

# How to handle partitions on UniFlex HD/SSD

Date	Name	Action
19Nov2020	OBH	Created

## Table of Contents

Preward.....	4
UniFlex partitions.....	5
diskinfo.....	6
16 MB diskinfo .....	6
64 MB diskinfo .....	7
Adding a partition to SLAVE disk.....	8
/etc/disk - program .....	8
/etc/disk commands .....	9
Boot-sector modified .....	11
Updating boot-sector.....	12
Restart system to read changed partition-table.....	12
Creating a filesystem in new partition .....	13
Mounting new partition for access.....	14
Checking free space on device.....	14

## Preword

This document contains information about **setting up partitions on UniFlex HD/SSD**.

## UniFlex partitions

Each drive (HD or SSD) can have up to 4 partitions. Size of partitions has been tested up to 256 MB. The **partition-table** for each drive is located as the last 16 bytes in the boot-sector ( here highlighted in yellow):

```

                                bootsector
00000000  16 00 19 0D 69 6D 61 67  65 3A 20 75 6E 69 66 6C  *....image: unifl*
00000010  65 78 00 00 00 00 00 00  00 0D 0A 00 FF 07 EA FD  *ex.....*
00000020  07 EC 8E 08 03 AD 9F F8  0A 10 8E 4E 20 AD 9F F8  *......N ...*
00000030  04 25 09 BD 09 E8 31 3F  26 F3 20 0F AD 9F F8 06  *.%....1?&. ....*
00000040  81 30 25 07 81 39 22 03  B7 08 12 8E 08 19 AD 9F  *.0%.9".....*
00000050  F8 0A B6 01 0A 81 FD 26  0E 0A 13 9E 14 A6 82 9F  *......&.....*
00000060  14 B7 01 0A B7 F4 0A CC  00 01 BD 09 35 EC 27 BD  *......5.'.*
00000070  09 1A 44 56 FD 07 E6 BD  09 63 10 26 00 B6 30 42  *..DV.....c.&..0B*
00000080  EC C4 27 13 10 8E 08 0B  C6 0E A6 80 A1 A0 26 07  *..'.....&.*
00000090  5A 26 F7 EC C4 20 17 BE  07 E6 30 1F BF 07 E6 10  *Z&... ....0....*
000000A0  27 00 91 33 C8 10 11 83  0C 00 26 D2 20 C9 BD 09  *'.3.....&. ...*
000000B0  35 BD 09 63 26 7E EC 4A  FD 07 E8 33 C8 18 8D 61  *5..c&~.J...3...a*
000000C0  1F 98 8D 5D 10 83 00 00  27 14 1F 01 8D 53 1F 98  *...]. ....'....S.*
000000D0  8D 4F 1F 02 8D 4B E7 A0  30 1F 26 F8 20 E0 7F 07  *.O...K..0.&. ...*
000000E0  EC 10 8E 09 17 BD 09 8E  26 4A FE 50 02 8D 18 C6  *......&J.P....*
000000F0  10 F7 07 EC 34 40 10 8E  09 17 BD 09 8E 35 40 26  *.....4@.....5@&*
00000100  02 8D 04 6E 9F 07 E8 10  8E 0D F0 8E 00 08 EC A1  *....n.....*
00000110  ED C1 30 1F 26 F8 39 00  00 00 44 56 44 56 44 56  *..0.&.9...DVDVDV*
00000120  39 11 83 0C 00 26 08 34  32 8D 38 35 32 26 03 E6  *9....&.42.852&...*
00000130  C0 39 35 06 3F 34 06 C3  00 0F 8D DE FD 07 E4 7F  *.95.?4.....*
00000140  07 E3 10 8E 07 E3 8D 46  35 06 26 E6 5A C4 07 86  *......F5.&.Z...*
00000150  40 3D C3 0C 00 1F 02 C3  00 09 FD 07 E1 86 0A B7  *@=.....*
00000160  07 E0 39 7D 07 E0 27 14  7A 07 E0 10 BE 07 E1 8E  *.9}...'.z.....*
00000170  0A 00 8D 1D 34 01 10 BF  07 E1 35 81 10 BE 07 E1  *....4.....5....*
00000180  8D 0C 26 AE FF 07 E1 86  80 B7 07 E0 20 D5 8E 0C  *..&..... ..*
00000190  00 34 10 FE 07 EA 8D 32  4F C6 40 FA 07 EC ED 4C  *.4.....2O.@....L*
000001A0  EC 4E C1 50 26 22 E6 22  ED 46 E6 21 ED 48 E6 A4  *.N.P&".F.!..H.*
000001B0  ED 4A 31 23 CC 00 01 ED  44 C6 C8 ED 4E A6 C8 18  *.Jl#....D...N...*
000001C0  85 80 27 F9 EC 4E C4 01  35 C0 B6 01 00 1F 89 44  *..'...N..5.....D*
000001D0  44 44 44 8A C0 A7 C8 13  58 58 58 58 34 04 1F 10  *DDD....XXXX4...*
000001E0  84 0F AA E0 ED C8 10 39  86 40 4A 26 FD 39 00 00  *......9.@J&.9...*
000001F0  A5 00 00 00 A5 FF FF FF  A5 FF FF FF A5 FF FF FF  *......z.....*
```

