

Table of Contents

- HDD/SSD Validation** 1
- SMART Tests: A Short Introduction** 1
- HDD/SSD Validation Via the FreeNAS OS Route** 2
- Conducting The HDD/SDD Validation Tests (Finally!) In FreeNAS Via An SSH Console** 2
- SMART Short Tests Via The SSH Console 2
- SMART Conveyance Tests Via The SSH Console 5
- SMART Long Tests Via The SSH Console 7
- Badblocks Tests Via The SSH Console 9
- FreeNAS Storage Volumes And A Known Problem With badblocks 9
- Detaching The Storage Volume Before A Badblocks Test (DESTRUCTIVE Method) 10
- Detaching The Storage Volume Before A Badblocks Test (Non-Destructive Method) 12
- Importing An Unencrypted Volume After A Non-Destructive Badblocks Test 12
- Importing An Encrypted Volume After A Non-Destructive Badblocks Test 13
- Destructive Badblocks Test Using tmux 15
- Non-Destructive Badblocks Test Using tmux 23
- Stopping A Badblocks Test In tmux 24
- Detaching a tmux Session 24
- Resuming A Session In tmux 25
- Getting Your Test Results** 27
- Making Sense of SMART Data 30

HDD/SSD Validation

HDD validation (in this case) basically involves 5 stages.

1. A SMART short test. This is a test that looks at certain aspects of the electrical and mechanical performance of the HDD. It is not a thorough test of the HDD. The tests take somewhere in the region of 2-5 minutes to complete.
2. A SMART conveyance test. This is a test performed on HDDs to check if they have survived transit without any damage. (I don't know how they differ from the short or long test if someone wants to give me the information Fester will try to add it.)
3. A SMART long test. Think of this as an extended version of the short test. It is a much more thorough test of the HDD and will include a surface scan of the drive. This test will take many hours to complete depending on the capacity of the HDD.
4. A Badblocks test. This is a test where every physical location on the HDD has a write/read test performed on it. The test consists of a block of data that gets written to every physical location on the HDD in sequence. Every physical location on the HDD is then read back also in sequence and each time at each location the value is checked to see if it is correct. This is one pass. The whole process if repeated with a different block of data, this is the second pass. The badblocks test uses 4 patterns by default. This test will take a very long time, usually between 24 hours to a few days depending on the capacity of the drive.
5. The SMART long test is repeated.

SMART Tests: A Short Introduction

SMART stands for Self-Monitoring Analysis and Reporting Technology.

A SMART test is a test a HDD or SDD can perform by itself on itself. These tests, often referred to as "self tests" are carried out by the HDD's/SDD's onboard firmware, not a separate piece of software running on the server as we have already seen.

The results of these tests are stored in the drives onboard non-volatile memory so they can be retrieved and utilised by simply interrogating the drive in the correct way.

However, to be able to use the SMART capabilities built into the drives we need a program or an OS that is capable of communicating with the built in SMART functions of the drive.

With such a program or OS present we can simply issue commands to invoke the firmware to initiate a SMART test and/or interrogate a SMART drive to obtain the results of that test (very convenient).

Only one SMART test can be performed per drive. So you cannot run the short test and the long test on the same drive simultaneously. Also the current SMART test must complete before another can be run on the same drive. If a SMART test is running on a drive and you start another then the current test is stopped and abandoned in favour of the newly requested test.

Fortunately, you can run SMART tests in parallel on different drives. So you could have any number of

drives all performing the short test at the same time, or the long test or a mix if you wish (i.e. some performing the short test and some performing the long test).

HDD/SSD Validation Via the FreeNAS OS Route

There is more than one way to carry out the HDD/SDD validation tests on the server.

A program specially written for this purpose could be used in a bootable form and run on the server.

However, the easiest way to conduct the HDD/SDD validation tests is to install the FreeNAS OS on the server. It has everything we need. This is not a proper installation of the OS, but just a test installation so we can conduct the SMART and badblocks tests needed.

There are a number of ways the FreeNAS OS can be installed, for example from a CD/DVD, or across a network using PXE. Fester favours a USB stick.

[Create the installer USB stick](#), [install FreeNAS](#), and [enable the SSH service](#), as described on the linked pages. When you first log in to the web GUI, you'll likely see the Initial Wizard; you can just exit out of this at this time.

Conducting The HDD/SDD Validation Tests (Finally!) In FreeNAS Via An SSH Console

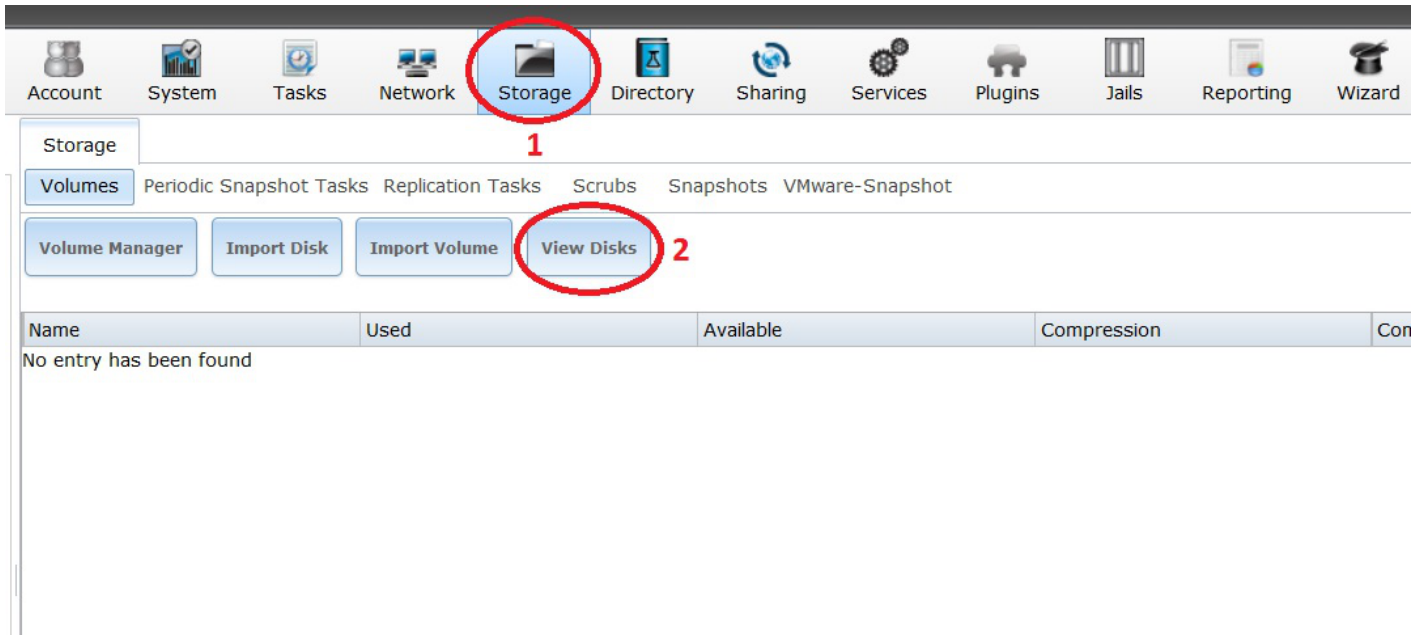
FreeNAS comes with certain software tools and capabilities built into it that will make the task of HDD/SDD validation much easier. This is why we needed to install it before conducting the tests.

The SSH console provides no tools for validation purposes, but does provide the means by which we can flexibly interact with the built in tools in FreeNAS to accomplish the validation tests. This is why we needed to set this up before carrying out the tests.

SMART Short Tests Via The SSH Console

Open up the FreeNAS web GUI in your browser and log in.

Go to the "Storage" page (1) and click the "View Disks" button (2).

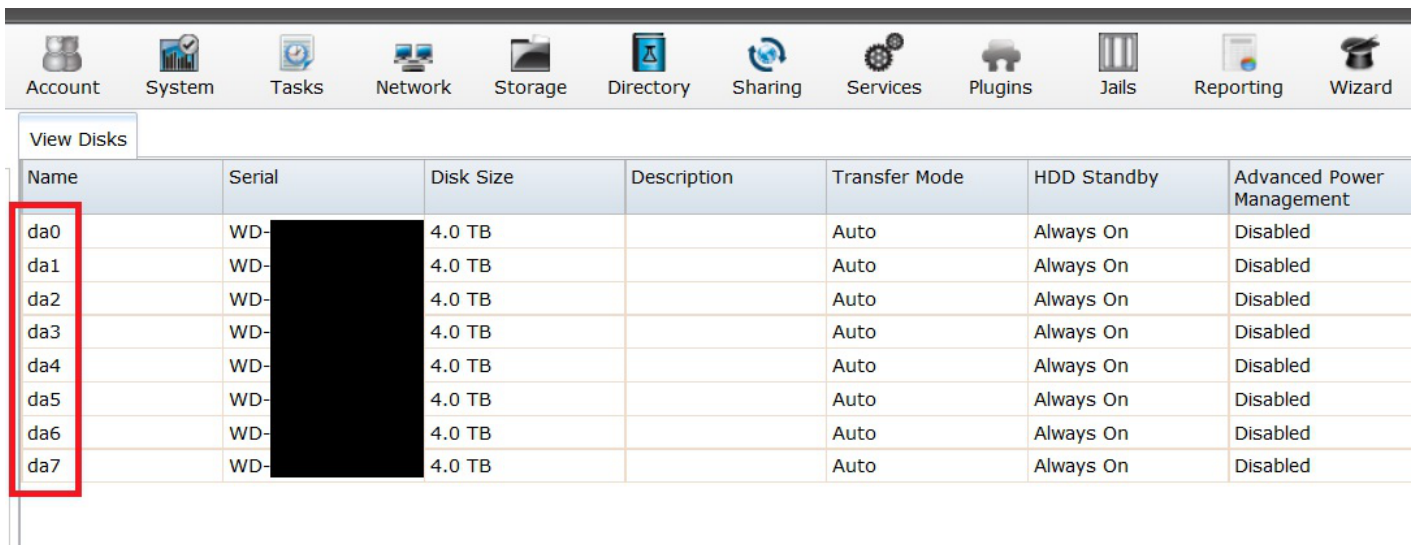


This should bring up a list of the storage HDDs (i.e. for data, not the OS) that are currently in your system.

Make a list of the names of each drive (shown in a red box in the screen shot) these will be needed soon.

(On Fester’s system this would be **da0 - da7**, giving a total of 8 HDDs.)

Incidentally, the name FreeNAS gives the OS HDD is **ada0**. If you have two OS drives (i.e. a mirrored configuration) this would be **ada0** and **ada1** respectively.



Start an SSH session and log in.

Where possible when entering commands it is easier and more accurate to use copy and paste. You can copy the text out of this document in the usual way (i.e. highlight the text, right click with the mouse and from the pop up menu select “Copy”) and then paste it into the PuTTY SSH console by simply right clicking with the mouse anywhere in the console window (the copied text will appear at the command prompt).

If you do it manually then commands entered at the prompt must be exactly as shown including all the spaces or they tend not to work.

Let us start by running the SMART short test on the OS drive labelled ada0 (in Fester's case this is the SSD drive).

At the command prompt type:

```
smartctl -t short /dev/ada0
```

You should get the following screen, the entered command is shown in the first red box (1) and the duration and completion time are shown in the second (2).

```
root@freenas:~
2016

FreeNAS (c) 2009-2015, The FreeNAS Development Team
All rights reserved.
FreeNAS is released under the modified BSD license.

For more information, documentation, help or support, go here:
http://freenas.org
Welcome to FreeNAS
[root@freenas] ~# smartctl -t short /dev/ada0 1
smartctl 6.4 2015-06-04 r4109 [FreeBSD 10.3-RC3 amd64] (local build)
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===
Sending command: "Execute SMART Short self-test routine immediately in off-line
mode".
Drive command "Execute SMART Short self-test routine immediately in off-line mod
e" successful.
Testing has begun.
Please wait 2 minutes for test to complete.
Test will complete after Sat Apr 23 08:37:57 2016 2

Use smartctl -X to abort test.
[root@freenas] ~#
```

(Do not worry about the fact that you cannot see any results or the test running. This is completely correct. The results are obtained later by entering another command at the command prompt after all the tests are completed.)

We now need to repeat this process for each drive in the system. We do not need to wait for this drive to complete its test before starting another on a different drive.

So at the command prompt enter the command to start the SMART short test for the next drive in your system (in Fester's case this is da0, the first storage drive).

```
smartctl -t short /dev/da0
```

Then do the same operation for the next drive, and the next, until all the drives are running the short

SMART test. In Fester's case this would be:

```
smartctl -t short /dev/da1
```

```
smartctl -t short /dev/da2
```

```
smartctl -t short /dev/da3
```

```
smartctl -t short /dev/da4
```

```
smartctl -t short /dev/da5
```

```
smartctl -t short /dev/da6
```

```
smartctl -t short /dev/da7
```

Make a note of the time when the last drive will complete the test and then go and get a cup of tea (or in Fester's case training Ferrets to make cheese cake).

When you are certain the last short test, on the last HDD has completed (you will know because you noted the completion time on the last test) then it is time to start the conveyance tests.

SMART Conveyance Tests Via The SSH Console

If you have exited the SSH session then start another and login.

Run the SMART conveyance test on the OS drive labelled ada0 (in Fester's case this is the SSD drive).

At the command prompt type in:

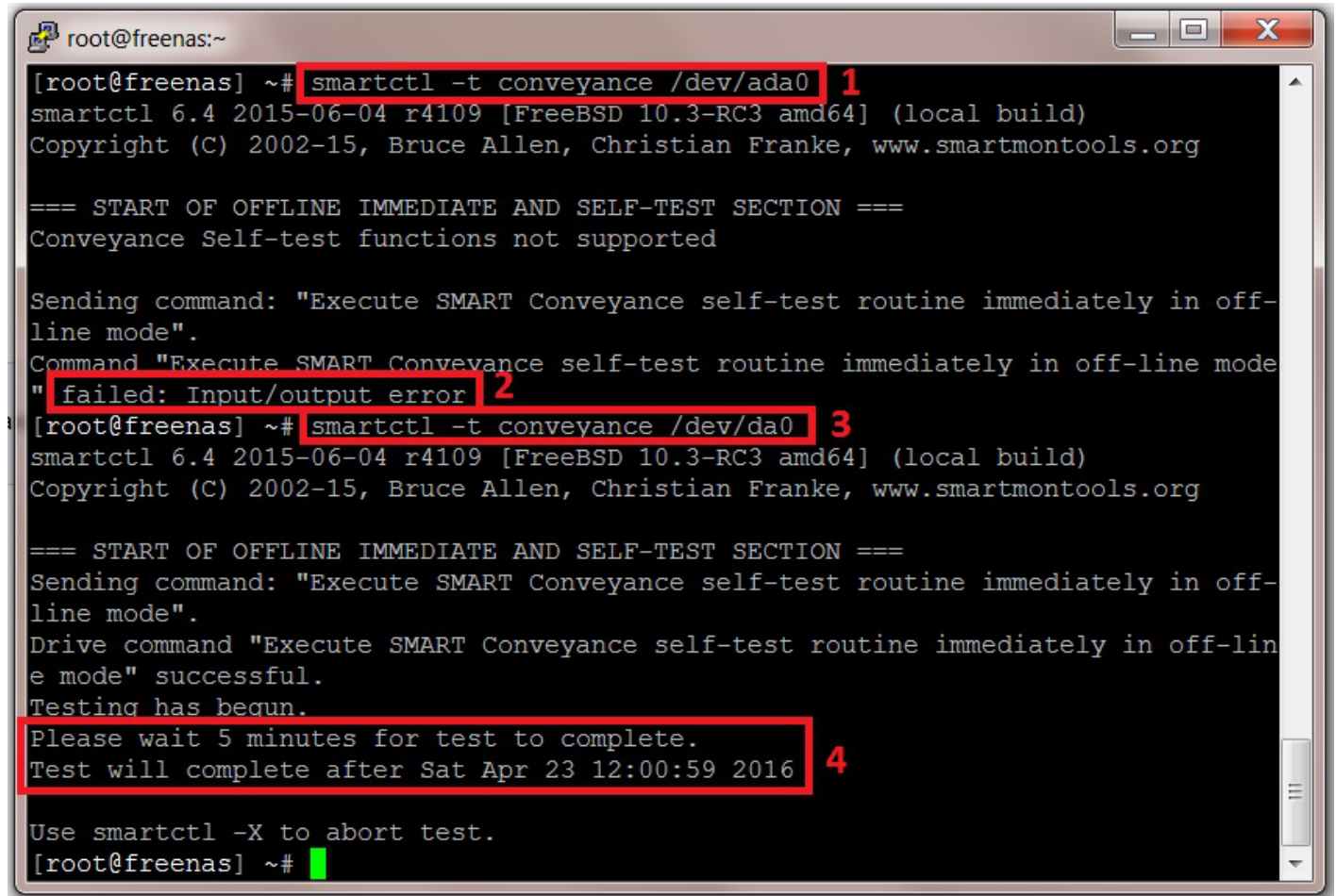
```
smartctl -t conveyance /dev/ada0
```

You should get the following screen, the entered command is shown in the first red box (1). However, the conveyance test failed on this drive due to an input/output error shown in the second red box (2) (some drives don't support conveyance tests, if yours does this just move on to the SMART long test).

