

Oracle9i

Release Notes

Release 2 (9.2.0.1.0) for HP 9000 Series HP-UX (64-bit)

May 2002

Part No. A97350-01

This document accompanies Oracle9i release 2 (9.2.0.1.0) for HP 9000 Series HP-UX (64-bit). Its contents supplement or supersede information in the installation guide for this release, or in the Oracle9i documentation library.

Topics:

- [System Requirements](#)
- [Documentation](#)
- [Installation Issues](#)
- [Product-Related Issues](#)
- [Post-Installation Issues](#)
- [Known Bugs](#)

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 3 GB.

Additional Software Patch Requirements

If you are installing Oracle Real Application Clusters, and if you are using the HMP protocol, then you must download the following patch from HP in addition to the patches listed in the installation guide.

ORACLE®

Copyright © 2002, Oracle Corporation
All rights reserved.

Oracle is a registered trademark, and Oracle7, Oracle8i, Oracle9i, Oracle*MetaLink*, Oracle Names, Oracle Transparent Gateway, PL/SQL, Pro*C, Pro*COBOL, Pro*FORTRAN and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

- PHNE_26551

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.

Oracle Universal Installer Version Update

Oracle*9i* release 2 (9.2.0.1.0), which is provided with the release, uses Oracle Universal Installer 2.2.0.12.0. This version number supersedes the version listed in the installation guide.

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:

- [32-bit O/S Support](#)
- [Multiple CD-ROM Installation](#)
- [runInstaller Script](#)
- [Installing Databases with Database Configuration Assistant](#)
- [Database Migration](#)

- [Installing with Response Files](#)

32-bit O/S Support

Do not run Oracle9i on a 32-bit operating system. Oracle9i release 2 (9.2.0.1.0) is offered in a 64-bit version only. If you try to run any 64-bit executables on a 32-bit operating system, then they will fail. If you run Oracle9i release 2 (9.2.0.1.0) on a 32-bit system, then you will see the following error message:

```
./oracle: Exec format error. Wrong Architecture.
```

Multiple CD-ROM Installation

During the installation of Oracle9i release 2 (9.2.0.1.0), you will be prompted to insert additional CD-ROMs from the set that make up Oracle9i release 2 (9.2.0.1.0). When prompted to mount the next CD-ROM, use the following procedure:

1. On the window where you have started the Installer, press Enter to go to the UNIX prompt, and then change the directory to your system's root directory. Log in as the `root` user by using the following commands:

```
$ cd /  
$ su root
```

2. Unmount and remove the CD-ROM from the CD-ROM drive with the following command:

```
# /usr/sbin/pfs_umount/SD_CDROM
```

3. Insert the required CD-ROM into the CD-ROM drive and mount it by using the following command:

```
# /usr/sbin/pfs_mount/SD_CDROM
```

4. Click OK to continue.

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 release 8.0.6 to release 9.2.0.1.0 and you have Oracle *interMedia* 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:

- [Character Sets](#)
- [Demo Schema](#)
- [Java Database Connectivity Driver](#)
- [Oracle Advanced Security](#)
- [Oracle Internet Directory \(OID\)](#)
- [Oracle Real Application Clusters](#)

Character Sets

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

Oracle9i NCHAR Datatypes

In Oracle9i 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 Oracle8i 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 Oracle8i.
2. Drop the SQL NCHAR columns.
3. Migrate the database to Oracle9i.
4. Import the SQL NCHAR columns in to Oracle9i.

AL24UTFFSS Character Set

Oracle9i 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.

Oracle9i 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 Oracle9i. 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.

Demo Schema

If you select a multibyte character set or UTF as the national character set in Oracle9i 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.

Java Database Connectivity Driver

The default behavior for the `ResultSet::getXXXStream()` has been modified to comply with the Java Database Connectivity (JDBC) specification so that the APIs return null values for database null `LONG/LONG RAW` values.

For Oracle8i release 8.1.x JDBC drivers, the default behavior was to return an empty stream for database null values. The Java property `jdbc.backward_compatible_to_8.1.7` allows the system to use the earlier JDBC default behavior when using the Oracle9i drivers and applies to Oracle9i JDBC Thin and OCI drivers.

For example, if the Java property is set at the virtual machine runtime, the following command will cause the Oracle9i JDBC drivers to return empty streams from calls to `ResultSet::getXXXStream()`:

```
java -Djdbc.backward_compatible_to_8.1.7 myJavaProgram
```

Oracle Advanced Security

If you install `jsse.jar` and `jcrt.jar` as extensions (located in `$JAVA_HOME/jre/lib/ext`), then you must also install `jssl-1_1.jar` in the same directory.

