

## Chapter 5. Virtual Storage

### Table of Contents

#### [5.1. Hard Disk Controllers](#)

#### [5.2. Disk Image Files \(VDI, VMDK, VHD, HDD\)](#)

#### [5.3. The Virtual Media Manager](#)

##### [5.3.1. Creating a Virtual Hard Disk Image](#)

##### [5.3.2. Creating a Virtual Optical Disk Image](#)

##### [5.3.3. Creating a Virtual Floppy Disk Image](#)

#### [5.4. Special Image Write Modes](#)

#### [5.5. Differencing Images](#)

#### [5.6. Cloning Disk Images](#)

#### [5.7. Host Input/Output Caching](#)

#### [5.8. Limiting Bandwidth for Disk Images](#)

#### [5.9. CD/DVD Support](#)

#### [5.10. iSCSI Servers](#)

#### [5.11. vboximg-mount: A Utility for FUSE Mounting a Virtual Disk Image](#)

##### [5.11.1. Viewing Detailed Information About a Virtual Disk Image](#)

##### [5.11.2. Mounting a Virtual Disk Image](#)

As the virtual machine will most probably expect to see a hard disk built into its virtual computer, Oracle VM VirtualBox must be able to present real storage to the guest as a virtual hard disk. There are presently three methods by which to achieve this:

- Oracle VM VirtualBox can use large image files on a real hard disk and present them to a guest as a virtual hard disk. This is the most common method, described in [Section 5.2, “Disk Image Files \(VDI, VMDK, VHD, HDD\)”](#).
- iSCSI storage servers can be attached to Oracle VM VirtualBox. This is described in [Section 5.10, “iSCSI Servers”](#).
- You can allow a virtual machine to access one of your host disks directly. This is an advanced feature, described in [Section 9.7.1, “Using a Raw Host Hard Disk From a Guest”](#).

Each such virtual storage device, such as an image file, iSCSI target, or physical hard disk, needs to be connected to the virtual hard disk controller that Oracle VM VirtualBox presents to a virtual machine. This is explained in the next section.

### 5.1. Hard Disk Controllers

In a computing device, hard disks and CD/DVD drives are connected to a device called a hard disk controller, which drives hard disk operation and data transfers. Oracle VM VirtualBox can emulate the most common types of hard disk controllers typically found in computing devices: IDE, SATA (AHCI), SCSI, SAS, USB-based, NVMe and virtio-scsi mass storage devices.

- **IDE (ATA)** controllers are a backwards-compatible yet very advanced extension of the disk controller in the IBM PC/AT (1984). Initially, this interface worked only with hard disks, but was later extended to also support CD-ROM drives and other types of removable media. In physical PCs, this standard uses flat ribbon parallel cables with 40 or 80 wires. Each such cable can connect two devices, called device 0 and device 1, to a controller. Typical PCs had two connectors for such cables. As a result, support for up to four IDE devices was most common: primary device 0, primary device 1, secondary device 0, and secondary device 1.

In Oracle VM VirtualBox, each virtual machine may have one IDE controller enabled, which gives you up to four virtual storage devices that you can attach to the machine. By default, one of these virtual storage devices, device 0 on the secondary channel, is preconfigured to be the virtual machine's virtual CD/DVD drive. However, you can change the default setting.

Even if your guest OS has no support for SCSI or SATA devices, it should always be able to see an IDE controller.

You can also select which exact type of IDE controller hardware Oracle VM VirtualBox should present to the virtual machine: PIIX3, PIIX4, or ICH6. This makes no difference in terms of performance, but if you import a virtual machine from another virtualization product, the OS in that machine may expect a particular controller type and crash if it is not found.

After you have created a new virtual machine with the **New Virtual Machine** wizard in VirtualBox Manager, you will typically see one IDE controller in the machine's **Storage** settings. The virtual CD/DVD drive will be attached to one of the four ports of this controller.

- **Serial ATA (SATA)** is a more recent standard than IDE. Compared to IDE, it supports both much higher speeds and more devices per controller. Also, with physical hardware, devices can be added and removed while the system is running. The standard interface for SATA controllers is called Advanced Host Controller Interface (AHCI).

Like a real SATA controller, Oracle VM VirtualBox's virtual SATA controller operates faster and also consumes fewer CPU resources than the virtual IDE controller. Also, this enables you to connect up to 30 virtual hard disks to one machine instead of just three, when compared to the Oracle VM VirtualBox IDE controller with a DVD drive attached.

For this reason, depending on the selected guest OS, Oracle VM VirtualBox uses SATA as the default for newly created virtual machines. One virtual SATA controller is created by default, and the default disk that is created with a new VM is attached to this controller.

### Warning

The entire SATA controller and the virtual disks attached to it, including those in IDE compatibility mode, will not be seen by OSes that do not have device support for AHCI. In particular, *there is no support for AHCI in Windows versions before Windows Vista*. Legacy Windows versions such as Windows XP, even with SP3 installed, will not see such disks unless you install additional drivers. It is possible to switch from IDE to SATA after installation by installing the SATA drivers and changing the controller type in the VM **Settings** window.

Oracle VM VirtualBox recommends the Intel Matrix Storage drivers, which can be downloaded from [http://downloadcenter.intel.com/Product\\_Filter.aspx?ProductID=2101](http://downloadcenter.intel.com/Product_Filter.aspx?ProductID=2101).

To add a SATA controller to a machine for which it has not been enabled by default, either because it was created by an earlier version of Oracle VM VirtualBox, or because SATA is not supported by default by the selected guest OS, do the following. Go to the **Storage** page of the machine's **Settings** window, click **Add Controller** under the Storage Tree box and then select **Add SATA Controller**. The new controller appears as a separate PCI device in the virtual machine, and you can add virtual disks to it.

To change the IDE compatibility mode settings for the SATA controller, see [Section 8.27, "VBoxManage storagectl"](#).

