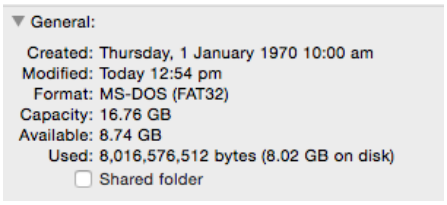
…Testing out other USB sticks on the c64 mini

- In light of so many reports of USB sticks failing on the C64 mini, I thought I'd try test out some of Camcam's collection.

I did a "**Get Info**" on it on my macbook, and it reports:



Ok, it already is MS-DOS (FAT32), a 16GB stick, I'll drop a file on it and see how it goes...

Aah, interesting, it didn't work, the disk dir listing shows up as "READONLY" and empty...

Ah ok then, I will run "**diskutil list**", it reports the disk as follows:

```
/dev/disk2
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:                            CAMCAM                 *16.8 GB    disk2
```

Aah interesting, only one entry here. I suspect there is no MBR and this is an un-partitioned drive, and it's looking to be a likely reason for why it is failing. But let me confirm this by assessing the drive-contents with dd first...

TROUBLESOME USB STICK

`dd if=/dev/disk2 bs=1 count=512 2> /dev/null | hexdump -C -v | less`

```
00000000  eb 58 90 4d 53 44 4f 53  35 2e 30 00 02 10 26 00  |.X.MSDOS5.0...&.|
00000010  02 00 00 00 00 f8 00 00  3f 00 ff 00 00 00 00 00  |........?.......|
00000020  00 00 f4 01 71 3e 00 00  00 00 00 00 02 00 00 00  |....q>..........|
00000030  01 00 06 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000040  00 00 29 03 83 4a 00 44  49 53 4b 5f 49 4d 47 20  |..)..J.DISK_IMG |
00000050  20 20 46 41 54 33 32 20  20 20 33 c9 8e d1 bc f4  |  FAT32   3.....|
00000060  7b 8e c1 8e d9 bd 00 7c  88 4e 02 8a 56 40 b4 08  |{......|.N..V@..|
00000070  cd 13 73 05 b9 ff ff 8a  f1 66 0f b6 c6 40 66 0f  |..s......f...@f.|
00000080  b6 d1 80 e2 3f f7 e2 86  cd c0 ed 06 41 66 0f b7  |....?.......Af..|
00000090  c9 66 f7 e1 66 89 46 f8  83 7e 16 00 75 38 83 7e  |.f..f.F..~..u8.~|
000000a0  2a 00 77 32 66 8b 46 1c  66 83 c0 0c bb 00 80 b9  |*.w2f.F.f.......|
000000b0  01 00 e8 2b 00 e9 48 03  a0 fa 7d b4 7d 8b f0 ac  |...+..H...}.}...|
000000c0  84 c0 74 17 3c ff 74 09  b4 0e bb 07 00 cd 10 eb  |..t.<.t.........|
000000d0  ee a0 fb 7d eb e5 a0 f9  7d eb e0 98 cd 16 cd 19  |...}....}.......|
000000e0  66 60 66 3b 46 f8 0f 82  4a 00 66 6a 00 66 50 06  |f`f;F...J.fj.fP.|
000000f0  53 66 68 10 00 01 00 80  7e 02 00 0f 85 20 00 b4  |Sfh.....~.... ..|
00000100  41 bb aa 55 8a 56 40 cd  13 0f 82 1c 00 81 fb 55  |A..U.V@........U|
00000110  aa 0f 85 14 00 f6 c1 01  0f 84 0d 00 fe 46 02 b4  |.............F..|
00000120  42 8a 56 40 8b f4 cd 13  b0 f9 66 58 66 58 66 58  |B.V@......fXfXfX|
00000130  66 58 eb 2a 66 33 d2 66  0f b7 4e 18 66 f7 f1 fe  |fX.*f3.f..N.f...|
00000140  c2 8a ca 66 8b d0 66 c1  ea 10 f7 76 1a 86 d6 8a  |...f..f....v....|
00000150  56 40 8a e8 c0 e4 06 0a  cc b8 01 02 cd 13 66 61  |V@............fa|
00000160  0f 82 54 ff 81 c3 00 02  66 40 49 0f 85 71 ff c3  |..T.....f@I..q..|
00000170  4e 54 4c 44 52 20 20 20  20 20 20 00 00 00 00 00  |NTLDR      .....|
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001a0  00 00 00 00 00 00 00 00  00 00 00 00 0d 0a 52 65  |..............Re|
000001b0  6d 6f 76 65 20 64 69 73  6b 73 20 6f 72 20 6f 74  |move disks or ot|
000001c0  68 65 72 20 6d 65 64 69  61 2e ff 0d 0a 44 69 73  |her media....Dis|
000001d0  6b 20 65 72 72 6f 72 ff  0d 0a 50 72 65 73 73 20  |k error...Press |
```

```
000001e0  61 6e 79 20 6b 65 79 20  74 6f 20 72 65 73 74 61  |any key to resta|
000001f0  72 74 0d 0a 00 00 00 00  00 ac cb d8 00 00 55 aa  |rt............U.|
```

