Chapter 1. First Steps

Table of Contents

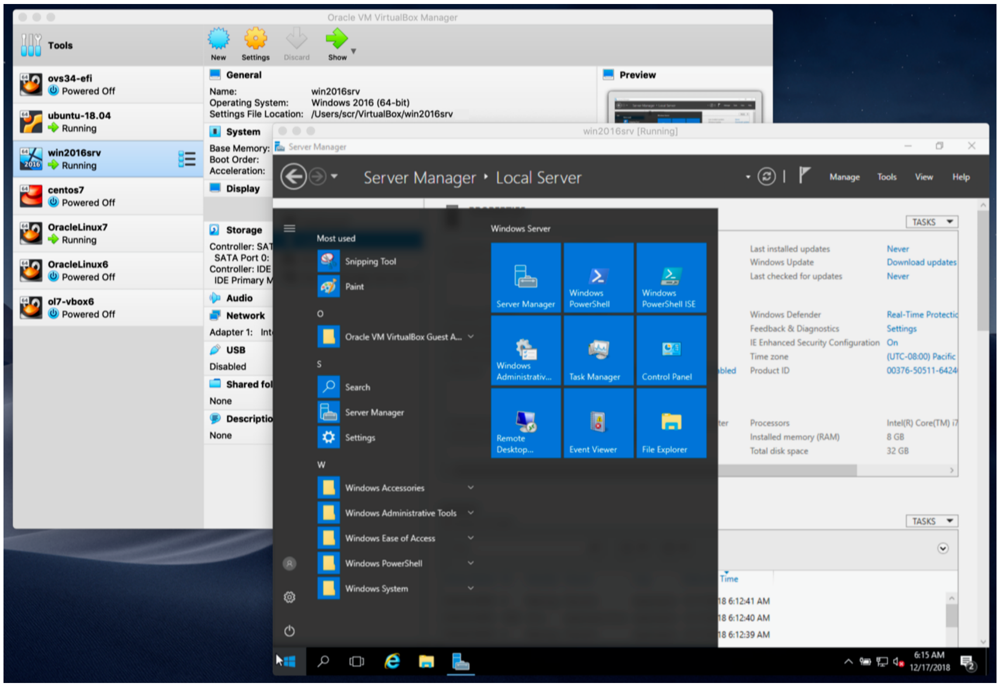
Welcome to Oracle VM VirtualBox.

Oracle VM VirtualBox is a cross-platform virtualization application. What does that mean? For one thing, it installs on your existing Intel or AMD-based computers, whether they are running Windows, Mac OS X, Linux, or Oracle Solaris operating systems (OSes). Secondly, it extends the capabilities of your existing computer so that it can run multiple OSes, inside multiple virtual machines, at the same time. As an example, you can run Windows and Linux on your Mac, run Windows Server 2016 on your Linux server, run Linux on your Windows PC, and so on, all alongside your existing applications. You can install and run as many virtual machines as you like. The only practical limits are disk space and memory.

Oracle VM VirtualBox is deceptively simple yet also very powerful. It can run everywhere from small embedded systems or desktop class machines all the way up to datacenter deployments and even Cloud environments.

The following screenshot shows how Oracle VM VirtualBox, installed on an Apple Mac OS X computer, is running Windows Server 2016 in a virtual machine window.

Figure 1.1. Windows Server 2016 Virtual Machine, Displayed on a Mac OS X Host



Name: win2016srv
Operating System: Windows 2016 (64-bit)
Settings File Location: /Users/scr/VirtualBox/win2016srv

In this User Manual, we will begin simply with a quick introduction to virtualization and how to get your first virtual machine running with the easy-to-use Oracle VM VirtualBox graphical user interface. Subsequent chapters will go into much more detail covering more powerful tools and features, but fortunately, it is not necessary to read the entire User Manual before you can use Oracle VM VirtualBox.

You can find a summary of Oracle VM VirtualBox's capabilities in Section 1.3, “Features Overview”. For existing Oracle VM VirtualBox users who just want to find out what is new in this release, see the Chapter 15, *Change Log*.

1.1. Why is Virtualization Useful?

The techniques and features that Oracle VM VirtualBox provides are useful in the following scenarios:

- **Running multiple operating systems simultaneously.** Oracle VM VirtualBox enables you to run more than one OS at a time. This way, you can run software written for one OS on another, such as Windows software on Linux or a Mac, without having to reboot to use it. Since you can configure what kinds of *virtual* hardware should be presented to each such OS, you can install an old OS such as DOS or OS/2 even if your real computer's hardware is no longer supported by that OS.

