CHAPTER 14: MISCELLANEOUS

by Volker Herminghaus

14.1 Disk Flags

There are several auxiliary flags that VxVM uses to maintain a disk's status or preferential use. All of them can be set and reset by the administrator, and some are maintained automatically. This section will discuss the flags, show you their intended (and unintended) uses, and how to set and reset the flags.

If you look at the following output, you will see some unusual states noted on the right-hand side, as well as two "broken" disk media, as you know them from the troubleshooting chapter. In fact, the latter two disks (adg01 and adg06) were simply offlined while deported, then the disk group was re-imported and thus the error triggered.

```
# vxdisk list
DEVICE       TYPE            DISK         GROUP        STATUS
[…]
c0t2d0s2     auto            -            -           offline
c0t3d0s2     auto:cdsdisk    adg02        adg          online reserved
c0t4d0s2     auto:cdsdisk    adg03        adg          online failing
c0t10d0s2    auto:cdsdisk    adg04        adg          online nohotuse
c0t11d0s2    auto:cdsdisk    adg05        adg          online spare
c0t12d0s2    auto            -            -           offline
-            -         adg01        adg          removed was:c0t2d0s2
-            -         adg06        adg          failed was:c0t12d0s2
```

V. Herminghaus and A. Sriba, *Storage Management in Data Centers*,
DOI: 10.1007/978-3-540-85023-6_14, © Springer-Verlag Berlin Heidelberg 2009

OK, let's sum up what we see, then make some sense of it and find out how these flags are set and reset. The following flags are set on one or more of the disks:

offline, online, reserved, failing, nohotuse, spare, removed, failed

Let's go through them and discuss them one by one:

OFFLINE/ONLINE

This flag is set by the administrator and used as a general software ON/OFF-switch for the disk medium. The administrator can simulate switching off a disk drive using this flag. The command to offline a disk is the following:

```
# vxdisk offline $ACCESSNAME
```

Remember the accessname is the name in the left most column of the **vxdisk list** output. Example:

```
# vxdisk offline c0t2d0
```

You can reverse the effect of this command by the using the opposing command to online the disk again:

```
# vxdisk online $ACCESSNAME
```

Example:

```
# vxdisk online c0t2d0
```

As long as a disk is in the offline state, it behaves just like a disk that is physically offline. I.e. the private region contents cannot be read and therefore the output of vxdisk list $ACCESSNAME is greatly reduced. Compare the outputs of the same command: vxdisk list c0t12d0, on an online and an offline disk:

```
# vxdisk online c0t12d0
# vxdisk list c0t12d0s2
Device:    c0t12d0s2
devicetag: c0t12d0
type:      auto
hostid:
disk:      name= id=1211724452.55.infra0
group:     name=adg id=1211817259.120.infra0
info:      format=cdsdisk,privoffset=256,pubslice=2,privslice=2
flags:     online ready private autoconfig
pubpaths:  block=/dev/vx/dmp/c0t12d0s2 char=/dev/vx/rdmp/c0t12d0s2
guid:      {ea26335e-1dd1-11b2-8dfd-080020c28592}
```

```
udid:      IBM%5FDNES-318350Y%5FDISKS%5F%20%20%20%20%20%20%20%20AKFQ6556
site:      -
version:   3.1
iosize:    min=512 (bytes) max=2048 (blocks)
public:    slice=2 offset=2304 len=35834496 disk_offset=0
private:   slice=2 offset=256 len=2048 disk_offset=0
update:    time=1216851961 seqno=0.52
ssb:       actual_seqno=0.1
headers:   0 240
configs:   count=1 len=1280
logs:      count=1 len=192
Defined regions:
 config   priv 000048-000239[000192]: copy=01 offset=000000 enabled
 config   priv 000256-001343[001088]: copy=01 offset=000192 enabled
 log      priv 001344-001535[000192]: copy=01 offset=000000 enabled
 lockrgn  priv 001536-001679[000144]: part=00 offset=000000
Multipathing information:
numpaths:   1
c0t12d0s2       state=enabled

# vxdisk offline c0t12d0s2
# vxdisk list c0t12d0s2
Device:    c0t12d0s2
devicetag: c0t12d0
type:      auto
flags:     offline private autoconfig
pubpaths:  block=/dev/vx/dmp/c0t12d0s2 char=/dev/vx/rdmp/c0t12d0s2
guid:      {ea26335e-1dd1-11b2-8dfd-080020c28592}
udid:      IBM%5FDNES-318350Y%5FDISKS%5F%20%20%20%20%20%20%20%20AKFQ6556
site:      -
Multipathing information:
numpaths:   1
c0t12d0s2       state=enabled
```

What you see is that all the information that is obtained by reading the private region of the disk is lost when the disk is offlined. In effect, the offlined disk looks much like a disk that is physically accessible, but does not have a private region. So this is how you turn disks on and off in software (after all, now they are virtualized disks, so you should be able to, shouldn't you?).