Whoa, ok, this is not an MBR sector, this is a VBR/boot-sector, confirming that this is an un-partitioned drive.

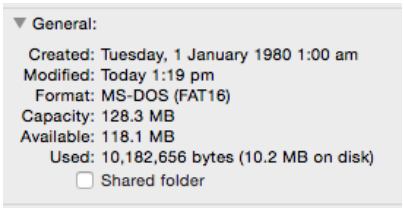
WORKING USB STICK

Ok then, here's my chance to compare the FAT32 structure of the working USB against the troublesome USB...

Here's the working USB stick:

All I can tell of it's maker is the term "**Miniking**".

Here's what "**Get Info**" says:



Wow, that's interesting, it is formatted as "**MS-DOS (FAT16)**", hehehe. It's a 128MB drive, so fairly small too.

Ok then, I'd like to assess the FAT structure of this working stick then...

I wanted to figure out the device-name in mac osx of the drives. I learnt online you can type "**diskutil list**" to learn this. My USB stick is shown as follows:

```
/dev/disk2
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *128.6 MB   disk2
   1:                 DOS_FAT_16                         128.6 MB   disk2s1
```

Hmm, ok then... This is looking like a partitioned drive, consisting of a single partition. So I could use dd to grab the info here and throw it into hexdump... Ok then, let's look at the first sector:

```
dd if=/dev/disk2 bs=1 count=512 2> /dev/null | hexdump -C -v | less
```

```
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
```

```
00000070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000000f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
00000190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001b0  00 00 00 00 00 00 00 00  54 53 c2 43 00 00 80 01  |........TS.C....|
000001c0  06 00 06 d9 30 17 35 00  00 00 00 cb d4 03 00 00  |....0.5.........|
000001d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
000001f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 55 aa  |..............U.|
```

Yep, this is the MBR, so it is a partioned drive with just a single partition entry on it. Let's assess the bytes within the 1st partition description:
80 01 06 00 06 d9 30 17 35 00 00 00 cb d4 03 00

- 80 = active/bootable partition
- 01 06 00 = chs address
  - break this out in binary form
  00000001 head = 1
  00000110 sector = 6
  00000000 cylinder = 0

  Note that:
  - cylinder range = 0 to 1023
  - head range = 0 to 255
  - sector range = 1 to 63

- To get an LBA address, wikipedia tells me the formula is as follows:
  CHS tuples can be mapped onto LBA addresses using the following formula:
  $A = (c \cdot N_{\text{heads}} + h) \cdot N_{\text{sectors}} + (s - 1)$,
  where A is the LBA address, N_{heads} is the number of heads on the disk, N_{sectors} is the maximum number of sectors per track, and (c, h, s) is the CHS address.

