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 tmux 9
Detaching The Storage Volume Before A Badblocks Test (DESTRUCTIVE Method) 9
Detaching The Storage Volume Before A Badblocks Test (Non-Destructive Method) 9
Importing An Unencrypted Volume After A Non-Destructive Badblocks Test 9
Importing An Encrypted Volume After A Non-Destructive Badblocks Test 9
Destructive Badblocks Test Using tmux 9
Non-Destructive Badblocks Test Using tmux 9
Stopping A Badblocks Test In tmux 9
Resuming A Session In tmux 9
Getting Your Test Results 9
Making Sense of SMART Data 9

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

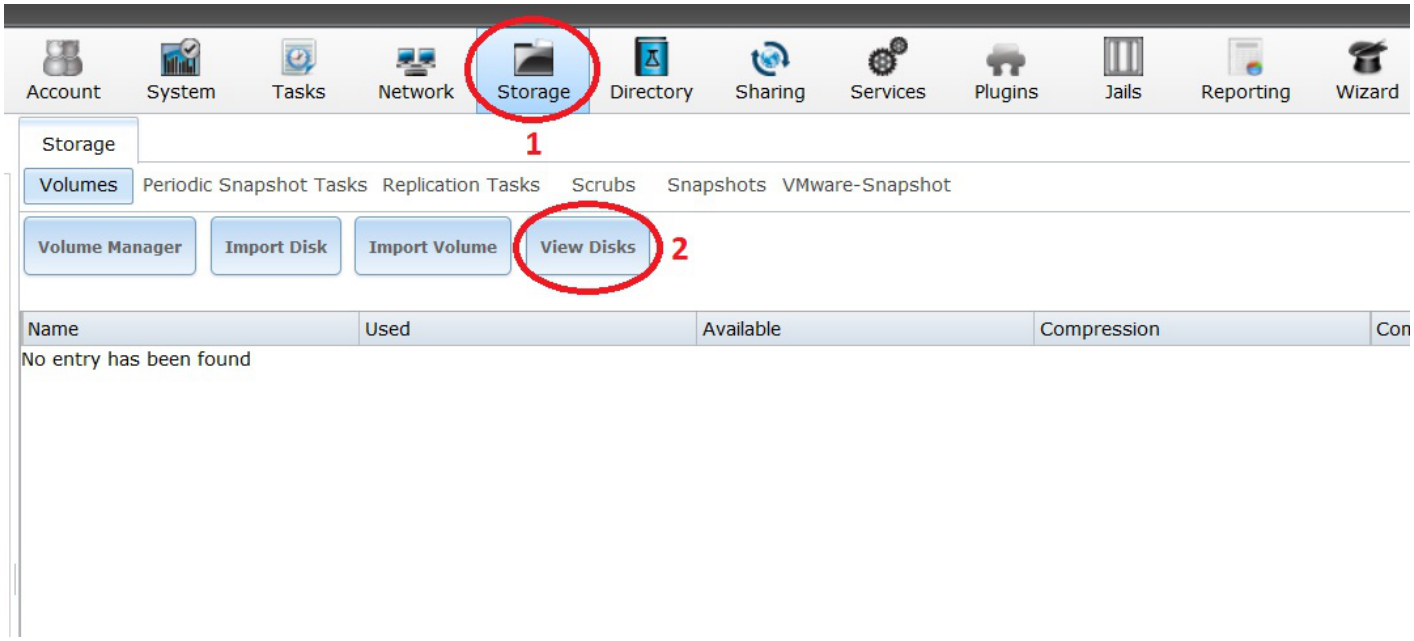
FreeNAS comes with certain software tools and capabilities built into it that will make the task of HDD/SDD validation much easier. This is why we needed to install it before conducting the tests.

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).

