Oracle9*i*

Release Notes

Release 2 (9.2.0.1.0) for Sun Solaris (32-bit)

May 2002

Part No. A97348-01

This document accompanies Oracle9*i* release 2 (9.2.0.1.0) for Sun Solaris. Its contents supplement or supersede information in the installation guide for this release, or in the Oracle9*i* documentation library.

Topics:

System Requirements

Except as noted here, system requirements are in the installation guide for this release, and are current as of the release date.

Hard Disk Space Requirements

The space requirements listed on the Available Products window apply to installations that include a database. If you select the Software Only configuration type, then you will require 2 GB.

Updated Requirements

Oracle Corporation updates these release notes online at the following site:

`http://docs.oracle.com`

If you need assistance with navigating the Oracle documentation site, refer to the following site:

`http://docs.oracle.com/instructions.html`

Refer also to the Certify Web Pages on Oracle*MetaLink*, which provide certified configuration information for Oracle and non-Oracle products. To access Certify:

1. Register or log in to Oracle*MetaLink* at the following web address:

   `http://metalink.oracle.com`

2. Select Product Lifecycle from the Oracle*Metalink* navigation bar.
3. Select Certifications in the Product Lifecycle window navigation bar.

Documentation

Additional product `README` files are located in their respective product directories under the `$ORACLE_HOME` directory and in the `$ORACLE_HOME/relnotes` directory.

Installation Issues

This section provides information about the following topics:

Multiple CD-ROM Installation

During installation of Oracle9*i* release 2 (9.2.0.1.0), you will be prompted to insert additional CD-ROMs from the set that make up Oracle9*i* release 2 (9.2.0.1.0). After inserting the requested disk, change the path in the *Disk Location* text box to reflect the root directory of the newly mounted CD-ROM.

For example, when you insert Disk 3 with a directory path of `/cdrom/orcl920_3`, change the path in the *Disk Location* dialog to `/cdrom/orcl920_3`.

runInstaller Script

Because it is necessary to insert and eject more than one CD-ROM during installation, you must not launch Oracle Universal Installer by running the `runInstaller` script from a shell where the current working directory is the CD-ROM mount point, or by clicking on the script in the *File Manager* window. In an X Window environment, it is possible to launch the Installer this way, but then the installation will fail because you will not be able to eject a software CD-ROM until you end the installation session.

Installing Databases with Database Configuration Assistant

Review the following information before running Database Configuration Assistant.

SYS and SYSTEM Password Change Requirement

If you use Database Configuration Assistant to create a database, be aware that you will be required to change the SYS and SYSTEM passwords at the end of the configuration process. This is a new security procedure designed to protect access to your data.

Database Migration

If you are upgrading from Oracle8 release 8.0.6 to Oracle9*i* release 2 (9.2.0.1.0) and you have Oracle *inter*Media installed on your system, then you cannot use Database Migration Assistant. You must migrate the database manually. For information on manual database migration, refer to *Oracle9i Database Migration Release 2 (9.2)*.

Installing with Response Files

For installation with a response file, the path to the response file must be the full path on the system. The Oracle Universal Installer does not handle relative paths properly.

Unzip Utility for Downloading and Installing Oracle Patches

An unzip utility is provided with Oracle9i Release 2 (9.2.0.1.0) for uncompressing Oracle patches downloaded from Oracle*MetaLink*. The utility is located in the following directory:

```
ORACLE_HOME/bin/
```

Product-Related Issues

This section provides information on the following topics:

Demo Schema

If you select a multibyte character set or UTF as the national character set in Oracle9*i* release 2 (9.2.0.1.0), then you must recreate the demo schema and the database installation.

For more information on creating schemas, schema dependencies and requirements, refer to the `readme.txt` file in the `$ORACLE_HOME/demo/schema` directory.

Character Sets

The following section provides information on restrictions and updates to character sets.