  Hmm, but in order for me to figure out this formula, I'd need to know Nheads and Nsectors... Bugger, how do I find that...

  I'm really not sure as yet, so let's give it a miss for now...

- `06` = partition type = FAT16B

- **d9 30 17** = last absolute sector in partition

  breaking into bits:

  11011001 head = 217
  00110000 sector = 48
  00010111 cylinder = 23

- **35 00 00 00** = LBA of first absolute sector in the partition
  - for now, let's assume sector-size = 512 bytes
  So, LBA start = 53*512 = 27136 = 0x6A00

    - As a test, I tried skipping into "**/dev/sda**" by 32256 bytes to see if that gives me the contents of "**/dev/sda1**":

      `dd if=/dev/disk2 bs=1 count=512 skip=27136 2> /dev/null | hexdump -C -v | less`

      ```
      00000000  eb 3e 90 2b 2e 4d 25 46  49 48 43 00 02 04 01 00  |.>.+.M%FIHC.....|
      00000010  02 00 02 00 00 f8 f5 00  30 00 da 00 35 00 00 00  |........0...5...|
      00000020  cb d4 03 00 80 01 29 00  00 00 00 00 00 00 00 00  |......)........|
      00000030  00 00 00 00 00 00 46 41  54 31 36 20 20 20 00 00  |......FAT16   ..|
      ```

      Compare this with the output of "**/dev/sda1**":

      `dd if=/dev/disk2s1 bs=1 count=512 2> /dev/null | hexdump -C -v | less`

```
00000000  eb 3e 90 2b 2e 4d 25 46  49 48 43 00 02 04 01 00  |.>.+.M%FIHC.....|
00000010  02 00 02 00 00 f8 f5 00  30 00 da 00 35 00 00 00  |........0...5...|
00000020  cb d4 03 00 80 01 29 00  00 00 00 00 00 00 00 00  |......).........|
00000030  00 00 00 00 00 00 46 41  54 31 36 20 20 20 00 00  |......FAT16   ..|
```

Aah, neato, they're the same, cool :) :)