RESERVED

The disk with the accessname **c0t3d0s2 (adg02)** carries the flag **reserved**. This flag is set by the administrator to keep VxVM from automatically allocating any storage from those disks that carry the flag. The command to set the **reserved** flag, or indeed any of the flags **reserved, failing, nohotuse**, or **spare**, is the following:

```
# vxedit -g $DG set $FLAG=on $Disk
```

Example:

```
# vxedit -g adg set reserved=on adg02
```

The result of setting the **reserved** flag is that if you do not specify this particular disk when creating, mirroring, or expanding volumes or when doing other actions that allocate storage (like moving a volume around, evacuating a disk or relayouting a volume), then it will not be used for storage allocation. However, as soon as you specify the disk using the "alloc=..." parameter it is again eligible for storage allocation.

SPARE

The disk with the accessname **c0t11d0s2 (adg05)** carries the flag **spare**. This flag is used to keep VxVM from automatically allocating any storage from those disks that carry the flag, just like the **reserved** flag we just discussed.

The difference between **reserved** and **spare** is that a disk flagged as **spare** will be preferred by VxVM's hot relocation functionality: Subdisks on a failed disks will be relocated to free disk space automatically. This process prefers disks flagged as **spare**, and excludes those marked with the **nohotuse** or **reserved** flag. Disks marked as **spare** will not be used for storage allocation unless they are specified using the "alloc=..." parameter.

As you will see below, the spare disk does not even show up in the free region list obtained by the **vxdg free** command. Remember the list of disks in the disk group looks like this:

```
# vxdisk list
DEVICE       TYPE            DISK         GROUP        STATUS
[…]
c0t2d0s2     auto            -            -            offline
c0t3d0s2     auto:cdsdisk    adg02        adg          online reserved
c0t4d0s2     auto:cdsdisk    adg03        adg          online failing
c0t10d0s2    auto:cdsdisk    adg04        adg          online nohotuse
c0t11d0s2    auto:cdsdisk    adg05        adg          online spare
c0t12d0s2    auto            -            -            offline
-            -         adg01        adg          removed was:c0t2d0s2
-            -         adg06        adg          failed was:c0t12d0s2
```

Not let's get the list of free extents (or regions):

```
# vxdg free
GROUP        DISK         DEVICE       TAG          OFFSET    LENGTH    FLAGS
adg          adg02        c0t3d0s2     c0t3d0       0         17702192  r
adg          adg04        c0t10d0s2    c0t10d0      0         17679776  n
```

Not let's get the list of free extents including the spare disk. We do so by passing the **-a** (all) switch to the **vxdg free** command.

```
# vxdg -a free
GROUP        DISK         DEVICE       TAG          OFFSET    LENGTH    FLAGS
adg          adg02        c0t3d0s2     c0t3d0       0         17702192  r
adg          adg04        c0t10d0s2    c0t10d0      0         17679776  n
adg          adg05        c0t11d0s2    c0t11d0      0         8378496   s
```

As you can see, specifying the **-a** flag allows the free region from the spare disk to show up. In theory, the same would happen if you specified the **-r** switch to **vxassist**. This is supposed (and documented in the **vxassist** man page) to override the spare flag and therefore allow allocation of free space on spare disks. However, it does not work in version 5.0 (Solaris) as the following example, based on the same set of disks and flags, shows:

```
# vxassist maxsize # Will use only adg04 as all others are flagged
Maximum volume size: 17678336 (8632Mb)
# vxassist -r maxsize # Will also use only adg04, which is a bug!
Maximum volume size: 17678336 (8632Mb)
```

The above shows that neither the **spare (adg05)** disk nor the **reserved (adg02)** disk has been added to the eligible storage pool by using the **-r** flag. The workaround for this bug is to explicitly specify the disk by name for vxassist allocation. This will override the spare flag and well as the reserved flag (but not the failing flag), yielding the desired result:

```
# vxassist -r maxsize alloc=adg04,adg02
Maximum volume size: 35381248 (17276Mb)
```

Passing specific disks on the command line overrides the flags previously set by the administrator. It does not, however, override another flag, which is usually set by the VxVM kernel (although it could be set manually, too): the failing flag

FAILING

If I/O to a subdisk leads to an unrecoverable read- or write-error, then the VxVM kernel automatically sets the failing-flag for the VM disk that the subdisk resides on. This flag by itself does not trigger any action on behalf of VxVM (but the underlying I/O error does trigger the relocation process unless the default has been changed), and it can indeed be set or reset by the administrator at will using the following command:

```
# vxedit -g $DG set failing=on $Disk
```

Example:

```
# vxedit -g adg set failing=on adg03
```

A failing disk will continue to be used as long as there are subdisks on it. The relocation process does try to move all subdisks away from a failing disk, but it may fail to allocate sufficient alternative storage. In that case, and while the relocation is in progress, I/O is

still done to as well as from the failing disk.