Oracle9*i* NCHAR Datatypes

In Oracle9*i* release 2 (9.2.0.1.0), the SQL NCHAR datatypes are limited to the Unicode character set encoding (UTF8 and AL16UTF16). Alternative character sets such as the fixed-width Asian character set JA16SJISFIXED in Oracle8*i* are no longer supported.

To migrate existing NCHAR, NVARCHAR, and NCLOB columns, export and import NCHAR columns, complete the following steps:

1. Export all SQL NCHAR columns from Oracle8*i*.
2. Drop the SQL NCHAR columns.
3. Migrate the database to Oracle9*i*.
4. Import the SQL NCHAR columns in to Oracle9*i*.

AL24UTFFSS Character Set

Oracle9*i* release 2 (9.2.0.1.0) does not support the Unicode character set AL24UTFFSS introduced in Oracle7. This character set was based on the Unicode standard 1.1, which is now obsolete.

Oracle9*i* release 2 (9.2.0.1.0) supports the Unicode database character sets AL32UTF8 and UTF8. These database character sets include the Unicode enhancements based on the Unicode standard 3.0.

To migrate the existing AL24UTFFSS database, upgrade your database character set to UTF8 before upgrading to Oracle9*i*. Oracle Corporation recommends that you use the Character Set Scanner for data analysis before attempting to migrate your existing database character set.

Character Set Scanner

Set the LD_LIBRARY_PATH variable to include the `$ORACLE_HOME/lib` directory before running the Character Set Scanner (csscan) from the `$ORACLE_HOME` directory. If you do not correctly set the LD_LIBRARY_PATH variable, then the csscan utility will fail.

Oracle Internet Directory (OID)

Review the following information if you intend to install Oracle Internet Directory (OID).

Starting Up OID Server

By default, the OID server is started on port 389. If this port is unavailable, then OID server is started on a different port, which is logged in the following file:

```
$ORACLE_HOME/ldap/install/oidca.out
```

Custom Installation and Global Database Name

When performing a custom Oracle Internet Directory installation, do not change the global database name or the Oracle SID.

Upgrade from Enterprise Edition Oracle9*i* or Oracle8*i*

If you have installed in the same ORACLE_HOME either Oracle Internet Directory release 3.0.1.x and the complete release of Oracle9*i* (9.0.1) Enterprise Edition, or Oracle Internet Directory 2.1.1.x and the complete release of Oracle 8*i* (8.1.7) Enterprise Edition, then you must first upgrade Oracle Internet Directory to the release 9.2.0.x.x version, and then upgrade as a separate step either Oracle9*i* Enterprise Edition Release 1 (9.0.1) or Oracle8*i* Release 3 (8.1.7) to Oracle9*i* Enterprise Edition Release 2 (9.2.0.x.x).

> **See Also:** *Oracle Internet Directory README* for more information on Oracle Internet Directory utilities, and necessary pre-upgrade and post-upgrade tasks.

Oracle Real Application Clusters

Review the following section if you will install Oracle Real Application Clusters.

Restrictions for Installing Real Application Clusters

The Cluster Manager implementation may not be able to handle 32-bit and 64-bit clients concurrently. This will prevent 32-bit and 64-bit Oracle Real Application Clusters executables from being used at the same time within the same cluster domain.

If a database is not set up with the Oracle Real Application Clusters option, then this restriction does not apply to the Oracle executables.

Real Application Clusters Pre-Installation Tasks

The following pre-installation instruction supplements the Solaris pre-installation procedure described under "Oracle Real Application Clusters" in Chapter 2, Pre-Installation, *Oracle9i Installation Guide Release 2 (9.2.0.1.0) for UNIX systems.*

Install the Oracle 9.2.0.1.0 UDLM patch for Sun clusters, which is located on the first Oracle9*i* release 2 (9.2.0.1.0) CD Pack. This patch must be installed on all nodes of the cluster on which you plan to run an Oracle9i release 2 (9.2.0.1.0) cluster database instance, even if you previously installed Oracle9*i* Release 1 (9.0.1) UDLM patch on these nodes.