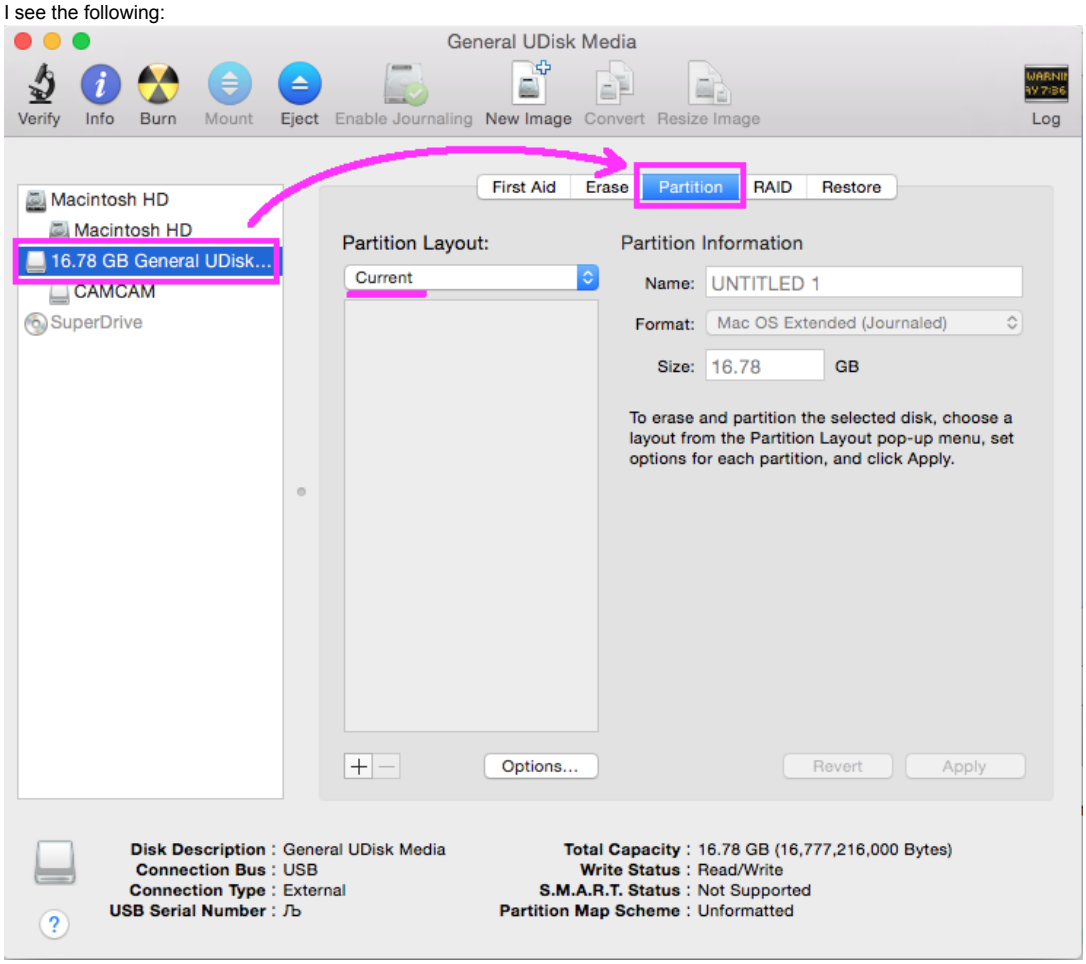
- **cb d4 03 00** = number of sectors in partition
  = 0x0003d4cb = 251083 sectors
  = 251083 x 512 = 128554496 bytes = 122 MB (approx)

Yeah, that all adds up fine...

- Hmm, ok then, the next thing I want to try is to convert my troublesome stick into a partitioned drive. I.e., force an MBR sector to be added somehow, via some partitioning tool. Wonder if any exists in the mac world...

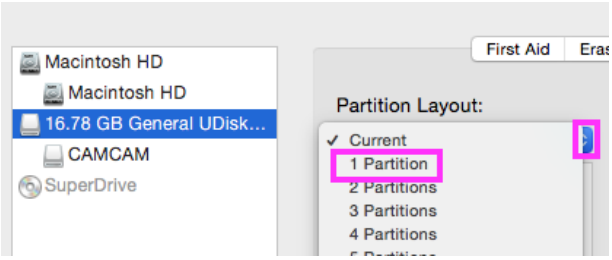
Actually, instead of that, I will try re-format the usb stick via my mac's formatter tool. This will be a good way to assess whether I encounter problems, as some mac users have reported (note that some mac users have had success too).

- Ok, I am running "**Disk Utility**"
- I click the entry for my USB stick "**16.78 GB General UDisk...**" in the left pane
- I click the "**Partition**" tab
- I see the following:

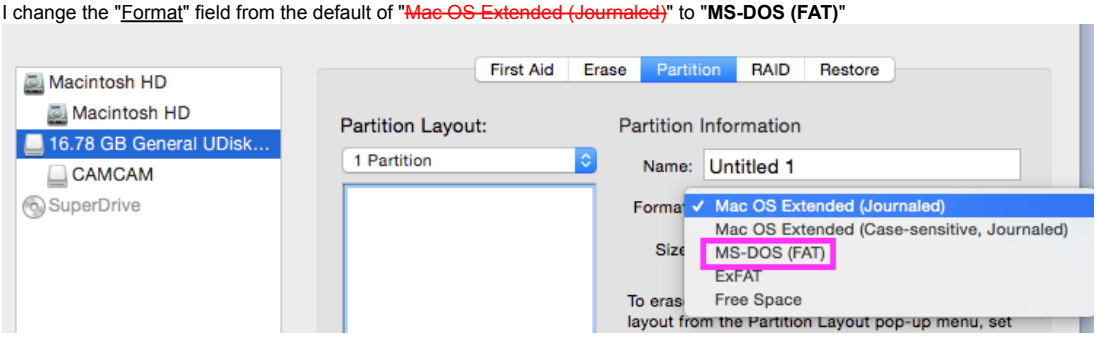


- For "Partition Layout", it just says "Current". To me, this appears to indicate that it is an unpartitioned drive.

