Administrating and Supporting ORACLE7 For A/UX - Part 1 (10/93)

Article Created: 28 October 1993

TOPIC ---

This technical information was provided by Oracle. This document was not written by Apple or tested for accuracy, but it's included because we feel you might find it useful.

This is part 1 of 2, of an Oracle technical note on Administrating and Supporting ORACLE7 for A/UX. To find part 2 or 2, search for an article titled, "Administrating and Supporting ORACLE7 For A/UX - Part 2 (10/93)."

DISCUSSION ---

Tips For Administrating and Supporting ORACLE7 For A/UX Desktop
Victor Grigorieff
September 29, 1993 Apple A/UX 3.0.1

Revised: 10/28/93 by Victor Grigorieff

This bulletin explains many of the tools and procedures used in setting up, administrating, and supporting ORACLE7 on A/UX 3.0.1. It is helpful to gather the pieces of information mentioned in this bulletin before trying other methods of diagnosing problems.

This bulletin describes several UNIX commands, but does not provide complete definitions of them. Please refer to the documentation from Apple Computer Inc., or the manual pages on-line for complete definitions of the commands described below.

Manual pages is an option which can be chosen at the time you install the A/UX operating system. If they are installed, you can access the information with the command called man.

Type 'man man' for a description of the man command.

Note: For clarity, some commands are enclosed in single quotes (') with the arguments they accept.

The commands and configuration files of importance are listed below, and are grouped by the type of task to which they are related.

Macintosh Environment & A/UX Management

If you are logged into the console of A/UX 3.0.1 in the Macintosh Finder environment, you there are several things which you need to know.

Command-Control-E If the Macintosh ever appears to freeze (You cannot type or move the mouse), DO NOT RESTART THE MACHINE. You can kill the Macintosh session by holding down Command-Control-E. The Command key is the key which looks like an Apple or a clover. Doing this will return you to the A/UX logon screen, but will not affect any other UNIX processes on the machine, like the Oracle database.

General Controls Under the Apple menu, there is an option for Control Panels. In the Control Panels folder is a General Controls Control Panel, which will let you set the date and time on the A/UX machine.

TextEditor If you need to edit a text file, and prefer the Macintosh interface, you can look in /: Applications for the program called TextEditor. It works like any other word processor on the Macintosh.

/etc/RELEASE_ID This text file tells you the version of A/UX you are running. A/UX 3.0.1 is required for ORACLE7 Server for A/UX.

Type the command 'cat /etc/RELEASE_ID' to list the file:

at /etc/hosts

```
**********************************
* A/UX 3.0.1                     *
* Friday, April 9, 1993          *
**********************************
```

User Management

id displays user and group IDs and names
If you are logged in as the oracle user, you will get output like this:

```
% id
uid=1000(oracle) gid=1000(dba)
```

This means that you are the oracle user, and your active group is dba. By default, the oracle user and dba group own the oracle database installation, meaning that this user can perform privileged operations. You must have dba privileges to

connect internal, and to startup or shutdown a database.

passwd passwd is the command to change the password for a user account. If I am logged in as oracle, and type 'passwd oracle', I will be asked for the current password (if there is one), and will then be allowed to enter a new password.

By default the oracle and root users have no password, so it is important to set passwords for those users.

su su is the command to become another user.

su - oracle

This command prompts you for a password, and then logs you in as the oracle user, while reading in the oracle user's environment. If you leave out the '-' between the su command and the user, you will not read in that user's environment properly. If you are logged in as some other user, and wish to logon as the oracle user, use the above command.

You usually become the oracle user when you want to start/stop the database, and perform routine maintenance.

Network Management

hostname - displays the name of the UNIX host. This is helpful in figuring out how to assemble a sql*net connect string for TCP/IP.

```
% hostname
squid
```

In this case, my hostname is 'squid'.

ping ping is a command which sends simple TCP/IP packets to a host to determine whether or not the host is functioning on the network. If you are having trouble connecting to a host with telnet or SQL*Net TCP/IP (ORA-6108 errors), then you should check to see if ping can reach the host.