For instructions on installing the UDLM patch, refer to the `README.udlm` file, which is located in the `/cdrom/racpatch` directory.

Real Application Clusters Custom Installation Requirement

If you plan to create an Oracle Enterprise Manager repository in an existing database, and you plan to use the DRSYS tablespace for the repository, then ensure that the DRSYS tablespace raw device data file has an additional 50 MB of free space. This is in addition to the 250 MB size documented for this raw device.

Real Application Clusters and Database Upgrade Assistant

If you use Database Upgrade Assistant to upgrade an earlier Oracle database version (the "source" database) to Oracle9*i* release 2 (9.2.0.1.0) (the "target" database), then the upgraded database will always use the server parameter file SPFILE by default to store `init.ora` file parameters. If the source database also uses SPFILE (either a cluster filesystem file or a shared raw device), then the upgraded target database also uses the same SPFILE.

If the source database does not use an SPFILE, then the target database uses a default server parameter file, `spfile.ora`, which is located in the `$ORACLE_HOME/dbs/` directory.

If your platform does not support a cluster file system, then you must move the SPFILE to a shared raw device, using the following procedure:

1. Create an SPFILE with the following commands:

   ```
   $ sqlplus "/ as sysdba"
   SQL> create pfile='?/dbs/initdbname.ora' from
   spfile='?/dbs/spfile.ora';
   SQL> create spfile='/dev/vx/rdsk/oracle_dg/dbname_spfile' from
   pfile='?/dbs/initdbname.ora';
   SQL> exit;
   ```

   where *dbname* is the name of your cluster database.

2. Go to the `$ORACLE_HOME/dbs` directory using the following command:

   ```
   $ cd $ORACLE_HOME/dbs
   ```

3. Create an `$ORACLE_HOME/dbs/initsid.ora` file, where *sid* is the system identifier of the instance on the node. The `initsid.ora` file must contain the following line:

   ```
   SPFILE='/dev/vx/rdsk/oracle_dg/dbname_spfile'
   ```

4. Copy the `initsid.ora` file to the remote nodes on which the cluster database has an instance with the following commands:

   ```
   $ rcp initsid.ora nodex:$ORACLE_HOME/dbs/initsidx.ora
   ```

   where *sidx* is the system identifier of the instance on node *x*. Repeat the preceding `rcp` command for each member node of the cluster database.

5. Restart the cluster database with the following command syntax:

   ```
   $ srvctl stop database -d dbname
   $ srvctl start database -d dbname
   ```

Real Application Clusters and Database Configuration Assistant

The following section provides information on using Database Configuration Assistant (DBCA) to create a Real Application Clusters database.

Placing Datafiles On a Shared Non-OFA Cluster Configuration

If your ORACLE_HOME directory is not on a shared cluster filesystem partition, but you want to place datafiles, controlfiles, redo log files, or other database files on a shared cluster filesystem partition, then invoke DBCA using the following syntax to create the cluster database:

```
$ dbca -datafileDestination pathname
```

where *pathname* is the location where you want files to be placed.

For example, to place datafiles in the path `/ora/oradata`, give the following command:

```
$ dbca -datafileDestination /ora/oradata
```

> **Note:** For optimal performance and data security, Oracle Corporation recommends that you configure your database in accordance with the Optimal Flexible Architecture (OFA) standard. For more information on OFA, refer to *Oracle9i Administrator's Reference for UNIX Systems*.

Real Application Clusters Instance Management

After you have created a cluster database using DBCA, SYSDBA privileges are revoked for all users. As SYSDBA, you must grant SYSDBA privileges explicitly to the database user account that you plan to use for adding or deleting an instance to or from the cluster database.

For example, to grant SYSDBA privileges to the administrative user SYS, issue the following commands:

```
$ sqlplus "/ as sysdba"
SQL> grant sysdba to sys;
SQL> exit;
```

Post-Installation Issues

This section presents issues that can occur during post-installation:

Control File Size Limits

In addition to the database, a number of other Oracle features use control files to record metadata. The maximum size of control files is limited by the size of the minimum data block size that your operating system permits. On Sun Solaris, the minimum data block size is 2048 bytes, and the maximum size of control files is 20000 database blocks.