- **Easier software installations.** Software vendors can use virtual machines to ship entire software configurations. For example, installing a complete mail server solution on a real machine can be a tedious task. With Oracle VM VirtualBox, such a complex setup, often called an *appliance*, can be packed into a virtual machine. Installing and running a mail server becomes as easy as importing such an appliance into Oracle VM VirtualBox.

- **Testing and disaster recovery.** Once installed, a virtual machine and its virtual hard disks can be considered a *container* that can be arbitrarily frozen, woken up, copied, backed up, and transported between hosts.

  On top of that, with the use of another Oracle VM VirtualBox feature called *snapshots*, one can save a particular state of a virtual machine and revert back to that state, if necessary. This way, one can freely experiment with a computing environment. If something goes wrong, such as prolems after installing software or infecting the guest with a virus, you can easily switch back to a previous snapshot and avoid the need of frequent backups and restores.

  Any number of snapshots can be created, allowing you to travel back and forward in virtual machine time. You can delete snapshots while a VM is running to reclaim disk space.

- **Infrastructure consolidation.** Virtualization can significantly reduce hardware and electricity costs. Most of the time, computers today only use a fraction of their potential power and run with low average system loads. A lot of hardware resources as well as electricity is thereby wasted. So, instead of running many such physical computers that are only partially used, one can pack many virtual machines onto a few powerful hosts and balance the loads between them.

1.2. Some Terminology

When dealing with virtualization, and also for understanding the following chapters of this documentation, it helps to acquaint oneself with a bit of crucial terminology, especially the following terms:

- **Host operating system (host OS).** This is the OS of the physical computer on which Oracle VM VirtualBox was installed. There are versions of Oracle VM VirtualBox for Windows, Mac OS X, Linux, and Oracle Solaris hosts. See Section 1.4, “Supported Host Operating Systems”.

  Most of the time, this manual discusses all Oracle VM VirtualBox versions together. There may be platform-specific differences which we will point out where appropriate.

- **Guest operating system (guest OS).** This is the OS that is running inside the virtual machine. Theoretically, Oracle VM VirtualBox can run any x86 OS. such as DOS, Windows, OS/2, FreeBSD, and OpenBSD. But to achieve near-native performance of the guest code on your machine, we had to go through a lot of optimizations that are specific to certain OSes. So while your favorite OS *may* run as a guest, we officially support and optimize for a select few, which include the most common OSes.

  See Section 3.1, “Supported Guest Operating Systems”.

- **Virtual machine (VM).** This is the special environment that Oracle VM VirtualBox creates for your guest OS while it is running. In other words, you run your guest OS *in* a VM. Normally, a VM will be shown as a window on your computer's desktop, but depending on which of the various frontends of Oracle VM VirtualBox you use, it can be displayed in full screen mode or remotely on another computer.

  In a more abstract way, internally, Oracle VM VirtualBox thinks of a VM as a set of parameters that determine its behavior. They include hardware settings, such as: how much memory the VM should have, what hard disks Oracle VM VirtualBox should virtualize through which container files, what CDs are mounted. They also include state information, such as: whether the VM is currently running, saved, if the VM has snapshots. These settings are mirrored in the VirtualBox Manager window, as well as the **VBoxManage** command. See Chapter 8, *VBoxManage*. In other words, a VM is also what you can see in its **Settings** dialog.

- **Guest Additions.** This refers to special software packages which are shipped with Oracle VM VirtualBox but designed to be installed *inside* a VM to improve performance of the guest OS and to add extra features. See Chapter 4, *Guest Additions*.

1.3. Features Overview

The following is a brief outline of Oracle VM VirtualBox's main features:

- **Portability.** Oracle VM VirtualBox runs on a large number of 32-bit and 64-bit host OS. See Section 1.4, “Supported Host Operating Systems”.

  Oracle VM VirtualBox is a so-called *hosted* hypervisor, sometimes referred to as a *type 2* hypervisor. Whereas a *bare-metal* or *type 1* hypervisor would run directly on the hardware, Oracle VM VirtualBox requires an existing OS to be installed. It can thus run alongside existing applications on that host.