So I ran the test on the next drive in the system with the following command:

```
smartctl -t conveyance /dev/da0
```

This is shown in the third red box (3) and now we see how it normally looks when the command is successful. The duration and completion time are shown in the fourth red box (4).



```
root@freenas:~  
[root@freenas] ~# smartctl -t conveyance /dev/ada0 1  
smartctl 6.4 2015-06-04 r4109 [FreeBSD 10.3-RC3 amd64] (local build)  
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org  
  
=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===  
Conveyance Self-test functions not supported  
  
Sending command: "Execute SMART Conveyance self-test routine immediately in off-  
line mode".  
Command "Execute SMART Conveyance self-test routine immediately in off-line mode"  
" failed: Input/output error 2  
[root@freenas] ~# smartctl -t conveyance /dev/da0 3  
smartctl 6.4 2015-06-04 r4109 [FreeBSD 10.3-RC3 amd64] (local build)  
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org  
  
=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===  
Sending command: "Execute SMART Conveyance self-test routine immediately in off-  
line mode".  
Drive command "Execute SMART Conveyance self-test routine immediately in off-lin  
e mode" successful.  
Testing has begun.  
Please wait 5 minutes for test to complete.  
Test will complete after Sat Apr 23 12:00:59 2016 4  
  
Use smartctl -X to abort test.  
[root@freenas] ~#
```

We now need to repeat this process for each drive in the system. We do not need to wait for this drive to complete its test before starting another on a different drive.

So at the command prompt enter the command to start the SMART conveyance test for the next drive in your system (in Fester's case this is da1, the second storage drive).

```
smartctl -t conveyance /dev/da1
```

Then do the same operation for the next drive, and the next, until all the drives are running the SMART conveyance test. In Fester's case this would be:

```
smartctl -t conveyance /dev/da2
```

```
smartctl -t conveyance /dev/da3
```

```
smartctl -t conveyance /dev/da4
```

```
smartctl -t conveyance /dev/da5
```

```
smartctl -t conveyance /dev/da6
```

```
smartctl -t conveyance /dev/da7
```

Make a note of the time when the last drive will complete the test and then go and get a cup of tea (or in

Fester's case cleaning cheese cake off the walls, bloody ferrets!).

When you are certain the last conveyance test, on the last HDD has completed (you will know because you noted the completion time on the last test) then it is time to start the long tests.

SMART Long Tests Via The SSH Console

If you have exited the SSH session then start another and login.

Run the SMART long test on the OS drive labelled ada0.

(Fester did not run this test on ada0 because the drive is an SSD drive. A surface scan on an SSD drive is pointless. The reasons why are beyond the scope of this guide and relate to the way in which SSDs handle a bad memory location using the built in hardware manager and over-provisioned memory).

At the command prompt type in:

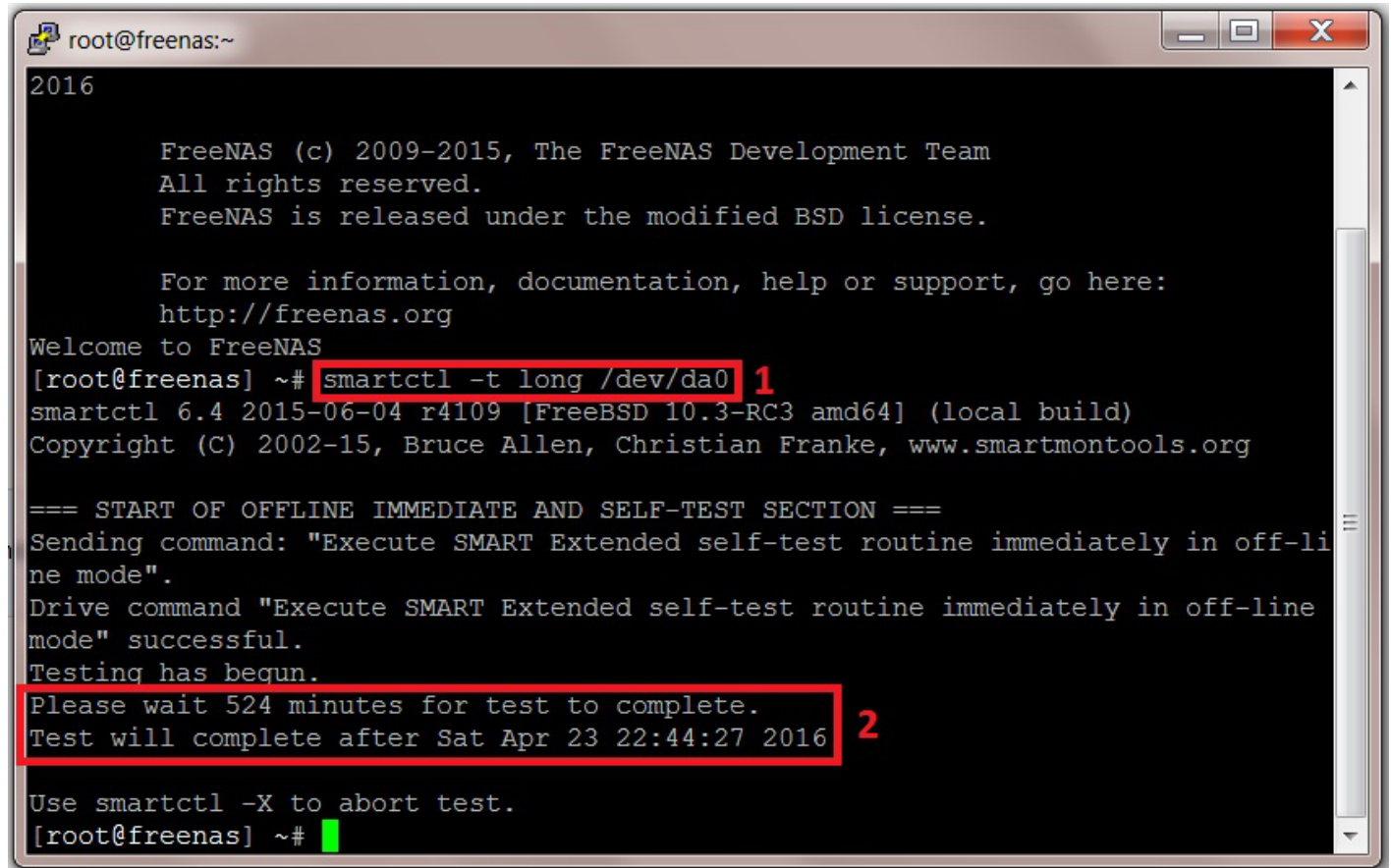
```
smartctl -t long /dev/ada0
```

I can't show you a screen shot of this on ada0 for reasons I have already explained. So let us go on to the next drive in the system and run the SMART long test on that.

At the command prompt type in:

```
smartctl -t long /dev/da0
```

You should get the following screen, the entered command is shown in the first red box (1) and the duration and completion time are shown in the second (2).



```
root@freenas:~  
2016  
  
FreeNAS (c) 2009-2015, The FreeNAS Development Team  
All rights reserved.  
FreeNAS is released under the modified BSD license.  
  
For more information, documentation, help or support, go here:  
http://freenas.org  
Welcome to FreeNAS  
[root@freenas] ~# smartctl -t long /dev/da0 1  
smartctl 6.4 2015-06-04 r4109 [FreeBSD 10.3-RC3 amd64] (local build)  
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org  
  
=== START OF OFFLINE IMMEDIATE AND SELF-TEST SECTION ===  
Sending command: "Execute SMART Extended self-test routine immediately in off-line mode".  
Drive command "Execute SMART Extended self-test routine immediately in off-line mode" successful.  
Testing has begun.  
Please wait 524 minutes for test to complete.  
Test will complete after Sat Apr 23 22:44:27 2016 2  
  
Use smartctl -X to abort test.  
[root@freenas] ~#
```

We now need to repeat this process for each drive in the system. We do not need to wait for this drive to complete its test before starting another on a different drive.

So at the command prompt enter the command to start the SMART long test for the next drive in your system (in Fester's case this is da1, the second storage drive).

```
smartctl -t long /dev/da1
```

Then do the same operation for the next drive, and the next, until all the drives are running the SMART long test. In Fester's case this would be:

```
smartctl -t long /dev/da2
```

```
smartctl -t long /dev/da3
```

```
smartctl -t long /dev/da4
```

```
smartctl -t long /dev/da5
```

```
smartctl -t long /dev/da6
```

```
smartctl -t long /dev/da7
```

Make a note of the time when the last drive will complete the test and then go and get several cups of tea (this one takes a while, most likely several hours).

When you are certain the last long test, on the last HDD has completed (you will know because you noted the completion time on the last test) then it is time to start the badblocks tests.

Badblocks Tests Via The SSH Console

The Badblocks test differs from the SMART tests in important ways.

Unlike the SMART test it is not a self-test. It is done using a piece of software built into the FreeNAS OS (it's actually part of FreeBSD which FreeNAS is built on).

This means if we end the SSH session we also terminate the Badblocks test. Due to the long period of time these tests take to complete it becomes seriously inconvenient to keep an SSH session open that long.

Another problem that occurs is when we start the Badblocks program we can no longer input commands into the SSH command prompt until Badblocks completes its test. Therefore, we cannot run Badblocks tests in parallel on different drives (unlike the SMART tests that can run concurrently).

This means we would need to run one Badblocks test at a time on each drive consecutively (i.e. run Badblocks on one drive and wait for that to complete. Then run it on the next drive and wait for that to complete, until all the drives had been tested).

Considering that this test can take anything from 24 hours to 2 - 4 days depending on the capacity of the drive, then the Badblocks test on an 8 drive system would take an inordinate amount of time (assuming 1 drive takes 3 days, an 8 drive system would take $3 \times 8 = 24$ days!).

So when conducting these tests we will use **tmux** which is a session multiplexer built into FreeNAS. A session multiplexer is a console that is capable of running more than one session at the same time. This means we can now run multiple instances of Badblocks in parallel on different drives (this reduces the 24 days to just 3 days).

Also when we close the SSH console, tmux on the FreeNAS system is kept open. It only closes properly when we formerly exit the tmux session. This means we do not need to keep the SSH console open for 3 days on the client computer (very convenient).

However, there are some caveats to be aware of when using tmux.

FreeNAS Storage Volumes And A Known Problem With badblocks

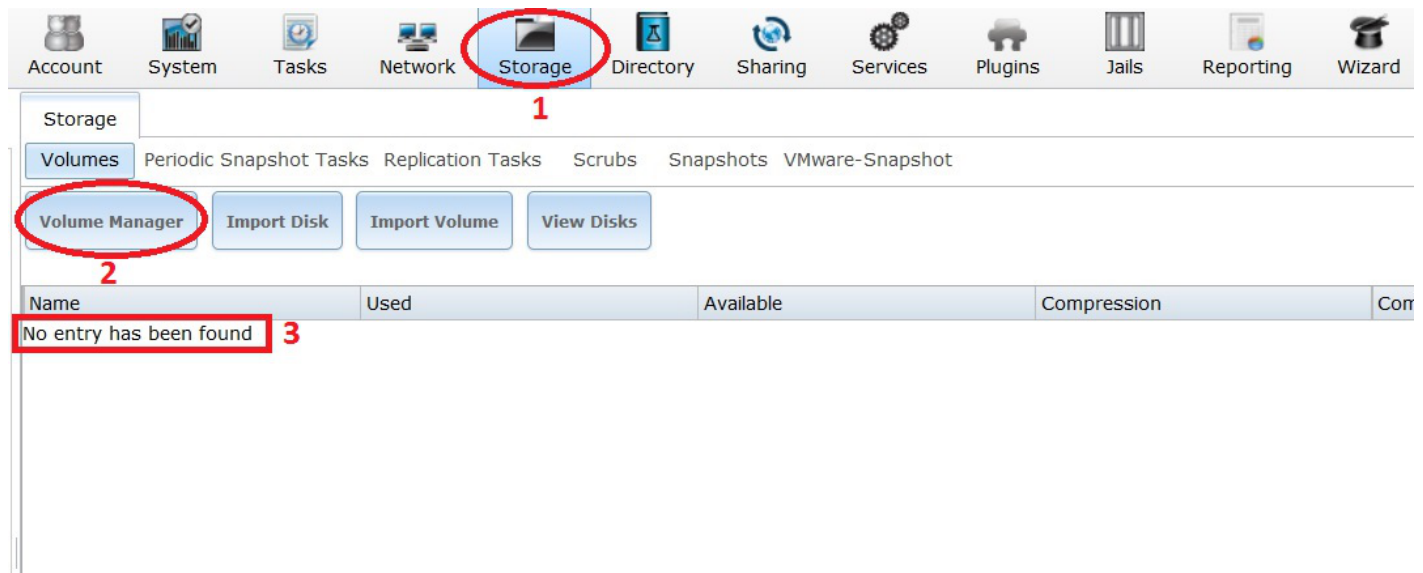
If you have a volume (this is a Zpool) created on the server using the "Volume Manager" in the FreeNAS web GUI, then it is essential to detach the volume before commencing any Badblocks tests.

This is because the FreeNAS OS does a series of small short writes to the volume (Fester does not know the how or why of this, if someone wants to provide some information I will try to include it in the guide so everyone can benefit).

This activity will mess up the Badblocks tests!

This is how to check if your system has a volume.

- Go to the “Storage” page (1).
- Click on the “Volume Manager” button (2).
- If you see text that states “No entry has been found” (3) then your system has no volume and you are good to go with the Badblocks tests.