Checking Soft and Hard File Size Limits For a Shell

Oracle9*i* software includes native support for file sizes greater than 2 GB. However, some system shells impose a lower limit. Limits set in the shell are referred to as soft limits. Maximum values for file sizes are referred to as hard limits. The soft limits value can be temporarily raised to the hard limits value. The file size shell limits for all users connected to an Oracle9*i* database must be higher than the size of the largest data file. Oracle Corporation recommends that the file size shell limit be set to `unlimited`, which means that the shell will not impose a files size limit. You can check your shell to determine whether it will impose limits by using the following commands:

- To check currently used (soft) shell limits, enter:

  ```
  $ ulimit -Sa
  ```

- To check the current maximum (hard) limits, enter:

  ```
  $ ulimit -Ha
  ```

- Multiply the `file (blocks)` value by 512 to obtain the maximum size imposed by the shell. A value of `unlimited` is the operating system default. Refer to your platform documentation for the exact value.

The following table lists the Oracle file size limits. Block size is defined in the `db_block_size` parameter of the `$ORACLE_HOME/dbs/init`*`sid`*`.ora` file.

File Type	Maximum Size in Bytes
data file with block size 2048	8,589,932,544
data file with block size 4096	17,179,865,088
data file with block size 8196	34,359,730,176
data file with block size 16,384	68,719,460,352
data file with block size 32,768	137,438,920,704
Import/Export file	2,147,483,647
SQL*Loader file	2,147,483,647

How to Determine Whether Segments or Tablespaces are Using Compression

The following section provides additional information about database management.

Segments and Compression Settings

To find out which database segments are using compression, log in to the database as the user SYS, and create the view `all_segs` with the following create or replace view statement:

```
SQL> create or replace view all_segs
     (owner, segment_name,
       partition_name, spare1
as
select u.name, o.name, o.subname, s.spare1
from sys.user$ u, sys.obj$ o, sys.ts$ ts, sys.sys_objects so,
     sys.seg$ s, sys.file$ f
where s.file# = so.header_file
   and s.block# = so.header_block
   and s.ts# = so.ts_number
   and s.ts# = ts.ts#
   and s.ts# = so.object_id
   and o.owner# = u.user#
   and s.type# = so.object_type_id
   and s.ts# = f.ts#
   and s.file# = f.relfile#
union all
select u.name, un.name, NULLL, NULL
from sys.user$ u, sys.ts$ ts, sys.undo $ un, sys.seg$ s,
     sys.file$ f
where s.file# = un.file#
   and s.block# = un.block
   and s.ts# = un.ts#
   and s.ts# = ts.ts#
   and s.user# = u.user#
   and s.type# in (1, 10)
   and un.status$ != 1
   and un.ts# = f.ts#
   and un.file# = f.relfile#
union all
select u.name, to_char(f.file#)|| '.' || to_char(s.block#), NULL, NULL
from sys.user$ u, sys.ts$ ts, sys.seg$ s, sys.file$ f
where s.ts# = ts.ts#
   and s.user# = u.user#
   and s.type# not in (1, 5, 6, 8, 10)
   and s.ts# = f.ts#
   and s.file# = f.relfile#
/
```

After creating this view, you can issue queries against the view to find out whether a segment currently is compressed, as illustrated in the following examples:

- To determine if a segment is currently compressed, apply the following predicate in a query to the column `spare1`:

  ```
  bitand(spare1, 2048) > 0
  ```

  For example, to see if segments currently are compressed, issue a statement similar to the following:

  ```
  SQL> select * from all_segs where bitand(spare1,2048) > 0;
  ```

- To determine if a segment contains any compressed blocks, apply the following predicate in a query:

  ```
  bitand(spare1, 4096) > 0
  ```