  To a very large degree, Oracle VM VirtualBox is functionally identical on all of the host platforms, and the same file and image formats are used. This enables you to run virtual machines created on one host on another host with a different host OS. For example, you can create a virtual machine on Windows and then run it under Linux.

  In addition, virtual machines can easily be imported and exported using the Open Virtualization Format (OVF), an industry standard created for this purpose. You can even import OVFs that were created with a different virtualization software. See Section 1.15, “Importing and Exporting Virtual Machines”.

- **No hardware virtualization required.** For many scenarios, Oracle VM VirtualBox does not require the processor features built into newer hardware like Intel VT-x or AMD-V. As opposed to many other virtualization solutions, you can therefore use Oracle VM VirtualBox even on older hardware where these features are not present. See Section 10.3, “Hardware vs. Software Virtualization”.

- **Guest Additions: shared folders, seamless windows, 3D virtualization.** The Oracle VM VirtualBox Guest Additions are software packages which can be installed *inside* of supported guest systems to improve their performance and to provide additional integration and communication with the host system. After installing the Guest Additions, a virtual machine will support automatic adjustment of video resolutions, seamless windows, accelerated 3D graphics and more. See Chapter 4, *Guest Additions*.

  In particular, Guest Additions provide for "shared folders", which let you access files from the host system from within a guest machine. See Section 4.3, “Shared Folders”.

- **Great hardware support.** Among others, Oracle VM VirtualBox supports the following:

  - **Guest multiprocessing (SMP).** Oracle VM VirtualBox can present up to 32 virtual CPUs to each virtual machine, irrespective of how many CPU cores are physically present on your host.

  - **USB device support.** Oracle VM VirtualBox implements a virtual USB controller and enables you to connect arbitrary USB devices to your virtual machines without having to install device-specific drivers on the host. USB support is not limited to certain device categories. See Section 3.11.1, “USB Settings”.

  - **Hardware compatibility.** Oracle VM VirtualBox virtualizes a vast array of virtual devices, among them many devices that are typically provided by other virtualization platforms. That includes IDE, SCSI and SATA hard disk controllers, several virtual network cards and sound cards, virtual serial and parallel ports and an Input/Output Advanced Programmable Interrupt Controller (I/O APIC), which is found in many modern PC systems. This eases cloning of PC images from real machines and importing of third-party virtual machines into Oracle VM VirtualBox.

  - **Full ACPI support.** The Advanced Configuration and Power Interface (ACPI) is fully supported by Oracle VM VirtualBox. This eases cloning of PC images from real machines or third-party virtual machines into Oracle VM VirtualBox. With its unique *ACPI power status support*, Oracle VM VirtualBox can even report to ACPI-aware guest OSes the power status of the host. For mobile systems running on battery, the guest can thus enable energy saving and notify the user of the remaining power, for example in full screen modes.

  - **Multiscreen resolutions.** Oracle VM VirtualBox virtual machines support screen resolutions many times that of a physical screen, allowing them to be spread over a large number of screens attached to the host system.

  - **Built-in iSCSI support.** This unique feature enables you to connect a virtual machine directly to an iSCSI storage server without going through the host system. The VM accesses the iSCSI target directly without the extra overhead that is required for virtualizing hard disks in container files. See Section 5.10, “iSCSI Servers”.

  - **PXE Network boot.** The integrated virtual network cards of Oracle VM VirtualBox fully support remote booting using the Preboot Execution Environment (PXE).

- **Multigeneration branched snapshots.** Oracle VM VirtualBox can save arbitrary snapshots of the state of the virtual machine. You can go back in time and revert the virtual machine to any such snapshot and start an alternative VM configuration from there, effectively creating a whole snapshot tree. See Section 1.11, “Snapshots”. You can create and delete snapshots while the virtual machine is running.

- **VM groups.** Oracle VM VirtualBox provides a groups feature that enables the user to organize and control virtual machines collectively, as well as individually. In addition to basic groups, it is also possible for any VM to be in more than one group, and for groups to be nested in a hierarchy. This means you can have groups of groups. In general, the operations that can be performed on groups are the same as those that can be applied to individual VMs: Start, Pause, Reset, Close (Save state, Send Shutdown, Poweroff), Discard Saved State, Show in File System, Sort.