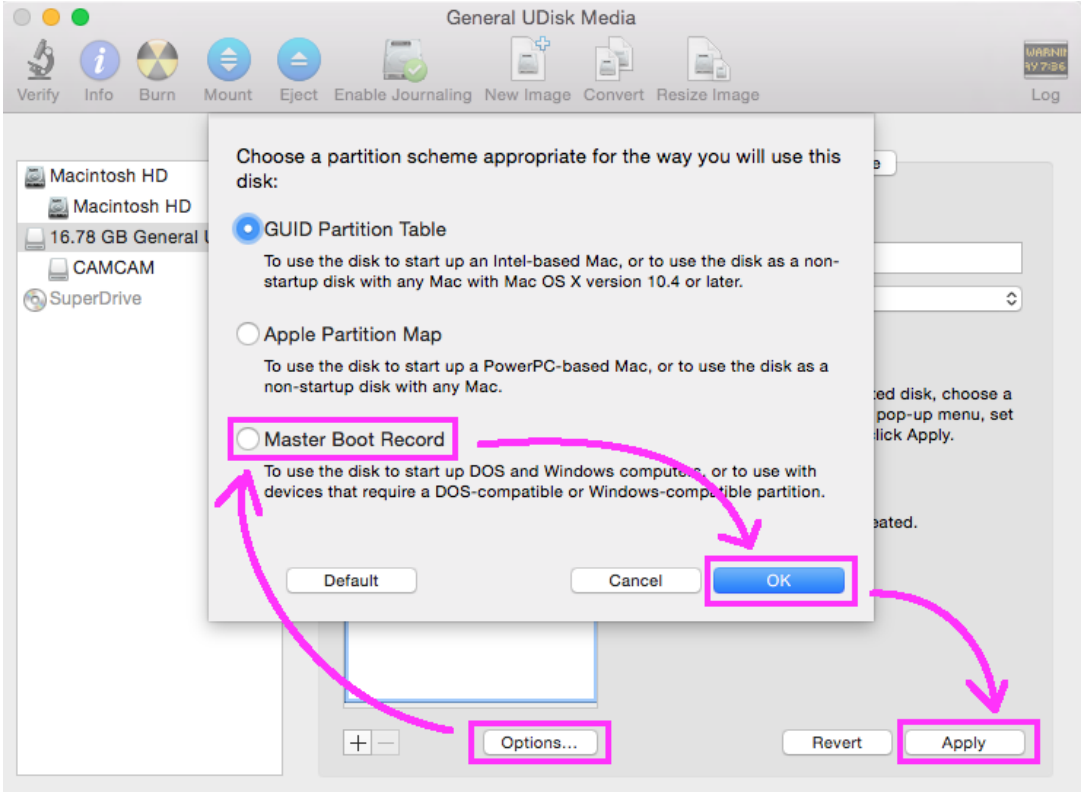
- I open up the combo-box to switch to "**1 Partition**"



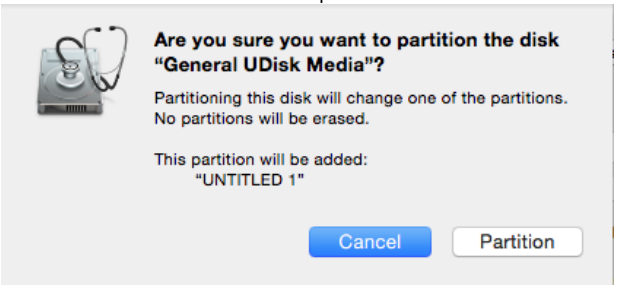
- I change the "Format" field from the default of "~~Mac OS Extended (Journaled)~~" to "**MS-DOS (FAT)**"