However, if your system has a volume then you must detach it before continuing.

Detaching The Storage Volume Before A Badblocks Test (DESTRUCTIVE Method)

This is the DESTRUCTIVE method of how to detach a volume.

This means that any and all data on the storage disks will be destroyed forever.

(Which according to the latest scientific research is apparently a long time!)

- Go to the “Storage” page (1).
- Click on the “Volume Manager” button (2).
- If you see entries similar to the screen shot (Fester’s volume is called “TestVolume”) (3) then your system has a volume which you must detach before you are good to go with the Badblocks tests.

The screenshot shows the Storage Manager interface. The 'Storage' menu item is circled in red and labeled '1'. Below it, the 'Volume Manager' button is circled in red and labeled '2'. A table of volumes is shown below, with the first row circled in red and labeled '3'.

Name	Used	Available	Compression	Com
TestVolume	1.2 MiB (0%)	29.0 TiB	-	-
TestVolume	665.4 KiB (0%)	20.0 TiB	lz4	1.00

- To detach the volume select it by clicking on it (it will turn blue when this is done) (1).
- Now click the “Detach Volume” button (2).
- Then tick the “Mark the disks as new (destroy data):” tick box (3)
- (THIS WILL DESTROY ANY AND ALL DATA YOU MAY HAVE ON THE STORAGE DRIVES, DON’T DO THIS IF YOU HAVE DATA YOU WISH TO KEEP).
- Now click the “Yes” button (4).

The screenshot shows the Storage Manager interface with a 'Detach Volume' dialog box open. The dialog box contains the following text:

You have 1.2 MiB of used space within this volume

Mark the disks as new (destroy data): (3)

TestVolume: Are you sure you want to detach?

Yes (4) Cancel

The 'TestVolume' row in the table is circled in black and labeled '1'. The 'Detach Volume' dialog box is circled in black and labeled '2'. The 'Yes' button is circled in black and labeled '4'. The 'Mark the disks as new (destroy data):' checkbox is circled in red and labeled '3'.

Detaching The Storage Volume Before A Badblocks Test (Non-Destructive Method)

- To detach the volume select it by clicking on it (it will turn blue when this is done) (1).
- Now click the “Detach Volume” button (2).
- DO NOT TICK the “Mark the disks as new (destroy data):” tick box (3)
- Now click the “OK” button (4).

The screenshot shows the FreeNAS web GUI. The 'Storage' section is active, displaying a table of volumes. The 'TestVolume' is selected, indicated by a red circle and the number '1'. A 'Detach Volume' dialog box is open, showing the volume name 'TestVolume' and its used space (1.2 MiB). The dialog asks 'Are you sure you want to detach?' and includes a checkbox for 'Mark the disks as new (destroy data):' which is unchecked, highlighted with a red circle and the number '3'. The 'Yes' button is highlighted with a red circle and the number '4'. In the bottom left corner, the 'Detach Volume' button is highlighted with a red circle and the number '2'.

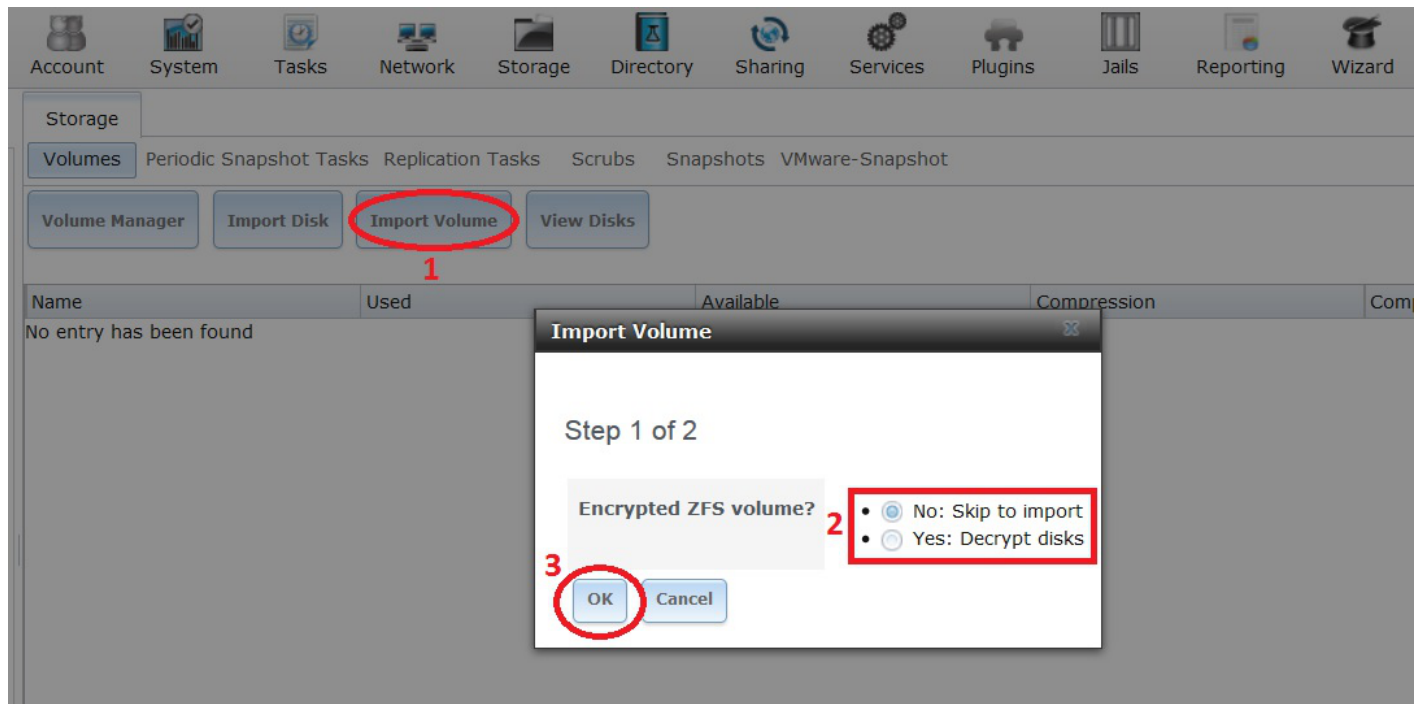
Name	Used	Available	Compression	Comp
TestVolume	1.2 MiB (0%)	29.0 TiB	-	-
TestVolume	665.4 KiB (0%)			1.00x

When you have carried out the **non-destructive** version of the Badblocks test (more on this in a moment) you will then need to reattach the volume.

Importing An Unencrypted Volume After A Non-Destructive Badblocks Test

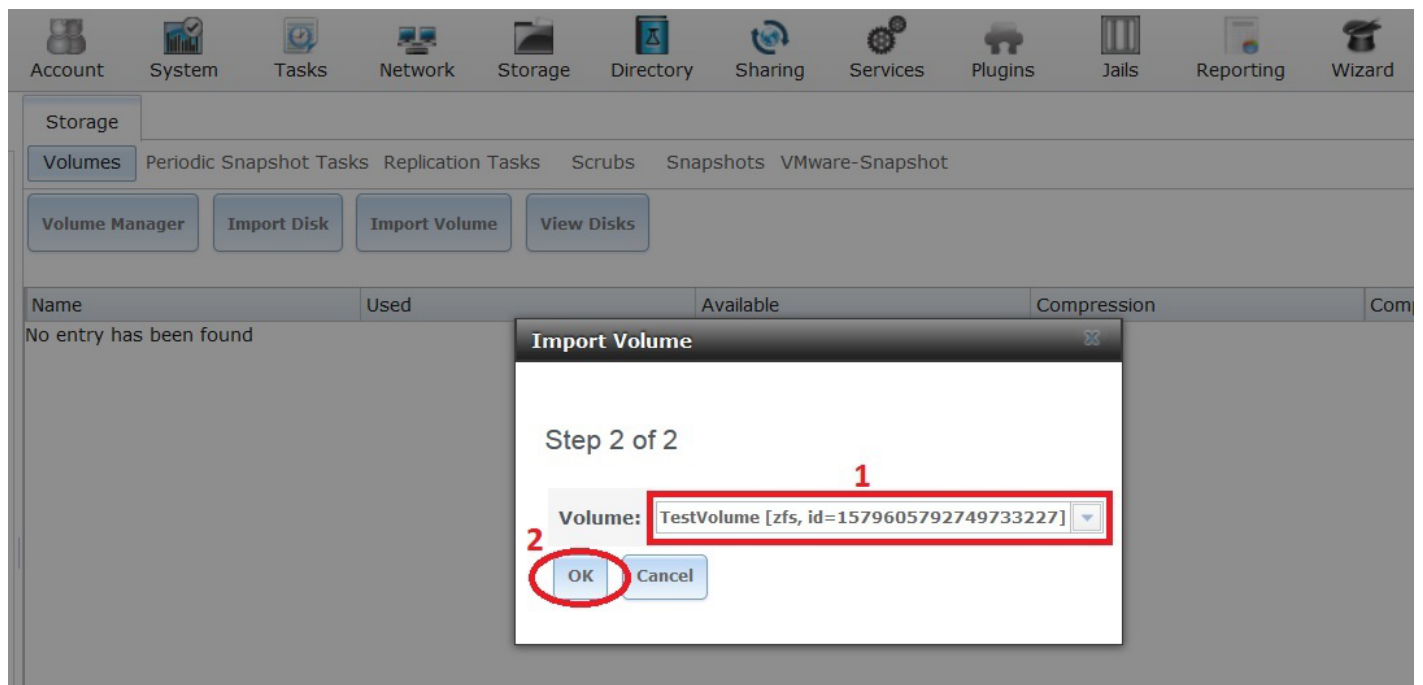
This is how to reattach a non-encrypted volume in the FreeNAS web GUI.

- Assuming you have selected the “Storage” page click on the “Import Volume” button (1).
- If the volume is not encrypted then click the “No: Skip to import” radio button (2).
- Now click the “OK” button (3).



This will take you to a second screen and step 2 of a 2 part process.

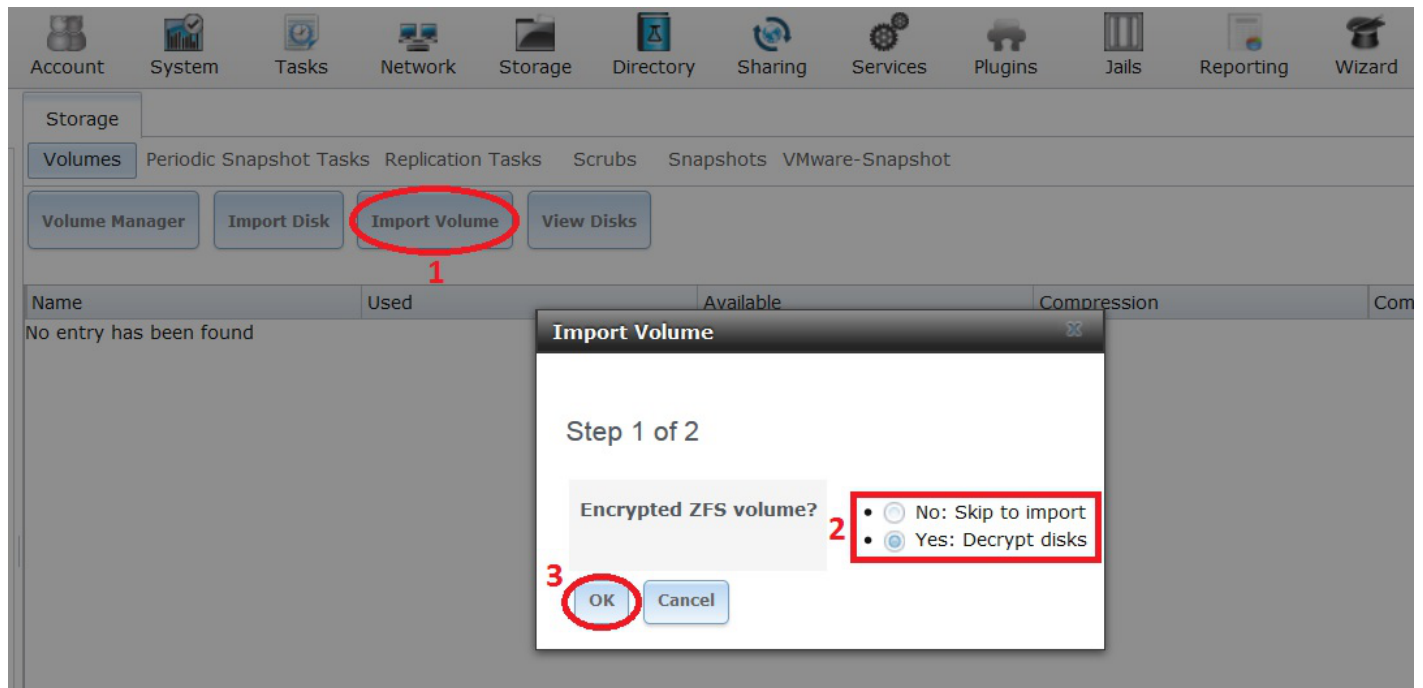
- In the “Volume:” drop down selection box (1) you should see your previously detached volume.
- With the correct volume selected click the “OK” button (2) and the volume should be imported momentarily.



Importing An Encrypted Volume After A Non-Destructive Badblocks Test

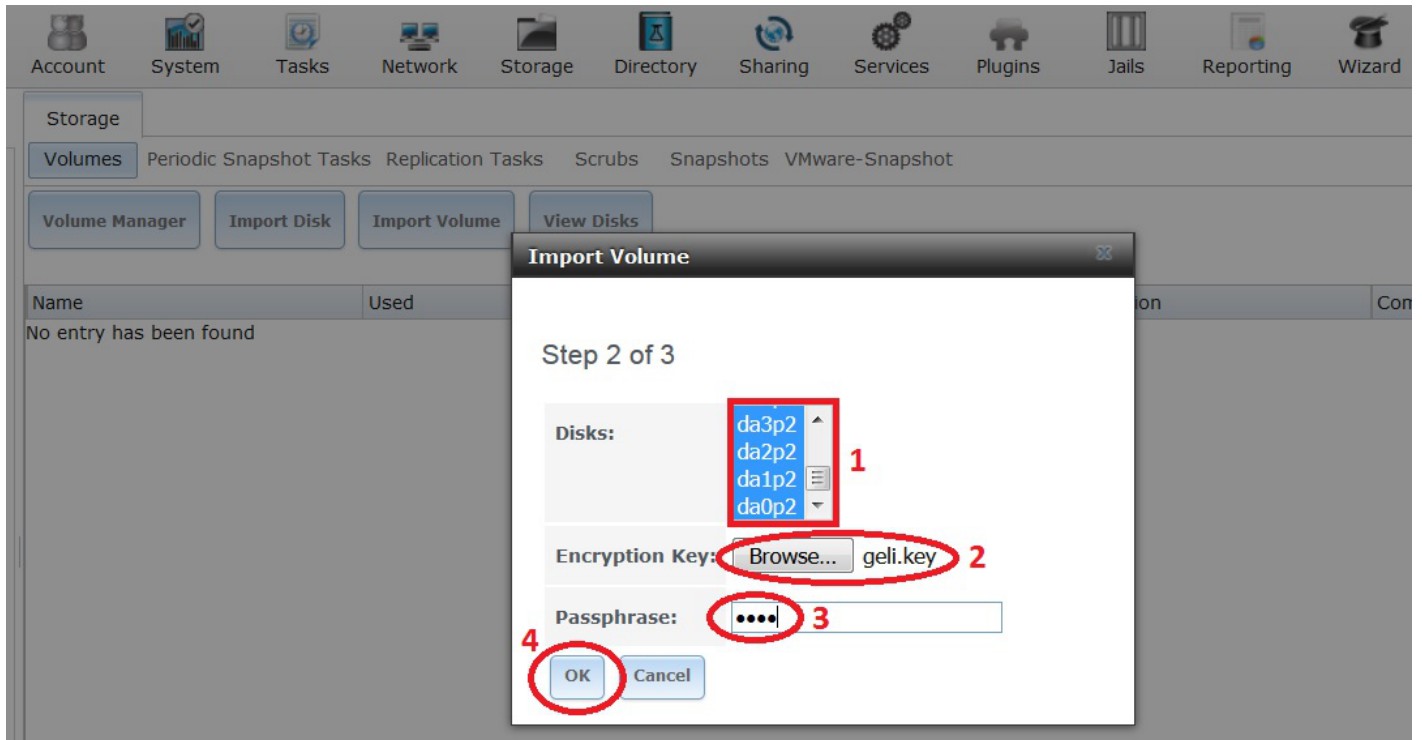
This is how to reattach an encrypted volume in the FreeNAS web GUI.