- **Clean architecture and unprecedented modularity.** Oracle VM VirtualBox has an extremely modular design with well-defined internal programming interfaces and a clean separation of client and server code. This makes it easy to control it from several interfaces at once. For example, you can start a VM simply by clicking on a button in the Oracle VM VirtualBox graphical user interface and then control that machine from the command line, or even remotely. See Section 1.17, "Alternative Front-Ends".

  Due to its modular architecture, Oracle VM VirtualBox can also expose its full functionality and configurability through a comprehensive **software development kit (SDK),** which enables integration of Oracle VM VirtualBox with other software systems. See Chapter 11, *Oracle VM VirtualBox Programming Interfaces*.

- **Remote machine display.** The VirtualBox Remote Desktop Extension (VRDE) enables high-performance remote access to any running virtual machine. This extension supports the Remote Desktop Protocol (RDP) originally built into Microsoft Windows, with special additions for full client USB support.

  The VRDE does not rely on the RDP server that is built into Microsoft Windows. Instead, the VRDE is plugged directly into the virtualization layer. As a result, it works with guest OSes other than Windows, even in text mode, and does not require application support in the virtual machine either. The VRDE is described in detail in Section 7.1, "Remote Display (VRDP Support)".

  On top of this special capacity, Oracle VM VirtualBox offers you more unique features:

  - **Extensible RDP authentication.** Oracle VM VirtualBox already supports Winlogon on Windows and PAM on Linux for RDP authentication. In addition, it includes an easy-to-use SDK which enables you to create arbitrary interfaces for other methods of authentication. See Section 7.1.5, "RDP Authentication".

  - **USB over RDP.** Using RDP virtual channel support, Oracle VM VirtualBox also enables you to connect arbitrary USB devices locally to a virtual machine which is running remotely on a Oracle VM VirtualBox RDP server. See Section 7.1.4, "Remote USB".

1.4. Supported Host Operating Systems

Currently, Oracle VM VirtualBox runs on the following host OSes:

- **Windows hosts (64-bit):**

  - Windows 7
  - Windows 8
  - Windows 8.1
  - Windows 10 RTM (1507) build 10240
  - Windows 10 November Update (1511) build 10586
  - Windows 10 Anniversary Update (1607) build 14393
  - Windows 10 Creators Update (1703) build 15063
  - Windows 10 Fall Creators Update (1709) build 16299
  - Windows 10 April 2018 Update (1803) build 17134
  - Windows 10 October 2018 Update (1809) build 17763
  - Windows Server 2008 R2
  - Windows Server 2012
  - Windows Server 2012 R2

    - Windows Server 2016
    - Windows Server 2019

- **Mac OS X hosts (64-bit):**

    - 10.12 (Sierra)
    - 10.13 (High Sierra)
    - 10.14 (Mojave)

  Intel hardware is required. See also Chapter 14, *Known Limitations*.

- **Linux hosts (64-bit).** Includes the following:

    - Ubuntu 16.04 LTS, 18.04 LTS and 18.10
    - Debian GNU/Linux 9 ("Stretch")
    - Oracle Linux 6 and 7
    - Redhat Enterprise Linux 6 and 7
    - Fedora 28 and 29
    - Gentoo Linux
    - SUSE Linux Enterprise server 12 and 15
    - openSUSE Leap 42.3 and 15.0

  It should be possible to use Oracle VM VirtualBox on most systems based on Linux kernel 2.6 or 3.x using either the Oracle VM VirtualBox installer or by doing a manual installation. See Section 2.3, “Installing on Linux Hosts”. However, the formally tested and supported Linux distributions are those for which we offer a dedicated package.

  Note that Linux 2.4-based host OSes are no longer supported.

- **Oracle Solaris hosts (64-bit only).** The following versions are supported with the restrictions listed in Chapter 14, *Known Limitations*:

    - Oracle Solaris 11

Note that the above list is informal. Oracle support for customers who have a support contract is limited to a subset of the listed host OSes. Also, any feature which is marked as *experimental* is not supported. Feedback and suggestions about such features are welcome.

1.5. Host CPU Requirements

SSE2 is required, starting with Oracle VM VirtualBox version 5.2.10 and version 5.1.24.

1.6. Installing Oracle VM VirtualBox and Extension Packs

Oracle VM VirtualBox comes in many different packages, and installation depends on your host OS. If you have installed software before, installation should be straightforward. On each host platform, Oracle VM VirtualBox uses the installation method that is most common and easy to use. If you run into trouble or have special requirements, see Chapter 2, *Installation Details* for details about the various installation methods.