- I then click the "**Options...**" button and change the default "~~GUID Partition Table~~" to "**Master Boot Record**"
  - Then click "**OK**" button
  - Followed by "**Apply**" button

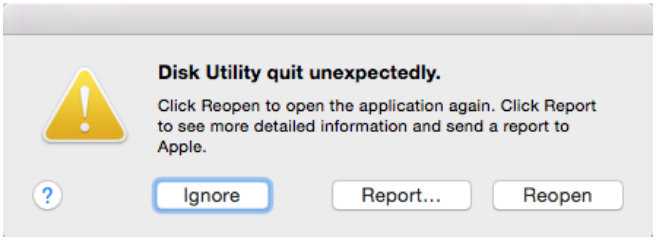


- I get a warning popup "**Are you sure you want to partition the disk "General UDisk Media"?**".
  - I click the "**Partition**" button to accept this.



Thankfully, I backed up the contents of the USB stick prior to doing this, so I'm not too worried if I lose everything on the stick...

Ah bugger, I got this error message though :(




```
Disk Utility quit unexpectedly.
Click Reopen to open the application again. Click Report to see more detailed information and send a report to Apple.
```

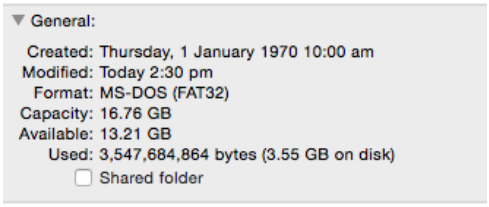
Aah, maybe I accidentally killed the app while it was still running, oops :) Let me try those steps again then...

Ok done. Now when I do "**diskutil list**", it reports:

```
/dev/disk2
   #:                       TYPE NAME                    SIZE       IDENTIFIER
   0:     FDisk_partition_scheme                        *16.8 GB    disk2
   1:                 DOS_FAT_32 UNTITLED 1             16.8 GB    disk2s1
```

Awesome, we now have a **disk2s1** entry for the first (and only) partition... My USB stick is now empty, so let me copy all my old contents back onto the stick now...

"**Get Info**" on the drive now reports the following, much the same as before...



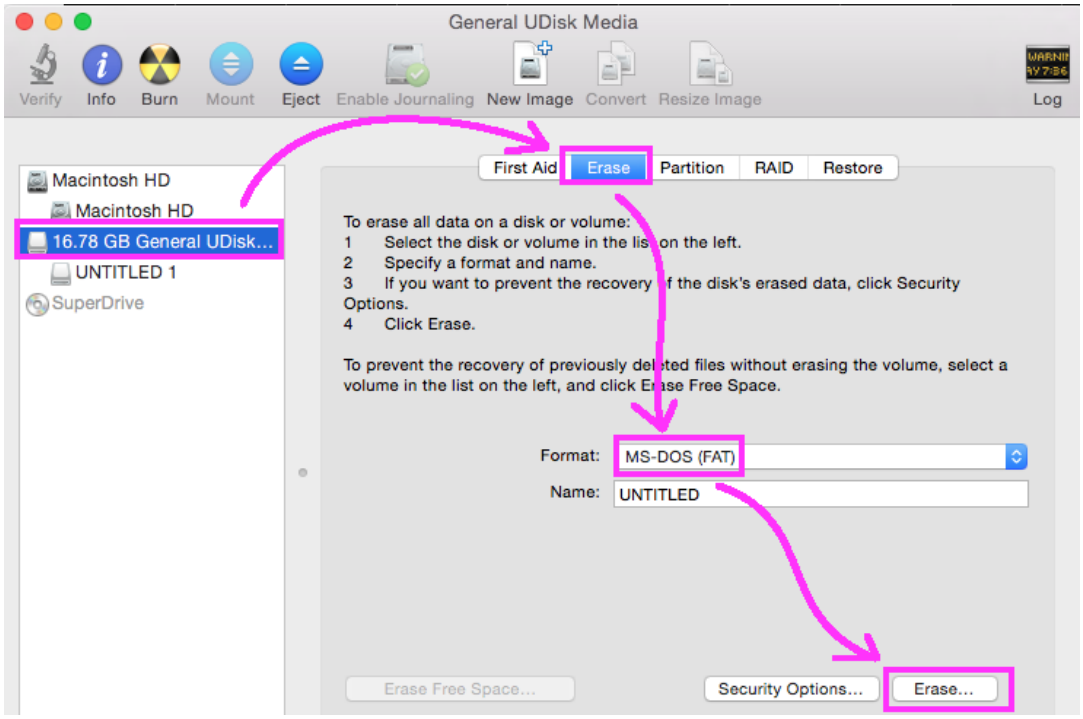
Now I'll try this stick on the c64 mini to see if it works...

