

Table of Contents

| | |
|--|----------|
| Creating Veritas volumes and Oracle tablespaces on PDB01..... | 1 |
| Step A: Create new Veritas volumes..... | 1 |
| Create new Veritas volumes using the VMSA..... | 1 |
| Change the O/S ownership of the new Veritas volumes..... | 3 |
| Step B: Create new Oracle tablespaces..... | 4 |
| Create the Oracle tablespaces..... | 4 |
| Create the Oracle users..... | 4 |

Creating Veritas volumes and Oracle tablespaces on PDB01

The following is a simple tutorial on the steps to follow to create dedicated tablespaces for specific users on PDB01. Two basic steps are needed: create a Veritas volume first, then define an Oracle tablespace using that volume.

The tutorial will cover the real-life example where Veritas volumes and dedicated tablespaces for the `CMS_TRANSFERMGMT` user need to be created. Two mirrored Veritas volumes of 2GB each will be created with names `pdb01_CMS_TRANSFERMGMT_DATA01` and `pdb01_CMS_TRANSFERMGMT_INDX01`. The two volumes will then be used, respectively, as datafiles for the two dedicated index and data tablespaces

`CMS_TRANSFERMGMT_DATA01` and `CMS_TRANSFERMGMT_INDX01` for the `CMS_TRANSFERMGMT` user.

More details about Veritas volumes on the PDB cluster can be found in the previous section on Veritas Volume Manager and T3 Disk Arrays on the PDB Cluster.

Step A: Create new Veritas volumes

The following operations should all be performed being logged in as `root` on the Master node of the Veritas Cluster Volume Manager (CVM) setup. Use the `vxctl -c mode` command to find out which node is the Master (as explained more in detail in the previous section).

Create new Veritas volumes using the VMSA

Launch the graphical Volume Manager Storage Administrator (VMSA) as `root` on the Master node of the Veritas CVM:

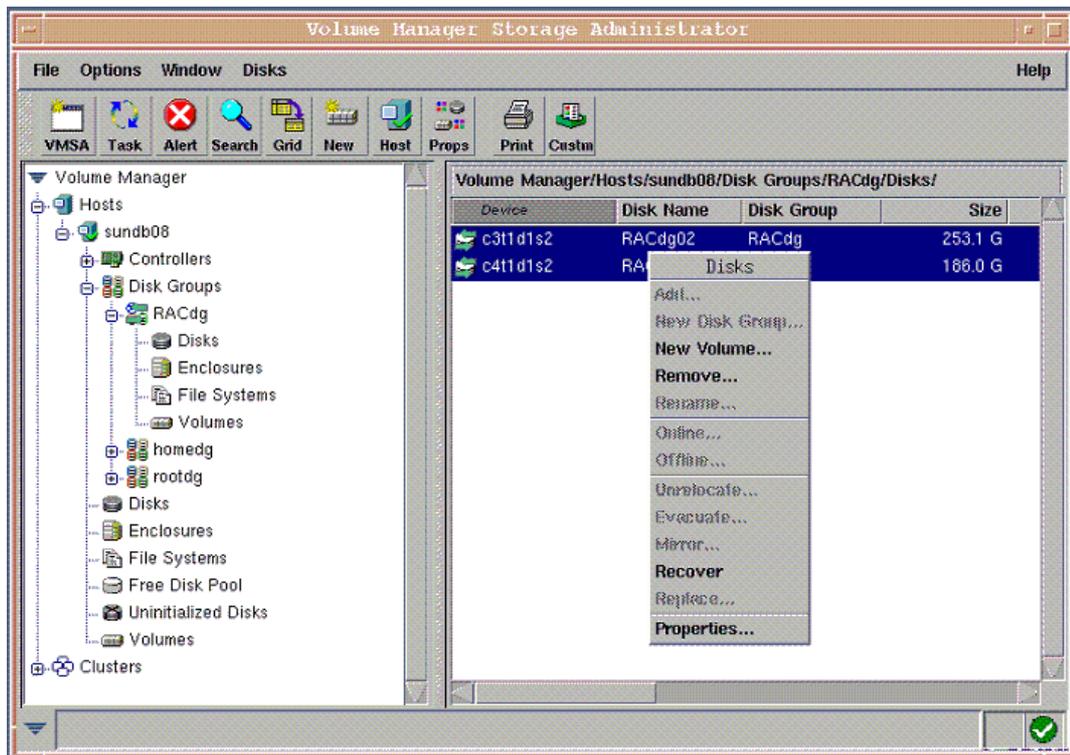
```
vmsa &
```

After inserting your root password for the Master node, this will open up the GUI with a button menu on the top, a navigator window on the left and another window on the right.