Oracle VM VirtualBox is split into the following components:

- **Base package.** The base package consists of all open source components and is licensed under the GNU General Public License V2.

- **Extension packs.** Additional extension packs can be downloaded which extend the functionality of the Oracle VM VirtualBox base package. Currently, Oracle provides a single extension pack, available from: http://www.virtualbox.org. The extension pack provides the following added functionality:

  1. The virtual USB 2.0 (EHCI) device. See Section 3.11.1, "USB Settings".

  2. The virtual USB 3.0 (xHCI) device. See Section 3.11.1, "USB Settings".

  3. VirtualBox Remote Desktop Protocol (VRDP) support. See Section 7.1, "Remote Display (VRDP Support)".

  4. Host webcam passthrough. See Section 9.6, "Webcam Passthrough".

  5. Intel PXE boot ROM.

  6. Experimental support for PCI passthrough on Linux hosts. See Section 9.5, "PCI Passthrough".

  7. Disk image encryption with AES algorithm. See Section 9.30, "Encryption of Disk Images".

  Oracle VM VirtualBox extension packages have a `.vbox-extpack` file name extension. To install an extension, simply double-click on the package file and a **Network Operations Manager** window is shown to guide you through the required steps.

  To view the extension packs that are currently installed, start the VirtualBox Manager, as shown in Section 1.7, "Starting Oracle VM VirtualBox". From the **File** menu, select **Preferences**. In the window that displays, go to the **Extensions** category. This shows you the extensions which are currently installed, and enables you to remove a package or add a new package.

  Alternatively, you can use the **VBoxManage** command line. See Section 8.43, "VBoxManage extpack".

1.7. Starting Oracle VM VirtualBox

After installation, you can start Oracle VM VirtualBox as follows:

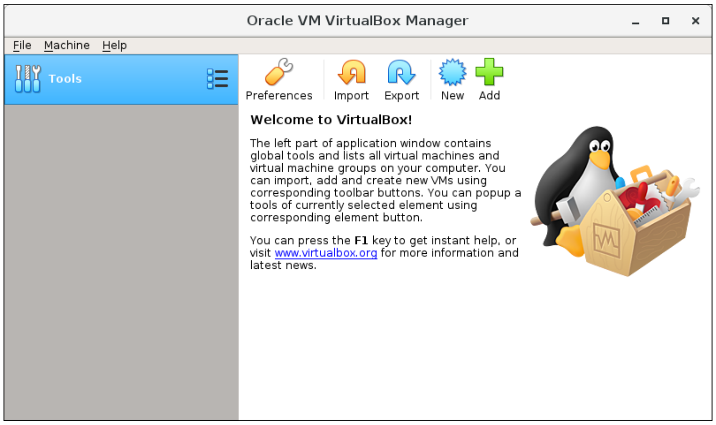
- On a Windows host, in the **Programs** menu, click on the item in the **VirtualBox** group. On Vista or Windows 7, you can also enter `VirtualBox` in the search box of the **Start** menu.

- On a Mac OS X host, in the Finder, double-click on the **VirtualBox** item in the Applications folder. You may want to drag this item onto your Dock.

- On a Linux or Oracle Solaris host, depending on your desktop environment, an Oracle VM VirtualBox item may have been placed in either the System or System Tools group of your **Applications** menu. Alternatively, you can enter `VirtualBox` in a terminal window.

When you start Oracle VM VirtualBox for the first time, a window like the following is displayed:

Figure 1.2. VirtualBox Manager Window, After Initial Startup



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

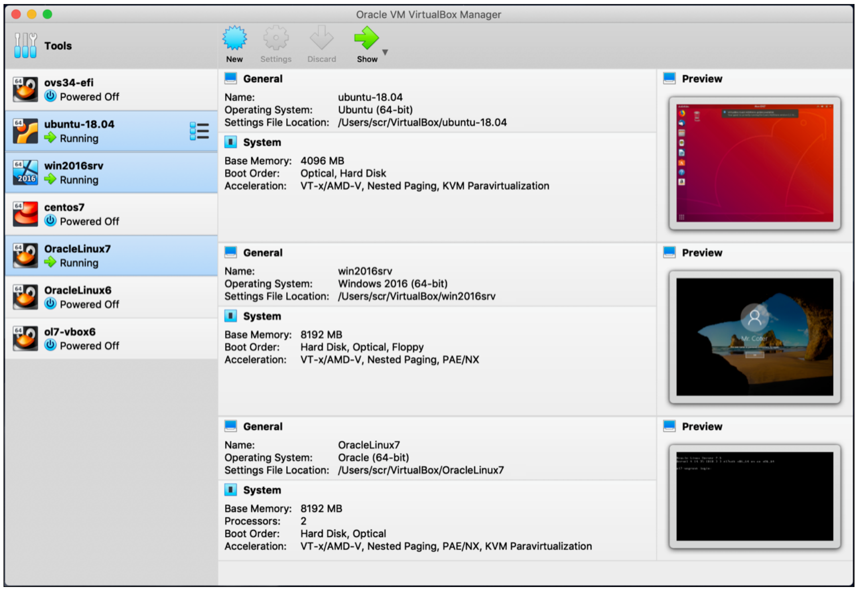
This window is called the **VirtualBox Manager**. The left pane will later list all your virtual machines. Since you have not yet created any virtual machines, this list is empty. The **Tools** button provides access to user tools, such as the Virtual Media Manager.

The pane on the right displays the properties of the currently selected virtual machine. Since you do not have any machines yet, the pane displays a welcome message.

The buttons on the right pane are used to create and work with VMs.

The following figure gives an idea of what Oracle VM VirtualBox might look like after you have created some VMs.

Figure 1.3. VirtualBox Manager Window, After Creating Virtual Machines



ovs34-efi Powered Off
ubuntu-18.04 Running
win2016srv Running
centos7 Powered Off
OracleLinux7 Running
OracleLinux6 Powered Off
ol7-vbox6 Powered Off

General
Name: ubuntu-18.04
Operating System: Ubuntu (64-bit)
Settings File Location: /Users/scr/VirtualBox/ubuntu-18.04

System
Base Memory: 4096 MB
Boot Order: Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, KVM Paravirtualization

General
Name: win2016srv
Operating System: Windows 2016 (64-bit)
Settings File Location: /Users/scr/VirtualBox/win2016srv

System
Base Memory: 8192 MB
Boot Order: Hard Disk, Optical, Floppy
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX

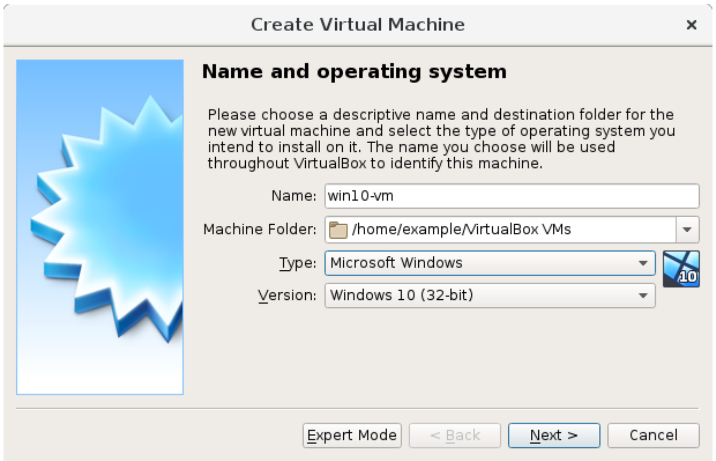
General
Name: OracleLinux7
Operating System: Oracle (64-bit)
Settings File Location: /Users/scr/VirtualBox/OracleLinux7

System
Base Memory: 8192 MB
Processors: 2
Boot Order: Hard Disk, Optical
Acceleration: VT-x/AMD-V, Nested Paging, PAE/NX, KVM Paravirtualization

1.8. Creating Your First Virtual Machine

Click **New** in the VirtualBox Manager window. A wizard is shown, to guide you through setting up a new virtual machine (VM).

Figure 1.4. Creating a New Virtual Machine: Name and Operating System



Name and operating system

Please choose a descriptive name and destination folder for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: win10-vm
Machine Folder: /home/example/VirtualBox VMs
Type: Microsoft Windows
Version: Windows 10 (32-bit)

On the following pages, the wizard will ask you for the bare minimum of information that is needed to create a VM, in particular:

1. The **Name** of the VM will later be shown in the machine list of the VirtualBox Manager window, and it will be used for the VM's files on disk. Even though any name can be used, bear in mind that if you create a few VMs, you will appreciate if you have given your VMs rather informative names."My VM" would thus be less useful than "Windows XP SP2 with OpenOffice", for example.

