This software was not developed for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It is the customer's responsibility to take all appropriate measures to ensure the safe use of such applications if the programs are used for such purposes.

This software/documentation contains proprietary information of Oracle Corporation; it is provided under a license agreement containing restrictions on use and disclosure and is also protected by copyright law. Reverse engineering of the software is prohibited.

If this software/documentation is delivered to a U.S. Government Agency of the Department of Defense, then it is delivered with Restricted Rights and the following legend is applicable:

**Restricted Rights Legend** Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of DFARS 252.227–7013, Rights in Technical Data and Computer Software (October 1988).

Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

If this software/documentation is delivered to a U.S. Government Agency not within the Department of Defense, then it is delivered with “Restricted Rights”, as defined in FAR 52.227–14, Rights in Data – General, including Alternate III (June 1987).

The information in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error–free.

Oracle, the Oracle logo, Pro*Ada, Pro*COBOL, Pro*FORTRAN, Pro*Pascal, and SQL*Net are registered trademarks of Oracle Corporation. Oracle7, Oracle Book, Oracle Multiprotocol Interchange, Oracle Names, Oracle Network Manager, Oracle Parallel Server, Oracle Server, Oracle Server Manager, PL/SQL, and Pro*C are trademarks of Oracle Corporation.

All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.
Preface

This Oracle7 Reference Addendum for Sun SPARC Solaris 2.x contains reference information that applies to all UNIX operating systems. It complements the Oracle7 Installation Guide for Sun SPARC Solaris 2.x.

The topics covered in this chapter are:

- Audience
- Document Conventions
- Customer Support and Documentation Sales
Audience

This document is intended for database administrators or others responsible for configuring and using Oracle tools on UNIX operating systems.

Additional Reading

If you are unfamiliar with the Oracle relational database management system, read the “Technical Introduction to the Oracle Server” chapter of Oracle7 Server Concepts. The chapter is a comprehensive introduction to the concepts and terminology used throughout Oracle documentation.

The Oracle Documentation Set

For an explanation of the Oracle documentation set and a complete listing of the Oracle7 Server documentation set, see Getting Started with Oracle7 for UNIX.

Document Conventions

Conventions used in this guide differ somewhat from those used in other Oracle documentation. Because UNIX is case-sensitive, commands and filenames are shown in boldface type, rather than uppercase letters.

Type Conventions

Following are the type conventions:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bold</strong></td>
<td>Boldface type indicates UNIX commands, directory names, pathnames, and filenames (for example, the <em>prefs.ora</em> file).</td>
</tr>
<tr>
<td>brackets [ ]</td>
<td>Words enclosed in brackets indicate key names (for example, Press [Return]).</td>
</tr>
<tr>
<td><em>italics</em></td>
<td>Italic type indicates a variable and is used for emphasis. It also indicates variable portions of filenames (for example, <em>sgadefx.dbf</em>).</td>
</tr>
<tr>
<td>UPPERCASE</td>
<td>Uppercase letters indicate Oracle commands and environment variables (for example, <em>ORACLE_HOME</em>).</td>
</tr>
</tbody>
</table>
Command Syntax

Commands appear in monospace type. Enter information precisely as it appears. Following are the syntax conventions for commands:

backslash \ A backslash indicates a line is too long to fit on the printed page. Either enter the line as printed (with a backslash) or enter it as a single line without a backslash.

\dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 \nbs=10b count=10000

braces { } Braces indicate required items.

.DEFINE { macro1 }

brackets [ ] Brackets indicate optional items.

.cvtcrt termname [outfile]

ellipsis ... An ellipsis indicates an arbitrary number of similar items.

.CHKV AL fieldname value1 value2 ... valueN

italics Italics type indicates a variable. Substitute a value for the variable.

library_name

vertical line | A vertical line indicates a choice within braces or brackets.

.SIZE filesize [K | M]

Icons

The following icons appear in the printed documentation. Icons do not appear in the online documentation.

Attention: The attention icon indicates important additional information.

OPS For Parallel Server: The OPS icon indicates information specific to the Oracle Parallel Server.

See Also: The book icon indicates a reference to another document, published by Oracle Corporation or another organization. The words “See Also” without an accompanying icon indicate a reference to another section of this document.

Suggestion: The suggestion icon indicates recommendations or hints.
Warning: The warning icon indicates an action that could damage your system.

Other Conventions

The term “Oracle7 Server” refers to the database server product from Oracle Corporation.

The term “oracle” refers to an executable or account by that name.

The term “oracle” refers to the owner of the Oracle software.

Unless otherwise stated, examples use the Bourne shell (sh(1)) syntax.

Customer Support and Documentation Sales

You can reach Oracle Worldwide Customer Support 24 hours a day.

In the USA: 1.415.506.1500

In Europe: +44 1344 860160

Supplying Technical Information

Please be prepared to supply the following information:

- your CSI number (This helps Oracle Corporation track problems for each customer.)
- the release numbers of the Oracle7 Server and associated products
- the operating system name and version number
- details of error numbers and descriptions (Write down the exact errors.)
- a description of the problem
- a description of changes made to the system

For installation-related problems please supply:

- a printout of the $ORACLE_HOME and $STAGE_HOME directories
- the directory path names of your installation staging area
- the $ORACLE_HOME directory
This information helps Oracle WorldWide Customer Support validate the information written to the installation log files.

**Documentation Sales and Client Relations**

In the United States:
- To order hardcopy documentation, call Documentation Sales: 1.800.252.0303
- For shipping inquiries, product exchanges, or returns, call Client Relations: 1.415.506.1500

In the United Kingdom:
- To order hardcopy documentation, call Oracle Direct Response: +44 990 332200
- For shipping inquiries and upgrade requests, call Customer Relations: +44 990 622300

Other European customers, please contact your local Oracle Support office for documentation or shipping inquiries.

**Your Comments are Welcome**

Please use the Reader’s Comment Form at the back of this document to convey your comments to us. You can also reach us at:

Technical Publications Manager  
Open Systems Operations and Tools  
Oracle Corporation  
500 Oracle Parkway  
Box 659103  
Redwood Shores, California 94065  
Phone: 1.415.506.7000  FAX: 1.415.506.7360  
E-mail: osdwrite@us.oracle.com

or

Technical Publications Manager  
Oracle Corporation  
European Porting Centre  
Maretimo Court  
Temple Road  
Blackrock  
Co Dublin  
Ireland
Contents

Chapter 1  Oracle7 Server ................................................................. 1 – 1
            Special Tools ................................................................. 1 – 2
            Memory Utilization ......................................................... 1 – 3
            Server Resource Limits .................................................. 1 – 4
            System Global Area ......................................................... 1 – 4
            I/O Types ........................................................................ 1 – 6
            Monitoring Performance .................................................... 1 – 7
            Tuning ............................................................................. 1 – 9
            Reducing Contention for Resources with Multiple Instances . 1 – 13
            Raw Devices ...................................................................... 1 – 17
            Support for Logical Volumes .............................................. 1 – 25
            Initialization Parameters .................................................... 1 – 25

Chapter 2  Oracle7 Interfaces ............................................................. 2 – 1
            Pro*Ada ........................................................................... 2 – 2
            Pro*C .............................................................................. 2 – 2
            Pro*COBOL ...................................................................... 2 – 3
            Pro*FORTRAN .................................................................. 2 – 4
            Pro*Pascal ........................................................................ 2 – 4
            SQL*Module for Ada .......................................................... 2 – 5
            SQL*Module for C ............................................................... 2 – 5
List of Tables

Table 1 – 1  Database Block Sizes and Corresponding MAXEXTENTS Values ..................................... 1 – 10
Table 1 – 2  Default Initialization Parameters ......................... 1 – 27
Table 4 – 1  Cluster Size Values .............................................. 4 – 2
Table 4 – 2  Database Size Values .............................................. 4 – 2
Table 4 – 3  Index Size Values .................................................. 4 – 3
Table 4 – 4  Block Size and Number of Extents ...................... 4 – 6
Table 4 – 5  CREATE CONTROLFILE Parameter Values ....... 4 – 7
Oracle7 Server

This chapter describes Solaris 2.x-specific aspects of Oracle7 Server system administration and tuning. It provides supplementary Solaris 2.x-specific information for the *Oracle7 Administrator’s Reference for UNIX*.

The topics covered in this chapter are:

- Special Tools
- Memory Utilization
- Server Resource Limits
- System Global Area
- I/O Types
- Monitoring Performance
- Tuning
- Reducing Contention for Resources with Multiple Instances
- Raw Devices
- Support for Logical Volumes
- Initialization Parameters
Special Tools

The following special tools are available with this release of the Oracle7 Server:

- Intimate Shared Memory
- Post-Wait Driver

Intimate Shared Memory

Intimate Shared Memory (ISM) is set to on by default. To turn ISM off, add the following line to your initSID.ora file:

```
use_ism = false
```

Post-Wait Driver

The Oracle Post-Wait driver implements an optimized mechanism of inter-process communication, without the overhead of signal handlers or semaphores. It improves performance for Oracle7 and can be used with Sun Solaris 2.4.

Suggestion: This optimization is only recommended for installations that use Oracle Parallel Server. You are not required to install the Post-Wait driver, even if you are installing Oracle Parallel Server. Oracle Corporation recommends that you consult Oracle Worldwide Customer Support Services before you install the Post-Wait driver.

To install the Post-Wait Driver Kernel Extension, follow the instructions in the $ORACLE_HOME/rdbms/lib/pw/README.PW file.

Attention: When you upgrade an Oracle system with the Post-Wait driver installed, make sure you upgrade the Post-Wait driver along with the system.
Memory Utilization

This section discusses Oracle7 Server memory usage.

Estimating Oracle7 Server Memory Usage

Before you start the Oracle7 Server, you must estimate virtual memory requirements. Use the following formula to measure virtual memory requirements:

Virtual memory needed =
size of the oracle executable text
+ size of the SGA
+ n \cdot (size of tool executables private data section
  + size of oracle executables uninitialized data section
  + 8192 bytes for the stack
  + 2048 bytes for the processes u area)

where \( n \) = number of background processes.

For each Oracle backend connection, use the following formula:

Virtual memory needed =
size of oracle executable data section
+ size of oracle executables uninitialized data section
+ 8192 bytes for the stack
+ 2048 bytes for processes u area
+ cursor area needed for the application

To get an executable’s text size, private data section size and uninitialized data section size (or bss), use the size command. Program text is only counted once, no matter how many times the program is invoked, because all Oracle executable text is always shared.

To compute actual Oracle physical memory usage while the database is up and users are connecting to it, use the ps command. Look for all the front end, server, and background Oracle process entries. For each entry, add the “real size of process” columns for the resident memory use subtotal. Now add the text size for the Oracle executable and every other Oracle tool executable running on the system to that subtotal. Remember to count executable sizes only once, regardless of how many times the executable was invoked.

