

## mark5 machines reboot themselves randomly

Starting with SDK9.4 mark5 machines have rebooted themselves randomly when running mk5daemon. The logs give no clear indication about the reason for the reboot however in general one finds many entries bz the EDAC services of the sort:

```
Jul  5 09:27:45 mark5fx05 kernel: EDAC MC0: 1 UE Read error on unknown
memory (branch:1 channel:1 slot:0 page:0x0 offset:0x0 grain:0 - Bank=0
RAS=0 CAS=0 FATAL Err=0x7 ((null)))
Jul  5 09:28:05 mark5fx05 kernel: EDAC MC0: 1 UE Read error on unknown
memory (branch:1 slot:0 page:0x0 offset:0x0 grain:0 - Rank=0 Bank=0 RAS=0
CAS=0, UE Err=0x1ff ((null)))
Jul  5 09:28:05 mark5fx05 kernel: EDAC MC0: INTERNAL ERROR: branch value
is out of range (2 >= 2)
```

This is very likely due to a known EDAC bug (see e.g. [https://www.thomas-krenn.com/de/wiki...Linux\\_Systemen](https://www.thomas-krenn.com/de/wiki...Linux_Systemen)). In any case the EDAC module should be blacklisted on all mark5 machines:

Identify the edac module:

```
lsmod | grep edac
```

Edit /etc/modprobe.d/blacklist.conf and insert the module found in the previous step

```
blacklist i5000_edac
```

Unload the kernel module:

```
rmmod i5000_edac
```

## m5dir fails with "illegal directory size"

When running m5dir and receiving an error message like this:

```
Module directory parse error 1 encountered: Illegal directory size
The directory size was 5244512
The best guess about the directory content is printed below:
This module contains a NeoLegacy directory (capable of storing up to
65536 scans):
  Number of scans on the directory = 342
  Directory version = 2 subversion = 7
  jive5ab format designator = Mark5B16DisksSDK9BankB
  DiFX signature = 1773171562
```

The binary directory was dumped to /tmp/dir.dump

```
Error: Directory read for module GSFC+026 unsuccessful, error code=-3
FYI: Not setting disk module state to Played for GSFC+026
```

fuse mount the module

```
fuseMk5 -f /tmp/dir.dump /mark5fxXX
```

Check that the scans are properly mounted and can be decoded e.g.

```
m5findformat /mark5fxXX/scan
or
printVDIFheader /mark5fxXX/scan
```

if everythin looks OK you can write back the binary directory created by m5dir to the module

```
writeuserdir /tmp/dir.dump
```

## Reconstructing a Mark5 user directory

When the user directory on a module has been corrupted for whatever reason, there are two possible ways to recover it, using utilities included in "fuseMk5". Place the problematic module as the *only* module into a Mark5 unit. Then:

If the module was previously imported into DiFX:

```
mkdir tmp; cd tmp
cp -a /cluster/difx/directories/<modulename>.dir .
/cluster/mark5/fuseMk5/fuseMk5-cvs/difxdirfile2userdir.py
<modulename>.dir newdir.bin
/cluster/mark5/fuseMk5/fuseMk5-cvs/fuseMk5 --udread newdir.bin /mnt/
diskpack
ls /mnt/diskpack
# If files under /mnt/diskpack looked reasonable you can write newdir.bin
onto module:
fusermount -u /mnt/diskpack
/cluster/mark5/fuseMk5/fuseMk5-cvs/writeuserdir newdir.bin
```

If only a FieldSystem log file exists:

```
mkdir tmp; cd tmp

cp <fieldsystemlog>.log fslog.log
(cat expt_part_2.log >> fslog.log) # if there are multiple experiments
on a module
(cat expt_part_3.log >> fslog.log) # ...

/cluster/mark5/fuseMk5/fuseMk5-cvs/fslog2userdir.py <fieldsystemlog>.log
newdir.dir

/cluster/mark5/fuseMk5/fuseMk5-cvs/fuseMk5 --udread newdir.bin /mnt/
diskpack
```

```
ls /mnt/diskpack
```

```
# If files under /mnt/diskpack looked reasonable you can write newdir.bin
onto module:
```

```
fusermount -u /mnt/diskpack
```

```
/cluster/mark5/fuseMk5/fuseMk5-cvs/writeuserdir newdir.bin
```

## Mark5 Module recovery

Starting from a certain Conduant SDK version (uncertain which), the Conduant card firmware has lost its ability to gracefully play back modules that contain one or more corrupt/dead disks. Such modules will either freeze the Mark 5, or will play back extremely slowly with zero data.

### [Recovery procedure](#)

## Upgrade SDK Version

insert a module into slot A

**BEWARE: data on the module might become deleted, so make sure it contains no valid data !**

log-into the mark5 machine as user root, then execute:

```
cd /usr/local/src/streamstor/linux/util/
./ssflash -u SDK9.3.ssf
```

make sure no errors are reported during the flashing process

run ssopen and sstest:

```
./ssinfo
./sstest
```

make sure no errors are reported

Update the sticker on the chassis of the mark5 to indicate the new version of SDK.