2. The **Machine Folder** is the location where VMs are stored on your computer. The default folder location is shown.

3. For **Operating System Type** select the OS that you want to install later. The supported OSes are grouped. If you want to install something very unusual that is not listed, select **Other**. Depending on your selection, Oracle VM VirtualBox will enable or disable certain VM settings that your guest OS may require. This is particularly important for 64-bit guests. See Section 3.1.2, “64-bit Guests”. It is therefore recommended to always set it to the correct value.

4. On the next page, select the **Memory (RAM)** that Oracle VM VirtualBox should allocate every time the virtual machine is started. The amount of memory given here will be taken away from your host machine and presented to the guest OS, which will report this size as the virtual computer's installed RAM.

   ### Caution

   Choose this setting carefully. The memory you give to the VM will not be available to your host OS while the VM is running, so do not specify more than you can spare. For example, if your host machine has 1 GB of RAM and you enter 512 MB as the amount of RAM for a particular virtual machine, while that VM is running, you will only have 512 MB left for all the other software on your host. If you run two VMs at the same time, even more memory will be allocated for the second VM, which may not even be able to start if that memory is not available. On the other hand, you should specify as much as your guest OS and your applications will require to run properly.

   A Windows XP guest will require at least a few hundred MB of RAM to run properly, and Windows Vista will not install with less than 512 MB. If you want to run graphics-intensive applications in your VM, you may require even more RAM.

   As a rule of thumb, if you have 1 GB of RAM or more in your host computer, it is usually safe to allocate 512 MB to each VM. In any case, make sure you always have at least 256 to 512 MB of RAM left on your host OS. Otherwise you may cause your host OS to excessively swap out memory to your hard disk, effectively bringing your host system to a standstill.

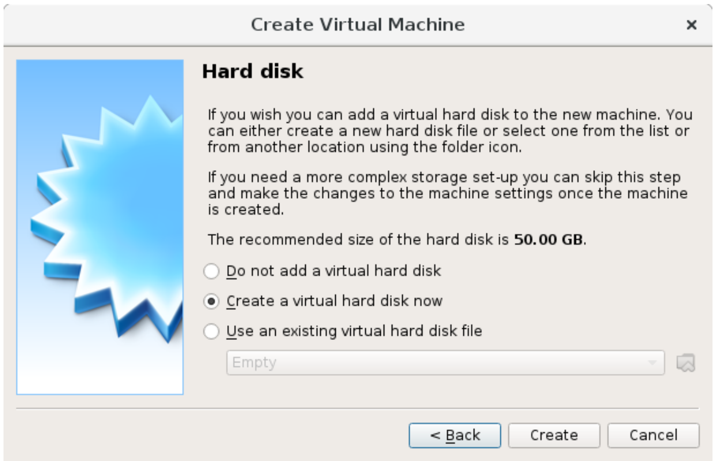
   As with the other settings, you can change this setting later, after you have created the VM.

5. Next, you must specify a **Virtual Hard Disk** for your VM.

   There are many and potentially complicated ways in which Oracle VM VirtualBox can provide hard disk space to a VM, see Chapter 5, *Virtual Storage*, but the most common way is to use a large image file on your "real" hard disk, whose contents Oracle VM VirtualBox presents to your VM as if it were a complete hard disk. This file represents an entire hard disk then, so you can even copy it to another host and use it with another Oracle VM VirtualBox installation.

   The wizard displays the following window:

   Figure 1.5. Creating a New Virtual Machine: Hard Disk



Create Virtual Machine

Hard disk

If you wish you can add a virtual hard disk to the new machine. You can either create a new hard disk file or select one from the list or from another location using the folder icon.

If you need a more complex storage set-up you can skip this step and make the changes to the machine settings once the machine is created.

The recommended size of the hard disk is **50.00 GB**.

○ Do not add a virtual hard disk

● Create a virtual hard disk now

○ Use an existing virtual hard disk file

Empty

At this screen, you have the following options:

- To create a new, empty virtual hard disk, click the **Create** button.
- You can pick an *existing* disk image file.