See Also: Your operating system documentation for a list of available switches for the ps command.
Server Resource Limits

Solaris 2.x inherits resource limits from the parent process (see getrlimit(2) in your operating system documentation). These limits apply to the Oracle7 Server shadow process that executes for user processes. The Solaris 2.x default resource limits are high enough for any Oracle7 Server shadow or background process. However, if these limits are lowered, the Oracle7 Server system could be affected. Discuss this with your Solaris 2.x system manager.

Disk quotas established for the Oracle dba user ID may hinder the operation of the Oracle7 Server system. Confer with your Oracle7 database administrator and the Solaris 2.x system manager before establishing disk quotas.

System Global Area

The System Global Area (SGA) is the Oracle structure that resides in shared memory. It contains static data structures, locks, and data buffers. Sufficient shared memory must be available to each oracle process to address the entire SGA.

The SGA topics are:

- Size Limits of the SGA
- Relocating the SGA

Size Limits of the SGA

The Oracle7 Server uses shared memory segments for the SGA.

The maximum size of a single shared memory region is specified by the Solaris 2.x parameter SHMMAX. An SGA that is 2048 KB can use four shared memory regions of 512 KB each.

See Also: The Solaris Answerbook.

If the size of the SGA exceeds the maximum size of a shared memory segment (SHMMAX), Oracle7 attempts to attach more contiguous segments to fulfill the requested SGA size. SHMSEG is the maximum number of segments that can be attached by a process. To attach the segments at contiguous addresses, SHMMAX must be set to its maximum value on systems where its size is limited.
Intimate Shared Memory (ISM) can cause some system problems when SHMMAX is smaller than the database SGA size.

Relocating the SGA

The address at which the SGA is attached affects the amount of virtual address space available for such things as database buffers in the SGA and cursors in the user’s application data area. After consulting Oracle Worldwide Support, perform the following steps to relocate the SGA:

1. Determine the valid virtual address range for attaching shared memory segments:
   
   $ tstshm

   **Attention:** If you are using Intimate Shared Memory (ISM), your system may experience problems when you run **tstshm**.

   In the resulting **tstshm** display, the lines “Lowest shared memory address” and “Highest shared memory address” indicate the valid range.

2. Check the “Segment boundaries” output of **tstshm** to determine the valid virtual address boundaries at which a shared memory segment can be attached. For example, some systems require that shared memory segments be on 512-byte boundaries.

3. Determine the size of your SGA.

   SGA size is displayed next to the heading “Total System Global Area” when your database system starts.

4. Move to the **$ORACLE_HOME/rdbms/lib** directory, and run **genksms** to generate the file **ksms.s**:

   $ cd $ORACLE_HOME/rdbms/lib
   $ genksms > ksms.s

5. Change the value of the symbol **sgabeg** by editing the file **ksms.s**.

   Following is the **ksms.s** file:

   ```
   sgabeg = 0xe0000000
   .global _ksmsgf_
   _ksmsgf_ = sgabeg+0
   .global _ksmvsg_
   _ksmvsg_ = sgabeg+4
   .
   .
   ```
The `sgabeg` symbol specifies the address where the SGA is attached. Change the following line:

```
sgabeg = 0xe0000000
```

to reflect the new address at which you want to attach the SGA. Use the following format:

```
sgabeg = address
```

**Note:** If you have chosen to hard-lock the SGA, make sure that the `sgabeg` is a multiple of 0x400000.

The `ksms.s` file contains a warning message about editing. You can ignore this message when you are relocating the SGA.

6. Shut down the existing Oracle database.
7. Rebuild the `oracle` executable in the `$ORACLE_HOME/rdbms/lib` directory:

```
$ make -f oracle.mk ioracle
```

Using `ioracle`:

- backs up the old executable (`oracle0`)
- assigns the correct privileges to the new `oracle` executable
- moves the new executable into the `$ORACLE_HOME/bin` directory

The result is a new Oracle kernel that loads the SGA at the address specified by `sgabeg`.

---

**I/O Types**

This section describes the following:

- Asynchronous I/O and Multiple Database Writers
- List I/O

**Asynchronous I/O and Multiple Database Writers**

Asynchronous I/O (AIO) is a standard Solaris 2.x feature which allows one Oracle7 Server database writer to write to multiple disks simultaneously, which increases disk throughput. With Solaris 2.x AIO, the DBWR process sends multiple writes to multiple drives without waiting for the previous writes to finish. It can use all disks.
simultaneously. With ten disks, for example, ten asynchronous writes can take place in the same amount of time as a single synchronous write.

Since the Oracle7 Server supports Solaris 2.x AIO, multiple database writers are usually unnecessary. Solaris 2.x AIO works on raw devices as well as UFS file systems.

Using Multiple Database Writers

To use multiple database writers, set the DB_WRITERS parameter in the \texttt{init\_sid.ora} file to the number of database writer processes desired and ensure the ASYNC\_WRITE parameter is either unset or set to FALSE.

List I/O

This is not a feature of Solaris 2.x.

Monitoring Performance

Before tuning the system, observe its normal behavior with some of the Solaris 2.x tools described in this section, as well as with the Server Manager MONITOR utility.

\textbf{For Parallel Server:} For the Oracle Parallel Server, dynamic performance tuning views are available to monitor and tune the system. These are described in Appendix C of the \textit{Oracle7 Parallel Server Concepts and Administration}.

This section describes the following:

\begin{itemize}
  \item Using Solaris 2.x Tools
  \item Monitoring Disk Performance
  \item Disk Performance Issues
\end{itemize}
Using Solaris 2.x Tools

The following operating system tools are available to monitor the performance of the Oracle7 Server under Solaris 2.x:

- cpustat
- dspst
- iostat
- osview
- perfmeter
- ps
- pstat
- sar
- syspic
- vmstat

These tools provide statistics for CPU usage, interrupts, swapping, paging, and context switching for the entire system, not just the oracle processes.

**See Also:** Your operating system documentation for more information about these commands.

Monitoring Disk Performance

To monitor disk performance, use sar –b and sar –u. Important sar –b columns for disk performance are:

- **bread/s, bwrit/s**: blocks read and blocks written (important for file system databases)
- **pread/s, pwrit/s**: partition reads and partition writes (important for raw partition database systems)

An important sar –u column for disk performance is:

- **%wio**: percentage of CPU time waiting on blocked I/O
Key indicators are:

- The sum of **bread**, **bwrit**, **pread** and **pwrite** indicates the state of the disk I/O subsystem. The higher the sum, the greater the potential for disk I/O bottlenecks. The larger the number of physical drives, the higher the sum threshold number can be. A good default value is no more than 40 for two drives and no more than 60 for four to eight drives.
- The **%rcache** should be greater than 90 and **%wcache** should be greater than 60. Otherwise, the system may be disk I/O bound.
- If **%wio** is consistently greater than 20, the system is I/O bound.

Disk Performance Issues

Oracle block sizes should either match disk block sizes, or be a multiple of disk block sizes.

If possible, do a file system check on the partition before using it for database files, then make a new file system to ensure that it is clean and unfragmented. Distribute disk I/O as evenly as possible and separate log files from database files.

Tuning

This section describes tuning the Oracle7 Server on Solaris 2.x. Topics discussed are:

- Block Size and File Size
- Associating Data Blocks with Instances
- Table Striping
- Reducing Database Fragmentation

Block Size and File Size

Although data storage space is often measured in megabytes (one MB = 1024 * 1024 bytes), the Solaris 2.x operating system and Oracle7 Server each perform input and output in units of data storage called **blocks**. The size of the operating system blocks is not necessarily equal to Oracle blocks.
Oracle7 Server block size is not fixed. You can set its value when creating a database by changing the DB_BLOCK_SIZE parameter in the initSID.ora file.

Changing the Oracle block size can change database performance, depending on the disk hardware, file system, and application. Experiment with different block sizes to determine the most efficient configuration. The default size of 2 KB should be adequate for most circumstances, but some performance benefits can be gained by modifying it.

Specifying Oracle Block Size

On Solaris 2.x, the default Oracle block size is 2 KB and the maximum block size is 16 KB.

You can set the actual block size to any multiple of 2 KB up to 16 KB, inclusive.

The optimal block size is typically the default, but varies with the applications. To create a database with a different Oracle block size, add the following line to the initSID.ora file:

db_block_size=new_block_size

To change block size, you must create a new database. Experiment with block size before transferring your data to the new database.

Maximum Number of Extents

The first block of each segment contains an extent map for each extent. The maximum number of extents allowed is therefore a function of the database block size and the size of each extent map entry.

To increase the limit, change the database block size. (This requires a full export/import). Table 1 – 1 shows the database block size relative to the greatest MAXEXTENTS value.

<table>
<thead>
<tr>
<th>Database Block Size</th>
<th>Greatest MAXEXTENTS Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 bytes</td>
<td>25</td>
</tr>
<tr>
<td>1 KB</td>
<td>57</td>
</tr>
<tr>
<td>2 KB</td>
<td>121</td>
</tr>
<tr>
<td>4 KB</td>
<td>249</td>
</tr>
<tr>
<td>8 KB</td>
<td>505</td>
</tr>
</tbody>
</table>

Table 1 – 1 Database Block Sizes and Corresponding MAXEXTENTS Values
Associating Data Blocks with Instances

If you are running the Oracle Parallel Server and have multiple Oracle instances, you can use free space lists. This allows transactions running on separate instances to insert and update data in the same table concurrently.

When an insert or update must locate free space in a table, the Oracle7 Server searches one of the free lists in the set associated with the instance running that transaction. If the free list does not contain a block with sufficient space, Oracle7 searches the master free list, not one of the other free lists for that instance.

If the master free list does not contain sufficient space, the Oracle7 Server allocates a new extent, if possible, and adds its space to the master free list. It is important to pre-allocate extents to tables; otherwise, the free space in an extent allocated by the Oracle7 Server is available to all instances and you lose the performance advantages of partitioning data between instances.

To prevent the Oracle7 Server from allocating new extents for a table, set MAXEXTENTS to the number of pre-allocated extents plus MINEXTENTS.

If the Oracle7 Server cannot find sufficient space on the master free list and cannot allocate a new extent, it returns error message ORA-01547:

failed to allocate extent of size num in tablespace ‘name’

This release of the Oracle7 Server allows you to explicitly allocate new space to a table and specify a database file from which to take the new space. You can also use the following new options of the SQL statements CREATE TABLE and ALTER TABLE to associate free space with particular instances:

- CREATE TABLE has two storage options, FREELIST GROUPS, that specifies the number of free list sets for the table, and another storage option, FREELISTS, that specifies the number of free lists per set.