Each entry is 4 bytes starting with 0xA5 and followed with a 3 hex byte value, that sets the offset of the partition in 512-byte blocks. The first 4 byte is partition 1, next 4 byte is partition 2 and so on. If a partition-offset is set to 0xFF 0xFF 0xFF, this indicates, that the partition is absent.

The default partition size supplied in the current disk-image is 16 MB.

## diskinfo

The program **diskinfo** in `/etc` can give you information about a disk.

Current devices/disks ( read partitions ) are as follow:

w0 = MASTER part 1  
 w1 = MASTER part 2  
 w2 = MASTER part 3  
 w3 = MASTER part 4

w4 = SLAVE part 1  
 w5 = SLAVE part 2  
 w6 = SLAVE part 3  
 w7 = SLAVE part 4

These are accessed through the `/dev/w0`, `/dev/w1` ..... `/dev/w7`

Pay attention, that w0, w1, w2 and w3 are related to the MASTER-disk on CPU09IDE-controller, where as w4, w5, w6 and w7 are related to the SLAVE-disk on CPU09IDE-controller.

## 16 MB diskinfo

So `/etc/diskinfo /dev/w0` will show you information on partition 1 on MASTER-disk:

```
++ /etc/diskinfo /dev/w0
```

```
Disk Information for "/dev/w0":
```

```
-----
```

```
Creation date: 04:35:44 Tue Mar 07 1989
```

```
Last update: 01:27:52 Mon May 18 1220
```

```
Total disk size: 15.6 Mb (31212 blocks)
```

```
- Fdn space: 472.5 Kb (945 blocks, 7560 fdns)
```

```
- File space: 14.1 Mb (28225 blocks)
```

```
- Swap space: 1.0 Mb (2040 blocks)
```

```
Free space: 4.5 Mb (8982 blocks)
```

```
Free fdns: 5755
```

```
Disk type: Hard disk
```

```
++
```

## 64 MB diskinfo

The following shows diskinfo on a 64MB partition on SLAVE-disk:

```
++ /etc/diskinfo /dev/w4

Disk Information for "/dev/w4":
-----
Creation date: 23:10:04 Sun May 17 1220
Last update: 13:43:52 Mon May 18 1220

Total disk size: 64.5 Mb (129024 blocks)
- Fdn space: 806.5 Kb (1613 blocks, 12904 fdns)
- File space: 63.7 Mb (127409 blocks)
- Swap space: 0.0 Kb (0 blocks)

Free space: 53.1 Mb (106198 blocks)
Free fdns: 12871

Disk type:  Hard disk

++
```

## Adding a partition to SLAVE disk

In order to add a new partition ( for now ) you have to 'patch' in the offset of the new partition to the bootsector on the drive, where you want to add it. For now, you can use **/etc/disk** program, as it allows you to read a block from disk, modify it and write it back, exactly what we need now. For playing around it has a 'readonly' mode by using option **'+r'**:

### **/etc/disk - program**

```
++ help disk
```

```
Syntax:  /etc/disk [<device>] [+hr]
```

```
Description:  Examine and change physical sectors on a disk.  The
                program prompts for single letter commands.  Default
                device is the root device.
```

```
Options:      +h          Start in hexadecimal mode instead of decimal.
                +r          Examine disk in read-only mode.
```

```
++
```

So to add a partition to a SLAVE-drive, you would use following command:

**/etc/disk /dev/w4 +h**

Remember that partition-table is located in boot-sector in **/dev/w0** for MASTER and **/dev/w4** for SLAVE

As my SLAVEDISK is a copy of MASTER-disk, I already have a partition 1, and want to add the offset for a partition 2. For this I need the size of partition 1, which was displayed from **'/etc/diskinfo /dev/w4'** and the disksize would be the numbers reported by:

```
Total disk size: 15.6 Mb (31212 blocks)
```

31212 decimal = 79EC hexadecimal.