  For example, to see which segments contain any compressed blocks, issue a statement similar to the following:

  ```
  SQL> select * from all_segs where bitand(spare1, 4096) > 0;
  ```

Tablespaces and Compression Settings

When you want to determine compression settings on a table space, log in as SYS, and create the view `compression_ts` with the following create or replace view statement:

```
SQL> create or replace view compression_ts (tablespace_name, flags) as
select ts.name, ts.flags from
sys.ts$ ts where ts.online$ !=3;
```

After creating this view, you can issue queries against it to find out the compression state of tablespaces, such as determining if a tablespace is currently set as DEFAULT COMPRESS, or DEFAULT NOCOMPRESS, as illustrated in the following examples:

- To determine if a tablespace is currently set as DEFAULT COMPRESS, use the following predicate:

  ```
  bitand(flags, 64) > 0
  ```

  For example, to see which tablespaces are currently DEFAULT COMPRESS, issue a statement similar to the following:

  ```
  SQL> select * from compression_ts where bitand(flags, 64) > 0
  ```

- To determine if a tablespace is currently set as DEFAULT NOCOMPRESS, use the following predicate:

  ```
  bitand(flags, 64) == 0
  ```

For example, to see which tablespaces are currently DEFAULT NOCOMPRESS, issue a statement similar to the following:

```
select * from compression_ts where bitand(flags, 64) == 0;
```

Known Bugs

The following is a list of known bugs that affect Oracle9*i* release 2 (9.2.0.1.0):

Error Messages Display Incorrectly in Japanese

Due to a problem in Java (Sun Microsystems bug identification number 4258198), Java cannot correctly display some error messages returned from UNIX operating systems. Oracle Corporation has assigned Oracle Bug identification number 2156506 to track this problem.

Thai Characters Display Incorrectly

Due to a problem in Java (Sun Microsystems bug identification number 4674864), Java cannot correctly display Thai characters. Oracle Corporation has assigned bug identification number 2342889 to track this problem.

To work around this problem, unset the LANG and LC_ALL environment variables.

Error Installing OLAP CWMLITE Tablespace

During installation, if you select Online Analytic Processing (OLAP) services, perform multiple installations on the same system, and create new databases during these installations, then CWMLite may have an invalid OLAP CWMLITE tablespace registry. Oracle Corporation has assigned bug identification number 2359208 to track this problem.

To work around this problem, use the following procedure after you have completed installation:

1. Ensure that the database and the listener are running.
2. Using the following command, start SQL*Plus as the administrative user SYS:

   ```
   sqlplus "/ as sysdba"
   ```

3. Using the following command, enable the display of text within the PL/SQL block:

   ```
   SQL> set serveroutput on;
   ```

4. Using the following command, verify whether the OLAP CWMLITE tablespace is valid:

```
SQL> execute
dbms_output.put_line(sys.dbms_registry.is_valid('AMD'));
```

   If the preceding command returns `0`, then the OLAP CWMLITE tablespace is invalid. Go to step 5.

   If the preceding command returns `1`, then the OLAP CWMLITE tablespace is valid, and no further testing needs to be done.

5. If the OLAP CWMLITE tablespace is invalid, turn on echoing with the following command:

```
SQL> execute cwm2_olap_manager.Set_Echo_on;
```

6. Validate the OLAP CMWLITE tablespace with the following command:

```
SQL> execute cwm2_olap_installer.Validate_CWM2_Install;
```

   After entering the preceding command, the OLAP CWMLITE registry is validated. During this process, screen messages list database objects such as Dimension, Dimension Attribute, and Level, and where these objects are created.

7. When the output stops, enter the following command to verify that the OLAP CWMLITE registry is now valid:

```
SQL> execute
dbms_output.put_line(sys.dbms_registry.is_valid('AMD'));
```

   If the preceding command returns `0`, then the OLAP CWMLITE registry is still invalid. Review your installation logs for other errors.

   If the preceding command returns 1, then the OLAP CWMLITE tablespace is valid, and no further testing needs to be done.