** Ping is used to check how the host is responding to TCP/IP
** packets, which are completely different than AppleTalk
** packets. If a client Macintosh can access the server using
** File Sharing, then that means that the network is physically
** intact. It DOES NOT tell you that TCP/IP is functioning
** properly. You should use ping to verify the TCP/IP setup.

ping can be run by typing '/usr/etc/ping some_host' where some_host is the name (or IP address) of a machine on the network. See the description of the /etc/hosts file below for more info.

```
/usr/etc/ping squid
PING squid.us.oracle.com: 56 data bytes
64 bytes from 139.185.81.95: icmp_seq=0. time=0. ms
64 bytes from 139.185.81.95: icmp_seq=1. time=0. ms
----squid.us.oracle.com PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/0
```

Since squid has been properly configured for TCP/IP and is running, TCP/IP packets travel to it and back with no problem. The ping command will keep sending packets until you hit control-C, at which point it gives you the PING Statistics.

If I attempt to ping a machine which is down, the ping command will keep sending packets until the host responds (which may never happen) or until you hit Control-C. If you do not see any lines like '64 bytes from...' after 10 seconds, hit control-C, and you will have output which looks like:

```
/usr/etc/ping bartman
PING bartman.us.oracle.com: 56 data bytes

----bartman.us.oracle.com PING Statistics----
12 packets transmitted, 0 packets received, 100% packet loss
```

In this case, bartman (a UNIX machine) has been turned off, and hence does not respond to the ping command. The 100% packet loss means that the host you are attempting to ping is not working, or is actually at a different address than the one you are trying.

If I am trying to make a database connection to bartman using a database connect string like 'T:bartman:SAMP', I would get the ORA-6108 error, indicating that the TCP/IP connection attempt failed. I could then use the ping command to find out if the machine named bartman is functioning on the network. If bartman does not respond to a ping, action should be taken to revive bartman. In this case, I would go and turn the machine on.

If bartman does respond to a ping, but not a SQL*Net TCP/IP connection attempt, then there could be several different things going on.

* The SQL*Net TCP/IP version 1 listener process (orasrv) may not be running. Check to see if orasrv is active on the machine, and if the /etc/services file has the correct entry for orasrv.

* Turn off the server and attempt to ping the server. This may sound like a silly thing to do, but if you ping a machine that is turned off and it actually responds, you know that

there is another machine using the same IP address. If this happens, you need to talk to your network administrator to determine a new unique address for your server.

telnet — telnet is a way to log into another machine on the network using TCP/IP. It is a useful debugging tool as well as a convenience.

```
telnet squid
Trying...
Connected to squid.us.oracle.com.
Escape character is '^]'.

Apple Computer A/UX (squid)

login:
```

Once you see the login prompt, type the UNIX userid you wish to use and hit return. You will then be asked for a password. Once you have successfully entered the password, you will be working with the remote machine you have logged into, until you type exit.

ftp — ftp is a command you can use to transfer files from one machine to another. If you are transferring anything other than a text file, remember to type the command 'binary' at the 'ftp> prompt.

/etc/hosts — This file lists all the IP addresses and host names for all machines that the A/UX server can connect to using TCP/IP.

```
127.0.0.1     loop local me localhost
139.185.81.95 squid.us.oracle.com squid
```

In this file is the default 127.0.0.1 which is an internal loopback entry, and the definition for this host, which is squid. 127.0.0.1 is a special address. Any machine that attempts to connect to that address will connect to itself.

The hosts file is used when the machine needs to figure out the IP address of a host, when it knows the hostname.

If you try to telnet to a host which is not in the hosts file, you will see:

```
telnet does_not_exist
does_not_exist: unknown host
```

/etc/services — This file should contain an entry for the SQL*Net TCP/IP version 1 server. If you do not have the proper

line in this file, you will get the ORA-6107 error while attempting to use SQL*Net TCP/IP version 1. The line should look like:

orasrv 1525/tcp oracle

/etc/NETADDRS This file lists the TCP/IP network configuration for the UNIX machine.

ao0 139.185.81.95 139.185.81.255 255.255.255.0

ao0 refers to the built-in Ethernet device of the server. 139.185.81.95 is the IP address of the server. 139.185.81.255 is the broadcast address of the server. 255.255.255.0 is the subnet mask for the server.

This file is created when you run the newconfig utility. Checking this file can be helpful to debug TCP/IP connectivity problems and address conflicts.

