The Oracle Magician

May 2002 *Volume I, number 2*

From the Editor

Welcome to the second issue of the magazine, The *Oracle Magician*! This quarterly newsletter focuses on various "tricks of the trade" in the Oracle world--from DBAs, architects, developers, designers, and report writers.

Thank you to the many notes and suggestions from readers. They are most appreciated!

In this issue is the second part of Brian Keating's series on raw devices. This article explores how to recover from a loss of a raw data "file."

Ideas, suggestion, and especially—*tricks*—are most welcome. Please email the editor with your suggestions.

We are also looking for writers to submit articles that explore interesting ideas of use to DBAs or designers.

Please send all ideas to Chris@OracleMagician.com

Chris Lawson

Editor

Recovery with Raw Devices

By Brian Keating

Editor's Note:

This concludes the series on using raw devices with Oracle. In the previous article, Brian provided an overview of raw devices. In this article, he explains how to recover from raw device loss.

Recovering Raw Devices

How you recover a raw device data "file" depends upon what has gotten deleted / corrupted - the logical volume itself or the character special device file that points to it. If just the character special device file has been deleted, then the data in the logical volume is OK - all you have to do is recreate the device file.

However, you cannot recreate the device file by copying the backup of that file into /dev, because as mentioned previously, the backup file is a normal UNIX file, not a character device file - and a character special device file is needed to access the logical volume.

Recreating a Character Special File

There are two ways to recreate the character special file for a logical volume. One way is to use the *mknod* command to manually recreate the device file. To use that command, though, you need to know the major and minor device numbers of the logical volume. If the block special device file for that logical volume still exists, you can find out the major and minor numbers of the device by looking at that file, since both the character and block special device files always have the same major and minor numbers (since they both point to the same device).

So, for example, let's say that an Oracle data file is built on the logical volume "testdb_admin" (which means that the name of the data file is "/dev/rtestdb_admin"), and the file /dev/rtestdb_admin gets accidentally

Continued on page 2

Table of Contents

From the Editor ..... Page 1
Using Oracle Raw Device Data Files (Part 2) ..... Page 1
Update: IOUG LIVE! 2002 ..... Page 4

Continued from page 1

deleted. The data in the logical volume is still OK - all you have to do is recreate rtestdb_admin.

The way to do this is to do a long listing of the block special device file for that logical volume to find out the device's major and minor numbers, then to use the *mknod* command to recreate the character special device file, and finally to *chown* the device file to be owned by oracle:dba. Note that the *mknod* command can only be run by root or a member of the system group.

"If both device files have been deleted, use LVM commands to rename the logical volume."

For example:

 # cd /dev

Observe the files, as shown in Figure 1.

 # mknod rtestdb_admin c 43 5

 # chown oracle:dba rtestdb_admin

Confirm the changes (see Figure 2).

If both the block and character special device files of a logical volume have been deleted, it is still possible to figure out what the major and minor numbers of the device are, but it is much more difficult (you have to dig them out of the UNIX kernel - and that is not a task for the faint of heart). Even if both of those files are deleted, though, it is still possible to access the logical volume through LVM (logical volume manager) commands. This leads us to the *second* way to recreate the character special file.

Renaming Logical Volume using LVM

If both device files have been deleted, use LVM commands to rename the logical volume in question to some other name. When you rename a logical volume, the block and character device files get recreated to reflect the new logical volume name automatically, and the new device files obviously have the correct major and minor numbers.

So, after you rename the logical volume to a new name, you can then rename it back to its original name, and after that both of the logical volume's device files will again be present. You can even use this solution if only one of the device files has been deleted. You still must remember to chown the device files to be owned by oracle:dba with this solution, though.

For example, let's say both the block and character special device files of the testdb_admin logical volume (from the

Continued on page 3

Figure 1

```
# ls -l testdb_admin

brw-rw----   1 oracle   dba   43,  5   May 21 15:50 testdb_admin
```

Figure 2

```
# ls -l rtestdb_admin

crw-rw----   1 oracle   dba   43,  5   May 26 13:56 rtestdb_admin
```

Continued from page 2

example above) have gotten accidentally deleted. Since both of the device files are gone, there is no easy way to determine what the major and minor numbers of the device are. So, what you would do is as follows:

Step 1. Shut down the Oracle instance that uses */dev/rtestdb_admin* as a data file (you will have to do a shutdown abort).