- ALTER TABLE has the option ALLOCATE EXTENT, which allocates and associates space with a particular set of free lists.

In a multi-instance server configuration, MAXINSTANCES for the database could be many times larger than FREELIST GROUPS for a table, so that many instances share one set of free lists.

See Also: The Oracle7 Server Administrator’s Guide for more information about CREATE TABLE and ALTER TABLE.
Table Striping

Table striping is the process of dividing the data for a large table into small portions and storing these portions in separate data files on separate disks. This permits multiple processes to access different portions of the table concurrently without disk contention. Striping is particularly helpful in optimizing random access to tables with many rows.

See Also: The Oracle7 Server SQL Reference for information on the ALLOCATE EXTENT DATAFILE parameters of the ALTER TABLE command.

Reducing Database Fragmentation

This section supplements the instructions for reducing database fragmentation in the Oracle7 Server Utilities, which contains more information about the Export and Import utilities.

To reduce database fragmentation:

1. Use the MONITOR command in Server Manager to check for active users.

2. Make sure all users have completed their transactions and do a full database export (FULL=Y) to back up the entire database.

3. Shut down each Oracle7 instance when all users are logged off.

4. When all instances are shut down, back up the database, redo log, and control files.

5. Recreate the database using the Server Manager command CREATE DATABASE.

6. Connect to oracle as SYS/password and run catalog.sql from the $/rdbms/admin directory.

   This file is run automatically when you use the Oracle Installer to create an Oracle7 instance and database. When you use Server Manager to create an Oracle7 system, you must run this file yourself.

7. Connect to oracle as system and run the catdbsyn.sql file from the $/rdbms/admin directory.

8. Create a rollback segment in the SYSTEM tablespace, as described in the Oracle7 Server Administrator’s Guide.

9. If you create private rollback segments using the SQL statement CREATE PRIVATE ROLLBACK SEGMENT, you must shut down
the database and then edit the initSID.ora parameter file. Use the initialization parameter ROLLBACK_SEGMENTS to identify the new segments. For example:

rollback_segments=(rb1 [,xxx,yyy,...])

where additional entries are delimited by commas. If you start up the database with this parameter file, the private rollback segments are enabled.

10. Start Oracle7 and run Import (FULL=Y) to import the exported database.

---

Reducing Contention for Resources with Multiple Instances

Running the Oracle7 Parallel Server on a multi-instance system can cause poor performance due to:

- Contention for CPU
- Contention for locks

Contention for CPU

If you started Oracle7 in SHARED mode, and have a number of instances running, ensure that the workload is balanced across the processors. For example, avoid running all the jobs on a single instance, or a few CPU-intensive jobs on one instance and some less intensive jobs on the other instances.

An overloaded instance can:

- Run out of processing power
- Have insufficient physical memory to hold all the data pages

When you connect to Oracle7 using SQL*Net TCP/IP, you can specify an ORACLE_SID or class name for each instance. If you specify a class name, an entry for that class name must exist in the /var/opt/oracle/oratab file. You can then connect to Oracle7 in either of the following ways:

- If you connect to an instance using SQL*Net TCP/IP with an ORACLE_SID, you connect directly to the specified instance; if that instance is overloaded, performance may be degraded. This option allows you to explicitly balance CPU load between instances. Divide CPU-intensive jobs among multiple instances.
• If you connect to a class name, you can connect to a group of instances. This option provides simplified administration of a larger number of equal-weight jobs.

For example:

$ svrmgrl
SVRMGR> CONNECT SYSTEM/MANAGER@T:node:SID1

This connects you to the instance associated with the sid SID1.

For example, if you have a class called GEN, and two instances on different processors with sids of SID1 and SID2 with class names set to GEN, you can connect to GEN using SQL*Net TCP/IP:

$ svrmgrl
SVRMGR> CONNECT SYSTEM/MANAGER@T:node:GEN

Connection requests are directed to both instances in turn.

Contenion for Locks

Oracle7 in SHARED mode allows multiple access to shared resources; to ensure data integrity, every database block is protected by a global lock. The lock/unlock operations are handled by the Distributed Lock Manager (DLM). The locking mechanisms impose some performance overhead, which you can minimize with careful tuning.

If a process in an instance attempts to change a database block, the instance must own the lock on this database block. If the lock is owned by another instance, the lock manager notifies the owner. The owner then returns the block to disk and releases the lock for the requesting process. The next process acquires the lock and reads a new copy of the block from the disk. This mechanism is called block-pinging. Whenever possible, avoid block-pinging to keep I/O costs low.

The following topics are described in this section:

• Checking for lock contention
• Retaining locks until they are required
• Assigning locks to files
• Clustering database blocks
• Tips for reducing contention for locks
Checking for Lock Contention

To find out which blocks are most subject to block-pinging, you can run `/rdbms/admin/catparr.sql` as SYS/password, and query the V$PING view. The affected blocks are those used concurrently by multiple instances.

This problem can be caused by contention for database blocks, or only for a lock on several blocks. The following sections discuss Oracle7 lock management techniques to improve this problem.

Retaining Locks Until They Are Required

Oracle7 includes a mechanism that allows one global lock to protect more than one database block. This means you can access many database blocks, without having to do a lock operation for each database block access.

If an Oracle7 instance has a lock on a database block (or group of blocks), it keeps the lock until another instance requests it. So if the database block leaves the instance cache, the lock is not released immediately. The instance can then re-access the same block without using the lock manager, since it still owns the lock.

Assigning Locks to Files

When Oracle7 must lock a block, it hashes the database block address to one of its resource names. The initialization parameter GC_DB_LOCKS allows you to specify the total number of different resource names to use in the system.

The initialization parameter GC_FILES_TO_LOCKS specifies how these resource names are spread over the database files.

If you have one datafile and one index file, and the majority of your transactions do not need to alter the index, you may want a high granularity of blocks-to-resource names for the datafile but not for the index file. For example:

```
GC_DB_LOCKS=1050 # Total no of resources
GC_FILES_TO_LOCKS="1=1000:2=50" # 1000 for datafile, 50 for index file
```

Here, for the datafile file#1, the first block is assigned to resource#1, the second block to resource#2, and so on, according to this formula:

```
resource_number = block_number MOD file_resources
```
Clustering Database Blocks

You can also cluster blocks so that a resource name covers several contiguous blocks. For example, if the datafile has 5,000 blocks, you may want to use a clustering factor of 5 so that a resource name covers only a single area of the file (instead of resource#1 covering blocks 1, 1001, 2001, etc.). For example:

```
GC_DB_LOCKS=1050 # Total no of resources
GC_FILES_TO_LOCKS="1=1000!5:2=50" # 1000 for datafile, # cluster factor of 5
```

The resource name used to protect a block is calculated as described above. Once the resource name is determined, Oracle7 uses a lock element to keep information about the status of the lock. Each resource name has its own lock element.

To find the correspondence between file numbers and filenames, query the V$DBFILE table.

Tips for Reducing Contention for Locks

- Organize the system so all processes working on the same dataset run on the same Oracle7 instance.
- Organize the data so large tables are stored in a separate tablespace. This tablespace can be stored on disk using a single large file. Also, you can group read-only tables together in a single tablespace in a single file, possibly spanning several disks. Indexes that do not change can also be stored in a large file.

  This allows you to protect files with read-only data with a single global lock, leaving lock space available for the changed blocks.

- Avoid having multiple processes working on different datasets that are stored in the same database blocks. To do this, you may need to recreate the tables and organize the data files differently using pctfree and pctused.

- Avoid contention on a table for concurrent multi-instance inserts, by creating the table with the storage parameters FREELIST GROUPS and/or FREELISTS set to a higher value. You may also need to allocate a separate extent for each instance to make the best use of free space lists. You can assign these extents to files if you have separate files in the tablespace.
Raw Devices

This section provides Solaris 2.x-specific information to supplement the discussion of raw devices in the “Raw Devices” section of Chapter 1, “Planning a Complex Database on UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

The following topics are described in this section:

• Disadvantages of Raw Devices
• Criteria for Using Raw Devices
• Guidelines for Using Raw Devices
• Setting Up Raw Devices
• Using Raw Devices
• Tuning the Solaris 2.x Buffer Cache Size
• Backing Up Raw Devices

Disadvantages of Raw Devices

Raw devices have the following disadvantages when used on Solaris 2.x:

• Using raw devices does not solve problems with ULIMIT that can arise when exporting tables larger than a megabyte. These tables would also have to be exported to raw devices (such as another disk partition).

• When raw devices and operating system files are mixed within an Oracle7 database, the operating system files must still be within the value of the LIMIT parameter.

• Using raw devices does not solve problems with LIMIT that can arise when reading the contents of the Oracle distribution media onto the disk.

• Clients with small systems usually cannot use sufficiently large raw device partitions. Disk partitions usually come in odd sizes that do not lend themselves to good database architecture.

• If a particular disk drive has intense I/O activity and performance would benefit from movement of an Oracle data
file to another drive, it is likely that no acceptably sized section exists on a drive with less I/O activity. Moving data files around, a common advantage of UNIX, may not be possible with raw devices.

• Adding space to a tablespace can be a difficult process in a raw device environment. Occasionally, all raw partitions are assigned data files at initial configuration time, leaving no raw storage to accommodate normal tablespace growth.

Criteria for Using Raw Devices

Consider the following criteria when deciding whether to use raw devices:

• Oracle7 Parallel Server Installation

—(same) ➔

• Raw Disk Partition Availability

((port 88op aix att hiosf ibmsp1 nec pcx sgi sol2 sun4 symp)) ➔

Oracle7 Parallel Server Installation

For Parallel Server: Each instance of OPS has individual log files. Therefore, in addition to the partitions required for the tablespaces and control files, each instance requires a minimum of three partitions for the log files. All these files must be on disks that can be shared by all nodes of a Solaris 2.x cluster. UNIX clusters do not provide access to a shared file system between all nodes of a cluster. As a result, all files associated with a database must be built on raw devices. Some hardware vendors providing UNIX clusters also provide logical volumes built on top of raw disks.

Raw Disk Partition Availability

Use raw devices for Oracle files if your site has at least as many raw disk partitions as Oracle tablespaces.

If the raw disk partitions are already formatted, match tablespace size to partition size as closely as possible to avoid wasting space.
Guidelines for Using Raw Devices

When creating raw disk partitions, observe the following guidelines:

- Three partitions for the log files of each instance
- One partition each for the following data files: SYSTEM, ROLLBACK, TEMP, USERS, TOOLS
- At least three partitions for data files

Configuration Planning

With logical volumes, you can create logical disks based on raw partition availability, since logical disks can be on more than one disk drive. The disk drives do not have to be re-formatted to obtain appropriate logical disk sizes.