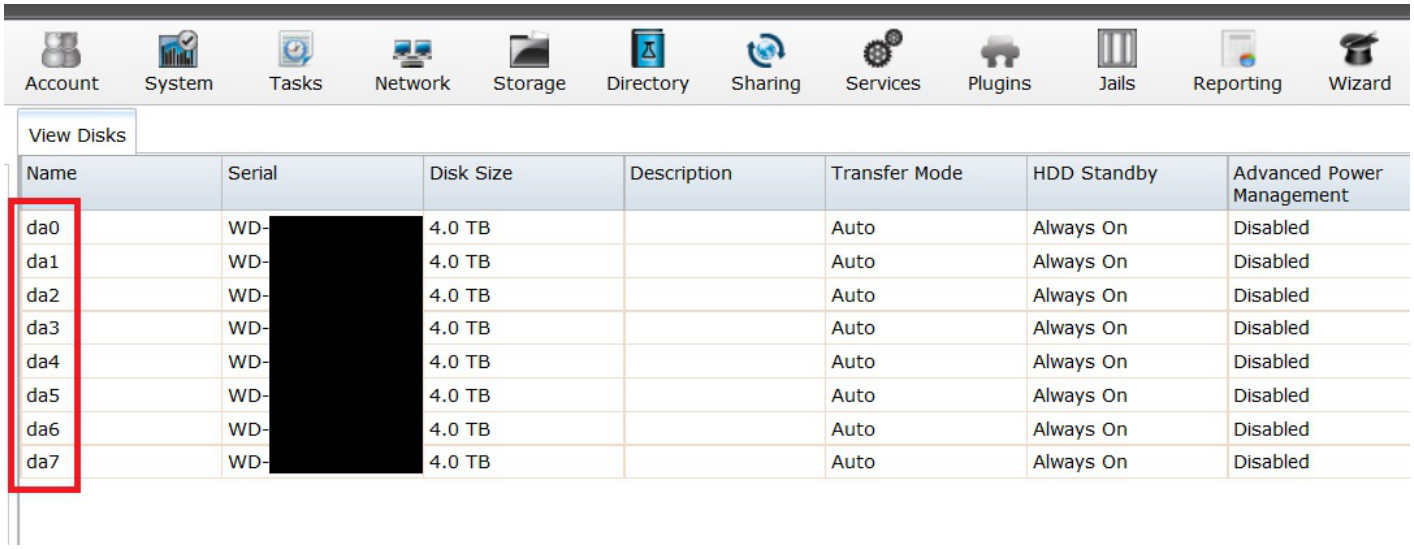


This should bring up a list of the storage HDDs (i.e. for data, not the OS) that are currently in your system.

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).

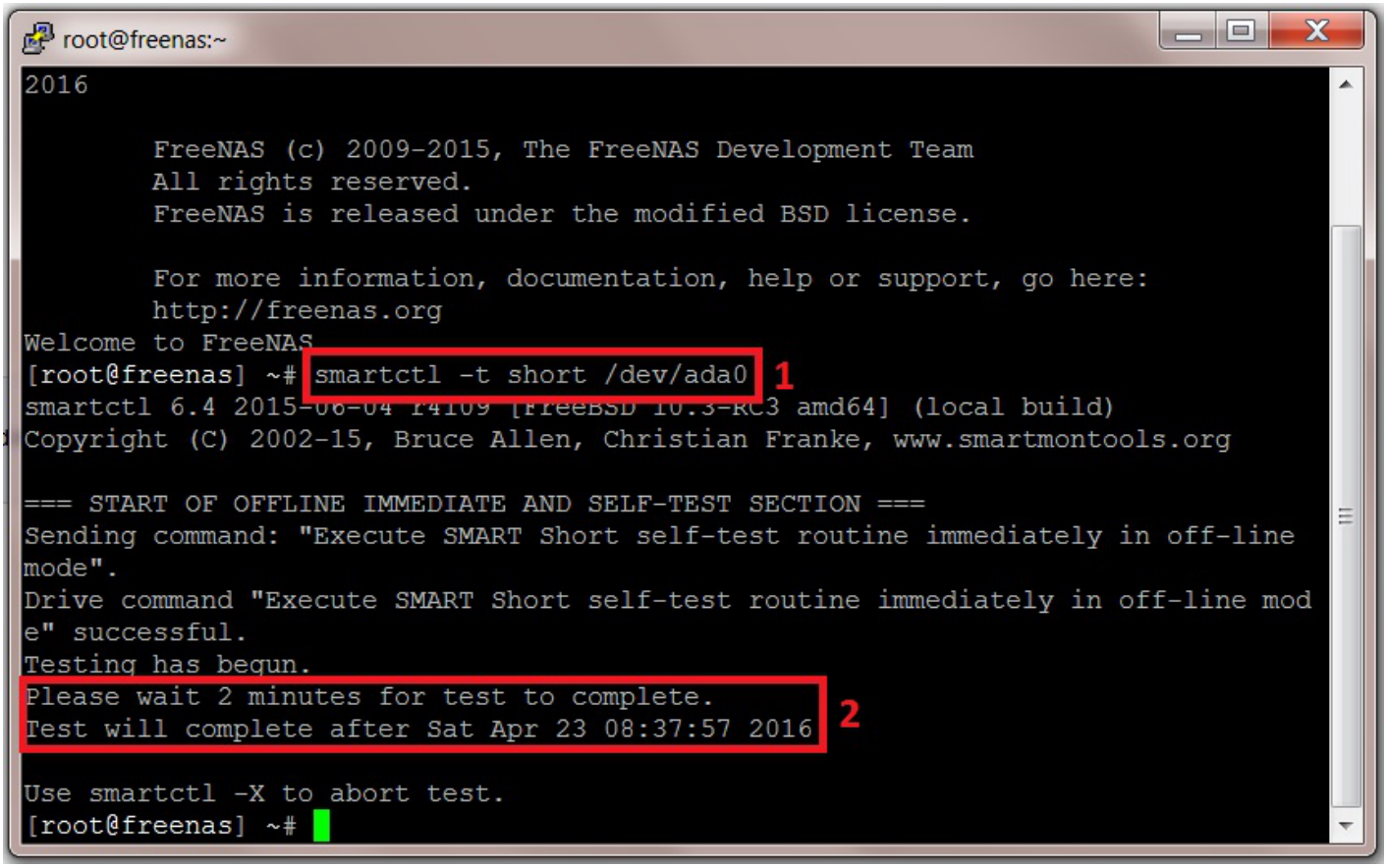
If you do it manually then commands entered at the prompt must be exactly as shown including all the spaces or they tend not to work.