- Assuming you have selected the “Storage” page click on the “Import Volume” button (1).
- If the volume is encrypted then click the “Yes : Decrypt disks” radio button (2).
- Now click the “OK” button (3).



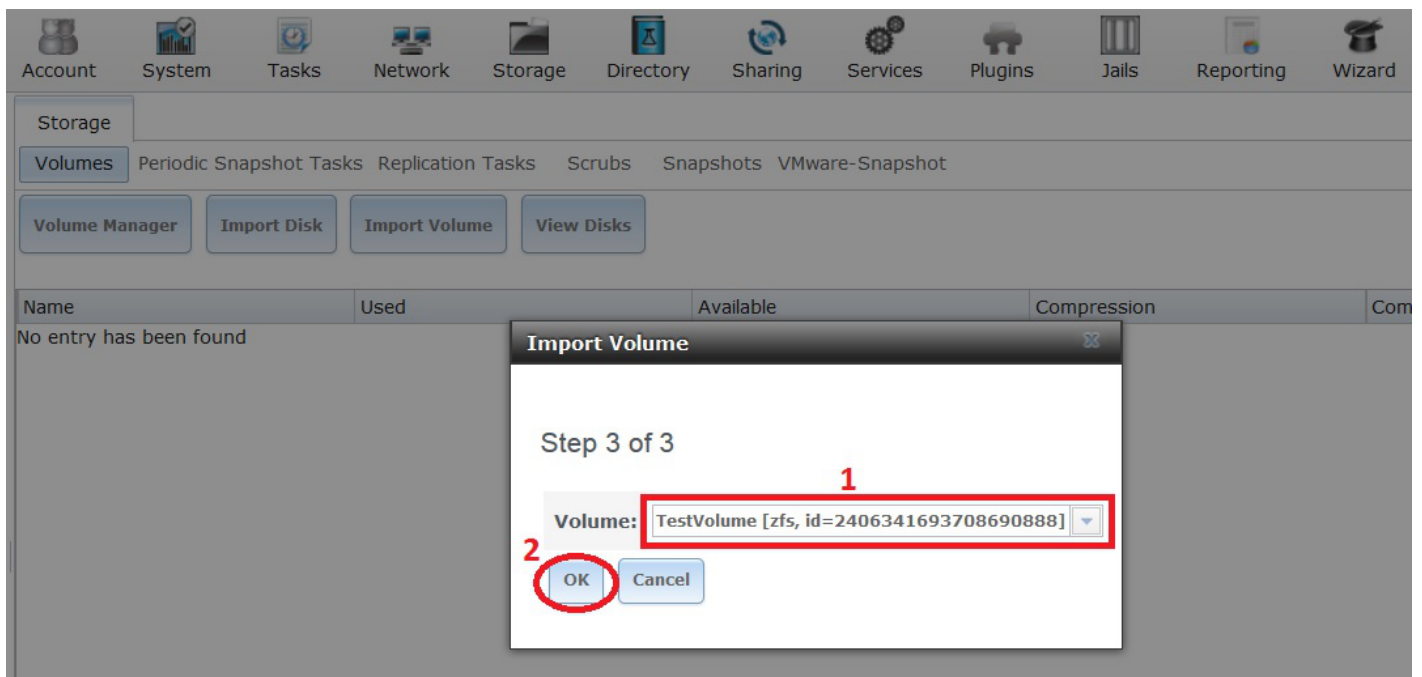
This will take you to a second screen and step 2 of a 3 part process.

- Select the disks that form the volume from the “Disks:” window (1) (in Fester’s case this was all of them).
- Now click the “Browse” button (2) and a window will pop up that allows you to load in your previously saved geli key (when creating encrypted volumes always make sure you save a recovery key).
- Navigate to the location of your key and load it into the FreeNAS GUI. If all goes well you should see it next to the “Browse” button (Fester’s shows up as geli.key) (2).
- Now type in the passphrase (which is a password you created when you made the encrypted volume), in the text box next to “Passphrase:” (3) (Fester very imaginatively used **test** here).
- Now click the “OK” button (4).



The third and final screen will now appear.

- In the “Volume:” drop down selection box (1) you should see your previously detached volume.
- With the correct volume selected click the “OK” button (2) and the volume should be imported momentarily.



Destructive Badblocks Test Using tmux

Start an SSH session and log in.

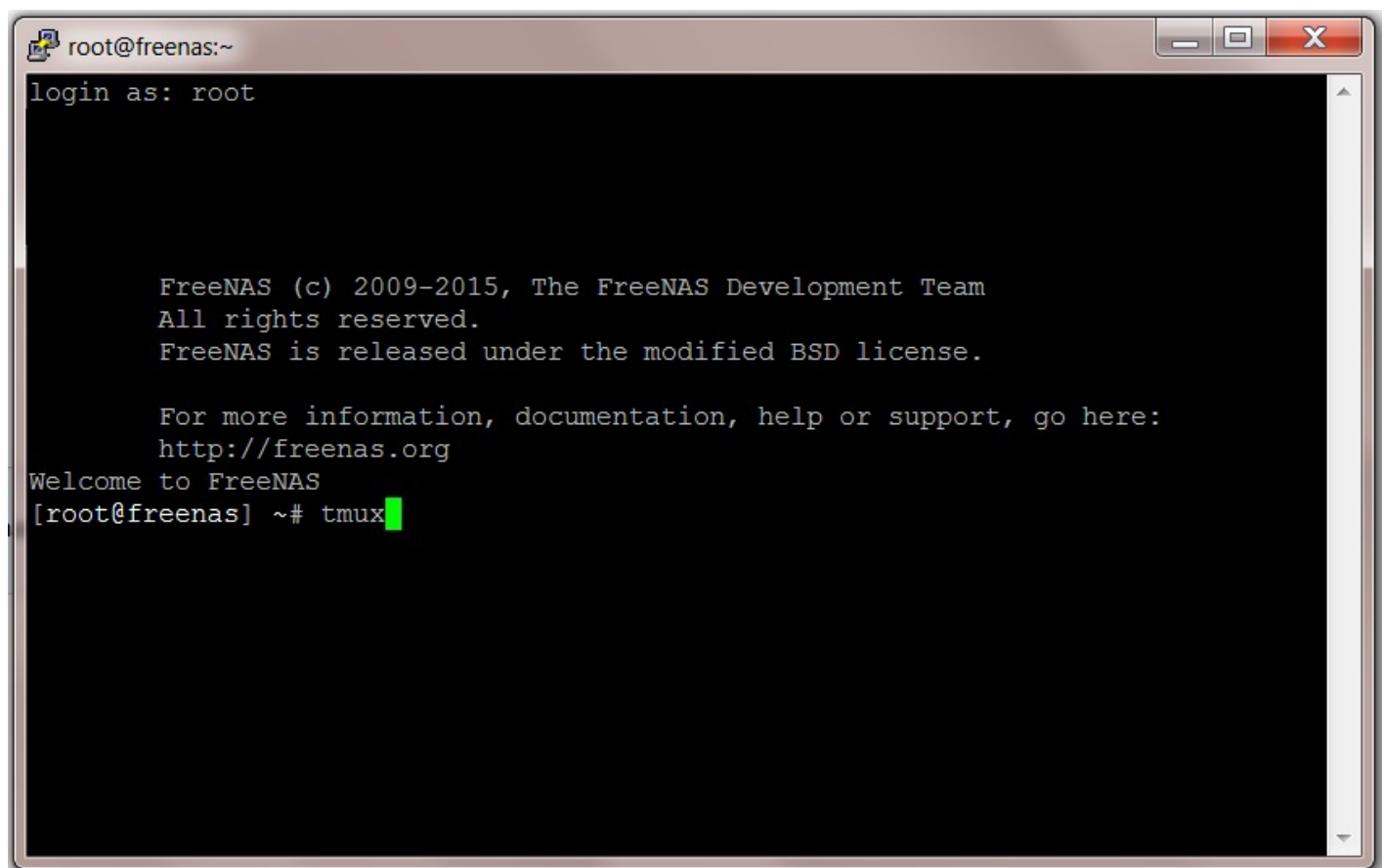
Before starting tmux we need to enable the kernel geometry debug flags, so type in this command at the command prompt.

```
sysctl kern.geom.debugflags=0x10
```

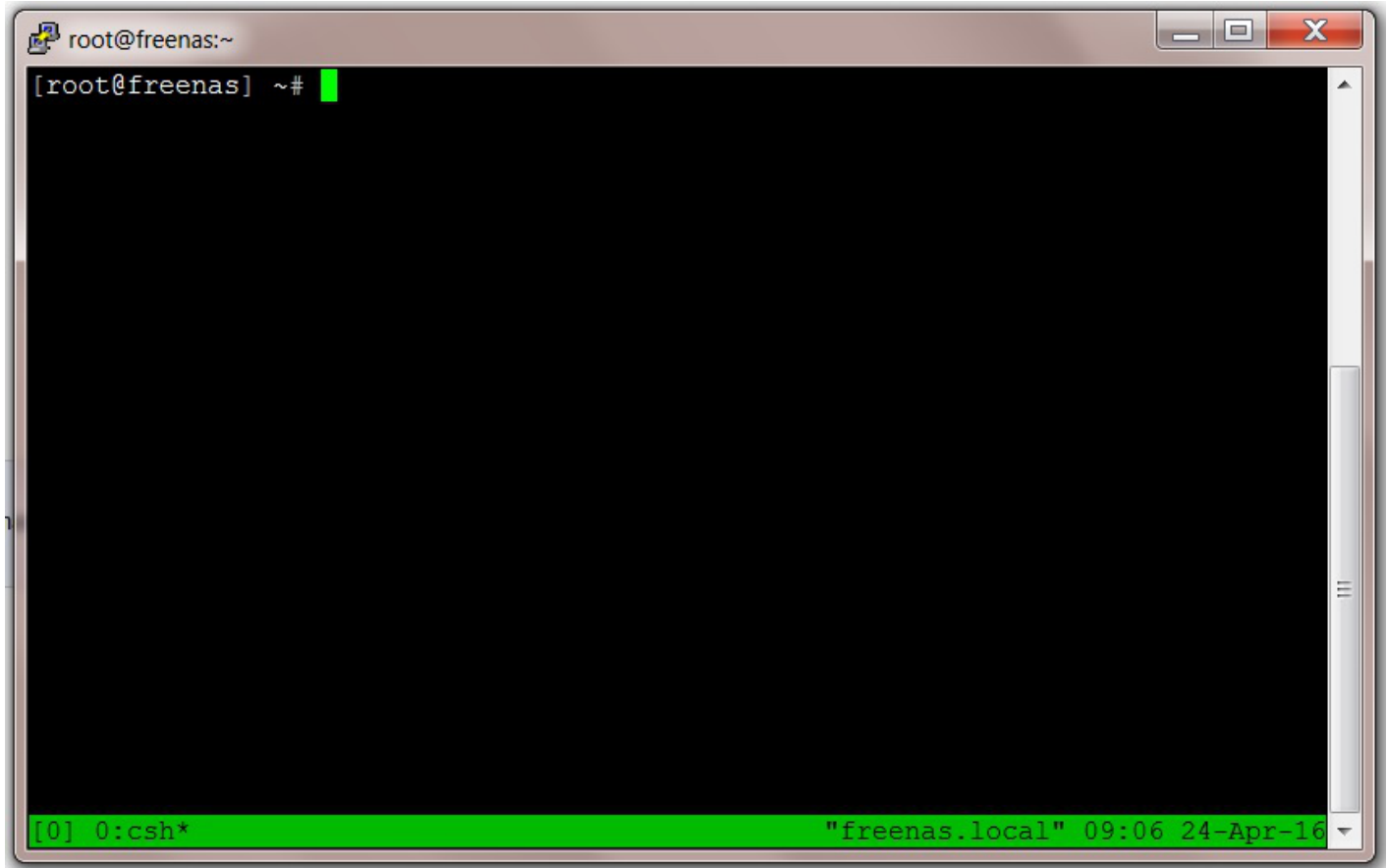
(When all the Badblocks tests are done the kernel geometry debug flags must be returned to their normal state. Thankfully no additional command is necessary, just reboot the server as this setting is not persistent and cannot survive the reboot.)

Now type the following command at the command prompt.

```
tmux
```

A screenshot of a terminal window titled 'root@freenas:~'. The terminal shows a login prompt 'login as: root' followed by the user 'root'. Below this, there is a copyright notice for FreeNAS (c) 2009-2015, The FreeNAS Development Team, and a link to the website 'http://freenas.org'. The prompt then changes to '[root@freenas] ~# tmux' with a green cursor at the end of the command. The terminal window has standard window controls (minimize, maximize, close) in the top right corner.

You should see a screen something like this. Notice the green band at the bottom of the screen, this is a tmux session.

A terminal window titled 'root@freenas:~' with standard window controls. The prompt is '[root@freenas] ~#' followed by a green cursor. The status bar at the bottom is green and contains the text '[0] 0:csh*' on the left and '"freenas.local" 09:06 24-Apr-16' on the right.

```
root@freenas:~  
[root@freenas] ~#  
[0] 0:csh* "freenas.local" 09:06 24-Apr-16
```

I will not be running the Badblocks test on ada0 (the SSD drive) there is no point as already explained and this is a destructive test (the FreeNAS OS is on this drive!).