Dynamic Performance Tuning

Disk performance can be optimized when the database is online by moving hot spots to cooler drives. Most hardware vendors who provide the logical disk facility also provide a graphical user interface that can be used for tuning.

Mirroring and Online Disk Replacement

Mirroring of logical volumes is possible and should be used to protect against loss of data. If one copy of a mirror fails, dynamic re-synchronization is possible. Some vendors also provide the ability to replace drives online in conjunction with the mirroring facility.

For Parallel Server: Logical volumes are available for drives associated with a single UNIX machine, as well as those that can be shared with more than one machine of a UNIX cluster. The latter allows for all files associated with the Oracle Parallel Server to be placed on these shared logical volumes.
Setting Up Raw Devices

⚠️ **Warning:** Do not attempt to set up raw devices without the help of an experienced system administrator and specific knowledge about the machine you are using.

To set up raw devices on your system:

1. Make sure the partitions you are adding are on a shared disk.
2. Determine the names of the free disk partitions.
   
   A free partition is one that is not used for a Solaris 2.x file system. That means that the partition follows these restrictions:
   
   - It is not listed when you execute the `/etc/mount` command.
   - It is not in use as a swap device.
   - It does not overlap a swap partition.
   - It is not in use by other Solaris 2.x applications (for example, other instances of Oracle).
   - It does not overlap a Solaris 2.x file system.
   - It does not use a space already used by UFS

   To find out whether a partition is free, obtain a complete map of the starting locations and sizes of the partitions on the device and check for free space. Note that some partitions may contain file systems that are currently not mounted and are not listed in the `/etc/mount` output.

⚠️ **Attention:** Make sure that the partition does not start at Cylinder 0.

3. Set up the raw device for use by the Oracle7 Server.

   Begin by verifying that the disk is partitioned. If not, use the operating system **Format** utility to partition it.

   Next, make sure that the partition is owned by the *oracle* software owner. If necessary, use **chown** to change its ownership on the block and character files for the device. For example:

   ```bash
   $ chown oracle /devices/iommu@f,e0000000/
      sbus@f,e0001000/espdma@f,400000/esp@f,800000/
      sd@5,0:a
   $ chown oracle /devices/iommu@f,e0000000/
      sbus@f,e0001000/espdma@f,400000/esp@f,800000/
      sd@5,0:a,raw
   ```
Use `chmod` to make the partition accessible only by the `oracle` software owner. For example:

```
$ chmod 600 /devices/iommu@f,e0000000/
sbus@f,e0001000/espdma@f,400000/esp@f,800000/
.sd@5,0:a
```

```
$ chmod 600 /devices/iommu@f,e0000000/
sbus@f,e0001000/espdma@f,400000/esp@f,800000/
.sd@5,0:a,raw
```

4. Create a symbolic link to the raw devices you require. For example:

```
$ ln -s /devices/10mmu@f,e0000000/sbus@f,e0001000/
espdra@/f,400000/esp@f,800000/sd@5,0:a,raw
/oracle_data/datafile.dbf
```

**Attention:** This symbolic link must be set up on each node of the Parallel Server. Check to be sure that no two symbolic links point to the same raw device.

5. Create or add the new partition to a new database.

From Server Manager, use the SQL statement `CREATE DATABASE` to create the database using the specified raw partition.

**Note:** The size of an Oracle database created in a raw partition must be at least two Oracle block sizes smaller than the size of the raw partition.

For example:

```
$ svrmgrl
SVRMGR> create database sid
SVRMGR> logfile '/oracle_data/log1.dbf' size 100K,
        'oracle_data/log2.dbf' size 100K
SVRMGR> datafile '/oracle_data/datafile.dbf' size 10000K
        reuse;
```

If you want instead to add the partition to a tablespace in an existing Oracle database, enter:

```
$ svrmgrl
SVRMGR> alter tablespace `tablespace_name` add datafile
        '/dev/rdsk/c0t1d0s6' size 10000K reuse;
```

You can use the same procedure to set up a raw device for the redo log files.
Using Raw Devices

When using raw devices, consider optimizing head movement and controller activity. Try to balance the load across separate drives and controllers. In a typical environment, you must consider four kinds of I/O:

- Solaris 2.x file system
- Solaris 2.x swapping and paging
- Oracle tablespace
- Oracle log file

To balance I/O:

1. Separate the redo logs and the database file(s) onto different drives.
   In environments with a high transaction rate, a lot of I/O activity is directed to the redo logs. If possible, put the redo logs alone on a disk drive.

2. Spread heavily used database files over different drives.
   If you create several tablespaces composed of files that reside on different drives, you prevent hot spots from occurring.
   You can also create a single tablespace composed of several files residing on different disks.

3. Separate tables and their indexes.
   If a SQL statement requires access to a table and its index, you can achieve greater throughput if you put the table and its index on separate drives.

4. Create more than one rollback segment and distribute the segments evenly across the drives.
   A high transaction rate environment requires heavier use of rollback segments. If you create additional segments across drives, you not only balance potentially heavy I/O, but avoid contention for the rollback segments themselves.
   Cache the rollback segments to avoid physical I/O.

5. Separate Oracle7 and Solaris 2.x I/O.
   If possible, put Solaris 2.x on a drive separate from Oracle I/O.
   You can also put raw partition files on the same drive as Solaris 2.x file system files.
Tuning the Solaris 2.x Buffer Cache Size

To take full advantage of raw devices, adjust the size of the Oracle7 buffer cache and, if memory is limited, the Solaris 2.x buffer cache.

The Solaris 2.x buffer cache is provided by the operating system. It holds blocks of data in memory while they are being transferred from memory to disk, or vice versa.

The Oracle7 buffer cache, on the other hand, is the area in memory that stores the database buffers for Oracle7. Since Oracle7 can use raw devices, it does not need to use the Solaris 2.x buffer cache.

When moving to raw devices, increase the size of the Oracle7 buffer cache. If the amount of memory on the system is limited, make a corresponding decrease in the Solaris 2.x buffer cache size.

Determining the Cache Hit Ratio

The cache hit ratio indicates the percentage of time a block is found in the cache and disk I/O is unnecessary. This ratio is determined by the formula:

\[
\frac{\text{logical\_reads} - \text{physical\_reads}}{\text{logical\_reads}}
\]

The number of logical reads is the sum of CONSISTENT_GETS plus DB_BLOCK_GETS.

The Oracle7 cache hit ratio is determined from the Server Manager monitor and statistics displays. It can also be determined from the user display for individual processes.

The Solaris 2.x command sar may help you determine which buffer caches should be increased or decreased:

- `sar -b` reports the Solaris 2.x buffer cache activity
- `sar -w` reports the Solaris 2.x swapping activity
- `sar -u` reports CPU utilization
- `sar -r` reports memory utilization
- `sar -p` reports the Solaris 2.x paging activity

Adjusting Cache Size

To adjust cache size:

- Increase Oracle7 cache size as long as the cache hit ratio goes up.
- Decrease cache sizes if the swapping/paging activity becomes high.
Backing Up Raw Devices

You can back up raw devices in one of two ways:

- Copy the raw device to a Solaris 2.x file and allow the normal backup procedures to copy it.
- Copy the raw disk device directly to tape.

Copying to a Solaris 2.x File

If you copy the raw device to a Solaris 2.x file and allow the normal backup procedures to copy it, you must still use `dd` to copy from the raw device.

Some Solaris 2.x backup programs ignore files over a certain size limit, so you must verify that the Solaris 2.x file is backed up.

Copying to a Raw Tape Device

To copy the raw disk device directly to the raw tape device, see the following example:

```bash
$ dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0
```

This command results in one operating system block per tape record. It assumes that the datafile uses the entire disk partition.

The following example shows more typical usage:

```bash
$ dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b count=10000
```

This command copies 10,000 blocks of the tape, 10 blocks per tape record (assuming 512-KB database blocks).

Use `dbfsize /dev/rdsk/c0t1d0s6` to determine the number of Oracle7 blocks in the database file. Add one Oracle7 block to that number for the database header block.

The `dd` command typically runs faster with a larger block size (bs=20 KB or bs=32 KB). Block sizes larger than 20 KB are discouraged when copying to tape, because the records are long and less reliable.

The `dd` command does not handle multiple tapes, so if the partition size exceeds one tape, use multiple `dd` commands. For example:

```bash
$ dd if=/dev/rdsk/c0t1d0s6 of=/dev/rst0 bs=10b count=10000
Change tapes, then enter:

$ dd if=/dev/rdsk/c0t1d0s6 skip=10000 of=/dev/rst0 bs=10b count=10000
```

$ dd if=/dev/rdsk/c0t1d0s6 skip=10000 of=/dev/rst0 bs=10b count=10000

If you block at 20 blocks per record, you can fit approximately 50,000 blocks on a standard 2400-foot tape at 1600 bpi.

For maximum speed, make the block size a multiple (or factor) of the number of blocks per track. Therefore, a block size of 16 is optimal for a device with 32 blocks per track.

Support for Logical Volumes

Solaris 2.x supports logical volumes as described in the Oracle7 Administrator’s Reference for UNIX.

Initialization Parameters

This section describes initialization parameters that can be modified in the init.ora file for your Oracle7 Server instance. Topics include:

- Default Initialization Parameter Values
- Asynchronous I/O
- Log Files
- Blocks
- Latching
- Parallel Server
- SGA
- Post-Wait Driver

See Also: Chapter 2, “Creating a Database”, in the Oracle7 Server Administrator’s Guide, and “Customizing the init.ora File” in Chapter 1, “Planning a Complex Database on UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

Default Initialization Parameter Values

The following table lists default initialization parameter values on Solaris 2.x. All Oracle7 Server instances assume these values if you do not specify different values for them in the init.ora file. Oracle Corporation recommends that you include in the init.ora file only those parameters that differ from the default values.
To display the current parameter values on your system, use Oracle Server Manager to execute the SQL statement SHOW PARAMETERS.

**See Also:** *Oracle7 Server Reference* for more information on these parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUDIT_FILE_DEST</td>
<td>$ORACLE_HOME/rdbms/audit</td>
</tr>
<tr>
<td>BACKGROUND_DUMP_DEST</td>
<td>$ORACLE_HOME/rdbms/log</td>
</tr>
<tr>
<td>COMMIT_POINT_STRENGTH</td>
<td>1</td>
</tr>
<tr>
<td>COMPATIBLE</td>
<td>See Oracle7 Server Migration</td>
</tr>
<tr>
<td>COMPATIBLE_NO_RECOVERY</td>
<td>See Oracle7 Server Migration</td>
</tr>
</tbody>
</table>
| CONTROL_FILES              | $ORACLE_HOME/dbs/ctrl1/sid.dbf  
                          | $ORACLE_HOME/dbs/ctrl2/sid.dbf  
<pre><code>                      | $ORACLE_HOME/dbs/ctrl3/sid.dbf |
</code></pre>
<p>| DB_BLOCK_BUFFERS           | 32            |
| DB_BLOCK_SIZE              | 2 KB          |
| DB_FILES                   | 30 (maximum of 1022) |
| DB_FILE_MULTIBLOCK_READ_COUNT | 4 (range of 1–32) |
| DB_FILE_SIMULTANEOUS_WRITES | 4             |
| DISTRIBUTED_TRANSACTIONS   | 8             |
| INIT_SQL_FILES             | $ORACLE_HOME/dbs/sql.bsq |
| INSTANCE_NUMBER             | 0             |
| LOG_ARCHIVE_BUFFER_SIZE    | 16            |
| LOG_ARCHIVE_BUFFERS        | 4             |
| LOG_ARCHIVE_DEST           | $ORACLE_HOME/dbs/arch |
| LOG_ARCHIVE_FORMAT         | %t_%s.dbf     |
| LOG_BUFFER                 | 4 x maximum block size |
| LOG_CHECKPOINT_INTERVAL    | 500 x (max block size/min. block size) |
| LOG_SMALL_ENTRY_MAX_SIZE   | 800           |
| MTS_LISTENER_ADDRESS       | (ADDRESS=address) See Chapter 9, “SQL*Net Version 2”, in the Oracle7 Administrator’s Reference for UNIX |
| MTS_MAX_DISPATCHERS        | 0             |
| MTS_MAX_SERVERS            | 0 (maximum of 50) |
| MTS_SERVERS                | 0             |
| NLS_LANGUAGE               | AMERICAN      |
| NLS_NUMERIC_CHARACTERS     | “.,” (derived from NLS_TERRITORY) |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLS_TERRITORY</td>
<td>AMERICA</td>
</tr>
<tr>
<td>OPEN_CURSORS</td>
<td>50 per process</td>
</tr>
<tr>
<td>OS_AUTHENT_PREFIX</td>
<td>ops$</td>
</tr>
<tr>
<td>PROCESSES</td>
<td>25</td>
</tr>
<tr>
<td>SHARED_POOL_SIZE</td>
<td>3500000 bytes</td>
</tr>
<tr>
<td>SORT_AREA_SIZE</td>
<td>65536</td>
</tr>
<tr>
<td>SORT_SPACEMAP_SIZE</td>
<td>512</td>
</tr>
<tr>
<td>TEMPORARY_TABLE_LOCKS</td>
<td>32</td>
</tr>
<tr>
<td>TRANSACTIONS_PER.Rollback_SEGMENT</td>
<td>16</td>
</tr>
<tr>
<td>USER_DUMP_DEST</td>
<td>$ORACLE_HOME/rdbms/log</td>
</tr>
</tbody>
</table>

Table 1 – 2 Default Initialization Parameters

Asynchronous I/O

ASYNC_WRITE

Default value: true
Range of values: true/false

Determines whether Oracle7 Server uses Solaris 2.x asynchronous I/O (AIO).

If this parameter is not set or set to FALSE, the Solaris 2.x AIO feature is not enabled for the Oracle7 Server. To turn on the Solaris 2.x AIO feature for the Oracle7 Server, set ASYNC_WRITE = TRUE and DB_WRITER = 1 in the initSID.ora file.

ASYNC_READ

Default value: true
Range of values: true/false

Enables or disables the Oracle7 Server from using Solaris 2.x asynchronous read I/O. If you do not use asynchronous read I/O on the system, set this parameter to 0. Asynchronous read is only used during Oracle7 recovery.
DB_WRITERS

Default value: 1
Range of values: 1–50

Specifies the number of database writer (DBWR) processes to be brought up when the database is started. The setting of this parameter depends on whether you use Solaris 2.x asynchronous I/O (AIO) or Oracle multiple database writers.

If you decide to use multiple database writers instead of asynchronous I/O, set this parameter to the appropriate number of database writers and set ASYNC_WRITE = FALSE. The number of database writers required is application-specific. However, this number should be at least equal to the number of disk drives used for holding tablespaces.

Note that each additional DBWR process consumes one more semaphore on the system, so the PROCESSES parameter in the init$SID.ora file and the Solaris 2.x semaphore parameters may need to be increased accordingly.

Log Files

LOG_ARCHIVE_DEST

Default value: $ORACLE_HOME/dbs/arch
Range of values: a valid path or device name

Specifies the name of the redo log archive file. This value must be a filename, not a directory location.

LOG_CHECKPOINT_INTERVAL

Default value: 4000
Range of values: 2–unlimited

Sets the number of redo log file blocks (operating system blocks, not Oracle7 blocks) required to trigger a checkpoint. A checkpoint ensures that all blocks modified since the previous checkpoint are written to disk. A checkpoint occurs when either the redo log file is full or the LOG_CHECKPOINT_INTERVAL is reached.
LOG_CHECKPOINT_TIMEOUT

Default value: 0 seconds

Range of values: 0–unlimited

Sets the maximum interval between checkpoints, in seconds, for a secondary database. This parameter does not override and is not overridden by the LOG_CHECKPOINT_INTERVAL parameter.

See Also: Oracle7 Server Administrator’s Guide for performance and tuning issues related to setting checkpoints.

LOG_FILES

Default value: 16

Range of values: 2–255

Sets the maximum number of redo log files that can be opened at runtime for this database. This is in effect every time an instance opens a database, and lasts until the database is closed.

This parameter is related to the MAXLOGFILES argument for the CREATE DATABASE statement, which is the absolute maximum number of redo log files for all instances. If neither LOG_FILES nor MAXLOGFILES is used, the default limit of the number of redo log files is the Oracle limit of 255 files. Otherwise, the default is 16 files.

LOG_BUFFER

Default value: 4 * DB_BLOCK_SIZE

Range of values: 2 KB–unlimited

Sets the number of bytes allocated to the redo log buffers in the System Global Area (SGA). In general, larger values reduce redo log file I/O, particularly if transactions are long or numerous. The default is four times the value of DB_BLOCK_SIZE. For efficiency, LOG_BUFFER should be a multiple of DB_BLOCK_SIZE.

When the system writes to the redo log buffers, it includes the block of altered bytes, the rollback segments block, and the database header data. Therefore, the entry in the redo log buffer is normally larger than the database buffers.
Blocks

CCF_IO_SIZE

Default value: 131072
Range of values: 1–131072
Determines the number of contiguous blocks per write when creating a database. The higher the number, the faster the database is created.

DB_BLOCK_SIZE

Default value: 2 KB
Range of values: 512–16 KB
Sets the size in bytes of an Oracle7 database block. The current value of DB_BLOCK_SIZE on the system can be found using the SQL command SHOW PARAMETERS.

DB_FILE_MULTIBLOCK_READ_COUNT

Default value: 8
Range of values: 1–32
Used for multi-block read operations. It sets the number of blocks to read when doing I/O during a sequential scan. This parameter is useful for performing a full table scan in which the WHERE clause does not refer to an indexed column.

USE_READV

Default value: false
Range of values: true/false
Enables you to use the readv system call to handle scattered read operations. If you set this parameter to TRUE, multi-block read operations are done through the readv call. If you set this parameter to FALSE, a buffer is allocated and scattered read operations are simulated by reading all blocks, followed by memory-to-memory copies.
Latching

LOG_SIMULTANEOUS_COPIES

**Default value:** CPU_COUNT  
**Range of values:** 0–unlimited  
Specifies the number of copy latches. If this is not specified, the default value is set to CPU_COUNT or set internally. Note that the SQL command SHOW PARAMETERS shows the value of this parameter as the default (which is 0), rather than the internally determined value.

SPIN_COUNT (for multiple CPUs only)

**Default value:** 2000  
**Range of values:** 0–unlimited  
Sets the number of times an Oracle7 process attempts to get a latch before sleeping. If the latch is busy, the Oracle7 Server spins and checks back with the latch up to SPIN_COUNT number of times until the latch is free. Once it attains SPIN_COUNT, the Oracle7 process sleeps for a set period of time, then tries again.

You can adjust performance by changing the value of SPIN_COUNT. A high value obtains the latch sooner than a low value, if the degree of latch contention is low. However, a high value can also use more CPU time. A good starting SPIN_COUNT value is 2000.

If there is a high level of latch contention, making it difficult to obtain a latch during the spin, set this number low so the process can relinquish the CPU to other processes.

TIMEOUTS

**Default value:** varies by machine  
**Range of values:** 0–unlimited  
Note that the TIMEOUTS column in the Server Manager display is system-specific in meaning. The number of TIMEOUTS is the number of times the process was unable to get a resource because it could not get a latch for that resource. This number is machine-specific.
Parallel Server

MAX_COMMIT_PROPAGATION_DELAY

Default value: 90000

Range of values: 0–90000

Indicates the maximum time in seconds allowed before the system commit number (SCN) held in the SGA of an instance is refreshed. Units are in hundredths of seconds. When user-level queries are issued in a Parallel Server environment, the SCN obtains a consistent read from the database. Since the system uses the value of the local SCN, held in the SGA, this may be inconsistent with the global SCN, maintained by the Distributed Lock Manager (DLM), or an SCN server.

The MAX_COMMIT_PROPAGATION_DELAY parameter specifies how often the local SCN value is refreshed. Setting the parameter to zero enforces a refresh for each user-level query. The default value of 90,000 (15 minutes) ensures that a refresh only occurs during a commit, or a data dictionary operation.

This parameter does not need to be identical in all instances.

SGA

PRE_PAGE_SGA

Default value: no

Range of values: yes/no

Touches all the SGA pages, bringing them into memory. This increases instance startup time and user login time, but reduces the number of possible page faults. The reduction in page faults allows the instance to reach its maximum performance capability quickly, rather than incrementally. It is most useful on systems with sufficient memory to hold all the SGA pages without degrading performance in other areas.
Post-Wait Driver

POST_WAIT_DEVICE

Default value: /devices/pseudo/pw@0:pw

Range of values: device_entry_point_name

Defines the device entry point to be used by the post-wait driver for the current instance. By default, this corresponds to the device entry point that Solaris 2.x creates for you.

⚠️ Warning: This parameter should only be used with Oracle Parallel Server and after consultation with Oracle World Wide Support. For support in the United States, call +1 415 506 1500. In Europe, call +44 344 860160.

USE_POST_WAIT_EXTENSION

Default value: false

Range of values: true/false

Enables/disables the use of the kernel extension that performs efficient post-wait operations in Oracle.

This feature requires that the post-wait driver extension is installed on the system. If this parameter is set to TRUE, the system first checks the extension. If it is not installed, Oracle uses semaphores for post-wait operations.

⚠️ Warning: This parameter should only be used with Oracle Parallel Server and after consultation with Oracle World Wide Support. For support in the United States, call +1 415 506 1500. In Europe, call +44 344 860160.
This chapter presents Solaris 2.x-specific information that supplements the section for each interface product in Chapter 8, “Oracle Precompilers and Interfaces”, in the Oracle7 Administrator’s Reference for UNIX.

The topics covered in this chapter are:

- Pro*Ada
- Pro*C
- Pro*COBOL
- Pro*FORTRAN
- Pro*Pascal
- SQL*Module for Ada
- SQL*Module for C
Pro*Ada

This section provides Solaris 2.x-specific additions to “Pro*Ada” in Chapter 8, “Oracle Precompilers and Interfaces”, in the Oracle7 Administrator’s Reference for UNIX.

Checking PATH

Make sure ada is in your PATH. Also, make sure the LM_LICENSE_FILE environment variable points to the correct license file for the Ada compiler. To check the setting of PATH:

```bash
$ echo $PATH
```

Linking Instructions

To link programs, set up the environment and then link the programs in the shell. This is true of both Oracle and user programs.

On Solaris 2.x the Oracle Call Interface supports the C language and Ada only.

Pro*C

This section provides Solaris 2.x-specific additions to “Pro*C” in Chapter 8, “Oracle Precompilers and Interfaces”, of the Oracle7 Administrator’s Reference for UNIX.

Linking Instructions

To link programs, you must set up the environment and then link the programs in the shell. This is true of both Oracle and user programs.

Error 1043 in Pro*C programs

If you receive error message 1043 while in your Pro*C program, you have tried to connect to the database with the CONNECT command too many times. Disconnect from the database with the COMMIT WORK RELEASE command before trying to connect again.
**Pro*COBOL**

This section provides Solaris 2.x-specific additions to “Pro*COBOL” in Chapter 8, “Oracle Precompilers and Interfaces”, in the *Oracle7 Administrator’s Reference for UNIX*.

**Relinking Pro*COBOL Executables**

Before relinking Pro*COBOL executables, you must change one of the specified object files in the `$COBDIR/coblib/liblist` file to resolve undefined symbols. In the `$COBDIR/coblib/liblist` file, replace the lines:

```
i/opt/SUNWspro/SC3.0.1/_fstd.o
p/opt/SUNWspro/SC3.0.1/_fstd.o
```

with:

```
i/opt/SUNWspro/SC3.0.1/lib/cg92/_fstd.o
p/opt/SUNWspro/SC3.0.1/lib/cg92/_fstd.o
```

*Note:* In the previous example, “SC3.0.1” represents a SPARC compiler version number. Be sure you enter the current version number when you perform this task.

**Setting COBDIR**

The Pro*COBOL precompiler requires that the COBDIR environment variable is set to the home directory of the compiler, and `$COBDIR/bin` is included in the PATH variable.

If you are using Micro Focus COBOL 3.2, you must set `LD_LIBRARY_PATH=$COBDIR/coblib` before building the sample Pro*COBOL demonstration programs.

**Sun Nihongo COBOL**

If you are using Sun Nihongo COBOL, rename the makefiles:

```
$ cd $ORACLE_HOME/procob/lib
$ mv procob.mk procob.mk.mf
$ cp procob.mk.nsun procob.mk
$ cp procob.mk $ORACLE_HOME/procob/demo
```
FORMAT Option

The FORMAT precompiler option is not available with this COBOL compiler.

---

Pro*FORTRAN

This section provides Solaris 2.x-specific additions to “Pro*FORTRAN” in Chapter 8, “Oracle Precompilers and Interfaces”, in the Oracle7 Administrator’s Reference for UNIX.

Linking Instructions

To link programs, set up the environment and link the programs in the shell. This is true of both Oracle and user programs.

Pro*FORTRAN is not supported on Solaris Intel.

---

Pro*Pascal

This section provides Solaris 2.x-specific additions to “Pro*Pascal” in Chapter 8, “Oracle Precompilers and Interfaces”, in the Oracle7 Administrator’s Reference for UNIX.

Linking Instructions

To link programs, set up the environment and then link the programs in the Bourne shell. This is true of both Oracle and user programs.

---

Sample Pascal Program

To make the sample Pascal program with native BSD behavior (without System V emulation):

$ make -f propas.mk samplefBSD
$ samplefBSD
SQL*Module for Ada

This section provides Solaris 2.x-specific additions to “SQL*Module for Ada and C” in Chapter 8, “Oracle Precompilers and Interfaces”, in the Oracle7 Administrator’s Reference for UNIX.

To run a demonstration of SQL*Module for Ada:

$ cd $ORACLE_HOME/mod/demo/ada
$ make -f moddemoada.mk SYSTEMPASS=system_password all

SQL*Module for C

This section provides Solaris 2.x-specific additions to “SQL*Module for Ada and C” in Chapter 8, “Oracle Precompilers and Interfaces”, in the Oracle7 Administrator’s Reference for UNIX.

Linking Instructions

To link programs, set up the environment and then link the programs in the shell. This is true of both Oracle and user programs.

To run a demonstration of SQL*Module for C:

$ cd $ORACLE_HOME/mod/demo/c
$ make -f moddemoc.mk SYSTEMPASS=system_password all
This chapter presents Solaris 2.x-specific information that supplements the section for each SQL*Net version 2 product in Chapter 9, “SQL*Net Version 2”, in the Oracle7 Administrator’s Reference for UNIX.

The topics covered in this chapter are:

- SQL*Net Version 2 Overview
- Oracle APPC/LU6.2 Protocol Adapter
- Oracle DECnet Protocol Adapter
SQL*Net Version 2 Overview

This section provides Solaris 2.x-specific additions to “Overview of SQL*Net Version 2 Products” in Chapter 9, “SQL*Net Version 2”, of the Oracle7 Administrator’s Reference for UNIX.

Topics include:

- Oracle Protocol Adapters
- SQL*Net Version 2 Files and Utilities

Oracle Protocol Adapters

Default Adapter

The IPC protocol adapter, which is implemented on top of UNIX Domain Sockets, is used for local connections to the multi-threaded server (MTS). The BEQ protocol adapter, implemented on top of UNIX pipes, cannot be configured. The BEQ protocol adapter spawns and connects processes via UNIX pipes.

Out-of-Band Breaks

Out-of-band breaks are not supported with SQL*Net version 2. You cannot cancel a query over a SQL*Net version 2 connection by entering [CTRL]-c.

SQL*Net Version 2 Files and Utilities

TNS Listener

This task is only necessary if you have a SQL*Net version 1 database link to a SQL*Net version 1 server that is contacted by a SQL*Net version 2 client. Set the TNS listener process setuid root for the SQL*Net version 2 listener on the intermediate node.

As root, set the TNS listener process setuid root:

```bash
# cd $ORACLE_HOME/bin
# chown root tnslsnr
# chmod 4751 tnslsnr
```
Oracle APPC/LU6.2 Protocol Adapter

This section provides Solaris 2.x-specific additions to “Oracle APPC/LU6.2 Protocol Adapter” in Chapter 9, “SQL*Net Version 2”, in the Oracle7 Administrator’s Reference for UNIX.

Using Oracle APPC/LU6.2 Protocol Adapter

Topics include:
- Specifying the APPC/LU6.2 address
- Solaris 2.x-Specific Listener

Specifying the APPC/LU6.2 Address

Once APPC/LU6.2 and the Oracle APPC/LU6.2 protocol adapter are installed for your system, use the APPC/LU6.2-specific parameters within the connect descriptors in the tnsnames.ora file to identify nodes within the APPC/LU6.2 community.

The APPC/LU6.2 protocol adapter parameters are defined in a connect descriptor for each node. Each connect descriptor contains several keyword=value pairs. The APPC/LU6.2-specific keywords can be entered in any order within the connect descriptor. The syntax for the APPC/LU6.2 protocol adapter connection is:

```plaintext
(ADDRESS=

[(COMMUNITY= community_name )]
(PROTOCOL= protocol )
(TP_NAME= transaction_program_name )
(LU_NAME= logical_unit_name )
(MODE= mode_name )
(PLU= partner_lu_name )
)
```

where:

- `community_name` specifies the protocol-specific community name when you also have the MultiProtocol Interchange product. The community name should be unique on the network and identify the community by a name that is indicative of the community. The name, however, should not contain the protocol name, in case multiple networks are integrated.
using the MultiProtocol Interchange and several community names use the same protocol.

**protocol** specifies the protocol adapter to be used; the value can be uppercase or lowercase. For APPC/LU6.2, the value is lu62.

**transaction_program_name** specifies the name of the transaction program to run at the target or the transaction program name to use when listening for incoming connection requests. This value is required.

**logical_unit_name** in tnsnames.ora, it specifies the name for the remote partner LU. When this keyword appears in listener.ora, it specifies the name of the local LU. LU_NAME can be ignored on many platforms or overridden by the values in other parameters. Due to the requirements of some APPC/LU6.2 implementations, LU_NAME should always specify the fully qualified LU name (that is, netid.lu_name).

**mode_name** defines the characteristics of sessions between logical units. The mode, along with the partner LU and the transaction program name, is specified in the ALLOCATE segments. The mode name must be common to both the local and partner LU. This value is required.

**partner_lu_name** specifies the name of the partner LU. This value is required on Solaris 2.x, and can be set to the TP_NAME/transaction_program_name.

### Solaris 2.x-Specific Listener

Solaris 2.x does not support the generic listener. To bring up the listener on the server side, run the ntlsnr command, whose syntax is:

```
ntlsnr start|stop -l luname -t tpname -m modename
```

**luname** in tnsnames.ora, it specifies the name for the remote partner LU. When this keyword appears in listener.ora, it specifies the name of the local LU. LU_NAME can be ignored on many platforms or overridden by the values in other parameters. Due to the requirements of some APPC/LU6.2 implementations, LU_NAME should always specify
the fully qualified LU name (that is, netid.lu_name).

\textit{tpname} specifies the name of the transaction program to run at the target or the transaction program name to use when listening for incoming connection requests.

\textit{mode} defines the characteristics of sessions between logical units. The mode, along with the partner LU and the transaction program name, is specified in the ALLOCATE segments. The mode name must be common to both the local and partner LU.

\section*{Oracle DECnet Protocol Adapter}

This section provides Solaris 2.x-specific additions to “Oracle DECnet Protocol Adapter” in Chapter 9, “SQL*Net Version 2”, in the \textit{Oracle7 Administrator’s Reference for UNIX}.

\section*{Using Oracle DECnet Protocol Adapter}

\subsection*{Connecting with Link Software}

Attempting connections using SunLink DNI 7.x may fail (generating a DNI error: BAD_OBJECT). If you get this error, retry the connection.