- **SCSI** is another established industry standard, standing for Small Computer System Interface. SCSI is as a generic interface for data transfer between all kinds of devices, including storage devices. SCSI is still used for connecting some hard disks and tape devices, but it has mostly been displaced in commodity

hardware. It is still in common use in high-performance workstations and servers.

Primarily for compatibility with other virtualization software, Oracle VM VirtualBox optionally supports LSI Logic and BusLogic SCSI controllers, to each of which up to fifteen virtual hard disks can be attached.

To enable a SCSI controller, on the **Storage** page of a virtual machine's **Settings** window, click **Add Controller** under the Storage Tree box and then select **Add SCSI Controller**. The new controller appears as a separate PCI device in the virtual machine.

### Warning

As with the other controller types, a SCSI controller will only be seen by OSes with device support for it. Windows 2003 and later ships with drivers for the LSI Logic controller, while Windows NT 4.0 and Windows 2000 ships with drivers for the BusLogic controller. Windows XP ships with drivers for neither.

- **Serial Attached SCSI (SAS)** is another bus standard which uses the SCSI command set. As opposed to SCSI physical devices, serial cables are used instead of parallel cables. This simplifies physical device connections. In some ways, therefore, SAS is to SCSI what SATA is to IDE: it enables more reliable and faster connections.

To support high-end guests which require SAS controllers, Oracle VM VirtualBox emulates a LSI Logic SAS controller, which can be enabled much the same way as a SCSI controller. At this time, up to 255 devices can be connected to the SAS controller.

### Warning

As with SATA, the SAS controller will only be seen by OSes with device support for it. In particular, *there is no support for SAS in Windows before Windows Vista*. So Windows XP, even SP3, will not see such disks unless you install additional drivers.

- The **USB mass storage device class** is a standard to connect external storage devices like hard disks or flash drives to a host through USB. All major OSes support these devices and ship generic drivers making third-party drivers superfluous. In particular, legacy OSes without support for SATA controllers may benefit from USB mass storage devices.

The virtual USB storage controller offered by Oracle VM VirtualBox works differently to the other storage controller types. While most storage controllers appear as a single PCI device to the guest with multiple disks attached to it, the USB-based storage controller does not appear as virtual storage controller. Each disk attached to the controller appears as a dedicated USB device to the guest.

### Warning

Bootting from drives attached using USB is only supported when EFI is used as the BIOS lacks USB support.

- **Non volatile memory express (NVMe)** is a standard for connecting non volatile memory (NVM) directly over PCI Express to lift the bandwidth limitation of the previously used SATA protocol for solid-state devices. Unlike other standards the command set is very simple in order to achieve maximum throughput and is not compatible with ATA or SCSI. OSes need to support NVMe devices to make use of them. For example, Windows 8.1 added native NVMe support. For Windows 7, native support was added with an update.

The NVMe controller is part of the extension pack.

**Warning**

Bootting from drives attached using NVMe is only supported when EFI is used as the BIOS lacks the appropriate driver.

- **Virtual I/O Device SCSI** is a standard to connect virtual storage devices like hard disks or optical drives to a VM. Recent Linux and Windows versions support these devices, but Windows needs additional drivers. Currently virtio-scsi controller support is experimental.

**Warning**

The virtio-scsi controller will only be seen by Oses with device support for it. In particular, *there is no built-in support in Windows*. So Windows will not see such disks unless you install additional drivers.

In summary, Oracle VM VirtualBox gives you the following categories of virtual storage slots:

- Four slots attached to the traditional IDE controller, which are always present. One of these is typically a virtual CD/DVD drive.
- 30 slots attached to the SATA controller, if enabled and supported by the guest OS.
- 15 slots attached to the SCSI controller, if enabled and supported by the guest OS.
- Up to 255 slots attached to the SAS controller, if enabled and supported by the guest OS.
- Eight slots attached to the virtual USB controller, if enabled and supported by the guest OS.
- Up to 255 slots attached to the NVMe controller, if enabled and supported by the guest OS.
- Up to 256 slots attached to the virtio-scsi controller, if enabled and supported by the guest OS.

Given this large choice of storage controllers, you may not know which one to choose. In general, you should avoid IDE unless it is the only controller supported by your guest. Whether you use SATA, SCSI, or SAS does not make any real difference. The variety of controllers is only supplied by Oracle VM VirtualBox for compatibility with existing hardware and other hypervisors.

## 5.2. Disk Image Files (VDI, VMDK, VHD, HDD)

Disk image files reside on the host system and are seen by the guest systems as hard disks of a certain geometry. When a guest OS reads from or writes to a hard disk, Oracle VM VirtualBox redirects the request to the image file.

Like a physical disk, a virtual disk has a size, or capacity, which must be specified when the image file is created. As opposed to a physical disk however, Oracle VM VirtualBox enables you to expand an image file after creation, even if it has data already. See [Section 8.31, “VBoxManage modifymedium”](#).

Oracle VM VirtualBox supports the following types of disk image files:

- **VDI.** Normally, Oracle VM VirtualBox uses its own container format for guest hard disks. This is called a Virtual Disk Image (VDI) file. This format is used when you create a new virtual machine with a new disk.
- **VMDK.** Oracle VM VirtualBox also fully supports the popular and open VMDK container format that is used by many other virtualization products, such as VMware.
- **VHD.** Oracle VM VirtualBox also fully supports the VHD format used by Microsoft.
- **HDD.** Image files of Parallels version 2 (HDD format) are also supported.

Due to lack of documentation of the format, newer versions such as 3 and 4 are not supported. You can however convert such image files to version 2 format using tools provided by Parallels.

Irrespective of the disk capacity and format, as mentioned in [Section 1.8, “Creating Your First Virtual Machine”](#), there are two options for creating a disk image: fixed-size or dynamically allocated.