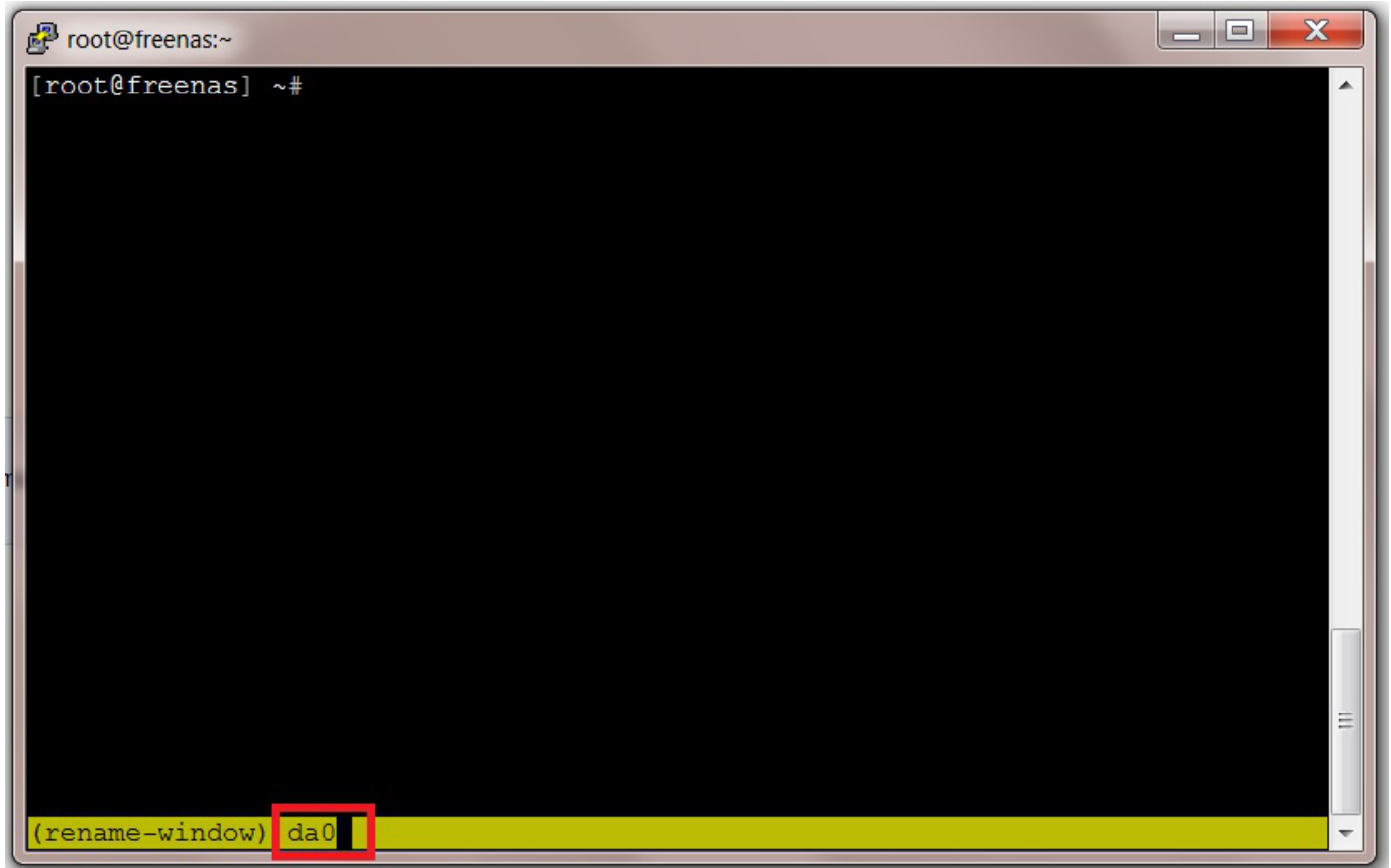
This leaves the 8 data storage drives to check.

This means I will need 8 sessions opened in tmux (open the number of sessions that suits your requirements).

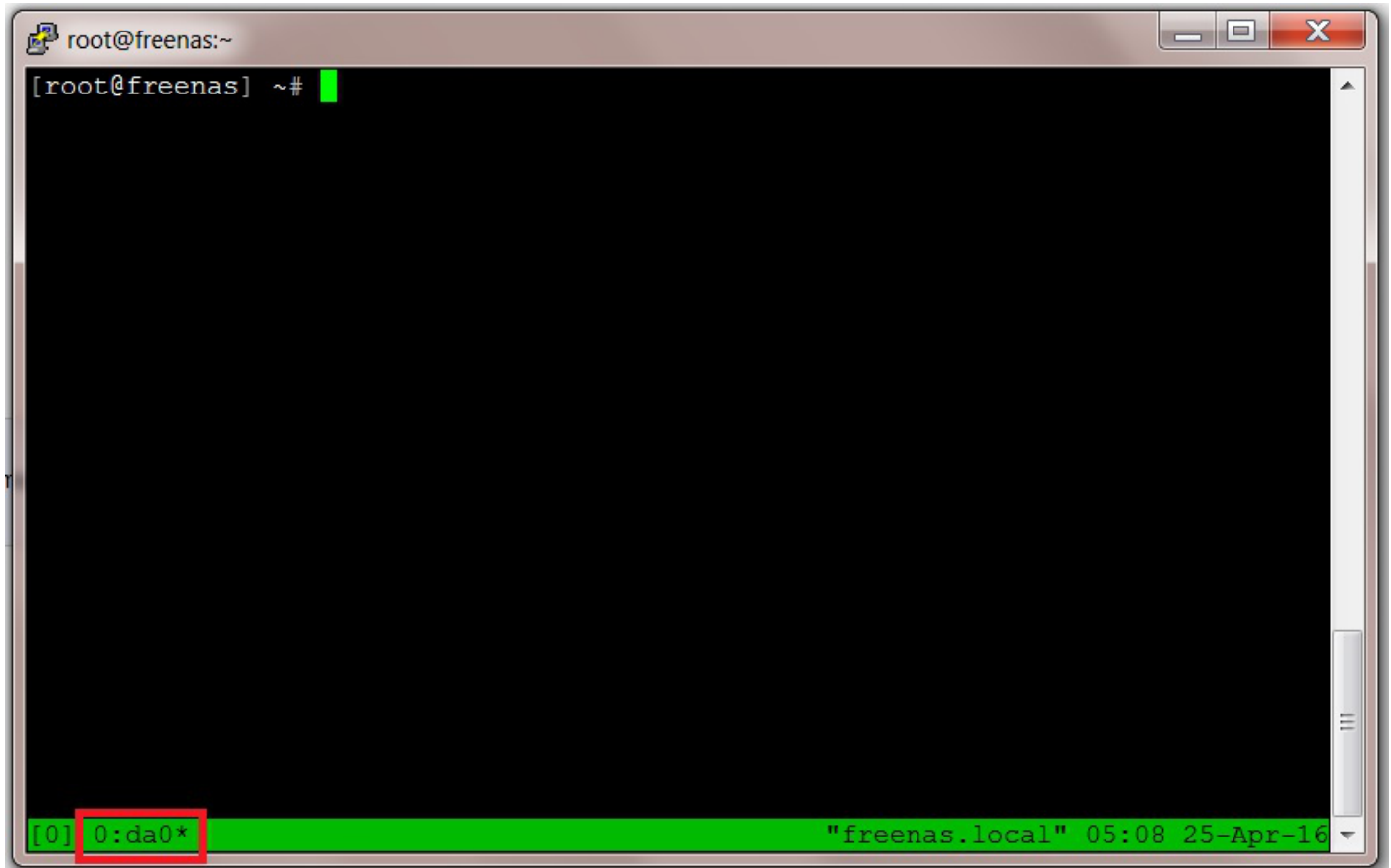
Let us start by renaming the current session in tmux to something more meaningful than "csh".

In the tmux window press the "Ctrl" and "b" keys together, release them and then press the "," key.

The bar at the bottom of the window should turn yellow and you can now delete the "csh" text and rename it (Fester called his "da0" after the drive that will be tested).



When you have typed in the new name press the "Return/Enter" key, the bar should now resort back to its original green colour and the session should be renamed.

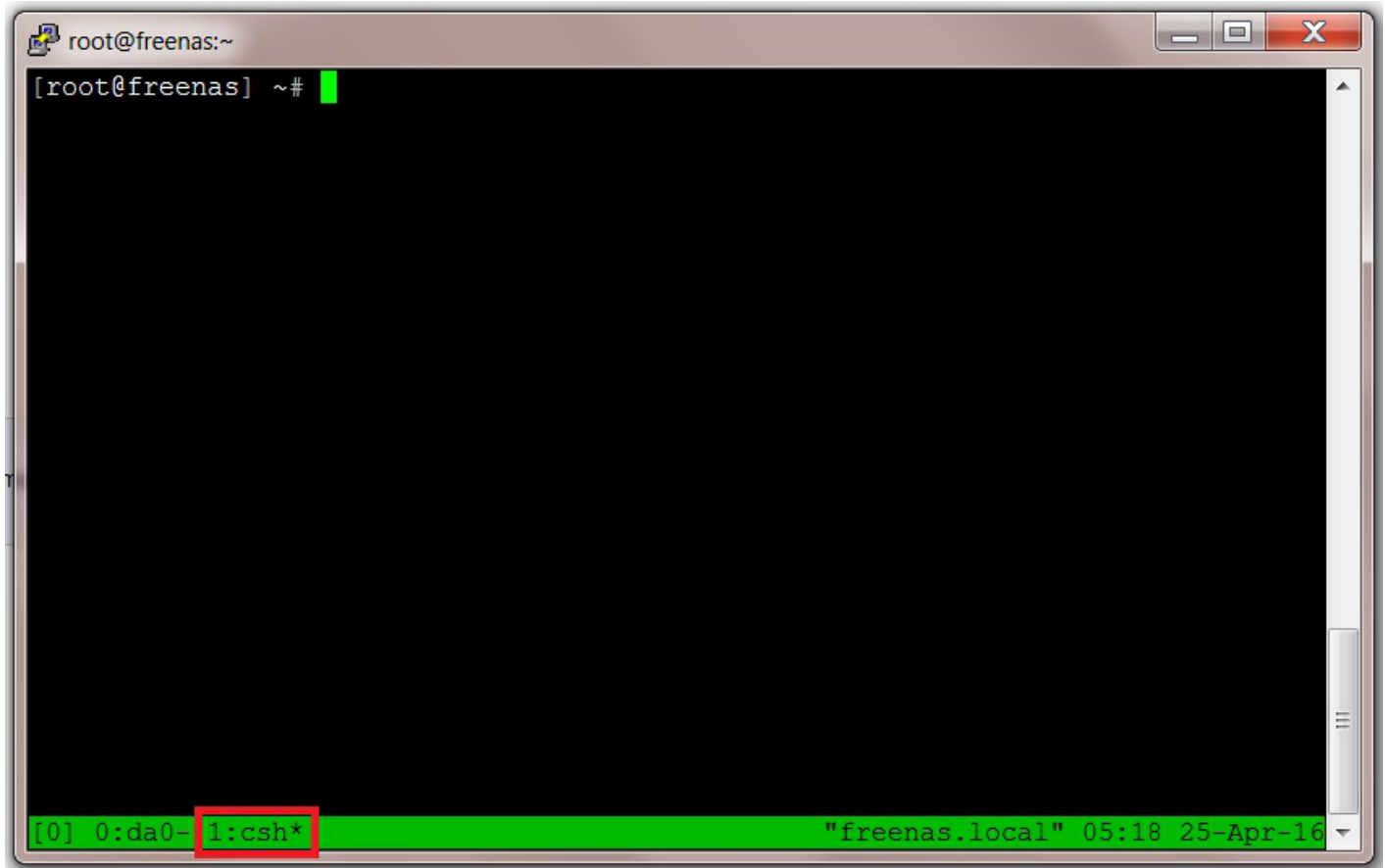


```
root@freenas:~  
[root@freenas] ~#  
[0] 0:da0* "freenas.local" 05:08 25-Apr-16
```

At this point we need to create an additional session and rename it for the next drive to be tested.

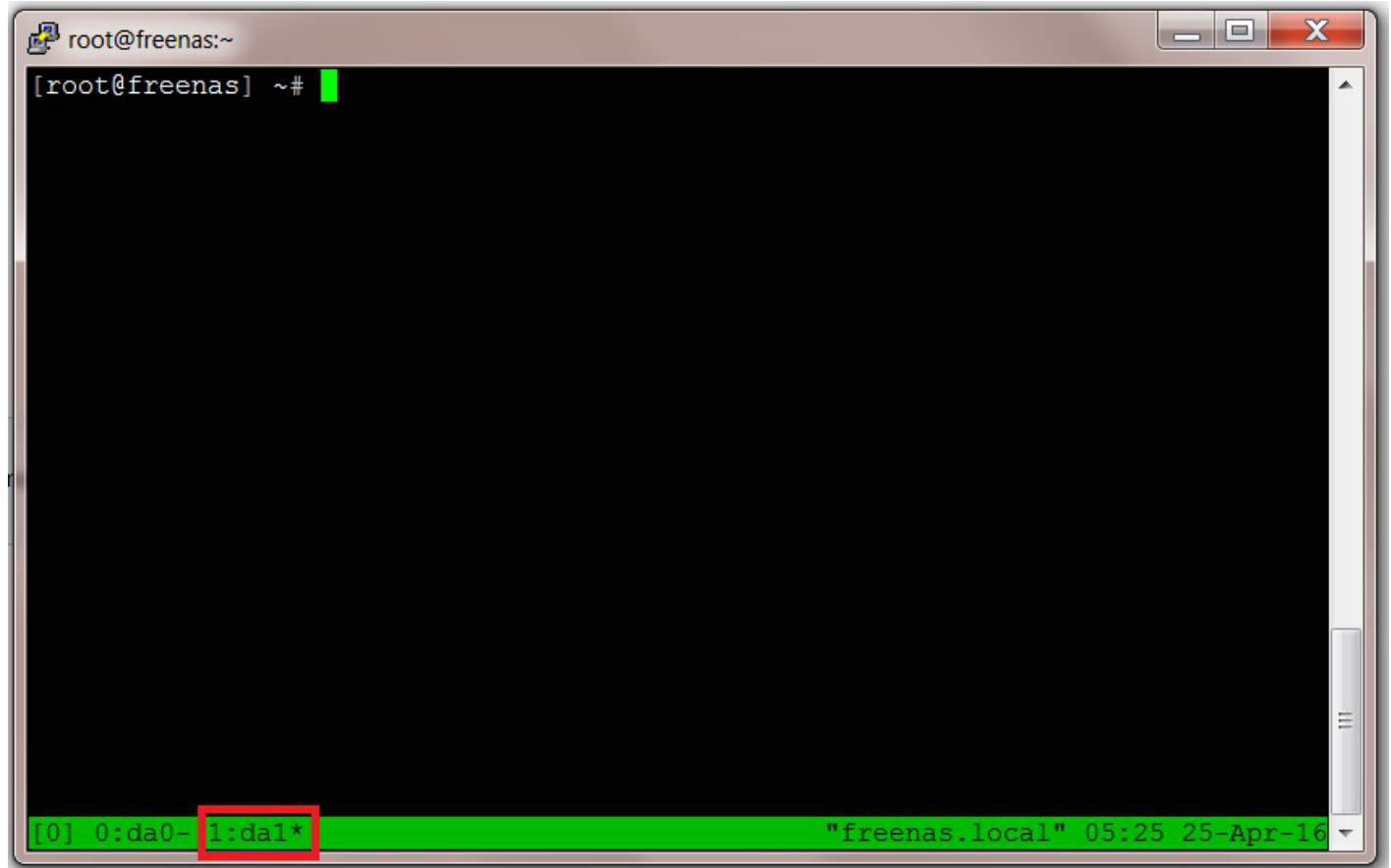
To create a new session press the “Ctrl” and “b” keys together, release them and then press the “c” key.