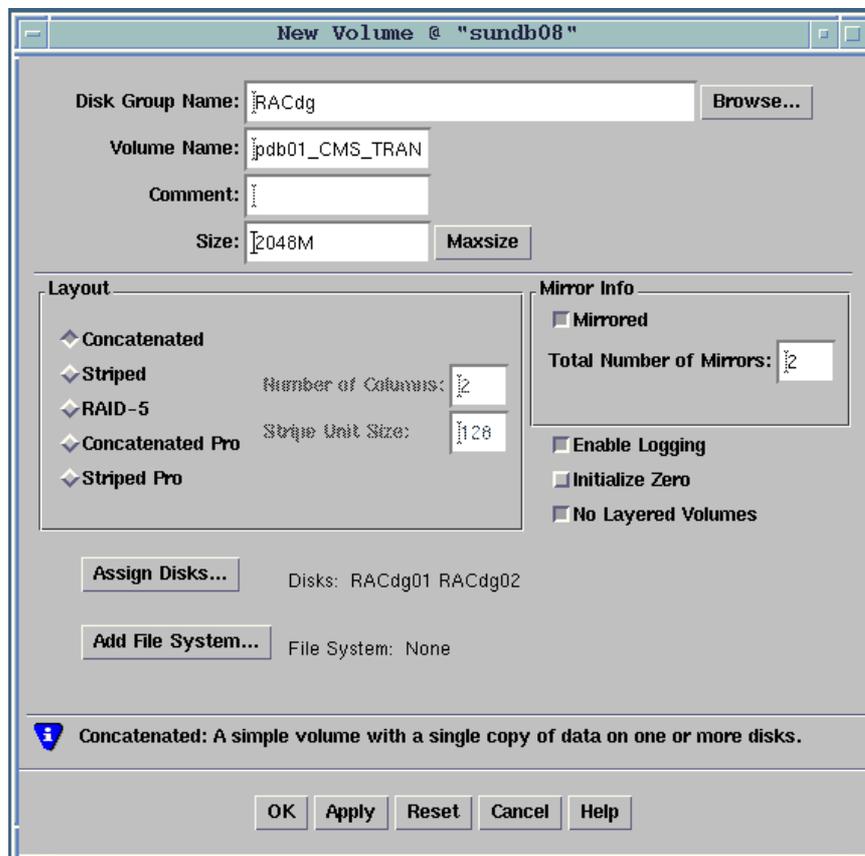
The simplest way to create volumes mirrored across the two T3 arrays is the following. In the navigator window, click on the "Disks" tab for the `RACdg` disk group. In the window on the right, this will display the two disks

`RACdg01` and `RACdg02` associated to this disk group. Using the Shift button, select both disks. Right-click to display the following menu, and select "New volume":

CreatePDB01Tablespace < PSSGroup < TWiki



Enter "pdb01_CMS_TRANSFERMGMT_DATA01" as volume name and "2048M" as size. Keep the default "Concatenated" layout selected and select the "Mirrored" configuration with the number of mirrors equal to the default value of 2. Keep all other default values ("Enable logging" and "No layered volumes" selected, "Initialize zero" not selected, disks RACdg01 and RACdg02, no file system), as in the following snapshot.



Click on "Apply" to create the new volume while keeping the window open. Modify the volume name to "pdb01_CMS_TRANSFERMGMT_INDX01", keeping all other selections as they are, and click on "Ok".

CreatePDB01Tablespace < PSSGroup < TWiki

This will also create the second new volume, and exit the window.

New volumes could also be created from the shell without using VMSA. The commands that are executed by the VMSA and that you could instead enter using the shell can be found by clicking on the "Tasks" button in the main GUI. In a new window, this will display a list of two tasks: their status should evolve from "Executing" to "Successful" while the window is open, unless there are any problems. Double-clicking on each task will display the task "Properties" in a new window, including the commands executed. In our example, the commands executed are the following:

```
/usr/sbin/vxassist -g RACdg make pdb01_CMS_TRANSFERMGMT_DATA01 2048M \  
  layout=mirror-concat,log nmirror=2 alloc="RACdg01 RACdg02"  
/usr/sbin/vxassist -g RACdg make pdb01_CMS_TRANSFERMGMT_INDX01 2048M \  
  layout=mirror-concat,log nmirror=2 alloc="RACdg01 RACdg02"
```

After checking that the two new volumes have appeared in the "Volumes" tab of the RACdg disk group, you can exit the VMSA graphical interface.

Change the O/S ownership of the new Veritas volumes

By default, the two volumes are created by the `root` user, that becomes their owner. In order for Oracle to be able to use them to define tablespaces, the `orapdm` user must own them instead.

Use the following commands to check that the two new volumes appear in the device tree, and to display their current owner:

```
ls -la /dev/vx/rdisk/RACdg/pdb01_CMS_TR*
```

returns the two character devices

```
crw----- 1 root root 86,30036 Feb 5 14:55 /dev/vx/rdisk/RACdg/pdb01_CMS_TRANSF  
crw----- 1 root root 86,30037 Feb 5 14:58 /dev/vx/rdisk/RACdg/pdb01_CMS_TRANSF
```

while

```
ls -la /dev/vx/dsk/RACdg/pdb01_CMS_TR*
```

returns the two block devices (i.e. those that can mounted to a mount directory),

```
brw----- 1 root root 86,30036 Feb 5 14:55 /dev/vx/dsk/RACdg/pdb01_CMS_TRANSF  
brw----- 1 root root 86,30037 Feb 5 14:58 /dev/vx/dsk/RACdg/pdb01_CMS_TRANSF
```