\textbf{See Also:} The \textit{SunLink DNI System Administrator’s Guide}. 

This chapter contains Solaris 2.x-specific information referenced by other Oracle7 Server manuals. This chapter is not intended for use on its own; it is an addendum to the Oracle7 Server documentation set.

This chapter provides Solaris 2.x-specific information for the following manuals:

- Oracle7 Server Administrator’s Guide
- Oracle7 Server Application Developer’s Guide
- Oracle7 Server Concepts
- Oracle7 Server Distributed Systems Volume 1: Distributed Data
- Oracle7 Server Distributed Systems Volume 2: Replicated Data
- Oracle7 Server Migration
- Oracle7 Parallel Server Concepts and Administration
- Oracle7 Server Reference
- Oracle7 Server SQL Reference
- Oracle7 Server Utilities
Oracle7 Server Administrator’s Guide

This section provides Solaris 2.x-specific information to supplement Appendix A in the Oracle7 Server Administrator’s Guide. These topics are listed alphabetically.

Calculating Cluster Size

Use the following figures for calculating cluster size using the formula in the Oracle7 Server Administrator’s Guide:

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed header size</td>
<td>68 bytes</td>
</tr>
<tr>
<td>Variable transaction header</td>
<td>24*INITRANS value for the table</td>
</tr>
<tr>
<td>Row directory</td>
<td>4 bytes per row of a clustered table</td>
</tr>
</tbody>
</table>

Table 4 – 1  Cluster Size Values

Calculating Database Limits

For the equation given in the Oracle7 Server Administrator’s Guide, use the following figures:

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database file size</td>
<td>5 MB</td>
</tr>
<tr>
<td>Database files</td>
<td>255 or the value of the DB_FILES parameter in the init.ora file</td>
</tr>
<tr>
<td>instances</td>
<td>255</td>
</tr>
<tr>
<td>locks</td>
<td>100 (DML_LOCKS parameters value)</td>
</tr>
<tr>
<td>MAXEXTENTS</td>
<td>57 for a 1KB block size, 121 for a 2 KB block size, 249 for a 4 KB block size, and 505 for an 8 KB block size</td>
</tr>
<tr>
<td>Redo log files</td>
<td>255</td>
</tr>
</tbody>
</table>

Table 4 – 2  Database Size Values

Data files are located in $ORACLE_HOME/dbs by default.
Calculating Index Size

Use the following figures for calculating the size required by an index using the formula in the Oracle7 Server Administrator’s Guide:

<table>
<thead>
<tr>
<th>Type</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed header size</td>
<td>113 bytes</td>
</tr>
<tr>
<td>Variable transaction header</td>
<td>$24 \times \text{INITRANS value for the index}$</td>
</tr>
<tr>
<td>Entry header</td>
<td>5 bytes</td>
</tr>
</tbody>
</table>

Table 4 – 3  Index Size Values

Oracle7 Server Application Developer’s Guide

This section provides Solaris 2.x-specific information to supplement Chapter 9, “Using Database Triggers”, and Chapter 11, “Managing Dependencies Among Schema Objects” in the Oracle7 Server Application Developer’s Guide.

All rdbms SQL scripts are located in the rdbms/admin directory.

Oracle7 Server Concepts

This section provides Solaris 2.x-specific information to supplement Oracle7 Server Concepts. These topics are listed in alphabetical order.

Authentication and Privileges

User authentication, administrator privileges, database roles, and other security issues are discussed in Chapter 5, “Administering Oracle7 on UNIX” in the Oracle7 Administrator’s Reference for UNIX.

Archiver Process

Archiving is discussed in Chapter 4, “Backing Up and Recovering Oracle7 on UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

Asynchronous I/O

Asynchronous I/O and its relation to Parallel Server performance are discussed in “I/O Types” on page 1 – 6.
Multiple Database Writers

Multiple database writers (DBWRs) are discussed in “I/O Types” on page 1 – 6.

**Oracle7 Server Distributed Systems Volume 1: Distributed Data**

This section provides Solaris 2.x-specific information to supplement *Oracle7 Server Distributed Systems Volume 1: Distributed Data*. These topics are listed in alphabetical order.

**Distributed Transactions**

Distributed transactions and related locking issues are discussed in “Task 7: Tuning Resource Contention for Oracle Parallel Server” in Chapter 3, “Tuning Oracle7 for UNIX” in the *Oracle7 Administrator’s Reference for UNIX*.

**The init.ora File**

The *init.ora* file is discussed in “Initialization Parameters” on page 1 – 25, and Chapter 1, “Planning a Complex Database on UNIX”, in the *Oracle7 Administrator’s Reference for UNIX*.

**Memory Requirements**

Memory requirements are discussed in “Task 3: Tuning Memory Management” in Chapter 3, “Tuning Oracle7 for UNIX” in the *Oracle7 Administrator’s Reference for UNIX*.

**XA Libraries**

Oracle*XA* support for shared libraries is discussed in “Shared Libraries and Shared Objects Support” in Chapter 8, “Oracle Precompilers and Interfaces”, in the *Oracle7 Administrator’s Reference for UNIX*.

**Library Names**

Library naming conventions are discussed in Chapter 2, “Oracle7 Architecture on UNIX”, in the *Oracle7 Administrator’s Reference for UNIX*. 
XA README.doc

The README.doc file located in the $ORACLE_HOME/xa/doc directory describes changes, bugs, or restrictions in the Oracle XA Library since the last version. The README.doc file is an outline supplement to the printed documentation.

XAVIEW.SQL Script

The XAVIEW.SQL script used to manually create the V$XATRANS$ view is located under $ORACLE_HOME/xa/admin/xaview.sql.

Oracle7 Server Distributed Systems Volume 2: Replicated Data

This section provides Solaris 2.x-specific information to supplement Oracle7 Server Distributed Systems Volume 2: Replicated Data.

Advanced Replication Option

To install a separate feature, such as the Advanced Replication Option, see the Oracle7 Installation Guide for Sun SPARC Solaris 2.x.

Oracle7 Server Migration

This section provides Solaris 2.x-specific information to supplement Oracle7 Server Migration.

Solaris 2.x-Specific Migration Tasks

All Solaris 2.x-specific migration tasks are discussed in Chapter 8, “Migrating to Oracle7”, in the Oracle7 Installation Guide for Sun SPARC Solaris 2.x.
MAXEXTENTS

Calculate the MAXEXTENTS parameter from the block size on your system:

<table>
<thead>
<tr>
<th>Block Size</th>
<th>Number of Extents</th>
<th>Platforms and Uses for Block Size (most can be set larger)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 KB</td>
<td>121</td>
<td>Sun, Hewlett-Packard, Sequent, most UNIX platforms</td>
</tr>
<tr>
<td>4 KB</td>
<td>249</td>
<td>IBM-AIX, Sequent (for large databases)</td>
</tr>
<tr>
<td>8 KB</td>
<td>507</td>
<td>for VLDBs</td>
</tr>
<tr>
<td>16 KB</td>
<td>1017</td>
<td>for decision-support VLDBs</td>
</tr>
</tbody>
</table>

Table 4–4 Block Size and Number of Extents

Memory Requirements

Memory requirements are discussed in “Disk Space and Memory Requirements” in Chapter 1, “Features and Requirements”, in the Oracle7 Installation Guide for Sun SPARC Solaris 2.x.

Oracle7 Installer

The Oracle7 Installer method of performing migration is discussed in the Oracle7 Installation Guide for Sun SPARC Solaris 2.x.

SGA

Shared memory segment size and the SGA are discussed in “Tip 1: Hold the SGA in a Single Shared Memory Segment” in Chapter 3, “Tuning Oracle7 for UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

Oracle7 Parallel Server Concepts and Administration

This section provides Solaris 2.x-specific information to supplement Chapter 19, “Administering Multiple Instances”, in Oracle7 Parallel Server Concepts and Administration.

User authentication, administrator privileges, database roles, and other security issues are discussed in Chapter 5, “Administering Oracle7 on UNIX” in the Oracle7 Administrator’s Reference for UNIX.
Oracle7 Server Reference

This section provides Solaris 2.x-specific information to supplement the Oracle7 Server Reference.

Initialization Parameters

Initialization parameters and their default values are discussed in “Initialization Parameters” on page 1 – 25.

SQL Scripts

All rdbms SQL scripts are located in the rdbms/admin directory.

Oracle7 Server SQL Reference

This section provides Solaris 2.x-specific information to supplement Chapter 4, “Commands”, in the Oracle7 Server SQL Reference.

CREATE CONTROLFILE Parameters

The values of the following parameters determine the size of control files for a database.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
<th>Maximum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXDATAFILES</td>
<td>30</td>
<td>1022</td>
</tr>
<tr>
<td>MAXINSTANCES</td>
<td>63</td>
<td>63</td>
</tr>
<tr>
<td>MAXLOGFILES</td>
<td>16</td>
<td>255</td>
</tr>
<tr>
<td>MAXLOGMEMBERS</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>MAXLOGHISTORY</td>
<td>100</td>
<td>63000</td>
</tr>
</tbody>
</table>

Table 4 – 5 CREATE CONTROLFILE Parameter Values

CREATE TRIGGER Parameters

All rdbms SQL scripts are located in the rdbms/admin directory.
STORAGE Parameters

Default storage parameter values are listed in “Task 6: Plan Space Management Parameters” in Chapter 1, “Planning a Complex Database on UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

Oracle7 Server Utilities

This section provides Solaris 2.x-specific information to supplement Oracle7 Server Utilities.

Error Messages

Solaris 2.x-specific error messages are documented in Appendix A, “Oracle Messages on UNIX,” in Oracle7 Administrator’s Reference for UNIX.

I/O Buffers

I/O performance problems are discussed in “Task 3: Tuning Memory Management” and “Task 4: Tuning Disk I/O” in Chapter 3, “Tuning Oracle7 for UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

Import

Import concerns and the Oracle7 Installer are discussed in Chapter 9, “Import/Export”, in the Oracle7 Installation Guide for Sun SPARC Solaris 2.x.

Naming Conventions

Oracle7 file naming conventions are discussed in Chapter 2, “Oracle7 Architecture on UNIX”, in the Oracle7 Administrator’s Reference for UNIX.