You should get something like this where “1:csh*” is the newly created session. Incidentally the asterisk just denotes the currently selected session.



```
root@freenas:~  
[root@freenas] ~#  
[0] 0:da0- 1:csh* "freenas.local" 05:18 25-Apr-16
```

Let us rename this session by pressing the “Ctrl” and “b” keys together, releasing them and then pressing the “,” key. Type in the new name and press the “Return/Enter” key (just as we did before, I called this one “da1” after the next drive to be tested).



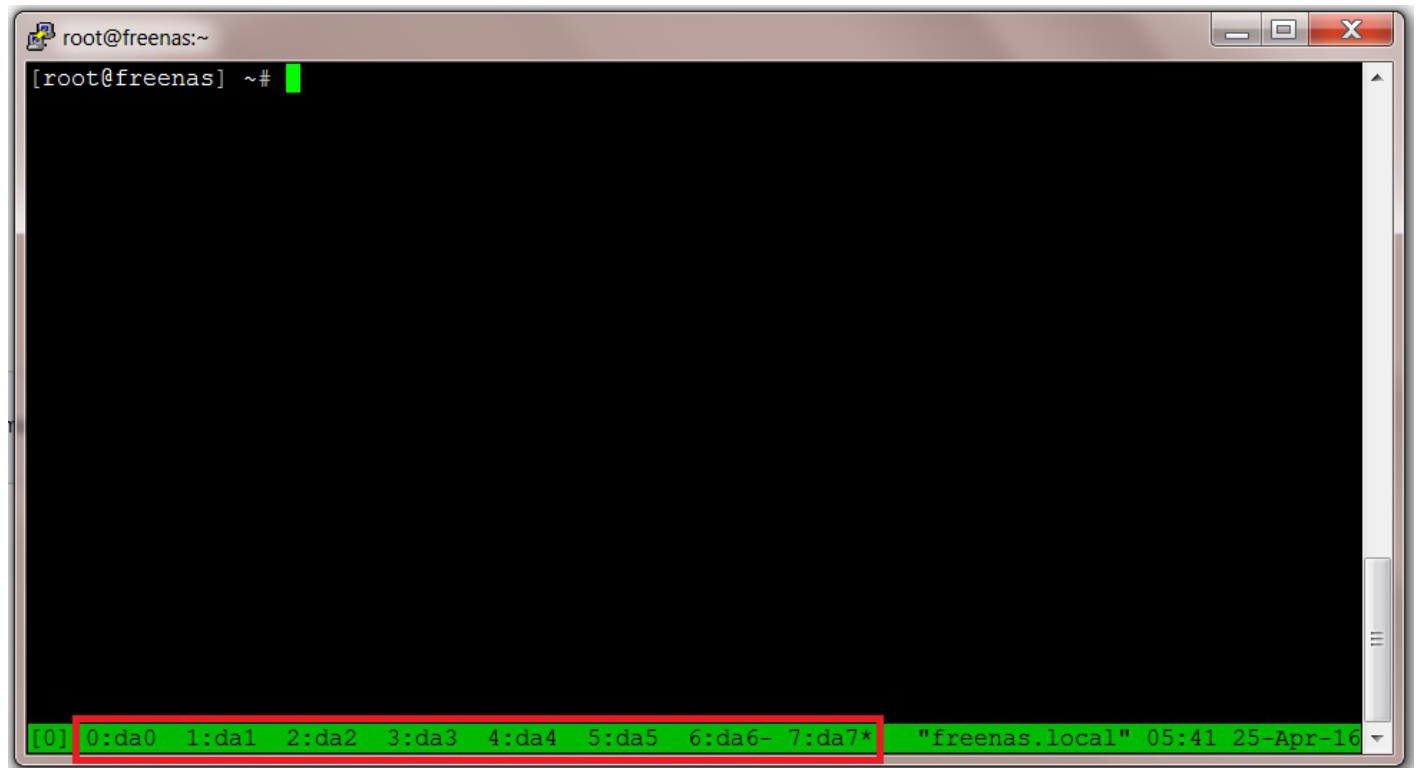
```
root@freenas:~  
[root@freenas] ~#  
[0] 0:da0- 1:da1* "freenas.local" 05:25 25-Apr-16
```

Navigation between the different sessions is achieved by pressing the “Ctrl” and “b” keys together, releasing them and then pressing the “n” key. This will take you to the next session along.

Alternatively you can also press the “Ctrl” and “b” keys together, release them and then press the “p” key. This will take you to the previous session.

By using the next and previous navigational keystroke combinations you can navigate through the different sessions, the asterisk signifying which session you are currently viewing.

Using the key combinations already explained let us create the remaining sessions needed and rename each one.



```
root@freenas:~
[root@freenas] ~#
[0] 0:da0 1:da1 2:da2 3:da3 4:da4 5:da5 6:da6- 7:da7* "freenas.local" 05:41 25-Apr-16
```

Now we can run the Badblocks tests from within tmux.

Navigate to the first session (i.e. "0:da0") and type in the following command at the prompt.

```
badblocks -ws /dev/da0
```

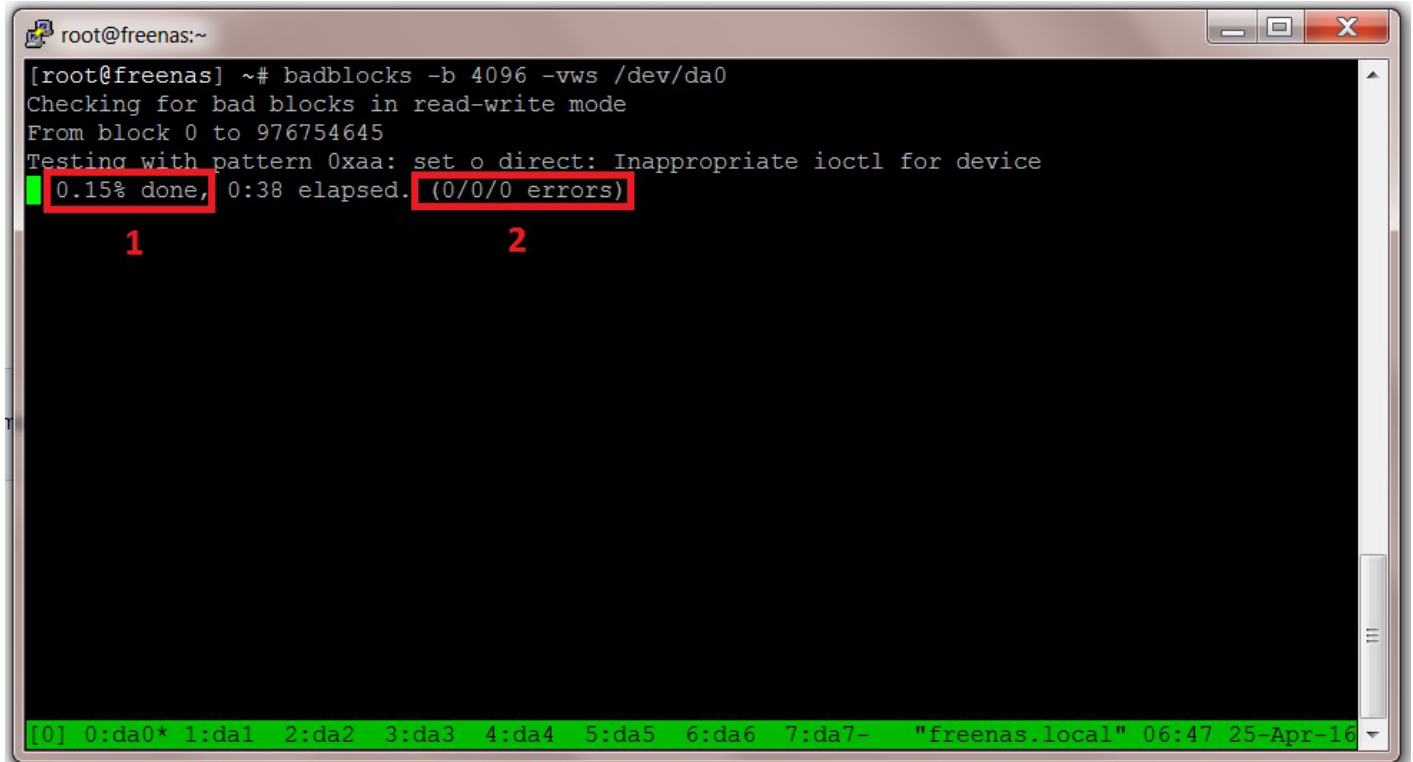
Fester uses a slightly different command to improve the efficiency of the tests with the WD40EFRX drives. These drives have a sector size of 4096 bytes (even though they report 512 bytes, naughty Western Digital). I also like a more verbose output from these tests so the command includes the `-v` switch. I include it here for informational purposes only. In addition to improving efficiency, the `"-b 4096"` below is required with larger disks (*i.e.*, larger than about 4 TB).

```
badblocks -b 4096 -vws /dev/da0
```

If the command executes properly you should see something like this.

You will see from the screen the completion progress expressed as a percentage (1) and any errors that have occurred expressed like this "(0/0/0 errors)" (2).

There should be zero errors throughout the test. If you get even one error then you should return the disk for testing.

A terminal window titled 'root@freenas:~' showing the execution of the 'badblocks' command. The command is 'badblocks -b 4096 -vws /dev/da0'. The output shows 'Checking for bad blocks in read-write mode', 'From block 0 to 976754645', and 'Testing with pattern 0xaa: set o direct: Inappropriate ioctl for device'. The progress bar shows '0.15% done, 0:38 elapsed. (0/0/0 errors)'. The terminal has a green status bar at the bottom with the text '[0] 0:da0* 1:da1 2:da2 3:da3 4:da4 5:da5 6:da6 7:da7- "freenas.local" 06:47 25-Apr-16'. Two red numbers '1' and '2' are placed below the first and second parts of the progress bar respectively.

```
root@freenas:~  
[root@freenas] ~# badblocks -b 4096 -vws /dev/da0  
Checking for bad blocks in read-write mode  
From block 0 to 976754645  
Testing with pattern 0xaa: set o direct: Inappropriate ioctl for device  
0.15% done, 0:38 elapsed. (0/0/0 errors)  
[0] 0:da0* 1:da1 2:da2 3:da3 4:da4 5:da5 6:da6 7:da7- "freenas.local" 06:47 25-Apr-16
```

Now navigate to the next session (in Fester's case that is "1:da1") and type this at the command prompt.

```
badblocks -ws /dev/da1
```

(Or Fester's variation if it suits you better, but remember to change the drive name from "da0" to "da1".)

Repeat this process of changing session and running the Badblocks command for every drive in your system that you want to test. In Fester's case this means running these commands while changing sessions each time.

```
badblocks -ws /dev/da2
```

```
badblocks -ws /dev/da3
```

```
badblocks -ws /dev/da4
```

```
badblocks -ws /dev/da5
```

```
badblocks -ws /dev/da6
```

```
badblocks -ws /dev/da7
```

Non-Destructive Badblocks Test Using tmux

To do a non-destructive badblocks test, follow the instructions above, but replace -w with -n. Example commands might look like:

```
badblocks -ns /dev/da0
```

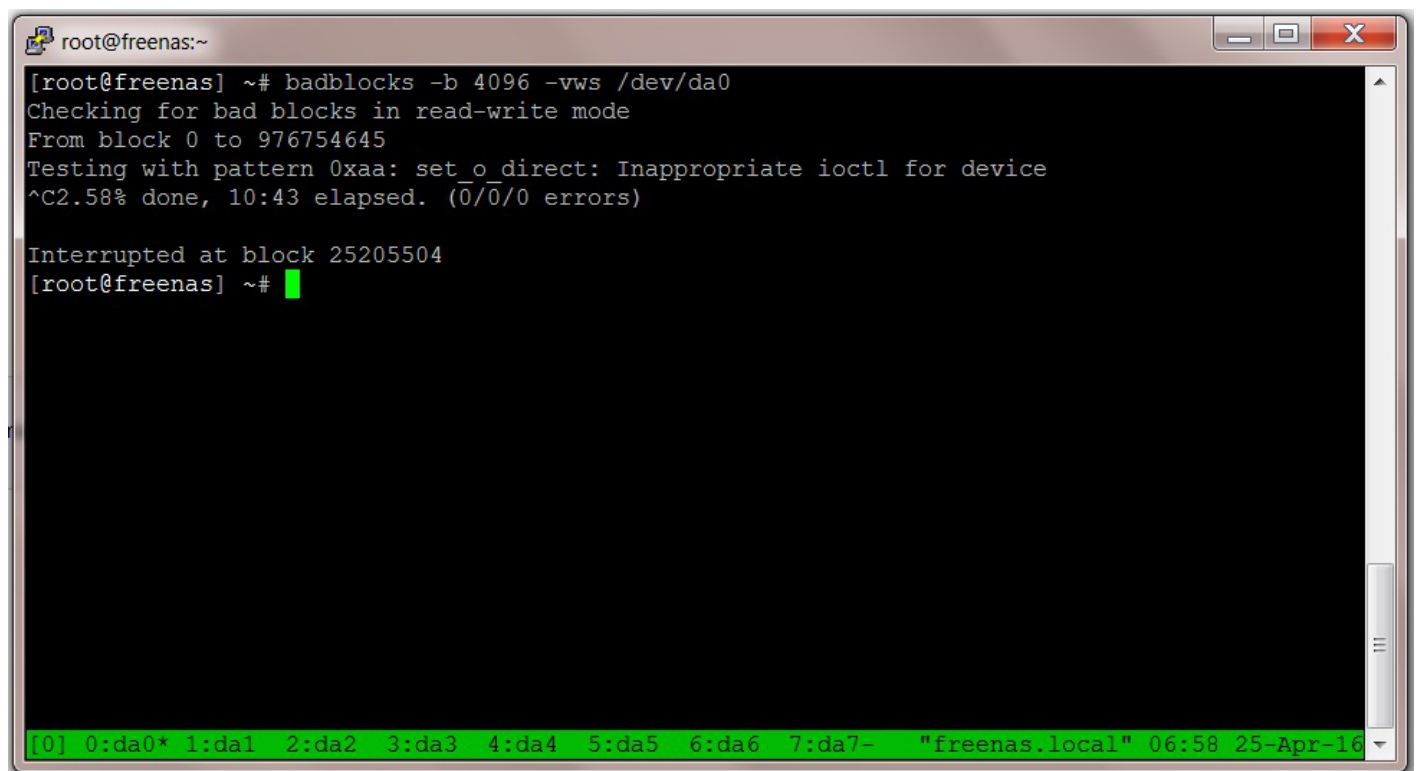
or

```
badblocks -b 4096 -nsv /dev/da1
```

This test is intended to be non-destructive—once it has completed, the data on your disk should be unharmed. Even so, I'd discourage running this on a disk with important data unless you have a good, readily-accessible backup.

Stopping A Badblocks Test In tmux

If for any reason you need to stop a Badblocks test then navigate to the applicable session at press the “Ctrl” and “c” keys together, then release them. This should stop the test.