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).



(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:

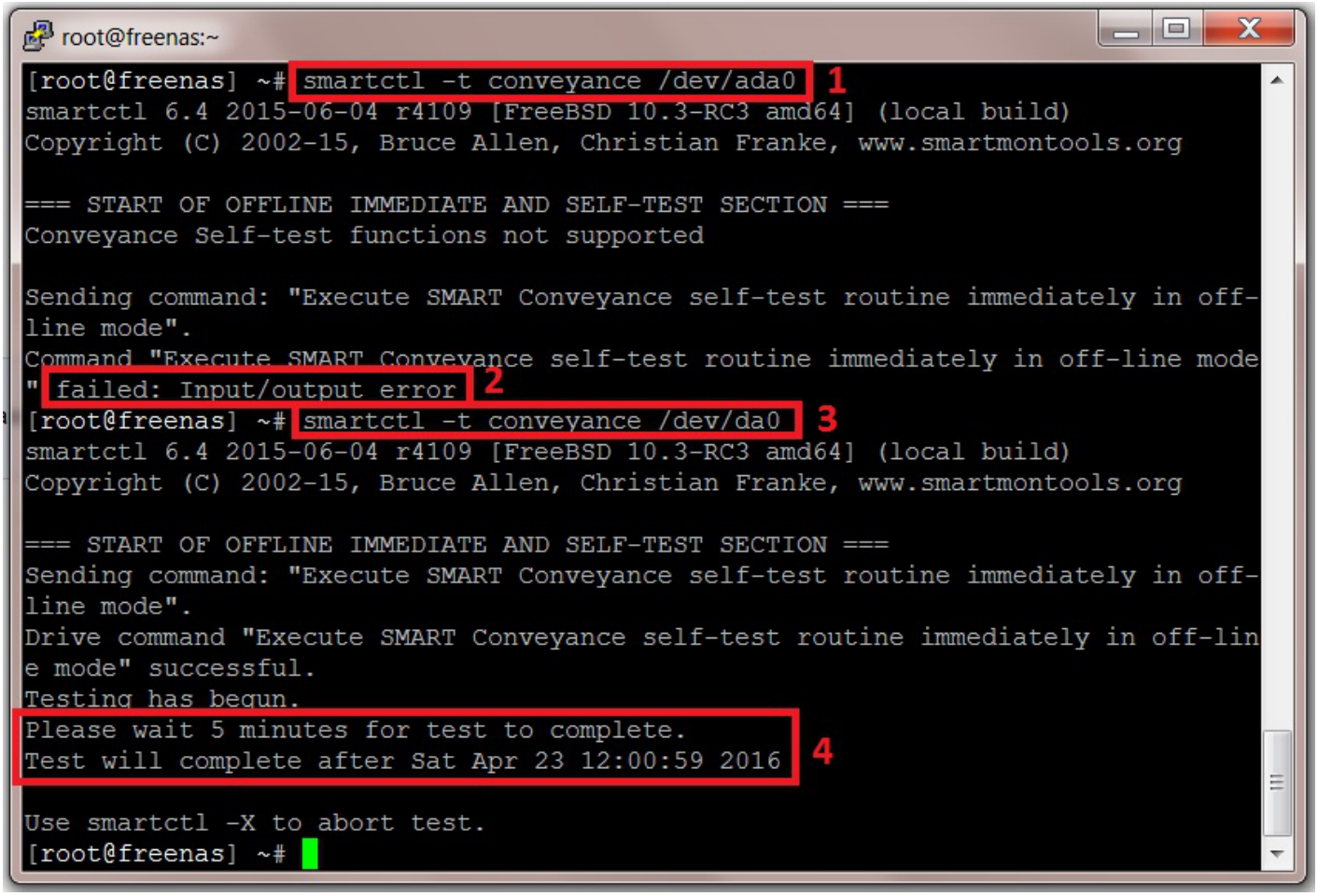
```
smartctl –t conveyance /dev/ada0
```

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).