The failing-flag can only be reset by specifying **failing=off** instead of **failing=on**. Because the VxVM flags are persisted onto the private region, the usual attempts by less experienced administrators (rebooting, running **vxdctl enable**, etc) do not bear fruit.

A disk that carries the **failing**-flag is not used for new subdisk allocation or subdisk extension by VxVM. There is also no way to force **vxassist** to allocate subdisk space from a disk marked as failing. It is possible using **vxmake sd**, but that is not of general interest, since the manual allocation of individual subdisks is not widely used and skips all reasonable tests (like excluding the use of failing disks :-).

If you are certain that a disk is erroneously labeled **failing**, then you can reset the flag using the command outlined above and use it. In the days of SAN-based LUNs seeing failing disks has become a rather seldom event. Disks are either present and reliable, or missing and accordingly appear as **failed** disks in the **vxdisk list** output.

FAILED

Volume Manager marks disks failed when the private region cannot be accessed. This is in contrast to failing disks, which is the state of a disk when only subdisk data cannot be accessed. Failure of I/O to the private region is more critical because VxVM cannot persist any meta data onto a disk in that case.

Because user data is meaningless without meta data pointing to it, the loss of the private region is considered a final blow to the disk, and it is therefore considered completely unusable. It does not appear in the main portion of the **vxdisk list** output any more, where all imported disk media are usually displayed next to their access names. Instead, it is listed as a record with no associated access record at the end of the **vxdisk list** output (see **adg06 in the last line** below).

```
# vxdisk list
DEVICE       TYPE            DISK         GROUP        STATUS
[…]
c0t2d0s2     auto            -            -            offline
c0t3d0s2     auto:cdsdisk    adg02        adg          online reserved
c0t4d0s2     auto:cdsdisk    adg03        adg          online failing
c0t10d0s2    auto:cdsdisk    adg04        adg          online nohotuse
c0t11d0s2    auto:cdsdisk    adg05        adg          online spare
c0t12d0s2    auto            -            -            offline
-            -         adg01        adg          removed was:c0t2d0s2
-            -         adg06        adg          failed was:c0t12d0s2
```

NOHOTUSE

A disk is never flagged **nohotuse** automatically, by VxVM itself. This option is meant exclusively for the administrator in order to exclude disks from serving as a target for hot relocation. If you want to prevent hot relocation altogether it is not the best solution to flag all of the disks **nohotuse**. Rather, it is preferable to keep the relocation daemon from starting up in the boot process. This saves computer resources and serves the same purpose.

REMOVED

This last flag is set by VxVM when a disk is removed from a Disk group even if there are still subdisks allocated from its public region. Information about these subdisks must not get lost when the disk is removed, and so the Disk group keeps the disk record inside the disk group, but flags the disk as **removed** in the private region data base.

```
# vxdisk list
DEVICE       TYPE            DISK         GROUP        STATUS
[…]
c0t2d0s2     auto            -            -            offline
c0t3d0s2     auto:cdsdisk    adg02        adg          online reserved
c0t4d0s2     auto:cdsdisk    adg03        adg          online failing
c0t10d0s2    auto:cdsdisk    adg04        adg          online nohotuse
c0t11d0s2    auto:cdsdisk    adg05        adg          online spare
c0t12d0s2    auto            -            -            offline
-            -         adg01        adg          removed was:c0t2d0s2
-            -         adg06        adg          failed was:c0t12d0s2
```

14.1.1 Summary

In this chapter we talked about states that can be flagged on a disk, like being reserved for, or excluded from hot relocation, becoming defective, being defective, or being removed from the disk group. We also learned that you can switch a disk on and off in software. But the flags themselves are only part of the game: It's how the VxVM state machine operates on them that makes understanding them worthwhile. For instance, you could temporarily exclude a number of disks from allocation simply by setting their **failing** or **reserved** flag, then resetting it later, after the allocation has been done.

As is the case with many UNIX software systems, VxVM is a very universal tool set that can be used as intended. But its mechanisms can also be used in ways the inventors may not have thought of. As long as you stay within reasonable limits, everything should still work fine.