/Repository 7.2 Installation Guide is available on MetaLink in Adobe Acrobat PDF format:

```
Top Tech Docs\Database\Rdb\CDD/Repository\Documentation Index\
  Installing Oracle CDD/Repository Release 7.2 for OpenVMS
```

and on OTN:

```
www.oracle.com/technology/documentation/rdb.html#cdd/
  Installing Oracle CDD/Repository Release 7.2 for OpenVMS
```

1.3 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.

1.3.1 DEFINE REPOSITORY Command Creates Excessively Large Files

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

This problem has been corrected in Oracle CDD/Repository release 7.2.0.2.

1.3.2 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 datatype signed longword initial_value –21`
CDO> show field test
Definition of field TEST
|   Datatype                signed longword
%SYSTEM-F–ACCVIO, access violation, reason mask=04, virtual addre`
4100, PC=FFFFFFFF84260D51, PS=0000001B
```

This problem has been corrected in Oracle CDD/Repository release 7.2.0.1.

1.3.3 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.

1.3.4 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.

1.3.5 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.

1.3.6 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 */
```

1.3.7 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.

| Contents