SQL Scripts

All rdbms SQL scripts are located in the rdbms/admin directory.
A
Ada
Pro*Ada, 2 – 2
SQL*Module for, 2 – 5
allocating, free space, 1 – 11
APPC/LU6.2 Protocol Adapter
specifying address, 3 – 3
updates, 3 – 3
archiving redo log files, destination, 1 – 28
ASYNC parameters
ASYNC_READ, 1 – 27
ASYNC_WRITE, 1 – 7, 1 – 27
asynchronous I/O
and DBWR, 1 – 7
and multiple database writers, 1 – 6
initialization parameters, 1 – 27
AUDIT_FILE_DEST parameter, 1 – 26

B
BACKGROUND_DUMP_DEST parameter, 1 – 26
BEQ protocol adapter, 3 – 2
block size, 1 – 9
blocks, initialization parameters, 1 – 30
buffer cache size, tuning, 1 – 23

C
C language
Pro*C, 2 – 2
SQL*Module for, 2 – 5
cache
hit ratio, 1 – 23
size, tuning, 1 – 23
CCF_IO_SIZE parameter, 1 – 30
cluster, estimating size, 4 – 2
COBDIR environment variable, 2 – 3
COBOL, Pro*COBOL, 2 – 3
COMMITS_POINT_STRENGTH parameter, 1 – 26
COMPATIBLE parameter, 1 – 26
COMPATIBLE_NO_RECOVERY parameter, 1 – 26
database, fragmentation, 1 – 12
limits, 4 – 2
database writers, and asynchronous I/O, 1 – 6
datablocks, associating with instances, 1 – 11
DB parameters
DB_BLOCK_SIZE, 1 – 26, 1 – 30
DB_FILE_MULTIBLOCK_READ_COUNT, 1 – 26, 1 – 30
DB_FILE_SIMULTANEOUS_WRITES, 1 – 26
DB_FILES, 1 – 26
DB_WRITERS, 1 – 7, 1 – 28
DB_BLOCK_BUFFERS parameter, 1 – 26
dbwr process, and asynchronous I/O, 1 – 7
DECnet Protocol Adapter, updates, 3 – 5
devices, raw, 1 – 17
disk
monitoring performance, 1 – 8
quotas, 1 – 4
DISTRIBUTED_TRANSACTIONS parameter, 1 – 26
dspst tool, 1 – 8

E
errors, 1043 in Pro*C, 2 – 2
extents, automatic allocation, 1 – 11

F
file size, 1 – 9
FORTRAN, Pro*FORTRAN, 2 – 4
fragmentation, reducing database, 1 – 12
free space lists
  associating with instances, 1 – 11
  locating space, 1 – 11

G
GC parameters
  GC_DB_LOCKS, 1 – 15
  GC_FILES_TO_LOCKS, 1 – 15

I
I/O
  asynchronous and multiple database writers, 1 – 6
  introduction, 1 – 6
  list, 1 – 7
index size, calculating, 4 – 3
INIT_SQL_FILES parameter, 1 – 26
initialization parameters
  defaults, 1 – 25
  list of, 1 – 25
  SHOW PARAMETERS command, 1 – 25
inserts, locating free space, 1 – 11
instance, associating with datablocks, 1 – 11
INSTANCE_NUMBER parameter, 1 – 26
Intimate Shared Memory, 1 – 2
iostat tool, 1 – 8
IPC protocol adapter, 3 – 2
ISM, Intimate Shared Memory, 1 – 2

L
latching, initialization parameters, 1 – 31
limits, resource, 1 – 4
list I/O, 1 – 7
locks, tuning for Parallel Server, 1 – 14
log files, initialization parameters, 1 – 28
LOG parameters
  LOG_ARCHIVE_BUFFER_SIZE, 1 – 26
  LOG_ARCHIVE_BUFFERS, 1 – 26
LOG_ARCHIVE_DEST, 1 – 26, 1 – 28
LOG_ARCHIVE_FORMAT, 1 – 26
LOG_BUFFER, 1 – 26, 1 – 29
LOG_CHECKPOINT_INTERVAL, 1 – 26, 1 – 28
LOG_CHECKPOINT_TIMEOUT, 1 – 29
LOG_FILES, 1 – 29
LOG_SIMULTANEOUS_COPIES, 1 – 31
LOG_SMALL_ENTRY_MAX_SIZE, 1 – 26
logical reads, 1 – 23
logical volumes, support, 1 – 25
LU6.2 Protocol Adapter, updates, 3 – 3

M
MAX_COMMIT_PROPAGATION_DELAY parameter, 1 – 32
MAXDATAFILES parameter, 4 – 7
MAXEXTENTS parameter, 1 – 10, 1 – 11
MAXINSTANCES parameter, 4 – 7
MAXLOGFILES parameter, 1 – 29, 4 – 7
MAXLOGHISTORY parameter, 4 – 7
MAXLOGMEMBERS parameter, 4 – 7
memory
  estimating usage, 1 – 3
  Intimate Shared Memory, 1 – 2
  shared, 1 – 4
  virtual, 1 – 3
MINEXTENTS parameter, 1 – 11
monitor latch display, 1 – 31
monitoring performance, tools, 1 – 8
MTS parameters
  MTS_LISTENER_ADDRESS, 1 – 26
  MTS_MAX_DISPATCHERS, 1 – 26
  MTS_MAX_SERVERS, 1 – 26
  MTS_SERVERS, 1 – 26
MultiProtocol Interchange, and APPC/LU6.2, 3 – 4

N
NLS_LANGUAGE parameter, 1 – 26
NLS_NUMERIC_CHARACTERS parameter, 1 – 26

Index – 10  Oracle7 Reference Addendum for Sun SPARC Solaris 2.x
NLS_TERRITORY parameter, 1 – 27

O
OPEN_CURSORS parameter, 1 – 27
Oracle, memory usage, 1 – 3
Oracle APPC/LU6.2 Protocol Adapter, 3 – 3
Oracle DECnet Protocol Adapter, updates, 3 – 5
Oracle Worldwide Technical Support, iv
how to contact in Europe, iv
how to contact in U.S.A., iv
Oracle7 Server, tuning, 1 – 9
OS_AUTHENT_PREFIX parameter, 1 – 27
osview, 1 – 8
out–of–band breaks, protocol adapters, 3 – 2

P
Parallel Server
contention for resources, 1 – 13
initialization parameters, 1 – 32
Pascal, Pro*Pascal, 2 – 4
perfmeter tool, 1 – 8
performance
monitoring tools, 1 – 8
Parallel Server, 1 – 13
partitioned data, 1 – 11
table striping, 1 – 12
Post–Wait driver, initialization parameters, 1 – 33
POST_WAIT_DEVICE parameter, 1 – 33
PRE_PAGE_SGA parameter, 1 – 32
precompilers, list of, 2 – 1
Pro*Ada
setting PATH, 2 – 2
updates, 2 – 2
Pro*C
error 1043, 2 – 2
updates, 2 – 2
Pro*CObOL
COBDIR environment variable, 2 – 3
FORMAT option, 2 – 4
relinking executables, 2 – 3
Sun Nihongo COBOL, 2 – 3
updates, 2 – 3
Pro*FORTRAN
linking, 2 – 4
updates, 2 – 4
Pro*Pascal
linking, 2 – 4
sample program, 2 – 4
updates, 2 – 4
PROCESSES parameter, 1 – 27
protocol adapters, list, 3 – 1
ps tool, 1 – 8
pstat tool, 1 – 8

R
ratio, cache hit, 1 – 23
raw devices
backing up, 1 – 24
buffer cache size, 1 – 23
creating partitions, 1 – 19
criteria, 1 – 18
disadvantages, 1 – 17
introduction, 1 – 17
README file, 4 – 5
redo log archive file, specifying, 1 – 28
resource limits, 1 – 4
sar command, 1 – 8, 1 – 23
Server Manager
monitor, 1 – 7
SHOW PARAMETERS, 1 – 25
SGA
initialization parameters, 1 – 32
relocating, 1 – 5
shared memory
Intimate Shared Memory, 1 – 2
SGA, 1 – 4
SHARED mode, tuning locks for Parallel
Server, 1 – 14
SHARED_POOL_SIZE parameter, 1 – 27
SORT parameters
  SORT_AREA_SIZE, 1 – 27
  SORT_SPACEMAP_SIZE, 1 – 27

space, allocating extents, 1 – 11
SPIN_COUNT parameter, 1 – 31

SQL*DBA
  monitor, 1 – 7
  SHOW PARAMETERS, 1 – 25

SQL*Module for Ada, updates, 2 – 5
SQL*Module for C
  demo, 2 – 5
  linking, 2 – 5
  updates, 2 – 5

SQL*Net
  list of products, 3 – 1
  updates, 3 – 2

STORAGE parameters, setting values, 4 – 8
Sun Nihongo COBOL, 2 – 3
syspic tool, 1 – 8
System Global Area (SGA)
  relocating, 1 – 5
  requirements, 1 – 4

T
  tables
    allocating extents, 1 – 11
    locating free space, 1 – 11
    storing data in separate files, 1 – 12
    striping, 1 – 12
  TEMPORARY_TABLE_LOCKS parameter, 1 – 27
  TIMEOUTS parameter, 1 – 31
  TNS listener, SQL*Net V2 files and utilities, 3 – 2
  tools
    cpustat, 1 – 8
dspst, 1 – 8
iostat, 1 – 8
perfmeter, 1 – 8
ps, 1 – 8
psstat, 1 – 8
syspic, 1 – 8
vmstat, 1 – 8

transaction, locating free space, 1 – 11
TRANSACTIONS_PER_ROLLBACK_SEGMENT parameter, 1 – 27

tuning
  Oracle7 Server, 1 – 9
  Parallel Server, 1 – 13
  table striping, 1 – 12

U
  updates, locating free space, 1 – 11
  USE_ISM parameter, 1 – 2
  USE_POST_WAIT_EXTENSION parameter, 1 – 33
  USE_READV parameter, 1 – 30
  USER_DUMP_DEST parameter, 1 – 27

V
  vmstat tool, 1 – 8
  V$XATRANS$ view, 4 – 5

X
  XA Library
    README file, 4 – 5
    XAVIEW.SQL location, 4 – 5
Reader’s Comment Form

Oracle7™ Reference Addendum for Sun SPARC Solaris 2.x
Release 7.3.3, Part No. A52289–01

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 about this manual?

Please indicate your comments or suggestions and the page numbers to which they apply:

____________________________________

____________________________________

____________________________________

Please send your comments to:

Technical Publications Manager
Open Systems Operations and Tools
Oracle Corporation
500 Oracle Parkway
Box 659103
Redwood Shores, CA 94065

or

Technical Publications Manager
Oracle Corporation
European Porting Centre
Marelimo Court
Blackrock
Co Dublin
Ireland

Phone: 1.415.506.7000    FAX: 1.415.506.7360    E–mail: cunderwo@us.oracle.com

If you would like a reply, please give your name, address, and telephone number:

____________________________________

____________________________________

____________________________________

Thank you for helping us improve our documentation.

For Customer Technical Support Services in the United States, call: 1.415.506.1500
For Customer Technical Support Services in Europe, call the Response Centre: +44 1344 860160