Wow, to my surprise it still doesn't work! :(When I do a dir listing, all I see is **0 "**... Really odd... Wonder what else it could be then...

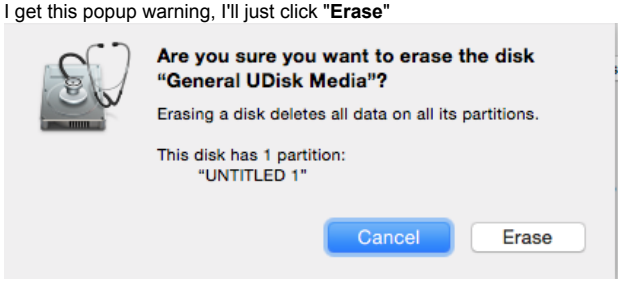
The USB drive has a huge dvd video on it too. So I will try erase that and just assure that the d64 is the only file on it...

Alrighty then, I'll do the following:

- Go to "**Disk Utility**" again.
- Select my "**16.78 GB General UDisk...**" drive on the left
- Select the "**Erase**" tab
- Assure that "Format" is set to "**MS-DOS (FAT)**"
- Click the "**Erase...**" button



- I get this popup warning, I'll just click "**Erase**"



Ok, the "**Get Info**" properties seem the same. I'll now copy across the .d64 file...

Aah, wait a sec, it looks like my .d64 file was a dud, as I see this same **0 "** directory listing in VICE :(Oops :)

Damn, looks like all my studies today were all for naught... :(

Aah wait, now I copied a working .d64 file over to the usb stick and then it worked! Awesome :)

So now, the only thing I need to do is to revert the USB stick to an un-partitioned drive and drop the .d64 file in again, to see if it works or not :)

Hmm, I think I might have to do this in linux then... with **mkfs.fat**?

Ok, I did this from my linux virtualbox with:

```
mkfs.msdos -I /dev/sdb
```

Awesome, that worked. I confirmed with dd+hexdump that there no longer is an MBR on the drive. I then copied over my working .d64 file...

Now to test on the device... Yep, when I do a **LOAD "$",8** dir listing, it now returns:

```
0 "READONLY      " 01 2A
```

Ok, I think this is now adding more weight to the theory that USB drives that are formatted without an MBR will fail on the device.

So it's looking like all USB devices should be formatted with an MBR.

- It looks like Mac OSX is capable of this, but it isn't the default, so the user will need to assure they've selected "**Master Boot Record**" in the partitioning/formatting options of "**Disk Utility**".

- I haven't assessed the Windows 10 formatter yet.
- I'd like to assess the 3rd party formatting tools various people have suggested, haven't gotten around to that yet either...