A screenshot of a terminal window titled "root@freenas:~". The terminal shows the execution of the command "badblocks -b 4096 -vws /dev/da0". The output indicates the test is checking for bad blocks in read-write mode from block 0 to 976754645, using a pattern of 0xaa. It reports "2.58% done, 10:43 elapsed. (0/0/0 errors)". The test is then interrupted at block 25205504, and the prompt returns to the user. A green status bar at the bottom of the terminal window shows session information: "[0] 0:da0* 1:da1 2:da2 3:da3 4:da4 5:da5 6:da6 7:da7- "freenas.local" 06:58 25-Apr-16".

```
root@freenas:~  
[root@freenas] ~# badblocks -b 4096 -vws /dev/da0  
Checking for bad blocks in read-write mode  
From block 0 to 976754645  
Testing with pattern 0xaa: set_o_direct: Inappropriate ioctl for device  
^C2.58% done, 10:43 elapsed. (0/0/0 errors)  
  
Interrupted at block 25205504  
[root@freenas] ~# █  
[0] 0:da0* 1:da1 2:da2 3:da3 4:da4 5:da5 6:da6 7:da7- "freenas.local" 06:58 25-Apr-16
```

Detaching a tmux Session

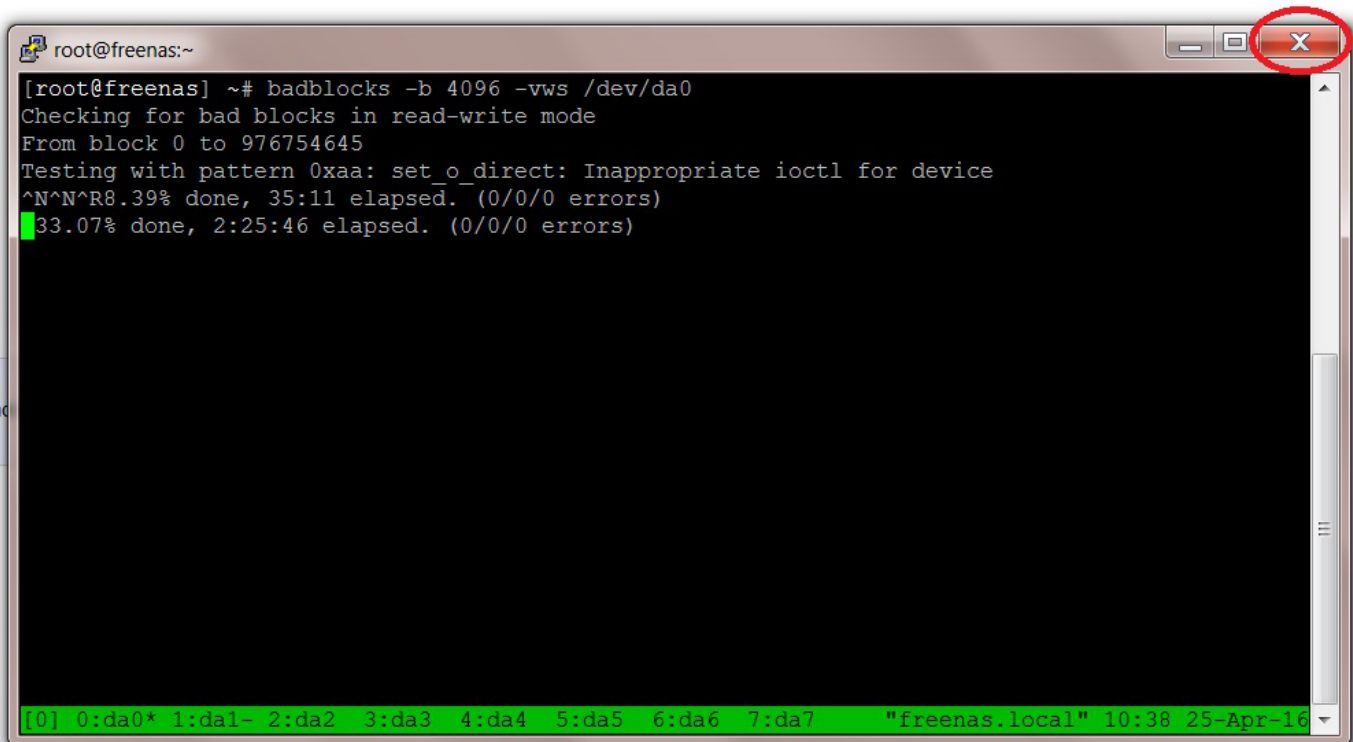
Once you've started all your tests running, you can “detach” your tmux session to let the tests run while you do something else with your SSH session (or just close the SSH connection entirely). To do this, press the “Ctrl” and “b” keys together, then press the “d” key. This will return you to your SSH session, without the tmux session (you'll notice that the green bar isn't present on the bottom of the screen).

Resuming A Session In tmux

Badblocks tests can take a long time (when the tests completed Fester was far from where he had started due to Continental Drift and Plate Tectonics).

You do not need to keep the terminal open or the client computer switched on in order to keep the tmux session running.

If you need to pack up for the night then just close the window that the sessions are running in (just don't shut down the server). Then get your ferrets to shut down your client computer and your pigeons to knock up a suitable night cap (I find a Multiple Orgasm very agreeable before bed).



```
root@freenas:~  
[root@freenas] ~# badblocks -b 4096 -vws /dev/da0  
Checking for bad blocks in read-write mode  
From block 0 to 976754645  
Testing with pattern 0xaa: set_o_direct: Inappropriate ioctl for device  
^N^N^R8.39% done, 35:11 elapsed. (0/0/0 errors)  
█33.07% done, 2:25:46 elapsed. (0/0/0 errors)  
[0] 0:da0* 1:da1- 2:da2 3:da3 4:da4 5:da5 6:da6 7:da7 "freenas.local" 10:38 25-Apr-16
```

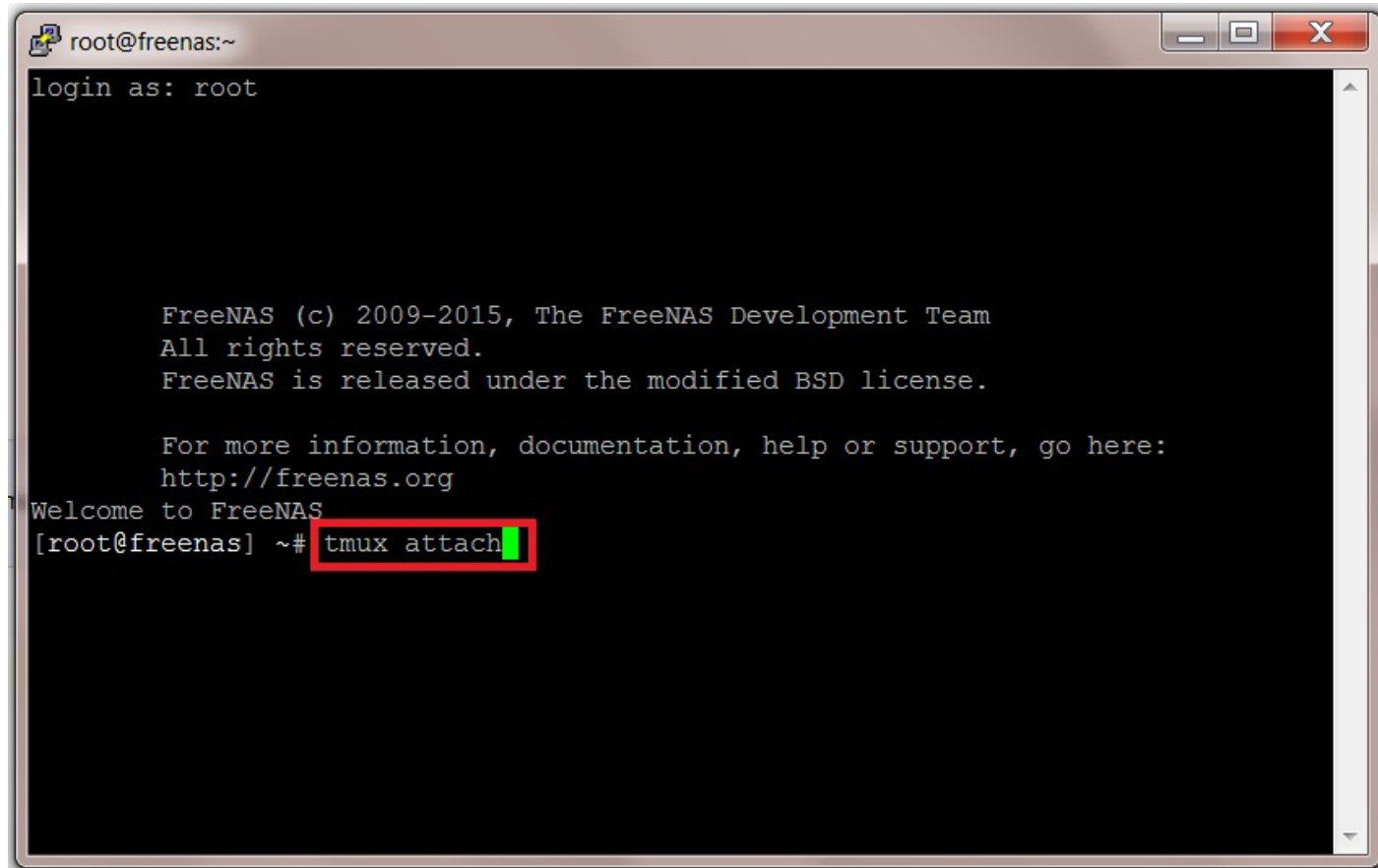
When you need to re-establish the connection with the tmux session/s simply start an SSH session and log in.

Type the following command in the command prompt.

```
tmux attach
```

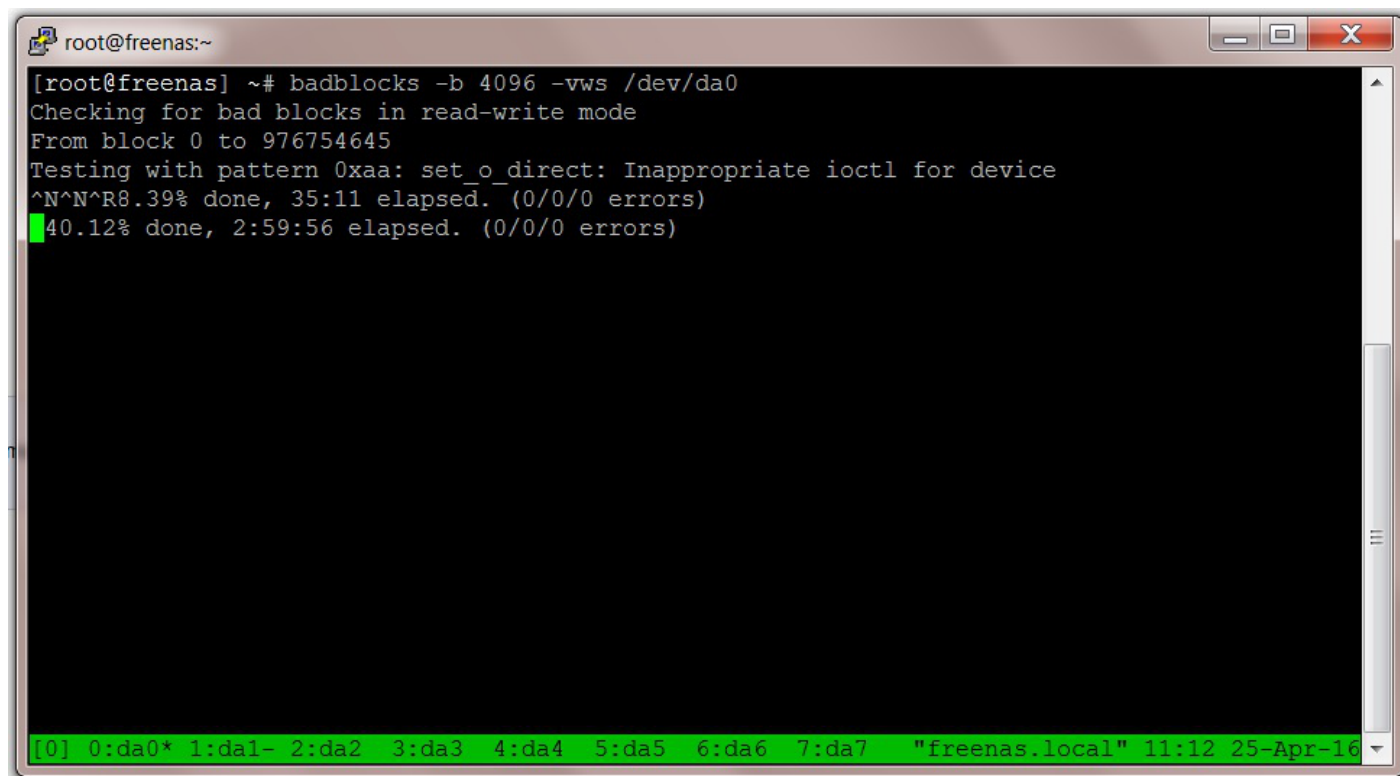
or simply

```
tmux a
```



```
root@freenas:~  
login as: root  
  
FreeNAS (c) 2009-2015, The FreeNAS Development Team  
All rights reserved.  
FreeNAS is released under the modified BSD license.  
  
For more information, documentation, help or support, go here:  
http://freenas.org  
Welcome to FreeNAS  
[root@freenas] ~# tmux attach
```

This should return you to the tmux session(s).



```
root@freenas:~  
[root@freenas] ~# badblocks -b 4096 -vws /dev/da0  
Checking for bad blocks in read-write mode  
From block 0 to 976754645  
Testing with pattern 0xaa: set_o_direct: Inappropriate ioctl for device  
^N^N^R8.39% done, 35:11 elapsed. (0/0/0 errors)  
40.12% done, 2:59:56 elapsed. (0/0/0 errors)  
  
[0] 0:da0* 1:da1- 2:da2 3:da3 4:da4 5:da5 6:da6 7:da7 "freenas.local" 11:12 25-Apr-16
```

When the tests are complete navigate to an open session, note the results if you need to and then type

the following into the command prompt.

```
exit
```

This will close that particular session in tmux.

Do this for each session in turn until you have exited all the sessions in tmux.

You will find that on exiting the last open session in tmux you will be returned to the standard SSH console (in Fester's case PuTTY).

Now reboot the server to reset the kernel geometry debug flags to their standard setting.

That's the Badblocks tests complete.

In order to complete all the HDD validation tests we must now repeat the SMART long tests. As this has already been documented I won't repeat it here. Just go back to the relevant section and repeat again.

Once the SMART long tests have completed then it is time to collect the results.

Getting Your Test Results

Getting your test results is always a tense moment.

(I remember such an instance in the doctor's examination room after an unforgettable trip to Bognor Regis, often referred to as "The Riviera of the South West". Unfortunately the doctor confirmed Fester had come back with more than just fond memories, but with the liberal application of a strong antibiotic cream Fester was as good as new in a couple of weeks.)

Here is how to get your results.