Use the following commands to change the ownership of the two volumes from user `root` to user `orapdm`:

```
/usr/sbin/vxedit -v set user=orapdm group=dba pdb01_CMS_TRANSFERMGMT_DATA01  
/usr/sbin/vxedit -v set user=orapdm group=dba pdb01_CMS_TRANSFERMGMT_INDX01
```

Use the previous commands to verify that the ownership was changed:

```
ls -la /dev/vx/rdisk/RACdg/pdb01_CMS_TR*
```

returns

```
crw----- 1 orapdm dba 86,30036 Feb 5 14:55 /dev/vx/rdisk/RACdg/pdb01_C  
crw----- 1 orapdm dba 86,30037 Feb 5 14:58 /dev/vx/rdisk/RACdg/pdb01_C
```

while

CreatePDB01Tablespace < PSSGroup < TWiki

```
ls -la /dev/vx/dsk/RACdg/pdb01_CMS_TR*
```

returns

```
brw-----      1 orapdm      dba              86,30036 Feb  5 14:55 /dev/vx/dsk/RACdg/pdb01_CM
brw-----      1 orapdm      dba              86,30037 Feb  5 14:58 /dev/vx/dsk/RACdg/pdb01_CM
```

Step B: Create new Oracle tablespaces

You can now create the CMS_TRANSFERMGMT_DATA01

and CMS_TRANSFERMGMT_INDX01 Oracle tablespaces for the CMS_TRANSFERMGMT user, using the two new Veritas volumes as datafiles.

Create the Oracle tablespaces

Connect to PDB01 as SYS AS SYSDBA and issue the following commands:

```
CREATE TABLESPACE CMS_TRANSFERMGMT_DATA01 LOGGING
DATAFILE '/dev/vx/rdisk/RACdg/pdb01_CMS_TRANSFERMGMT_DATA01'
SIZE 2047M REUSE
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;
```

```
CREATE TABLESPACE CMS_TRANSFERMGMT_INDX01 LOGGING
DATAFILE '/dev/vx/rdisk/RACdg/pdb01_CMS_TRANSFERMGMT_INDX01'
SIZE 2047M REUSE
EXTENT MANAGEMENT LOCAL AUTOALLOCATE
SEGMENT SPACE MANAGEMENT AUTO;
```

This may take a few minutes (the larger the tablespace, the longer it will take).

Note that Veritas volumes of size 2048 MB were created, but the Oracle tablespaces must be created of lower sizes 2047 MB. This is because Oracle needs a little overhead inside the datafile for its own purposes, that is not available as storage space in the tablespace. If you attempt to create tablespaces of the same size as that of the Veritas volumes (2048 MB in this case), you will get an error message such as:

```
ORA-01119: error in creating database file '/dev/vx/rdisk/RACdg/pdb01_CMS_TRANSFERMGMT_DATA01'
ORA-27042: not enough space on raw partition to fullfill request
Additional information: 2
```

Conversely note also that, if tablespaces of sizes smaller than the available Veritas volume size were created, the only way to make sensible use of the remaining space on the Veritas volumes would be to define the tablespaces with the AUTOEXTEND option.

Create the Oracle users

You may now create the CMS_TRANSFERMGMT user, with quota unlimited on its dedicated tablespaces, and default data tablespace its dedicated data tablespace. Connect to PDB01 as SYS AS SYSDBA

and issue the following commands:

```
create user CMS_TRANSFERMGMT identified by timbarrass
default tablespace CMS_TRANSFERMGMT_DATA01
temporary tablespace TEMP
quota unlimited on CMS_TRANSFERMGMT_DATA01
quota unlimited on CMS_TRANSFERMGMT_INDX01;
```

CreatePDB01Tablespace < PSSGroup < TWiki

```
grant designer to CMS_TRANSFERMGMT;
```

Finally, for a large fraction of projects, including this one, you may also find it useful to create "writer" and "reader" accounts. Typically, these accounts would have only very little quota on the general purpose tablespaces (for instance, to create a `PLAN_TABLE` for optimization studies):

```
create user CMS_TRANSFERMGMT_WRITER identified by timbarrass
default tablespace DATA01
temporary tablespace TEMP
quota 10M on DATA01;
```

```
grant designer to CMS_TRANSFERMGMT_WRITER;
```

```
create user CMS_TRANSFERMGMT_READER identified by timbarrass
default tablespace DATA01
temporary tablespace TEMP
quota 10M on DATA01;
```

```
grant designer to CMS_TRANSFERMGMT_READER;
```

It would then be *the responsibility of the user of the main account* to grant these accounts the privileges to execute DML/SELECT (but no DDL) and SELECT statements respectively on its own schema. This can only be done *after the schema has been created*, because it is impossible to grant such privileges on ALL tables of any one user.

(Page compiled by Andrea Valassi on 2004-02-04 from invaluable conversations with Magnus Lubeck)

This topic: PSSGroup > CreatePDB01Tablespace

Topic revision: r1 - 2005-12-07 - unknown



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback