


```

/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16438/Xp3 # ls -la
total 6170640
drwxr-xr-x 1 root root      2100 Feb 25 12:12
drwxr-xr-x 1 root root      1680 Feb 25 12:14
-rw-r----- 1 root root 16863232 Feb 25 12:12 Xp3-000001-delta.vmdk
-rw-r----- 1 root root      809 Feb 25 12:12 Xp3-000001.vmdk
-rw-r----- 1 root root 4300516505 Feb 25 12:05 Xp3-Snapshot1.vmsn
-rw-r----- 1 root root 42949672960 Feb 25 11:59 Xp3-flat.vmdk
-rw-r----- 1 root root      8684 Feb 25 12:12 Xp3.nvram
-rw-r----- 1 root root      534 Feb 18 13:52 Xp3.vmdk
-rw-r--r-- 1 root root      384 Feb 25 12:00 Xp3.vmsd
-rwxr-xr-x 1 root root      2607 Feb 25 12:12 Xp3.vmsx
-rw-r--r-- 1 root root      258 Feb 17 15:20 Xp3.vmsxf
-rw-r--r-- 1 root root 145555 Feb 17 15:20 vmware-1.log
-rw-r--r-- 1 root root 147968 Feb 17 15:26 vmware-2.log
-rw-r--r-- 1 root root 148266 Feb 18 10:05 vmware-3.log
-rw-r--r-- 1 root root 156057 Feb 25 12:12 vmware.log
/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16438/Xp3 #

```

Xp3 vm and snapshot

```

cd datastore2
ls -la

```

As you can see now, there are more than one vmdk disk files, the flat vmdk, the delta vmdk and the vmdk descriptor file (the Xp3-000001.vmdk file in our example), along with the memory snapshot and various control and log files. What we should use here as a source file is the descriptor file that points to our snapshot. If you didn't have a snapshot, the source file would be the vmdk descriptor file of your vm, Xp3.vmdk.

Now, what I need to do is clone my Xp3 snapshot to a new Xp4 vm. First create the destination folder in datastore2 folder and type in the following commands

```

mkdir Xp4

```

and clone the disk file

```

vmkfstools -i /vmfs/volumes/datastore2/Xp3/Xp3-000001.vmdk /vmfs/volumes/datastore2/Xp4/Xp4.vmdk -d thin

```

if it was a single vm without a snapshot, you should run

```

vmkfstools -i /vmfs/volumes/datastore2/Xp3/Xp3.vmdk /vmfs/volumes/datastore2/Xp4/Xp4.vmdk -d thin

```

if you list the files in the destination folder you will see the new flat file and the descriptor file.

```

/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16438/Xp4 # vmkfstools -i /vmfs/volumes/datastore2/Xp3/Xp3-000001.vmdk /vmfs/volumes/datastore2/Xp4/Xp4.vmdk -d thin
Destination disk format: VMFS thin-provisioned
Cloning disk '/vmfs/volumes/datastore2/Xp3/Xp3-000001.vmdk'...
Clone: 100% done.
/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16438/Xp4 # ls -la
total 1948680
drwxr-xr-x 1 root root      560 Feb 25 12:20
drwxr-xr-x 1 root root      1680 Feb 25 12:14
-rw-r----- 1 root root 42949672960 Feb 25 12:20 Xp4-flat.vmdk
-rw-r----- 1 root root      534 Feb 25 12:20 Xp4.vmdk
/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16438/Xp4 #

```

cloned disk files

The actual syntax of vmkfstools command is

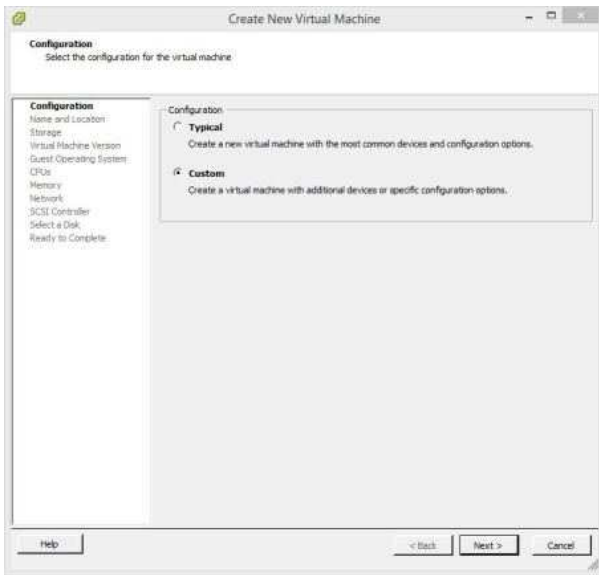
```

vmkfstools -i source_path destination_path -d disk_format -a adapter_type

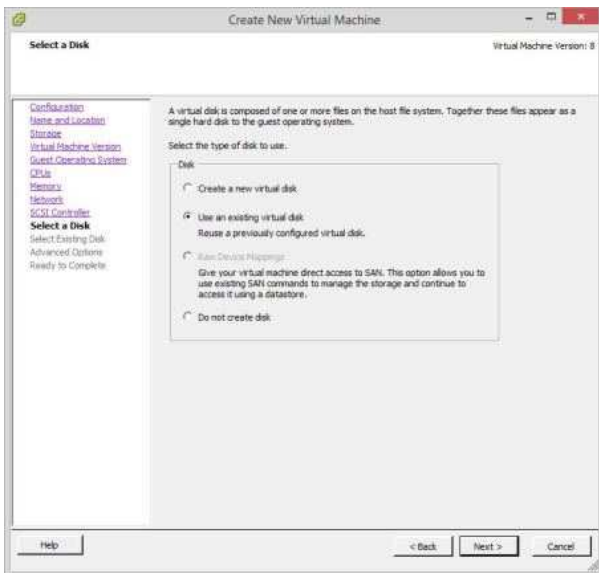
```

type man vmkfstools for more on this command

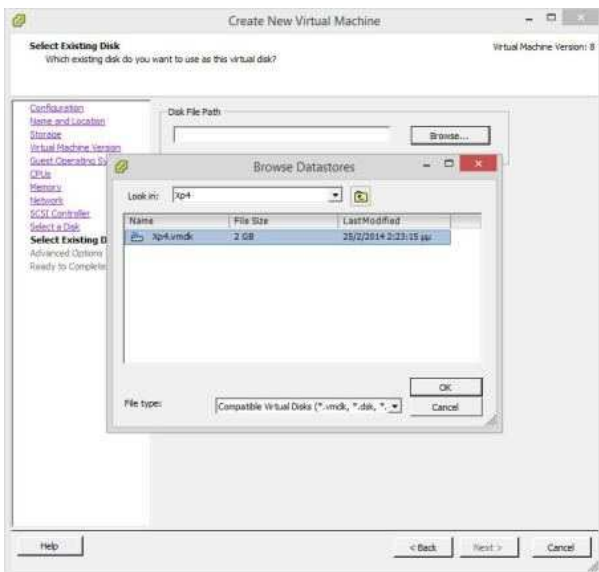
Finally create a new vm and force the use of an existing disk file.



create new vm



use existing virtual disk



select the cloned vmdk

You are ready to power it on.



power on

if you list the folder of the cloned vm you will see something like that

```
/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16439/Xp4 # ls -la
total 2325928
drwxr-xr-x 1 root root      1820 Feb 25 12:23
drwxr-xr-x 1 root root      1680 Feb 25 12:14
-rw-r----- 1 root root 268435456 Feb 25 12:23 Xp4-337f9563.vswp
-rw-r----- 1 root root 42949672960 Feb 25 12:39 Xp4-flat.vmdk
-rw-r----- 1 root root      8684 Feb 25 12:23 Xp4.nvram
-rw-r----- 1 root root      534 Feb 25 12:23 Xp4.vmdk
-rw-r----- 1 root root      0 Feb 25 12:28 Xp4.vmsd
-rwxr-xr-x 1 root root    2592 Feb 25 12:28 Xp4.vmsx
-rw-r----- 1 root root      0 Feb 25 12:23 Xp4.vmx.lock
-rw-r----- 1 root root     258 Feb 25 12:23 Xp4.vmxif
-rwxr-xr-x 1 root root    2566 Feb 25 12:23 Xp4.vmsx
-rw-r----- 1 root root   121310 Feb 25 12:27 vmware.log
-rw-r----- 1 root root    114294784 Feb 25 12:23 vmx-Xp4-863995235-1.vswp
/vmfs/volumes/530209bb-0af2bf8e-69d7-c81f66e16439/Xp4 #
```

cloned vm files

Don't forget to use sysprep command in windows to generalize your new virtual machine. More on this at the end of my previous [article](#).

Reference: [VMWare's website article](#)

Cloning and converting virtual machine disks with vmkfstools (1028042)

Purpose

This article provides information and instructions on the use of the `vmkfstools` command to convert virtual machine disks from one type to another.

Resolution

The `vmkfstools` command offers the ability to clone virtual machine content and also convert from one virtual machine disk (`.vmdk`) format into another.

Note: The host operating system chosen to perform the conversion may not necessarily support running of virtual machines via the output format defined. `vmkfstools` maintains the possibility of exporting virtual disks for use in other VMware products which support alternative disk formats.

To convert a virtual machine disk from one type to another:

1. Shut down the virtual machine. Virtual machine disk files are locked while in-use by a running virtual machine.
2. Log into the VMware vSphere Management Assistant (vMA). Alternatively, open a command-line interface after deploying the VMware vSphere CLI (vCLI). If terminal access is required:
 - o For VMware ESX 4.x and 3.x, log in via the terminal or SSH. For additional information, see [Connecting to an ESX host using a SSH client \(1019852\)](#).

- For VMware ESXi 4.1 and ESXi 5.x, see [Using Tech Support Mode in ESXi 4.1 and ESXi 5.0 \(1017910\)](#).
- For VMware ESXi 4.0 and 3.x, see [Tech Support Mode for Emergency Support \(1003677\)](#).

3. Run the following `vmkfstools` command to clone the disk from one format to another:

From the ESX/ESXi terminal:

```
vmkfstools -i input output -d format -a adaptertype
```

From vMA/vCLI:

```
vmkfstools --server <vCenter or ESX IP/Hostname> -i input output -d format -a adaptertype
```

For example, to clone `examplevm.vmdk` from `Datastore` to `Datastore 2` and thin-provision the destination copy:

From the ESX/ESXi terminal:

```
vmkfstools -i "/vmfs/volumes/Datastore/examplevm/examplevm.vmdk"
"/vmfs/volumes/Datastore 2/newexamplevm/newexamplevm.vmdk" -d thin -a
buslogic
```

From vMA/vCLI:

```
vmkfstools --server 10.21.49.14 -i '[Datastore] examplevm/examplevm.vmdk'
'[Datastore 2] newexamplevm/newexamplevm.vmdk' -d thin -a buslogic
```

Warning: If virtual machine snapshots or delta disks are present, ensure that the source chosen is the current snapshot delta disk. Failing to do so results in an outdated destination copy. However, when using the command, the corresponding descriptor file for the latest delta disk needs to be used when cloning the disk.

For example to clone and/or convert virtual machine disk with snapshot (otherwise known as a delta disk), `examplevm-000001.vmdk` and like above, from `Datastore` to `Datastore 2`, with a thin-provisioned destination copy, run this command:

From the ESX/ESXi terminal:

```
vmkfstools -i "/vmfs/volumes/Datastore/examplevm/examplevm-000001.vmdk"
"/vmfs/volumes/Datastore 2/newexamplevm/newexamplevm.vmdk" -d thin -a
buslogic
```

From vMA/vCLI:

```
vmkfstools --server 10.21.49.14 -i '[Datastore] examplevm/examplevm-
000001.vmdk' '[Datastore 2] newexamplevm/newexamplevm.vmdk' -d thin -a
buslogic
```

Note the selection of `examplevm-000001.vmdk`, as opposed to `examplevm.vmdk` for the input file. Snapshot delta disk files are appended a sequencing number, differentiating one snapshot delta disk iteration or level from another. Selecting the most recent delta disk ensures that the most recently stored data is cloned to the destination.

Note: Running the `vmkfstools -i` command creates a cloned drive with an LSI controller, even if the source disk is using VMware Paravirtual. This results in the virtual machine failing to boot. To resolve this issue, change the controller type to the same as the source.

For steps to identify a virtual machine's current snapshot, see [Confirming a virtual machine's snapshot delta disk layout \(1027887\)](#).

Additional Information

Commonly-used disk format options for `vmkfstools`:

- `zeroedthick`

This is the default option for virtual machine disks created and stored on a VMFS datastore.

- `thin`

This is the default option for virtual machine disks stored on an NFS datastore. It allocates and commits space on demand, growing as more virtual disk space is used.

Note: The configured size of thin-provisioned disks is correctly displayed using the `ls` command. The disk space consumed by disk files on a datastore is determined using the `du` command.

- `eagerzeroedthick`

This is a specified format required for cross-host virtual machine clustering and VMware Fault Tolerance (FT) virtual machines. All blocks for the virtual machine disk are allocated and committed with written zeroes at the time of creation. Existing disks can be inflated to this format without destroying existing data. For more information, see [Enabling clustering features for an existing virtual disk by converting in place \(1035823\)](#) and [Enabling clustering features for an existing virtual disk by converting while copying \(1003491\)](#).

- `rdmp`

This is the default Raw Device Mapping format with Physical compatibility mode. Most SCSI commands are passed-through to the guest operating system to/from a mapped physical raw LUN. This is required for cross-host virtual machine clustering; both virtual machines share the same mapping file. This format does not support virtual machine snapshots. For more information, see [Converting a virtual disk into a Raw Device Mapping \(3443266\)](#).

- `rdm`

This is an additional available Raw Device Mapping format with virtual compatibility mode set. A subset of SCSI commands are passed-through to the guest operating system to/from a mapped physical raw LUN. An added benefit of this format is the support of virtual machine snapshots. For more information, see [Converting a virtual disk into a Raw Device Mapping \(3443266\)](#).

- `2gbsparse`

This format is compatible with Hosted (desktop) products such as VMware Fusion, Player, Server, Workstation, etc. Virtual machine disks are spanned across several 2GB extents to eliminate potential cross-platform file system compatibility issues. For example, FAT32 supports a maximum file size of 4GB. This format is not supported for running virtual machines in VMware ESX/ESXi 4.x and ESXi 5.x.

Note: For SDK purposes, this property is known as `sparse2GB` in ESXi 5.0 and later. For more information, see [Scope of the sparse2GB property in the vSphere API \(2044714\)](#) and the *VirtualDiskType* section of the [vSphere Web Services SDK Documentation](#).

Additional information, types, and parameters are available in the `vmkfstools` manual pages and [product documentation](#).

- Review the manual pages on VMware ESX 3.x and 4.x by running: `man vmkfstools`
- Review command information on ESXi 3.x and 4.x hosts or vMA and vCLI by running: `vmkfstools --help`
- For much more information, review *Appendix C: Using vmkfstools* from the *ESX Configuration Guide* in the [product documentation](#) pages for your respective version of VMware ESX/ESXi.