# Re: 2.4.23+atalib2 sil3112A write errors

***Source:*** http://linux.derkeiler.com/Mailing−Lists/Kernel/2004−01/2161.html

---

***From:*** Bryan Andersen (*bryan_at_bogonomicon.net*)
***Date:*** 01/09/04

```
Date:   Thu, 08 Jan 2004 17:45:24 −0600
To: Bryan Andersen <bryan@nerdvest.com>
```

Sorry for the false alarm. I'm now suspecting hardware errors. I took a look at the actual errors introduced into the files. When I octal dump both the source file and destination file and diff them I'm only seeing a short string of bytes changed. This is telling me I'm getting multibit errors slipping though. Time to start swapping cables and hardware. Is there a way to get the driver to output messages when errors are seen rather than messages for all transfers?

This is an example of the differences seen.

72922,72923c72922,72923
< 4357500 064047 120335 134421 006545 137622 023477 043620 164561
< 4357520 106514 023757 026270 043466 051034 117400 174451 127107

```
−−−
 > 4357500 064047 120335 134421 006545 136566 035207 015425 133770
 > 4357520 012031 012055 171154 045114 015406 000160 017147 035214
− Bryan
Bryan Andersen wrote:
> I'm seeing silent write errors with two Seagate 160GB drives on a
> sil3112A SATA controller on an ASUS A7N8X−Deluxe motherboard, but I'm
> not seeing any read errors.  Each drive is on it's own cable.  Kernel is
> 2.4.23 release with the 2.4.23−libata2 patch and some patches for using
> MythTV applied.  (I also see the same problem under 2.4.24+libata only)
>  I'm also not seeing any error messages in any log files or the kernel
> dmesg output.  The error rate looks to be around 1 in a million blocks.
>  My current guess is the block or blocks just didn't get written by the
> drive as when the system reads in the blocks with the bad data I'm not
> seeing any read error messages.
>
> I am now running the tests again using jfs as the filesystem rather than
> ext3 to rule out the file system as the cause.  As part of this test I'm
> also zeroing the disks before the test and then checking for non zero
> data and how large the non zero data blocks are.  Given enough time I
> may run this test a couple of times to see if the positions of the bad
> data move about or stay put.
>
> How the first testes were run.
>
> I used "cp −a * /data10" as root to copy the data (150GB worth) to the
> disk under test.  Then I created md5sum lists for all files in both the
> source and destination and sorted and compaired the lists.  Differences
> were found between the lists, some files and directories were missing,
```

> or corrupted.  On fscking the disks some inodes were found corrupted.  I
> then copied only the corrupted files and after a few iterations I
> finally got a copy that was the same as the source.  I then ran a
> program to repeatadly md5sum checksum all the fi