- **Fixed-size.** If you create a fixed-size image, an image file will be created on your host system which has roughly the same size as the virtual disk's capacity. So, for a 10 GB disk, you will have a 10 GB file. Note that the creation of a fixed-size image can take a long time depending on the size of the image and the write performance of your hard disk.
- **Dynamically allocated.** For more flexible storage management, use a dynamically allocated image. This will initially be very small and not occupy any space for unused virtual disk sectors, but will grow every time a disk sector is written to for the first time, until the drive reaches the maximum capacity chosen when the drive was created. While this format takes less space initially, the fact that Oracle VM VirtualBox needs to expand the image file consumes additional computing resources, so until the disk file size has stabilized, write operations may be slower than with fixed size disks. However, after a time the rate of growth will slow and the average penalty for write operations will be negligible.

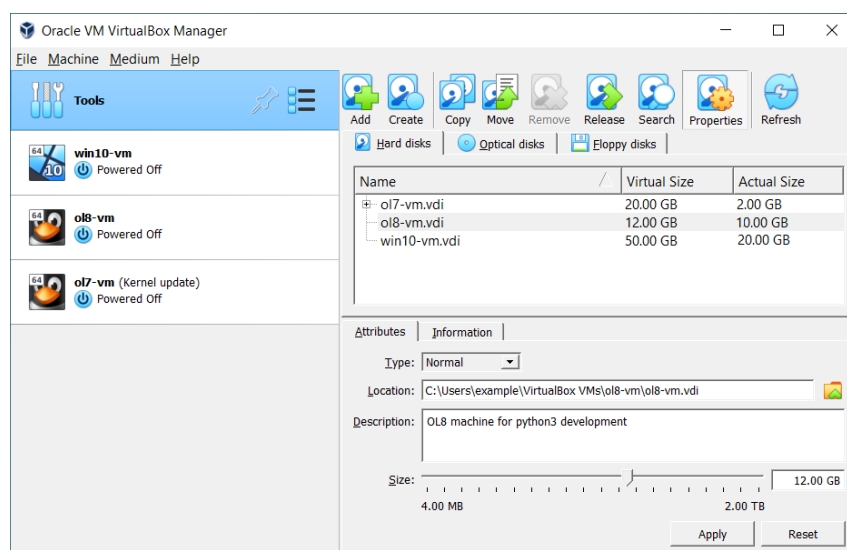
### 5.3. The Virtual Media Manager

Oracle VM VirtualBox keeps track of all the hard disk, CD/DVD-ROM, and floppy disk images which are in use by virtual machines. These are often referred to as *known media* and come from two sources:

- All media currently attached to virtual machines.
- Registered media, for compatibility with legacy Oracle VM VirtualBox versions.

The known media can be viewed and changed using the **Virtual Media Manager** tool, which you access by clicking **Media** on the global **Tools** menu in VirtualBox Manager.

**Figure 5.1. The Virtual Media Manager, Showing Hard Disk Images**



The known media are conveniently grouped in separate tabs for the supported formats. These formats are:

- Hard disk images, either in Oracle VM VirtualBox's own Virtual Disk Image (VDI) format, or in the third-party formats listed in [Section 5.2, “Disk Image Files \(VDI, VMDK, VHD, HDD\)”](#).
- CD/DVD images in standard ISO format.

- Floppy images in standard RAW format.

For each image, the Virtual Media Manager shows you the full path of the image file and other information, such as the virtual machine the image is currently attached to.

The Virtual Media Manager enables you to do the following:

- **Add** an image to the known media.
- **Create** a new disk image.
  - For hard disks, the **Create Virtual Hard Disk** wizard is shown. See [Section 5.3.1, “Creating a Virtual Hard Disk Image”](#).
  - For optical disks, the **VISO Creator** tool is shown. See [Section 5.3.2, “Creating a Virtual Optical Disk Image”](#).
  - For floppy disks, the **Floppy Disk Creator** tool is shown. See [Section 5.3.3, “Creating a Virtual Floppy Disk Image”](#).
- **Copy** an image to create another one.

For virtual hard disks, you can specify one of the following target types: VDI, VHD, or VMDK.

- **Move** an image to another location.

A file dialog prompts you for the new image file location.

When you use the Virtual Media Manager to move a disk image, Oracle VM VirtualBox updates all related configuration files automatically.

#### Note

Always use the Virtual Media Manager or the **VBoxManage modifymedium** command to move a disk image.

If you use a file management feature of the host OS to move a disk image to a new location, run the **VBoxManage modifymedium --setlocation** command to configure the new path of the disk image on the host file system. This command updates the Oracle VM VirtualBox configuration automatically.

- **Remove** an image from the known media. You can optionally delete the image file when removing the image.
- **Release** an image to detach it from a VM. This action only applies if the image is currently attached to a VM as a virtual hard disk.
- **Clear** all inaccessible disk images from the list. The disk images are released from the VMs they are attached to and removed from the known media.

#### Note

This option is for optical disks and floppy disks only.

- **Search** for an image by name or UUID.
- View and edit the **Properties** of a disk image.

Available properties include the following:

- **Type:** Specifies the snapshot behavior of the disk. See [Section 5.4, “Special Image Write Modes”](#).
  - **Location:** Specifies the location of the disk image file on the host system. You can use a file dialog to browse for the disk image location.
  - **Description:** Specifies a short description of the disk image.
  - **Size:** Specifies the size of the disk image. You can use the slider to increase or decrease the disk image size.
  - **Information:** Specifies detailed information about the disk image.
- **Refresh** the property values of the selected disk image.

To perform these actions, highlight the medium in the Virtual Media Manager and then do one of the following:

- Click an icon in the Virtual Media Manager toolbar.
- Right-click the medium and select an option.

Use the **Storage** page in a VM's **Settings** window to create a new disk image. By default, disk images are stored in the VM's folder.

