

Convert RAW partition to VDI image with VirtualBox on OS X

Some time ago, I [set up a multi-boot environment on my Mac](#), where I could use my BootCamp-Partition for both booting Windows natively on the Mac as well as virtualized from within VirtualBox.

Now, I wanted to create a clone of the RAW partition to do some testing.

As usual, there's more than one way to accomplish this. I'm going with the CLI way, as it's feast and easier in my opinion.

Before you start with this anyway, I shall recomment you first to take a backup and second, make sure you don't have lot's of temp files WITHIN the BOOTCAMP partition. Also it's a good idea, to defragment it first. This will save you some additional space and the converted VDI disk will be slightly smaller at the end.

So, first we need to identify, which device number our BOOTCAMP partition resides in.

Do this simply from a Terminal by running the command 'diskutil info disk0' (If you have a standard setup, your setup should reside on 'disk0'. If you do some remote/iSCSI booting or use an USB drive to boot from, then your disk-id may be different. This is beyond this topic.)

```
new-host-2:Image Gianpaolo$ diskutil info disk0
/new/disk0
#          TYPE NAME              SIZE      IDENTIFIER
0:      GUID_partition_scheme      128.0 GB   disk0s0
1:      EFI                        200.0 MB   disk0s1
2:      Apple_APFS_Macintosh_HD     33.5 GB   disk0s2
3:      Microsoft_Basic_Data_BOOTCAMP 33.7 GB   disk0s3
```

You can easily identify your BOOTCAMP partition, as it reads "Microsoft Basic Data" in it's TYPE description. Now look our for the last row which says "IDENTIFIER".

Note the name it says there ('disk0s4' in my example).

Now you may take the slow and easy way or the fast and complicated way. The later being the more geek-like approach ;-)

The slow and easy way

Now change to a temporary directory, where you have enough room to store the disk image. Enough room in our case means, twice as much as your BOOTCAMP partition is in size. In my example, it's roughly around 35G, so you should have some 70G free at least. You can always check your remaining free disk space on the commad line with the 'df' command:

```
new-host-2:Image Gianpaolo$ df -h
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/disk0s2    179Gi  62Gi  111Gi   79%    /
devfs           128Ki  128Ki  0Bi   100%   /dev
map -hosts      0Bi   0Bi   0Bi   100%   /net
map auto_home   0Bi   0Bi   0Bi   100%   /home
```

Then I created a new directly called 'Image' below my home directory to hold the image.

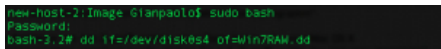
```
new-host-2:~ Gianpaolo$ mkdir Image
new-host-2:~ Gianpaolo$ cd Image
```

To actually dump the RAW partition, you need to have super-user privileges. So it's easiest to run 'sudo bash' to get a super-user shell like this:

```
sudo bash
```

Then run this command to dump the RAW partition into a file. Remember to use the correct IDENTIFIER as noted above ('disk0s4' in my case).

```
dd if=/dev/disk0s4 of=Win7RAW.dd
```



```
new-host-2:Image Gianpaolo@ sudo bash
Password:
bash-3.2# dd if=/dev/disk0s4 of=Win7RAW.dd
```

This will take ample of time. Unfortunately, 'dd' does not print it's progress to the console. So if you want to know the progress, you can always open a second Terminal and run these command from there:

```
sudo bash
while [ : ] ; do killall -SIGINFO dd && sleep 5 || exit ; done
```

This will send a 'SIGINFO' signal to 'dd' every five seconds and cause it to print it's progress to the Terminal as shown below. The loop will terminate on it's own once 'dd' is finished.

```
8925687808 bytes transferred in 373.295433 secs (23910520 bytes/sec)
```

```
17634581+0 records in
```

```
17634580+0 records out
```

When you're done with capturing the image, it's time to convert it to a virtual disk usage for VirtualBox.

Do so by the 'VBoxManage' command like this:

```
VBoxManage convertfromraw Win7RAW.dd Win7.vdi --format VDI
```

This will convert the RAW image into a 'VDI' disk. Additionally, if the RAW partition space was not completely filled up, your VDI disk should end up considerably smaller than the original RAW image.

As soon as this conversion is finished, you can delete the RAW image.

```
rm Win7RAW.dd
```

As you see, this method is straight forward, but takes double the space and almost double the time to create the VDI image.

Look into the next method to gain some speed, but also a bit more complexity ;-)

The fast and complicated way

So you end up here. Maybe because you want to spare some time, or simply don't have enough free disk space to hold two images.

So, let's start again on the Terminal by gathering information. Remember the IDENTIFIER of the BOOTCAMP partition? It was 'disk0s4' for me ...

Now let's get some detail about this partition, as we need to know it's exact size in bytes.

The 'diskutil' command can provide us with this information, but only if it's run in so called 'plist-mode'. By invoking 'diskutil' like this, you'll end up with an XML-formated output with lots of detail information.

```
bash-3.2# diskutil info -plist /dev/disk0s4
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Bootable</key>
<true/>
<key>BusProtocol</key>
<string>SATA</string>
<key>CanBeMadeBootable</key>
<false/>
<key>CanBeMadeBootableRequiresDestroy</key>
<false/>
<key>Content</key>
<string>Microsoft Basic Data</string>
<key>DeviceBlockSize</key>
<integer>512</integer>
<key>DeviceIdentifier</key>
<string>disk0s4</string>
<key>DeviceNode</key>
<string>/dev/disk0s4</string>
<key>DeviceTreePath</key>
<string>IODeviceTree:/PCI0@0/SATA@1F,2/PRT0@0/PMP@0</string>
<key>Ejectable</key>
<false/>
<key>FileSystemName</key>
<string>NTFS</string>
<key>FileSystemType</key>
<string>ntfs</string>
<key>FileSystemUserVisibleName</key>
<string>Windows NT File System (NTFS)</string>
<key>FreeSpace</key>
<integer>0</integer>
<key>GlobalPermissionsEnabled</key>
<false/>
<key>IOKitSize</key>
<integer>34688991232</integer>
<key>Internal</key>
<true/>
<key>MediaName</key>
<string>BOOTCAMP</string>
<key>MediaType</key>
<string>Generic</string>
<key>MountPoint</key>
<string></string>
<key>ParentWholeDisk</key>
```

```
<string>disk0</string>
<key>RAIDMaster</key>
<false/>
<key>RAIDSlice</key>
<false/>
<key>SMARTStatus</key>
<string>Verified</string>
<key>SolidState</key>
<true/>
<key>SupportsGlobalPermissionsDisable</key>
<false/>
<key>SystemImage</key>
<false/>
<key>TotalSize</key>
<integer>34688991232</integer>
<key>VolumeName</key>
<string>BOOTCAMP</string>
<key>VolumeUUID</key>
<string>7028E14A-6363-480D-8AAD-57609FBA87CB</string>
<key>WholeDisk</key>
<false/>
<key>Writable</key>
<true/>
<key>WritableMedia</key>
<true/>
<key>WritableVolume</key>
<false/>
</dict>
</plist>
```

Look out for the keyword 'TotalSize' and the take the integer value it provides ('34688991232' in this example).

Now dump your RAW partition directly into a VDI image like this:

```
dd if=/dev/disk0s4 | VBoxManage convertfromraw stdin Win7.vdi 34688991232 --format VDI
```



Further readings on this topic are located at [VirtualBox Forums](#).