To be on 'the safe side', we will use 0x007A00 as offset.

After issuing command:

**/etc/disk /dev/w4 +h**

You now see the command-prompt from the 'disk'-utility:

```
[blk 1 hex]>
```



Entering a '?' prints info about commands available ( quite powerful ):

[/etc/disk commands](#)

```
[blk 1 hex]> ?
```

```
a <no_1> [+] <no_2>      : add two numbers
b [<block>]              : [read] and display block
c                        : change contents of fdn
d                        : use decimal mode
e <start>[-]<end>         : examine partial block (also t,m,b)
f [<fdn>]                : [read] and display fdn
h                        : use hexadecimal mode
i                        : display info about disk
m <offset>              : modify current block or fdn
n                        : read and display next block or fdn
q                        : quit
r <filename>            : read file into current block or fdn
s                        : switch to block or fdn without dump
w [<file or number>]     : write to file or specified location
x <number>              : translate number
z [<fill_value>]         : zero [or fill] current block or fdn
+ <UniFLEX_command>     : execute a UniFLEX command
?                        : obtain help
$<hex_number>           : read and display specified block
#<decimal_number>       : read and display specified block
<number>                : read and display specified block
<CTRL-R>                : repeat last command
```

```
[blk 1 hex]>
```

We need to read 'block zero' from the device, in order to have the boot-sector:

```
[blk 1 hex]> b 0
```

Block 0 hex:

```

      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
      -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0000  16 00 19 0D 69 6D 61 67 65 3A 20 75 6E 69 66 6C  ....image: unifl
0010  65 78 00 00 00 00 00 00 00 0D 0A 00 FF 07 EA FD  ex.....
0020  07 EC 8E 08 03 AD 9F F8 0A 10 8E 4E 20 AD 9F F8  ....N ...
0030  04 25 09 BD 09 E8 31 3F 26 F3 20 0F AD 9F F8 06  .%.1?&. ....
0040  81 30 25 07 81 39 22 03 B7 08 12 8E 08 19 AD 9F  .0%..9".....
0050  F8 0A B6 01 0A 81 FD 26 0E 0A 13 9E 14 A6 82 9F  ....&.....
0060  14 B7 01 0A B7 F4 0A CC 00 01 BD 09 35 EC 27 BD  ....5.'.
0070  09 1A 44 56 FD 07 E6 BD 09 63 10 26 00 B6 30 42  ..DV.....c.&..0B
0080  EC C4 27 13 10 8E 08 0B C6 0E A6 80 A1 A0 26 07  ..'.....&.
0090  5A 26 F7 EC C4 20 17 BE 07 E6 30 1F BF 07 E6 10  Z&... ..0....
00A0  27 00 91 33 C8 10 11 83 0C 00 26 D2 20 C9 BD 09  '..3.....&. ...
00B0  35 BD 09 63 26 7E EC 4A FD 07 E8 33 C8 18 8D 61  5..c&~.J...3...a
00C0  1F 98 8D 5D 10 83 00 00 27 14 1F 01 8D 53 1F 98  ...]....'....S..
00D0  8D 4F 1F 02 8D 4B E7 A0 30 1F 26 F8 20 E0 7F 07  .O...K..0.&. ...
00E0  EC 10 8E 09 17 BD 09 8E 26 4A FE 50 02 8D 18 C6  ....&J.P....
00F0  10 F7 07 EC 34 40 10 8E 09 17 BD 09 8E 35 40 26  ....4@.....5@&
0100  02 8D 04 6E 9F 07 E8 10 8E 0D F0 8E 00 08 EC A1  ...n.....
0110  ED C1 30 1F 26 F8 39 00 00 00 44 56 44 56 44 56  ..0.&.9...DVDVDV
0120  39 11 83 0C 00 26 08 34 32 8D 38 35 32 26 03 E6  9....&.42.852&..
0130  C0 39 35 06 3F 34 06 C3 00 0F 8D DE FD 07 E4 7F  .95.?4.....
0140  07 E3 10 8E 07 E3 8D 46 35 06 26 E6 5A C4 07 86  ....F5.&.Z...
0150  40 3D C3 0C 00 1F 02 C3 00 09 FD 07 E1 86 0A B7  @=.....
0160  07 E0 39 7D 07 E0 27 14 7A 07 E0 10 BE 07 E1 8E  ..9}...'.z.....
0170  0A 00 8D 1D 34 01 10 BF 07 E1 35 81 10 BE 07 E1  ....4.....5.....
0180  8D 0C 26 AE FF 07 E1 86 80 B7 07 E0 20 D5 8E 0C  ..&.....
0190  00 34 10 FE 07 EA 8D 32 4F C6 40 FA 07 EC ED 4C  .4.....2O.@....L
01A0  EC 4E C1 50 26 22 E6 22 ED 46 E6 21 ED 48 E6 A4  .N.P&".".F.!..H..
01B0  ED 4A 31 23 CC 00 01 ED 44 C6 C8 ED 4E A6 C8 18  .J1#....D...N...
01C0  85 80 27 F9 EC 4E C4 01 35 C0 B6 01 00 1F 89 44  ..'..N..5.....D
01D0  44 44 44 8A C0 A7 C8 13 58 58 58 58 34 04 1F 10  DDD.....XXXX4...
01E0  84 0F AA E0 ED C8 10 39 86 40 4A 26 FD 39 00 00  ....9.@J&.9...
01F0  A5 00 00 00 A5 FF FF FF A5 FF FF FF A5 FF FF FF  ....z.....
[blk 0 hex]>

```

The 3 bytes, that we want to change from 0xFF 0xFF 0xFF to 0x00 0x7A 0x00 are here highlighted in **yellow** starting at offset 0x01F5. To change them issue command:

```
m 1f5
```

Then type 00 followed by carriage return, then 7A followed by carriage return, then 00 followed by carriage return and the a carriage return to exit 'modify-mode'.

```
[blk 0 hex]> m 1f5
01F5 FF 00
01F6 FF 7A
01F7 FF 00
01F8 A5
[blk 0 hex]>
```

Now display boot-sector in the edit-buffer, to check changes:

### Boot-sector modified

```
[blk 0 hex]> e 0-1ff
      0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F  0123456789ABCDEF
      -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
0000  16 00 19 0D 69 6D 61 67 65 3A 20 75 6E 69 66 6C ....image: unifl
0010  65 78 00 00 00 00 00 00 00 0D 0A 00 FF 07 EA FD ex.....
0020  07 EC 8E 08 03 AD 9F F8 0A 10 8E 4E 20 AD 9F F8 .....N ...
0030  04 25 09 BD 09 E8 31 3F 26 F3 20 0F AD 9F F8 06 .%.1?&. ....
0040  81 30 25 07 81 39 22 03 B7 08 12 8E 08 19 AD 9F .0%..9".....
0050  F8 0A B6 01 0A 81 FD 26 0E 0A 13 9E 14 A6 82 9F .....&.....
0060  14 B7 01 0A B7 F4 0A CC 00 01 BD 09 35 EC 27 BD .....5.'..
0070  09 1A 44 56 FD 07 E6 BD 09 63 10 26 00 B6 30 42 ..DV.....c.&..0B
0080  EC C4 27 13 10 8E 08 0B C6 0E A6 80 A1 A0 26 07 ..'.....&.
0090  5A 26 F7 EC C4 20 17 BE 07 E6 30 1F BF 07 E6 10 Z&... ..0.....
00A0  27 00 91 33 C8 10 11 83 0C 00 26 D2 20 C9 BD 09 '..3.....&. ...
00B0  35 BD 09 63 26 7E EC 4A FD 07 E8 33 C8 18 8D 61 5..c&~.J...3...a
00C0  1F 98 8D 5D 10 83 00 00 27 14 1F 01 8D 53 1F 98 ...]....'....S...
00D0  8D 4F 1F 02 8D 4B E7 A0 30 1F 26 F8 20 E0 7F 07 .O...K..0.&. ...
00E0  EC 10 8E 09 17 BD 09 8E 26 4A FE 50 02 8D 18 C6 .....&J.P....
00F0  10 F7 07 EC 34 40 10 8E 09 17 BD 09 8E 35 40 26 ....4@.....5@&
0100  02 8D 04 6E 9F 07 E8 10 8E 0D F0 8E 00 08 EC A1 ...n.....
0110  ED C1 30 1F 26 F8 39 00 00 00 44 56 44 56 44 56 ..0.&.9...DVDVDV
0120  39 11 83 0C 00 26 08 34 32 8D 38 35 32 26 03 E6 9....&.42.852&..
0130  C0 39 35 06 3F 34 06 C3 00 0F 8D DE FD 07 E4 7F .95.?4.....
0140  07 E3 10 8E 07 E3 8D 46 35 06 26 E6 5A C4 07 86 .....F5.&.Z...
0150  40 3D C3 0C 00 1F 02 C3 00 09 FD 07 E1 86 0A B7 @=.....
0160  07 E0 39 7D 07 E0 27 14 7A 07 E0 10 BE 07 E1 8E ..9}...'.z.....
0170  0A 00 8D 1D 34 01 10 BF 07 E1 35 81 10 BE 07 E1 ....4.....5.....
0180  8D 0C 26 AE FF 07 E1 86 80 B7 07 E0 20 D5 8E 0C ..&.....
0190  00 34 10 FE 07 EA 8D 32 4F C6 40 FA 07 EC ED 4C .4.....2O.@....L
01A0  EC 4E C1 50 26 22 E6 22 ED 46 E6 21 ED 48 E6 A4 .N.P&".".F.!..H..
01B0  ED 4A 31 23 CC 00 01 ED 44 C6 C8 ED 4E A6 C8 18 .J1#....D...N...
01C0  85 80 27 F9 EC 4E C4 01 35 C0 B6 01 00 1F 89 44 ..'.N..5.....D
01D0  44 44 44 8A C0 A7 C8 13 58 58 58 58 34 04 1F 10 DDD....XXXX4...
01E0  84 0F AA E0 ED C8 10 39 86 40 4A 26 FD 39 00 00 .....9.@J&.9..
01F0  A5 00 00 00 A5 00 7A 00 A5 FF FF FF A5 FF FF FF .....z.....
[blk 0 hex]>
```