You can copy hard disk image files to other host systems and then import them in to VMs from the host system. However, some Windows guest OSes may require that you configure the new VM in a similar way to the old one.

#### Note

Do not simply make copies of virtual disk images. If you import such a second copy into a VM, Oracle VM VirtualBox issues an error because Oracle VM VirtualBox assigns a universally unique identifier (UUID) to each disk image to ensure that it is only used one time. See [Section 5.6, “Cloning Disk Images”](#). Also, if you want to copy a VM to another system, use the Oracle VM VirtualBox import and export features. See [Section 1.15, “Importing and Exporting Virtual Machines”](#).

### 5.3.1. Creating a Virtual Hard Disk Image

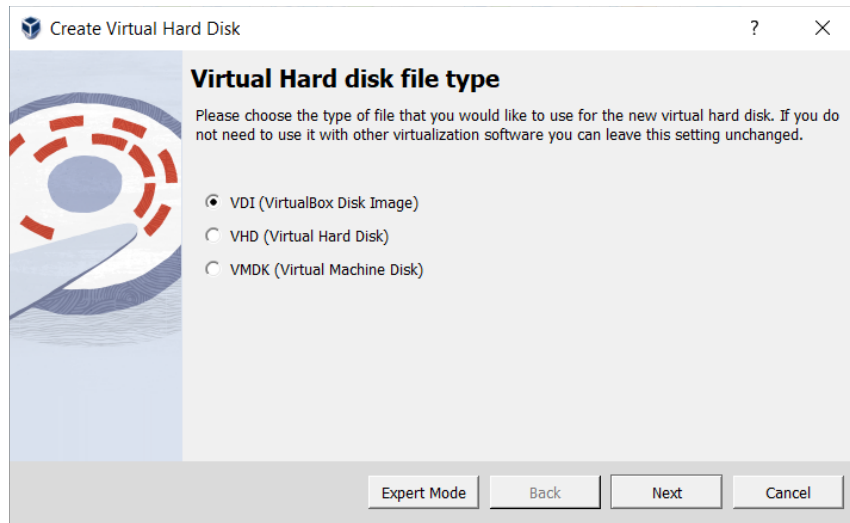
Use the **Create Virtual Hard Disk** wizard to create a hard disk image.

1. Display the **Hard Disks** tab in Virtual Media Manager and click **Create**.

The **Create Virtual Hard Disk** wizard is shown.

**Figure 5.2. Create Virtual Hard Disk Wizard**





2. On the **Virtual Hard Disk File Type** page, select a file type for the new virtual hard disk image.

Click **Next**.

3. On the **Storage on Physical Hard Disk** page, select whether the size of the virtual hard disk file is dynamically allocated or is of fixed size.

Click **Next**.

4. On the **File Location and Size** page, configure the location of the virtual hard disk file and use the slider to set the size limit for the virtual hard disk.

Click **Finish** to create the virtual hard disk file.

The virtual hard disk image is created in the specified location and added to the **Hard Disks** tab in Virtual Media Manager.

### 5.3.2. Creating a Virtual Optical Disk Image

Use the **VISO Creator** tool to create a virtual optical disk image. This enables you to create a virtual ISO from selected files on the host.

1. Display the **Optical Disks** tab in Virtual Media Manager and click **Create**.

The **VISO Creator** tool is shown.

2. Create the virtual ISO file.

- a. Configure the name of the ISO file.

Click **Configuration** and enter a name in the **Viso Name** field.

- b. Add files to your virtual ISO.

In the **Host File System** pane, select files to copy from the host system to the virtual ISO.

Click **Add Items To VISO**. The files are displayed in the **VISO Content** pane.

The following file operations are also available:

- To create folders on the virtual ISO, click **Create New Directory**.
- To remove files from the virtual ISO, select files in the **VISO Content** pane and click **Remove**



### Items From VISO.

- To remove *all* files from the virtual ISO, click **Reset the VISO Content**.

3. Create the virtual ISO image.

Click **Create**.

A virtual ISO file with the specified name and content is created.

### 5.3.3. Creating a Virtual Floppy Disk Image

Use the **Floppy Disk Creator** tool to create a floppy disk image.

1. Display the **Floppy Disks** tab in Virtual Media Manager and click **Create**.

The **Floppy Disk Creator** tool is shown.

2. Configure the following settings:

- **File Path:** The name and location of the floppy disk image.
- **Size:** Select from the list of supported floppy disk sizes.
- **Format Disk as FAT 12:** This is the default format used for most floppy disks. For an unformatted disk, do not select this option.

3. Create the floppy disk image file.

Click **Create**.

The floppy disk image is created in the specified location and added to the **Floppy Disks** tab in Virtual Media Manager.

## 5.4. Special Image Write Modes

For each virtual disk image supported by Oracle VM VirtualBox, you can determine separately how it should be affected by write operations from a virtual machine and snapshot operations. This applies to all of the aforementioned image formats (VDI, VMDK, VHD, or HDD) and irrespective of whether an image is fixed-size or dynamically allocated.

By default, images are in *normal* mode. To mark an existing image with one of the non-standard modes listed below, use **VBoxManage modifymedium**. See [Section 8.31, “VBoxManage modifymedium”](#). Alternatively, use **VBoxManage storageattach** to attach the image to a VM and specify the `--mttype` argument. See [Section 8.26, “VBoxManage storageattach”](#).

The available virtual disk image modes are as follows:

- **Normal images** have no restrictions on how guests can read from and write to the disk. This is the default image mode.

When you take a snapshot of your virtual machine as described in [Section 1.11, “Snapshots”](#), the state of a normal hard disk is recorded together with the snapshot, and when reverting to the snapshot, its state will be fully reset.

The image file itself is not reset. Instead, when a snapshot is taken, Oracle VM VirtualBox *freezes* the image file and no longer writes to it. For the write operations from the VM, a second, *differencing* image file is created which receives only the changes to the original image. See [Section 5.5, “Differencing Images”](#).

While you can attach the same normal image to more than one virtual machine, only one of these virtual machines attached to the same image file can be executed simultaneously, as otherwise there would be conflicts if several machines write to the same image file.

- **Write-through hard disks** are completely unaffected by snapshots. Their state is *not* saved when a snapshot is taken, and not restored when a snapshot is restored.
- **Shareable hard disks** are a variant of write-through hard disks. In principle they behave exactly the same. Their state is *not* saved when a snapshot is taken, and not restored when a snapshot is restored. The difference only shows if you attach such disks to several VMs. Shareable disks may be attached to several VMs which may run concurrently. This makes them suitable for use by cluster filesystems between VMs and similar applications which are explicitly prepared to access a disk concurrently. Only fixed size images can be used in this way, and dynamically allocated images are rejected.

### Warning

This is an expert feature, and misuse can lead to data loss, as regular filesystems are not prepared to handle simultaneous changes by several parties.

- **Immutable images** only remember write accesses temporarily while the virtual machine is running. All changes are lost when the virtual machine is powered on the next time. As a result, as opposed to Normal images, the same immutable image can be used with several virtual machines without restrictions.

Creating an immutable image makes little sense since it would be initially empty and lose its contents with every machine restart. You would have a disk that is always unformatted when the machine starts up. Instead, you can first create a normal image and then later mark it as immutable when you decide that the contents are useful.

If you take a snapshot of a machine with immutable images, then on every machine power-up, those images are reset to the state of the last (current) snapshot, instead of the state of the original immutable image.

### Note

As a special exception, immutable images are *not* reset if they are attached to a machine in a saved state or whose last snapshot was taken while the machine was running. This is called an *online snapshot*. As a result, if the machine's current snapshot is an online snapshot, its immutable images behave exactly like the a normal image. To reenble the automatic resetting of such images, delete the current snapshot of the machine.

Oracle VM VirtualBox never writes to an immutable image directly at all. All write operations from the machine are directed to a differencing image. The next time the VM is powered on, the differencing image is reset so that every time the VM starts, its immutable images have exactly the same content.

The differencing image is only reset when the machine is powered on from within Oracle VM VirtualBox, not when you reboot by requesting a reboot from within the machine. This is also why immutable images behave as described above when snapshots are also present, which use differencing images as well.

If the automatic discarding of the differencing image on VM startup does not fit your needs, you can turn it off using the `autoreset` parameter of **VBoxManage modifymedium**. See [Section 8.31, "VBoxManage modifymedium"](#).

- **Multiattach mode images** can be attached to more than one virtual machine at the same time, even if these machines are running simultaneously. For each virtual machine to which such an image is attached, a differencing image is created. As a result, data that is written to such a virtual disk by one machine is not seen by the other machines to which the image is attached. Each machine creates its own write history of

the multiattach image.

Technically, a multiattach image behaves identically to an immutable image except the differencing image is not reset every time the machine starts.

This mode is useful for sharing files which are almost never written, for instance picture galleries, where every guest changes only a small amount of data and the majority of the disk content remains unchanged. The modified blocks are stored in differencing images which remain relatively small and the shared content is stored only once at the host.

- **Read-only images** are used automatically for CD/DVD images, since CDs/DVDs can never be written to.

The following scenario illustrates the differences between the various image modes, with respect to snapshots.

Assume you have installed your guest OS in your VM, and you have taken a snapshot. Later, your VM is infected with a virus and you would like to go back to the snapshot. With a normal hard disk image, you simply restore the snapshot, and the earlier state of your hard disk image will be restored as well and your virus infection will be undone. With an immutable hard disk, all it takes is to shut down and power on your VM, and the virus infection will be discarded. With a write-through image however, you cannot easily undo the virus infection by means of virtualization, but will have to disinfect your virtual machine like a real computer.

You might find write-through images useful if you want to preserve critical data irrespective of snapshots. As you can attach more than one image to a VM, you may want to have one immutable image for the OS and one write-through image for your data files.

## 5.5. Differencing Images

The previous section mentioned differencing images and how they are used with snapshots, immutable images, and multiple disk attachments. This section describes in more detail how differencing images work.

A differencing image is a special disk image that only holds the differences to another image. A differencing image by itself is useless, it must always refer to another image. The differencing image is then typically referred to as a *child*, which holds the differences to its *parent*.

When a differencing image is active, it receives all write operations from the virtual machine instead of its parent. The differencing image only contains the sectors of the virtual hard disk that have changed since the differencing image was created. When the machine reads a sector from such a virtual hard disk, it looks into the differencing image first. If the sector is present, it is returned from there. If not, Oracle VM VirtualBox looks into the parent. In other words, the parent becomes *read-only*. It is never written to again, but it is read from if a sector has not changed.

Differencing images can be chained. If another differencing image is created for a virtual disk that already has a differencing image, then it becomes a *grandchild* of the original parent. The first differencing image then becomes read-only as well, and write operations only go to the second-level differencing image. When reading from the virtual disk, Oracle VM VirtualBox needs to look into the second differencing image first, then into the first if the sector was not found, and then into the original image.

There can be an unlimited number of differencing images, and each image can have more than one child. As a result, the differencing images can form a complex tree with parents, siblings, and children, depending on how complex your machine configuration is. Write operations always go to the one *active* differencing image that is attached to the machine, and for read operations, Oracle VM VirtualBox may need to look up all the parents in the chain until the sector in question is found. You can view such a tree in the Virtual Media Manager.

**Figure 5.3. Differencing Images, Shown in Virtual Media Manager**

Name	Virtual Size	Actual Size
Win7.vdi	20,00 GB	12,57 GB
Win8-EFI.vdi	25,00 GB	3,00 MB
Win8.vdi	25,00 GB	9,50 GB
XP.vdi	10,00 GB	7,54 GB
{91b25d96-9794-49fe-aa60-417c4d671f76}.vdi	10,00 GB	2,00 MB
{78ab3b27-6f37-4040-a77f-ff59fa57bb25}.vdi	10,00 GB	2,00 MB
{4e36fca0-3ef2-492f-8bf8-94e4a6ea89be}.vdi	10,00 GB	2,00 MB
{b40e402d-29c6-4052-88ca-0a8ee8f9ee3...}	10,00 GB	2,00 MB

Type:	Normal
Location:	/Users/vbox/VirtualBox VMs/Windows/XP/XP.vdi
Format:	VDI
Storage details:	Dynamically allocated storage
Attached to:	XP (Snapshot 1)
UUID:	b3ac6430-f770-4128-b94e-6b9569d629e9

In all of these situations, from the point of view of the virtual machine, the virtual hard disk behaves like any other disk. While the virtual machine is running, there is a slight run-time I/O overhead because Oracle VM VirtualBox might need to look up sectors several times. This is not noticeable however since the tables with sector information are always kept in memory and can be looked up quickly.

Differencing images are used in the following situations:

- **Snapshots.** When you create a snapshot, as explained in the previous section, Oracle VM VirtualBox *freezes* the images attached to the virtual machine and creates differencing images for each image that is not in *write-through* mode. From the point of view of the virtual machine, the virtual disks continue to operate before, but all write operations go into the differencing images. Each time you create another snapshot, for each hard disk attachment, another differencing image is created and attached, forming a chain or tree.

In the above screenshot, you see that the original disk image is now attached to a snapshot, representing the state of the disk when the snapshot was taken.

If you *restore* a snapshot, and want to go back to the exact machine state that was stored in the snapshot, the following happens:

- Oracle VM VirtualBox copies the virtual machine settings that were copied into the snapshot back to the virtual machine. As a result, if you have made changes to the machine configuration since taking the snapshot, they are undone.
- If the snapshot was taken while the machine was running, it contains a saved machine state, and that state is restored as well. After restoring the snapshot, the machine will then be in Saved state and resume execution from there when it is next started. Otherwise the machine will be in Powered Off state and do a full boot.
- For each disk image attached to the machine, the differencing image holding all the write operations since the current snapshot was taken is thrown away, and the original parent image is made active again. If you restored the root snapshot, then this will be the root disk image for each attachment. Otherwise, some other differencing image descended from it. This effectively restores the old machine state.

If you later *delete* a snapshot in order to free disk space, for each disk attachment, one of the differencing images becomes obsolete. In this case, the differencing image of the disk attachment cannot simply be deleted. Instead, Oracle VM VirtualBox needs to look at each sector of the differencing image and needs to copy it back into its parent. This is called "merging" images and can be a potentially lengthy process, depending on how large the differencing image is. It can also temporarily need a considerable amount of extra disk space, before the differencing image obsoleted by the merge operation is deleted.

- **Immutable images.** When an image is switched to immutable mode, a differencing image is created as well. As with snapshots, the parent image then becomes read-only, and the differencing image receives all the write operations. Every time the virtual machine is started, all the immutable images which are attached to it have their respective differencing image thrown away, effectively resetting the virtual machine's virtual disk with every restart.

## 5.6. Cloning Disk Images

You can duplicate hard disk image files on the same host to quickly produce a second virtual machine with the same OS setup. However, you should *only* make copies of virtual disk images using the utility supplied with Oracle VM VirtualBox. See [Section 8.32, “VBoxManage clonemedium”](#). This is because Oracle VM VirtualBox assigns a UUID to each disk image, which is also stored inside the image, and Oracle VM VirtualBox will refuse to work with two images that use the same number. If you do accidentally try to reimport a disk image which you copied normally, you can make a second copy using the **VBoxManage clonevm** command and import that instead.

Note that Linux distributions identify the boot hard disk from the ID of the drive. The ID Oracle VM VirtualBox reports for a drive is determined from the UUID of the virtual disk image. So if you clone a disk image and try to boot the copied image the guest might not be able to determine its own boot disk as the UUID changed. In this case you have to adapt the disk ID in your boot loader script, for example `/boot/grub/menu.lst`. The disk ID looks like the following:

```
scsi-SATA_VBOX_HARDDISK_VB5cfdb1e2-c251e503
```

The ID for the copied image can be determined as follows:

```
hdparm -i /dev/sda
```

## 5.7. Host Input/Output Caching

Oracle VM VirtualBox can optionally disable the I/O caching that the host OS would otherwise perform on disk image files.

Traditionally, Oracle VM VirtualBox has opened disk image files as normal files, which results in them being cached by the host OS like any other file. The main advantage of this is speed: when the guest OS writes to disk and the host OS cache uses delayed writing, the write operation can be reported as completed to the guest OS quickly while the host OS can perform the operation asynchronously. Also, when you start a VM a second time and have enough memory available for the OS to use for caching, large parts of the virtual disk may be in system memory, and the VM can access the data much faster.

Note that this applies only to image files. Buffering does not occur for virtual disks residing on remote iSCSI storage, which is the more common scenario in enterprise-class setups. See [Section 5.10, “iSCSI Servers”](#).

While buffering is a useful default setting for virtualizing a few machines on a desktop computer, there are some disadvantages to this approach:

- Delayed writing through the host OS cache is less secure. When the guest OS writes data, it considers the data written even though it has not yet arrived on a physical disk. If for some reason the write does not happen, such as power failure or host crash, the likelihood of data loss increases.
- Disk image files tend to be very large. Caching them can therefore quickly use up the entire host OS cache. Depending on the efficiency of the host OS caching, this may slow down the host immensely, especially if several VMs run at the same time. For example, on Linux hosts, host caching may result in Linux delaying all writes until the host cache is nearly full and then writing out all these changes at once, possibly stalling VM execution for minutes. This can result in I/O errors in the guest as I/O requests time out there.

- Physical memory is often wasted as guest OSes typically have their own I/O caches, which may result in the data being cached twice, in both the guest and the host caches, for little effect.

If you decide to disable host I/O caching for the above reasons, Oracle VM VirtualBox uses its own small cache to buffer writes, but no read caching since this is typically already performed by the guest OS. In addition, Oracle VM VirtualBox fully supports asynchronous I/O for its virtual SATA, SCSI, and SAS controllers through multiple I/O threads.

Since asynchronous I/O is not supported by IDE controllers, for performance reasons, you may want to leave host caching enabled for your VM's virtual IDE controllers.

For this reason, Oracle VM VirtualBox enables you to configure whether the host I/O cache is used for each I/O controller separately. Either select the **Use Host I/O Cache** check box in the **Storage** settings for a given virtual storage controller, or use the following **VBoxManage** command to disable the host I/O cache for a virtual storage controller:

```
VBoxManage storagectl "VM name" --name <controllername> --hostiocache off
```

See [Section 8.27, “VBoxManage storagectl”](#).

For the above reasons, Oracle VM VirtualBox uses SATA controllers by default for new virtual machines.

## 5.8. Limiting Bandwidth for Disk Images

Oracle VM VirtualBox supports limiting of the maximum bandwidth used for asynchronous I/O. Additionally it supports sharing limits through bandwidth groups for several images. It is possible to have more than one such limit.

Limits are configured using **VBoxManage**. The example below creates a bandwidth group named Limit, sets the limit to 20 MB per second, and assigns the group to the attached disks of the VM:

```
VBoxManage bandwidthctl "VM name" add Limit --type disk --limit 20M
VBoxManage storageattach "VM name" --storagectl "SATA" --port 0 --device 0 --type hdd
--medium disk1.vdi --bandwidthgroup Limit
VBoxManage storageattach "VM name" --storagectl "SATA" --port 1 --device 0 --type hdd
--medium disk2.vdi --bandwidthgroup Limit
```

All disks in a group share the bandwidth limit, meaning that in the example above the bandwidth of both images combined can never exceed 20 MBps. However, if one disk does not require bandwidth the other can use the remaining bandwidth of its group.

The limits for each group can be changed while the VM is running, with changes being picked up immediately. The example below changes the limit for the group created in the example above to 10 MBps:

```
VBoxManage bandwidthctl "VM name" set Limit --limit 10M
```

## 5.9. CD/DVD Support

Virtual CD/DVD drives by default support only reading. The medium configuration is changeable at runtime. You can select between the following options to provide the medium data:

- **Host Drive** defines that the guest can read from the medium in the host drive.
- **Image file** gives the guest read-only access to the data in the image. This is typically an ISO file.
- **Empty** means a drive without an inserted medium.

Changing between the above, or changing a medium in the host drive that is accessed by a machine, or changing



an image file will signal a medium change to the guest OS. The guest OS can then react to the change, for example by starting an installation program.

Medium changes can be prevented by the guest, and Oracle VM VirtualBox reflects that by locking the host drive if appropriate. You can force a medium removal in such situations by using the VirtualBox Manager or the **VBoxManage** command line tool. Effectively this is the equivalent of the emergency eject which many CD/DVD drives provide, with all associated side effects. The guest OS can issue error messages, just like on real hardware, and guest applications may misbehave. Use this with caution.

### Note

The identification string of the drive provided to the guest, displayed by configuration tools such as the Windows Device Manager, is always VBOX CD-ROM, irrespective of the current configuration of the virtual drive. This is to prevent hardware detection from being triggered in the guest OS every time the configuration is changed.

The standard CD/DVD emulation enables reading of standard data CD and DVD formats only. As an experimental feature, for additional capabilities, it is possible to give the guest direct access to the CD/DVD host drive by enabling *passthrough* mode. Depending on the host hardware, this may potentially enable the following things to work:

- CD/DVD writing from within the guest, if the host DVD drive is a CD/DVD writer
- Playing audio CDs
- Playing encrypted DVDs

To enable host drive passthrough you can use the `--passthrough` option of the **VBoxManage storageattach** command. See [Section 8.26, “VBoxManage storageattach”](#).

Even if passthrough is enabled, unsafe commands, such as updating the drive firmware, will be blocked. Video CD formats are never supported, not even in passthrough mode, and cannot be played from a virtual machine.

On Oracle Solaris hosts, passthrough requires running Oracle VM VirtualBox with real root permissions due to security measures enforced by the host.

## 5.10. iSCSI Servers

iSCSI stands for *Internet SCSI* and is a standard that supports use of the SCSI protocol over Internet (TCP/IP) connections. Especially with the advent of Gigabit Ethernet, it has become affordable to attach iSCSI storage servers simply as remote hard disks to a computer network. In iSCSI terminology, the server providing storage resources is called an *iSCSI target*, while the client connecting to the server and accessing its resources is called an *iSCSI initiator*.

Oracle VM VirtualBox can transparently present iSCSI remote storage to a virtual machine as a virtual hard disk. The guest OS will not see any difference between a virtual disk image (VDI file) and an iSCSI target. To achieve this, Oracle VM VirtualBox has an integrated iSCSI initiator.

Oracle VM VirtualBox's iSCSI support has been developed according to the iSCSI standard and should work with all standard-conforming iSCSI targets. To use an iSCSI target with Oracle VM VirtualBox, you must use the command line. See [Section 8.26, “VBoxManage storageattach”](#).

## 5.11. vboximg-mount: A Utility for FUSE Mounting a Virtual Disk Image



**vboximg-mount** is a command line utility for Mac OS and Linux hosts that provides raw access to an Oracle VM VirtualBox virtual disk image on the host system. Use this utility to mount, view, and optionally modify the disk image contents.

The utility is based on Filesystem in Userspace (FUSE) technology and uses the VirtualBox runtime engine. Ensure that Oracle VM VirtualBox is running on the host system.

### Note

When using **vboximg-mount**, ensure that the following conditions apply:

- The disk image is not being used by any other systems, such as by guest VMs.
- No VMs are running on the host system.

Raw access using FUSE is preferred over direct loopback mounting of virtual disk images, because it is snapshot aware. It can selectively merge disk differencing images in an exposed virtual hard disk, providing historical or up-to-date representations of the virtual disk contents.

**vboximg-mount** enables you to view information about registered VMs, their attached disk media, and any snapshots. Also, you can view partition information for a disk image.

The **vboximg-mount** command includes experimental read-only access to file systems inside a VM disk image. This feature enables you to extract some files from the disk image without starting the VM and without requiring third-party file system drivers on the host system. FAT, NTFS, ext2, ext3, and ext4 file systems are supported.

Use the `--help` option to view information about the **vboximg-mount** command usage. The complete command reference is described in [Section 8.55, “vboximg-mount”](#).

When **vboximg-mount** mounts an Oracle VM VirtualBox disk image, it creates a one level deep file system at a mount point that you specify. The file system includes a device node that represents the synthesized disk image as a readable or readable-writeable bytestream. This bytestream can be mounted either by using the host OS or by using other FUSE-based file systems.

## 5.11.1. Viewing Detailed Information About a Virtual Disk Image

The following examples show how to use the **vboximg-mount** command to view information about virtual disk images.

The following command outputs detailed information about all registered VMs and associated snapshots:

```
$ vboximg-mount --list --verbose
```

```
-----
VM Name:    "macOS High Sierra 10.13"
UUID:      3887d96d-831c-4187-a55a-567c504ff0e1
Location:   /Volumes/work/vm_guests/macOS High Sierra 10.13/macOS High Sierra 10.13.vbox
-----
HDD base:   "macOS High Sierra 10.13.vdi"
UUID:      f9ea7173-6869-4aa9-b487-68023a655980
Location:   /Volumes/work/vm_guests/macOS High Sierra 10.13/macOS High Sierra 10.13.vdi

Diff 1:
  UUID:      98c2bac9-cf37-443d-a935-4e879b70166d
  Location:   /Volumes/work/vm_guests/macOS High Sierra 10.13/
  Snapshots/{98c2bac9-cf37-443d-a935-4e879b70166d}.vdi
Diff 2:
  UUID:      f401f381-7377-40b3-948e-3c61241b1a42
  Location:   /Volumes/work/vm_guests/macOS High Sierra 10.13/
  Snapshots/{f401f381-7377-40b3-948e-3c61241b1a42}.vdi
```

```

-----
HDD base:  "simple_fixed_disk.vdi"
UUID:      ffb4d7e-1277-489d-8173-22ca7660773d
Location:  /Volumes/work/vm_guests/macOS High Sierra 10.13/simple_fixed_disk.vdi

Diff 1:
  UUID:      aecab681-0d2d-468b-8682-93f79dc97a48
  Location:  /Volumes/work/vm_guests/macOS High Sierra 10.13/
  Snapshots/{aecab681-0d2d-468b-8682-93f79dc97a48}.vdi
Diff 2:
  UUID:      70d6b34d-8422-47fa-8521-3b6929a1971c
  Location:  /Volumes/work/vm_guests/macOS High Sierra 10.13/
  Snapshots/{70d6b34d-8422-47fa-8521-3b6929a1971c}.vdi
-----
VM Name:   "debian"
UUID:      5365ab5f-470d-44c0-9863-dad532ee5905
Location:  /Volumes/work/vm_guests/debian/debian.vbox
-----
HDD base:  "debian.vdi"
UUID:      96d2e92e-0d4e-46ab-a0f1-008fdbf997e7
Location:  /Volumes/work/vm_guests/debian/ol7.vdi

Diff 1:
  UUID:      f9cc866a-9166-42e9-a503-bbfe9b7312e8
  Location:  /Volumes/work/vm_guests/debian/Snapshots/
  {f9cc866a-9166-42e9-a503-bbfe9b7312e8}.vdi

```

The following command outputs partition information about the specified disk image:

```
$ vboximg-mount --image=f9ea7173-6869-4aa9-b487-68023a655980 --list
```

Virtual disk image:

```

Path: /Volumes/work/vm_guests/macOS High Sierra 10.13/macOS High Sierra 10.13.vdi
UUID: f9ea7173-6869-4aa9-b487-68023a655980

```

#	Start	Sectors	Size	Offset	Type
1	40	409599	199.9M	20480	EFI System
2	409640	67453071	32.1G	209735680	Hierarchical File System Plus (HFS+)
3	67862712	1269535	107.8M	34745708544	Apple Boot (Recovery HD)

### 5.11.2. Mounting a Virtual Disk Image

The following steps show how to use the **vboximg-mount** command to mount a partition of a virtual disk image on the host OS.

1. Create a mount point on the host OS. For example:

```
$ mkdir macos_sysdisk
```

2. Show partition information about the virtual disk image.

```
$ vboximg-mount --image=uuid --list
```

where *uuid* is the UUID of the disk image.

3. Use **vboximg-mount** to perform a FUSE mount of a partition on the virtual disk image. For example:

```
$ vboximg-mount --image=uuid -p 2 macos_sysdisk
```

where *uuid* is the UUID for the disk image.

In this example, partition 2 is mounted on the `macos_sysdisk` mount point. The mount includes all snapshots

for the disk image.

4. Use the host OS to mount the `vhdd` device node. The FUSE-mounted device node represents the virtual disk image.

```
$ ls macos_sysdisk
  macOS High Sierra 10.13.vdi  vhdd
$ sudo mount macos_sysdisk/vhdd /mnt
```