Step 2. Rename the *testdb_admin* logical volume to another name that is not used by any other logical volume (in this example I'll use foo_bar) with the following command:

 # chlv -n 'foo_bar' testdb_admin

This will cause two new device files, *foo_bar* and *rfoo_bar*, to be created with the correct major and minor device numbers, as shown in Figure 3.

Step 3. You can then rename the *foo_bar* logical volume back to its original name of *testdb_admin*:

 # chlv -n 'testdb_admin' foo_bar

Step 4. That will rename the device files back *to testdb_admin* and *rtestdb_admin*. You must remember to chown them to be owned by oracle:dba, though.

 # chown oracle:dba *testdb_admin

Check the files, as shown in Figure 4.

5. Restart the Oracle instance.

If Logical Volume Lost

If a logical volume itself gets destroyed or corrupted (for example, if you lose the disk that the logical volume lives on), and if that logical volume has a data file built on top of it, then the commands to recover that data file are as follows (for this example, we'll assume that the testdb_admin logical volume has been destroyed):

Step 1. Recreate the testdb_admin logical volume. The command to do this will vary, of course, depending on the volume group that the logical volume is in, the number of logical partitions it has, etc., but in this example the command is as follows:

 # mklv -y 'testdb_admin' -t 'raw' testdbvg 13

Continued on page 4

> "Since both of the device files are gone, there is no easy way to determine what the major and minor numbers of the device are."

Figure 3

```
# ls -l *foo_bar

brw-rw---- 1 root   system 43,  5 May 26 14:34 foo_bar
crw-rw---- 1 root   system 43,  5 May 26 14:34 rfoo_bar
```

Figure 4

```
# ls -l *testdb_admin

crw-rw----  1 oracle  dba  43, 5 May 26 14:35 rtestdb_admin
brw-rw----  1 oracle  dba  43, 5 May 26 14:35 testdb_admin
```

Continued from page 3

Step 2. *Chown* the special device files of the *testdb_admin* logical volume:

```
# cd /dev

# chown oracle:dba *testdb_admin

# ls -l *testdb_admin

crw-rw----   1 oracle   dba    43,  5 May 26 14:57 rtestdb_admin
brw-rw----   1 oracle   dba    43,  5 May 26 14:58 testdb_admin
```

Step 3. Copy the most recent backup of the *rtestdb_admin* data file onto the character special device file that points to the *testdb_admin* logical volume. Note: when you copy a normal UNIX file onto another already existing normal UNIX file, the contents of the second file get overwritten with the contents of the first file.

That is not what happens when you copy a normal UNIX file onto an existing character device file, though - what happens in that case is the data in the normal file gets read into the device that the character device file points to.

Note: UNIX operating systems store information about logical volumes at the beginning of each logical volume; for example, AIX stores information about each logical volume in the first 4096 bytes. So, when you run the dd command to copy the backup file, you must ensure that you do not overwrite that information; you must skip the area that contains that information with the "skip" argument in dd. If you don't do this, the entire logical volume may become completely unusable. Check with your system administrator to determine the location of logical volume information on other flavors of UNIX.

In this example, I have a backup of the rtestdb_admin data file in /apps/oracle:

```
$ dd if=/apps/oracle/rtestdb_admin
of=/dev/rtestdb_admin bs=4096 skip=1
```

Depending on the size of the data file in question, this command may take a while to complete – in my tests, a 200-megabyte data file took about 6 minutes to copy into the appropriate logical volume.

Step 4. The *testdb_admin* logical volume will then have whatever data was in the rtestdb_admin data file at the point in time that the backup was taken. Of course, you may have to apply archived redo logs to this data file if it is on a different system change number than the other data files in the database, but the procedures involved in applying archived redo logs are the same whether you are using cooked files or raw devices for your data files.

Brian Keating is a Senior Oracle DBA with Database Specialists of San Francisco. Brian can be reached at Briankeating@Juno.com.

IOUG LIVE! 2002 Update

Chris Lawson presented two sessions at IOUG-Live:

- *Don't Shutdown that Database! Use Oracle 9i Online Object Redefinition Instead* (with Roger Schrag)
- *Don't Settle for Mere Competence. Be the Magician!*

Both presentations are available on the OracleMagician.com web site.

The Oracle Magician

Editor: Chris Lawson

Contributing Editor:
Brian Keating

Layout: Wizard Press