Highlighted in **green** you can see the offset to partition **2**, which can be accessed through **/dev/w5** -device.

### Updating boot-sector

We now need to write back the edit-buffer to disk:

```
[blk 0 hex]> w  
Write buffer to block 0 hex? y  
[blk 0 hex]>
```

Now quit disk-utility:

```
[blk 0 hex]> q  
++
```

In order for UniFlex to recognize the new offset to partition **2** on **SLAVE**-device, you need to reboot the system, so on 'console':

### Restart system to read changed partition-table

**shutup –**

wait for prompt

**stop**

hit reset and boot again.

## Creating a filesystem in new partition

Before UniFlex can access the new partition 'enabled' in the boot-sector, we need to create a file-system on **/dev/w5**.

We do that, by using the **mkfs** command ( short for MaKe FileSystem ) and the command needs a few options beside the **/dev/w5** – devicename:

**mkfs /dev/w5 +c=2048 +s=63**

It will take a couple of minutes as it write the 'filesystem'-information to the device. By changing values for **+c** and **+s** you can set the size you want ( c times s gives total blocks ) , but be carefull. It is not adviced to create the partition bigger than 256 MB:

```
++ mkfs /dev/w5 +c=2048 +s=63
```

**/etc/diskinfo** can display info about new 'disk' in partition **2** on **SLAVE**-disk.

```
++ /etc/diskinfo /dev/w5
```

```
Disk Information for "/dev/w5":
```

```
-----
```

```
Creation date: 16:11:19 Mon May 18 1220
```

```
Last update: 16:11:19 Mon May 18 1220
```

```
Total disk size: 64.5 Mb (129024 blocks)
```

```
- Fdn space: 806.5 Kb (1613 blocks, 12904 fdns)
```

```
- File space: 63.7 Mb (127409 blocks)
```

```
- Swap space: 0.0 Kb (0 blocks)
```

```
Free space: 63.7 Mb (127408 blocks)
```

```
Free fdns: 12903
```

```
Disk type: Hard disk
```

```
++
```

### Mounting new partition for access

In order to access the new partition **2** in the **SLAVE**-drive, you have to create a new folder and 'mount' to that. After successful mounting, accessing that folder means accessing the new partition:

```
++ mkdir /mnt5
++ mount /dev/w5 /mnt6
++ cp /bin/asmb /mnt5
++ ll /mnt5
total 28
-rwx--x 1 root      14274 May 18 16:24 asmb
++ ll /bin/asmb
-rwx--x 1 bin       14274 Sep 15  1987 /bin/asmb
++
```

### Checking free space on device

```
++ df /dev/w5

120720 free blocks (61808640 bytes)
12887 free fdns

++
```