UNIX Command Management

man man is a program which displays the help and documentation available for a command.

man <some_command>

will show you the description of the command, one page at a time. As you are browsing the description, you hit the space bar to advance a page, and the return key to advance one line.

apropos gives a listing of all commands which have a certain text string in their description. This is useful if you remember what a command does, but not its name.

For example, you know you need to reconfigure your UNIX kernel to set up certain network services. If you forget which command to use, type:

apropos kernel

It will give a listing of commands and descriptions, which have the word 'kernel' in them. from that list, you determine that the command you were looking for was:

newconfig(1M) - generates an up-to-date kernel

After finding a command with apropos, check the man page on the command before executing it. It is a good idea to be aware of all of the options available before running a command.

more more can be called on a file, to display the file a

screen at a time. It can also have the output of a command piped through it to show output a screen at a time.

```
ps -ef | more
```

This will give you a long listing of processes, a page at a time.

| (pipe) The '|' is the UNIX pipe operator. It sends the output of the first command through the second command.

```
some_command | some_other_command
```

some_command will not send its output to the screen, but instead, the output will be read and processed by some_other_command.

grep grep is a text-searching utility that returns all lines that contain the search condition.

```
grep <some_condition> <some_file>
```

This searches <some_file> for <some_condition>, and returns all lines that match the search criterion.

```
% grep squid /etc/hosts
139.185.81.95 squid.us.oracle.com squid
```

This command lists all lines in the /etc/hosts file with 'squid' in them. When combined with the 'hostname' command above, this is a great way to find out what the A/UX machine's IP address is. In this example, 139.185.81.95 is my IP address, which identifies my machine on the TCP/IP network.

wc wc counts characters, words, and lines in a file. It can be very helpful to pipe the output of another program through wc.

```
ps -ef | wc
134   1339   8817
```

'ps -ef' gives a long listing of all processes on the machine, and wc tells you that there are 134 lines in that output. This means that there are 134 processes currently running.

head head <some_file> displays the first 10 lines of a file. This is useful if you just want to peek at the beginning of a file.

'ps -ef | head' lists the first 10 processes.

tail tail is like head, except it returns the last 10 lines.

which which will tell you exactly what function or executable will be executed, given a particular command.

'which cc' (This will tell me '/bin/cc' if the C compiler is
             installed, or tell me 'cc: Command not found' if
             the C compiler is not installed.)

Process & System Management

ps -ef gives a long listing of all processes running on the UNIX machine. There are usually MANY processes running on a UNIX machine, so you usually want to pipe the output through 'more' or 'grep'.

```
ps -ef | more
```

This will show you all of the processes a screen at a time. Hit the space bar to see the next page.

```
ps -ef | grep oracle
```

This will give you a full listing of all processes owned by oracle, or with oracle in their name.

```
% ps -ef | grep oracle
   UID   PID  PPID  C    STIME TTY      TIME COMMAND
oracle  1520  1474  5 17:27:48 qa       0:00 ps -ef
oracle  1521  1474  2 17:27:48 qa       0:00 grep oracle
oracle 20307     1  0  Aug 20  p2       0:00 ora_pmon_SAMP
oracle 20308     1  0  Aug 20  p2       0:01 ora_dbwr_SAMP
oracle 20309     1  0  Aug 20  p2       0:05 ora_lgwr_SAMP
oracle 20310     1  0  Aug 20  p2       0:45 ora_smon_SAMP
oracle 20311     1  0  Aug 20  p2       0:00 ora_reco_SAMP
daemon  1473 21954  4 17:08:27 ?        0:26 oracleSAMP
                                                    T:I,1024,5
oracle  1531  1530  0 17:34:55 qa       0:00 oracleSAMP
                                                    AT:I,0,5
oracle  1474  1456  0 17:08:34 qa       0:01 -csh
```

Note: I pasted the titles of the columns into this document for clarity. The UID is the owner of the process. The COMMAND is the name of the process.

In this example, you see that there are 5 processes with the format ora_XXXX_SAMP. These are the processes that make up the ORACLE database named SAMP. The 'oracleSAMP T:I,1024,5' process is called a TCP/IP shadow process, and means that there is a remote user connected to the SAMP database using SQL*Net TCP/IP. The 'oracleSAMP AT:I,0,5' process is an AppleTalk shadow process, representing a remote connection using SQL*Net AppleTalk.

From time to time, you may have to kill these shadow processes. 'ps -ef | grep oracle' will help you find them.

kill — kill can be used to terminate a process that is no longer needed on your machine.

```
kill <PID>
```

where PID is the process ID of the process you wish to terminate. The PID can be determined by looking at the second column of the ps -ef listing explained above.

For example, as part of the 'ps -ef' listing, I see:

```
daemon  1473 21954  4 17:08:27 ?        0:26 oracleSAMP
                                                    T:I,1024,5
```

If I know that no one is connected to the SAMP database using TCP/IP, I can kill the orphaned shadow process with the command:

```
kill 1473
```

Then the process will be dead, and the database will clean up after the connection. If the process does not die, you can use 'kill -9 <PID>' instead, which will definitely kill the process.

& — The ampersand character, when added onto any other command, will run the program in the background, which means that you could run another command in the current window.

```
atksrv Squid
```

would normally just run the program, and you could not type any other commands (except control-c to stop the program).

```
atksrv Squid &
```

runs the same command in the background, meaning that you can run other programs.

jobs — lists the jobs which are currently in the background.

```
jobs
[1]  + Running                 atksrv MyServer
```

% — This command is used to bring a bring a background process to the foreground. In the above example, there was a background process called 'atksrv MyServer'. Typing % at the prompt would bring the 'atksrv MyServer' process back to the foreground.

If there are multiple processes in the background, you can use '%2' to bring the second process in the list to the foreground.

pstat -m pstat -m lists current system status, including the amount of physical memory which is currently free.

uptime uptime tells you how long the UNIX machine has been running, and what the current load average is on the machine. It also tells you the current time, which lets you know if you need to reset the clock on your server.

```
uptime
7:28am  up 54 days, 13:59,  19 users,  load average: 2.29 2.32
2.20
```

The load average tells you how many processes are actively using the processor. At the time I ran this command, there were 130 processes on the machine, but only two of them were active. If I have 5 remote Macintosh clients actively start using one of the databases, the 5 shadow processes would become active, and the load would jump from about 2 to about 7.

ipcs -a ipcs -m lists the status of all attached blocks of shared memory. An Oracle database uses a block of shared memory to store the SGA (Shared Global Area).

```
ipcs -m
IPC status from /dev/kmem as of Wed Sep 15 07:35:18 1993
T     ID     KEY        MODE       OWNER    GROUP
Shared Memory:
m   2100 0x744c4f57 --rw-------     root     root
m   1501 0x00028d92 --rw-r-----   oracle      dba
m    402 0x00020386 --rw-r-----   oracle      dba
m    203 0x00025a8e --rw-r-----   oracle      dba
m    404 0x0002c003 --rw-r-----   oracle      dba
m    605 0x0002f58f --rw-r-----   oracle      dba
```

The first entry is owned by root, with the root group. This segment is used by the UNIX kernel, and is OK.

Note: the order of the entries in this list does not matter.

The other 5 segments here are blocks which were obtained by Oracle databases on the machine. I know this because the owner is oracle and the group is dba. I can then determine which Oracle databases are running by looking for processes with the letters 'dbwr' in the name, as that is a process which is required by the rdbms.

```
ps -ef | grep dbwr
root   10355 10242  2 07:39:46 pb       0:00 grep dbwr
oracle  4196     1  0  Sep  2  ?        0:02 ora_dbwr_S6A
oracle  5856     1  0  Sep  8  ?        0:02 ora_dbwr_S6C
oracle  4211     1  0  Sep  2  ?        0:02 ora_dbwr_S6B
oracle  4232     1  0  Sep  2  q4       0:02 ora_dbwr_SAMP
```

Looking at the right column, I see 4 processes with the name ora_dbwr_???, where ??? is the name of the database (SID). The databases S6A,S6B,S6C, and SAMP are active on this machine.

I have 4 running databases, and 5 segments of shared memory attached by Oracle databases. This generally means that a database was terminated abnormally (by killing the processes or a severe crash). That last block of shared memory will not be freed until the system is rebooted.

In situations where you have blocks of shared memory that were not freed, you should reboot the system as soon as you can.

Keywords: <None>

==========

19960215 11:05:19.00

Tech Info Library Article Number: 14335