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 Oracle9i or Oracle8i

If you have installed in the same ORACLE_HOME either Oracle Internet Directory release 3.0.1.x and the complete release of Oracle9i (9.0.1) Enterprise Edition, or Oracle Internet Directory 2.1.1.x and the complete release of Oracle8i (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 Oracle9i Enterprise Edition Release 1 (9.0.1) or Oracle8i release 3 (8.1.7) to Oracle9i 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 following restrictions apply for this release:

- 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.

- If you are installing Oracle9i release 2 (9.2.0.1.0) Real Applications Clusters on a cluster that already contains an ORACLE_HOME for a previous release of Real Application Clusters, then you must run the Oracle Universal Installer from the cluster node with the oraInventory installation registry. Doing this ensures that product installation inventories are synchronized on the nodes with information about existing ORACLE_HOME directories.

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 Oracle9i 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/oracle_vg/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/oracle_vg/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;
```

Oracle Real Application Clusters and Upgrading from Oracle8i

If you are upgrading from Oracle8i, then set the `instance_number` initialization parameter in the `init.ora` file for all instances.

Oracle Real Application Clusters and Hyper Messaging Protocol (HMP)

Complete the following tasks prior to using HMP:

Ensure HF Software is Installed and Verified Ensure that you have the HyperFabric software installed, and verify that it exists. The HF software product ID number is B6257AA. Verify that it exists with the following command:

```
swlist | grep -i B6257AA
```

Configure HyperFabric Patch Use the following command to verify that the HyperFabric patch is configured properly before installing Oracle9i:

```
/opt/clic/bin/clic_stat
```

If the HyperFabric patch is configured properly, then the response to this command will list all nodes in the cluster. If cluster members are missing, then they will not appear in the response.

Note: The Hyper Messaging Protocol is not supported on V-class HP systems.

Tune HMP parameters Oracle Corporation recommends that you review parameters and tune them according to HP guidelines for optimal HMP performance. A list of HMP-tunable parameters can be found in the following file:

```
/opt/clic/lib/skgxp/skclic.conf
```

Relink Oracle With HMP Oracle9i release 2 (9.2.0.1.0) uses UDP as the default mode for IPC mechanism. To enable Oracle to use HMP and to relink the Oracle binary with HMP, complete the following steps:

1. Set up your environment on all nodes
2. Shut down the database
3. Enter the following commands on all nodes:

```
$ cd $ORACLE_HOME/rdbms/lib
$ make -F ins_rdbms.mk ipc_hms ioracle
```

Platform-Specific Product Information

The following product information in this section supersedes the information in the installation guide for Oracle9i release 2 (9.2.0.1.0) on HP-UX.

- Precompiler Options:
 - Pro*COBOL (32-bit and 64-bit) are supported.
 - Pro*FORTRAN (32-bit and 64-bit) are supported.
 - SQL Module for Ada is not supported.
- Oracle Advanced Security:
 - Radius challenge response authentication is not supported.
 - CyberSafe is not supported.
 - DCE Integration is not supported.
- JDBC/OCI Interfaces:
 - Oracle JDBC Thin Driver for JDK 1.4 is not supported.
 - Oracle JDBC/OCI Driver for JDK 1.4 is not supported.

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 HP-UX, the minimum data block size is 2048 bytes, and the maximum size of control files is 20000 database blocks.

Support for 32-bit Client Applications

Oracle9i (64-bit) provides support for both 32-bit clients as well as 64-bit clients. By default, all demos and clients provided with this release link and run in 64-bit mode. You can, however, build 32-bit and 64-bit clients in the same \$ORACLE_HOME.

The following combinations will run and link successfully in Oracle9i (64-bit):

- 32-bit applications against a 64-bit Oracle Server
- 64-bit applications against a 64-bit Oracle Server

The 64-bit client shared library is:

```
$ORACLE_HOME/lib/libclntsh.sl
```

The 32-bit client shared library is:

`$ORACLE_HOME/lib32/libclntsh.sl`

Building 32-bit Pro*C Customer Applications

Both 32-bit and 64-bit customer applications are supported in this release. Refer to the following files for further information:

`$ORACLE_HOME/precomp/demo/demo_proc.mk`
`$ORACLE_HOME/precomp/demo/demo_proc32.mk`

Building 32-bit OCI Customer Applications

Both 32-bit and 64-bit Oracle Call Interface (OCI) customer applications are supported in this release. Refer to the following file for further information:

`$ORACLE_HOME/rdbms/demo/demo_rdbms.mk`

32-bit Files and Directories

In Oracle9i (64-bit) for HP-UX (64-bit), the following directories contain 32-bit executables and libraries:

- `$ORACLE_HOME/lib32`
- `$ORACLE_HOME/rdbms/lib32`
- `$ORACLE_HOME/hs/lib32`
- `$ORACLE_HOME/network/lib32`
- `$ORACLE_HOME/precomp/lib32`

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.relfid#
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.relfid#
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.relfid#
/

```

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 Oracle9i release 2 (9.2.0.1.0):

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 CWMLITE 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.