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:

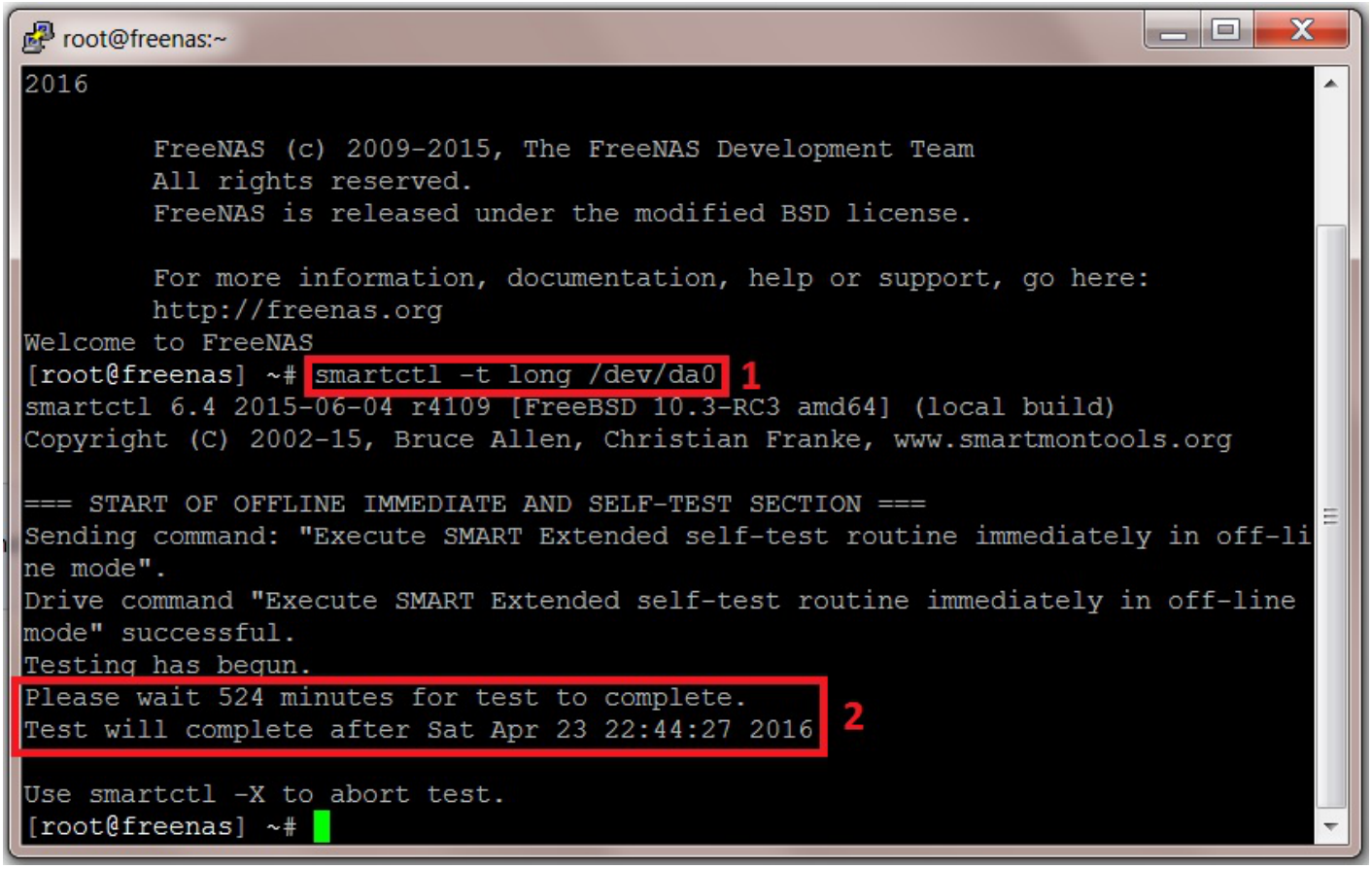
```
smartctl –t long /dev/ada0
```

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).



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

FreeNAS Storage Volumes And A Known Problem With tmux

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

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

Importing An Unencrypted Volume After A Non-Destructive Badblocks Test

Importing An Encrypted Volume After A Non-Destructive Badblocks Test

Destructive Badblocks Test Using tmux

Non-Destructive Badblocks Test Using tmux

Stopping A Badblocks Test In tmux

Resuming A Session In tmux

Getting Your Test Results

Making Sense of SMART Data

From:
https://familybrown.org/dokuwiki/ - **danb35's Wiki**

Permanent link:
https://familybrown.org/dokuwiki/doku.php?id=fester:hvalid_hdd&rev=1498317323

Last update: **2017/06/24 15:15**