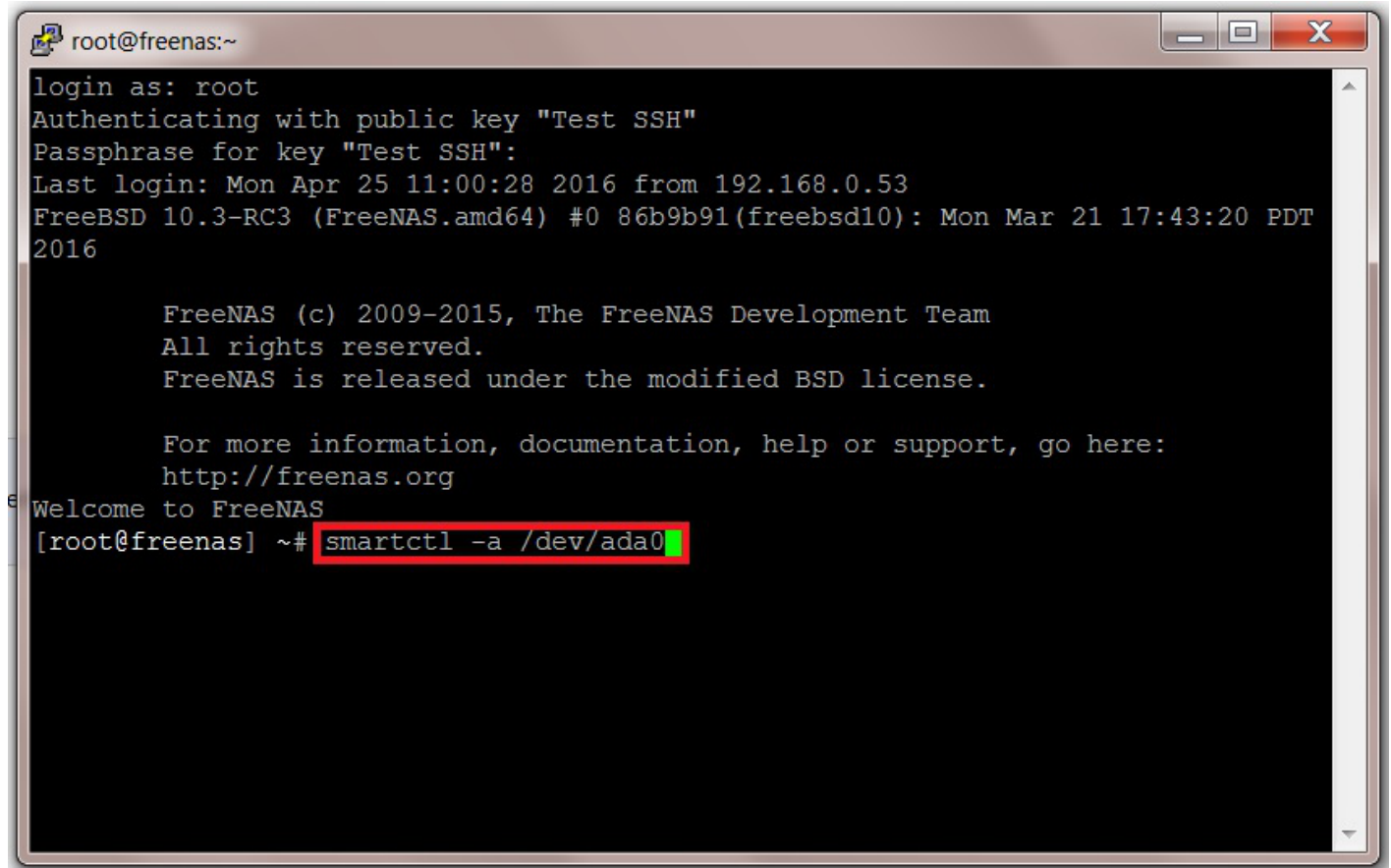
(Do not start this section until all HDD tests have been completed.)

Open an SSH console and log in.

We are going to issue a command to each HDD/SDD in succession that will interrogate and retrieve the results of the tests stored in each drives memory using SMART commands.

At the command prompt type in the following command using the name of the first drive you want to interrogate (in Fester's case this is ada0).

```
smartctl -a /dev/ada0
```

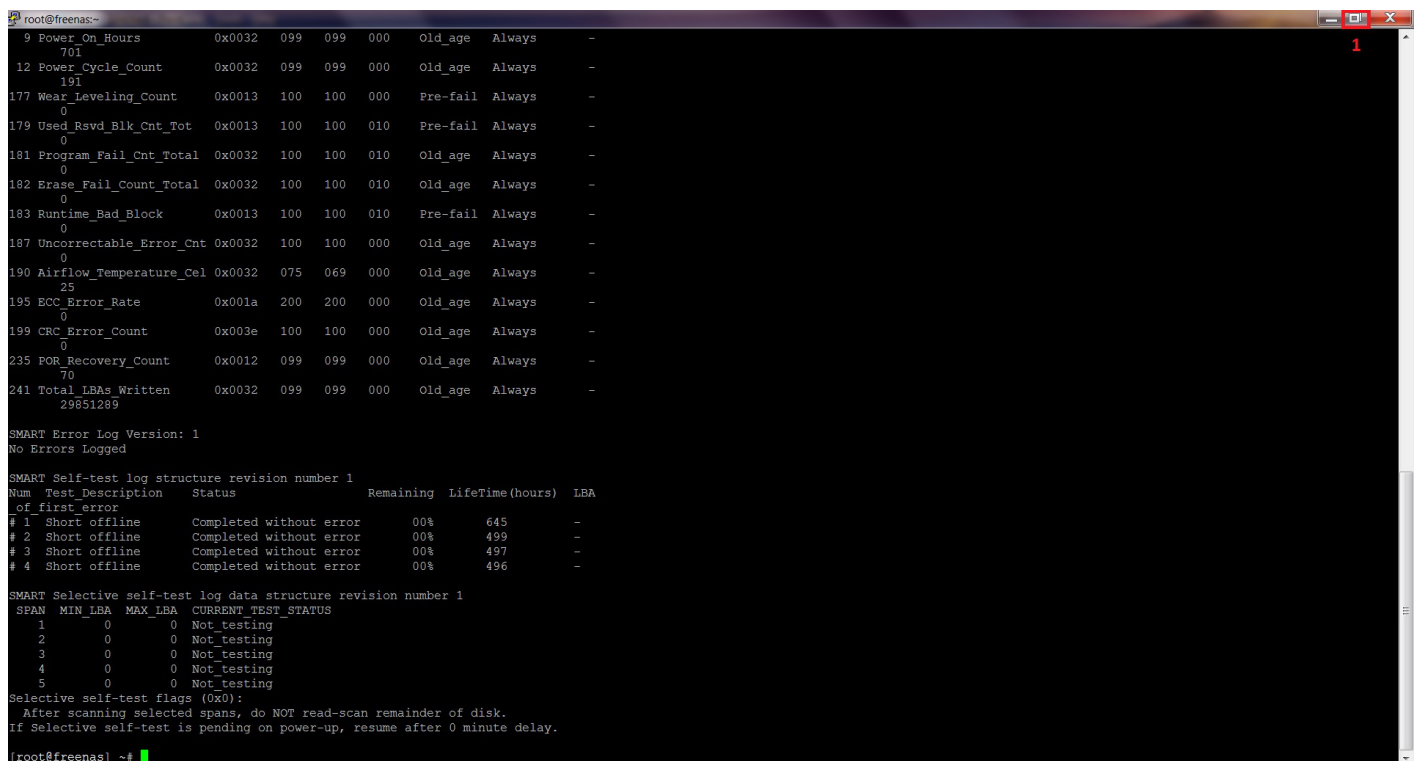


```
root@freenas:~
login as: root
Authenticating with public key "Test SSH"
Passphrase for key "Test SSH":
Last login: Mon Apr 25 11:00:28 2016 from 192.168.0.53
FreeBSD 10.3-RC3 (FreeNAS.amd64) #0 86b9b91(freebsd10): Mon Mar 21 17:43:20 PDT
2016

FreeNAS (c) 2009-2015, The FreeNAS Development Team
All rights reserved.
FreeNAS is released under the modified BSD license.

For more information, documentation, help or support, go here:
http://freenas.org
Welcome to FreeNAS
[root@freenas] ~# smartctl -a /dev/ada0
```

This should produce the following screen with the test results. The window displaying the information has been maximised (1) so it is easier to read.



```
root@freenas:~
9 Power_On_Hours          0x0032 099 099 000 Old_age Always -
701
12 Power_Cycle_Count     0x0032 099 099 000 Old_age Always -
191
177 Wear_Leveling_Count   0x0013 100 100 000 Pre-fail Always -
0
179 Used_Rsvd_Blk_Cnt_Tot 0x0013 100 100 010 Pre-fail Always -
0
181 Program_Fail_Cnt_Total 0x0032 100 100 010 Old_age Always -
0
182 Erase_Fail_Count_Total 0x0032 100 100 010 Old_age Always -
0
183 Runtime_Bad_Block      0x0013 100 100 010 Pre-fail Always -
0
187 Uncorrectable_Error_Cnt 0x0032 100 100 000 Old_age Always -
0
190 Airflow_Temperature_Cel 0x0032 075 069 000 Old_age Always -
25
195 ECC_Error_Rate        0x001a 200 200 000 Old_age Always -
0
199 CRC_Error_Count       0x003e 100 100 000 Old_age Always -
0
235 POR_Recovery_Count    0x0012 099 099 000 Old_age Always -
70
241 Total_LBAs_Written    0x0032 099 099 000 Old_age Always -
29851289

SMART Error Log Version: 1
No Errors Logged

SMART Self-test log structure revision number 1
# Num Test_Description Status Remaining LifeTime(hours) LBA
of first_error
# 1 Short offline Completed without error 00% 445 -
# 2 Short offline Completed without error 00% 439 -
# 3 Short offline Completed without error 00% 437 -
# 4 Short offline Completed without error 00% 436 -

SMART Selective self-test log data structure revision number 1
#SPAN MIN_LBA MAX_LBA CURRENT_TEST_STATUS
1 0 0 Not_testing
2 0 0 Not_testing
3 0 0 Not_testing
4 0 0 Not_testing
5 0 0 Not_testing

Selective self-test flags (0x0):
After scanning selected spans, do NOT read-scan remainder of disk.
If Selective self-test is pending on power-up, resume after 0 minute delay.

[root@freenas] ~#
```

At this point Fester copies the information and pastes it into a text editor for ease of use.

If you want to do this then select the text in the SSH console by clicking with the left mouse button where you want to begin, hold it down and then highlight the text you want to include.

When you have done this press the "Ctrl" button and the "v" button together. This keystroke will copy the highlighted text into the clip board.

Open the text editor you wish to use (Fester uses Notepad in Windows) and paste it into the text into the editor.

You now need to repeat this process for the next drive in your system.

At the command prompt type in the following command using the name of the next drive you want to interrogate (in Fester's case this is da0).

```
smartctl -a /dev/da0
```

This will produce the next set of results in the SSH console. Copy and paste as before (if you want to).

Now repeat the process for the next drive and the next until all the drives have been interrogated and their data copied and pasted.

(In this way you will build up a list of each drives test results in a single text file that can be saved for examination later.)

In Festers case this would mean issuing the following commands in the SSH console.

```
smartctl -a /dev/da1
```

```
smartctl -a /dev/da2
```

```
smartctl -a /dev/da3
```

```
smartctl -a /dev/da4
```

```
smartctl -a /dev/da5
```

```
smartctl -a /dev/da6
```

```
smartctl -a /dev/da7
```

These commands produce copious amounts of information about the drives. If you want something a little less gregarious then use this command instead (don't forget to change the drive name each time, and note the capital -A rather than the lowercase -a).

```
smartctl -A /dev/ada0
```

This should produce a screen that looks something like this (much more compact).

```

root@freenas:~# smartctl -A /dev/ada0
smartctl 6.4 2015-06-04 r4109 [FreeBSD 10.3-RC3 amd64] (local build)
Copyright (C) 2002-15, Bruce Allen, Christian Franke, www.smartmontools.org

=== START OF READ SMART DATA SECTION ===
SMART Attributes Data Structure revision number: 1
Vendor Specific SMART Attributes with Thresholds:
ID# ATTRIBUTE_NAME          FLAG     VALUE WORST THRESH TYPE      UPDATED  WHEN_FAILED RAW_VALUE
  5 Reallocated_Sector_Ct     0x0033   100    100   010   Pre-fail Always    -         0
  9 Power_On_Hours            0x0032   099    099   000   Old_age  Always    -        702
 12 Power_Cycle_Count         0x0032   099    099   000   Old_age  Always    -        191
177 Wear_Leveling_Count       0x0013   100    100   000   Pre-fail Always    -         0
179 Used_Rsvd_Blk_Cnt_Tot    0x0013   100    100   010   Pre-fail Always    -         0
181 Program_Fail_Cnt_Total    0x0032   100    100   010   Old_age  Always    -         0
182 Erase_Fail_Count_Total    0x0032   100    100   010   Old_age  Always    -         0
183 Runtime_Bad_Block         0x0013   100    100   010   Pre-fail Always    -         0
187 Uncorrectable_Error_Cnt   0x0032   100    100   000   Old_age  Always    -         0
190 Airflow_Temperature_Cel  0x0032   075    069   000   Old_age  Always    -         25
195 ECC_Error_Rate            0x001a   200    200   000   Old_age  Always    -         0
199 CRC_Error_Count          0x003e   100    100   000   Old_age  Always    -         0
235 POR_Recovery_Count        0x0012   099    099   000   Old_age  Always    -         70
241 Total_LBAs_Written        0x0032   099    099   000   Old_age  Always    -       29851289

[root@freenas] ~#

```

So you have now gathered your results, but they make about as much sense as a bacon butty at a bar mitzvah.

What now?

Making Sense of SMART Data

When looking at SMART data from a SMART storage device certain entries in the table are not important in terms of data integrity and health. They just give general information (e.g. Model, serial number, etc) and other types of information that could be useful in certain circumstances.

Other entries are very important and should immediately ring alarms bells if certain values are present.

In terms of HDD/SDD hardware validation these are the entries in the SMART data you need to scrutinise.

ID#	ATTRIBUTE_NAME	FLAG	VALUE	WORST	THRESH	TYPE	UPDATED	WHEN_FAILED	RAW_VALUE
1	Raw_Read_Error_Rate	0x002f	200	200	051	Prefail	Always	-	0
5	Reallocated_Sector_Ct	0x0033	200	200	140	Prefail	Always	-	0
7	Seek_Error_Rate	0x002e	200	200	000	Old_age	Always	-	0
10	Spin_Retry_Count	0x0032	100	100	000	Old_age	Always	-	0
11	Calibration_Retry_Count	0x0032	100	100	000	Old_age	Always	-	0

196	Reallocated_Event_Count	0x0032	200	200	000	Old_age	Always	-	0
197	Current_Pending_Sector	0x0032	200	200	000	Old_age	Always	-	0
198	Offline_Uncorrectable	0x0030	100	253	000	Old_age	Always	-	0
199	UDMA_CRC_Error_Count	0x0032	200	200	000	Old_age	Always	-	0

If you get any value other than zero in the “RAW VALUE” for these entries you should be suspicious of this drive and may need to return the device for testing depending on the manufacturer’s warranty.

Another area you should look at is the “SMART Self-test log structure”. Here is an example. It will tell you if the drive passed its tests. Any result other than “Completed without error” is cause for concern.

SMART Self-test log structure revision number 1

Num	Test_Description	Status	Remaining LifeTime(hours)	LifeTime(hours)	LBA_of_first_error
# 1	Extended offline	Completed without error	00%	503	-
# 2	Conveyance offline	Completed without error	00%	494	-
# 3	Short offline	Completed without error	00%	75	-

(If Fester is misinformed about interpreting SMART data or has omitted something important please let me know and I will try to put it in the guide or you could replace this or any section with your own?)

That’s the HDD/SDD validation completed. Now it is time to reinstall FreeNAS and create a basic server.

From:

<https://www.familybrown.org/dokuwiki/> - danb35's Wiki

Permanent link:

https://www.familybrown.org/dokuwiki/doku.php?id=fester:hvalid_hdd

Last update: **2017/06/24 17:46**