  The drop-down list presented in the window lists all disk images which are currently remembered by Oracle VM VirtualBox. These disk images are currently attached to a virtual machine, or have been attached to a virtual machine.

  Alternatively, click on the small **folder icon** next to the drop-down list. In the displayed file dialog, you can click **Add** to select any disk image file on your host disk.

If you are using Oracle VM VirtualBox for the first time, you will want to create a new disk image. Click the **Create** button.

This displays another window, the **Create Virtual Hard Disk Wizard** wizard. This wizard helps you to create a new disk image file in the new virtual machine's folder.

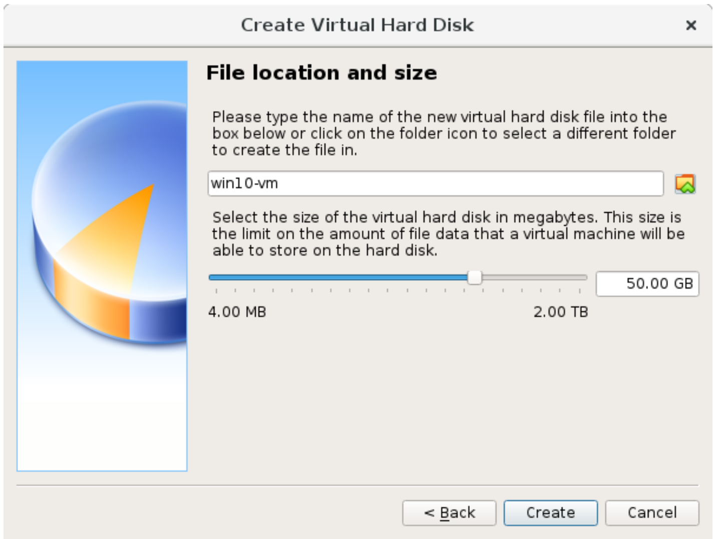
Oracle VM VirtualBox supports the following types of image files:

- A **dynamically allocated file** will only grow in size when the guest actually stores data on its virtual hard disk. It will therefore initially be small on the host hard drive and only later grow to the size specified as it is filled with data.
- A **fixed-size file** will immediately occupy the file specified, even if only a fraction of the virtual hard disk space is actually in use. While occupying much more space, a fixed-size file incurs less overhead and is therefore slightly faster than a dynamically allocated file.

For details about the differences, see Section 5.2, “Disk Image Files (VDI, VMDK, VHD, HDD)”.

To prevent your physical hard disk from running full, Oracle VM VirtualBox limits the size of the image file. Still, it needs to be large enough to hold the contents of your OS and the applications you want to install. For a modern Windows or Linux guest, you will probably need several gigabytes for any serious use. The limit of the image file size can be changed later, see Section 8.24, “VBoxManage modifymedium”.

Figure 1.6. Creating a New Virtual Machine: File Location and Size



Create Virtual Hard Disk

File location and size

Please type the name of the new virtual hard disk file into the box below or click on the folder icon to select a different folder to create the file in.

win10-vm

Select the size of the virtual hard disk in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard disk.

50.00 GB

After having selected or created your image file, click **Next** to go to the next page.

6. Click **Create**, to create your new virtual machine. The virtual machine is displayed in the list on the left side of the VirtualBox Manager window, with the name that you entered initially.

Note

After becoming familiar with the use of wizards, consider using the Expert Mode available in some wizards. Where available, this is selectable using a button, and speeds up the process of using wizards.

1.9. Running Your Virtual Machine

To start a virtual machine, you have several options:

- Double-click on the VM's entry in the list in the VirtualBox Manager window.
- Select the VM's entry in the list in the VirtualBox Manager window, and click **Start** at the top of the window.
- Go to the `VirtualBox VMs` folder in your system user's home directory. Find the subdirectory of the machine you want to start and double-click on the machine settings file. This file has a `.vbox` file extension.

Starting a virtual machine displays a new window, and the virtual machine which you selected will boot up. Everything which would normally be seen on the virtual system's monitor is shown in the window. See the screenshot image in Chapter 1, *First Steps*.

In general, you can use the virtual machine as you would use a real computer. There are couple of points worth mentioning however.

1.9.1. Starting a New VM for the First Time

When a VM is started for the first time, the **First Start Wizard**, is displayed. This wizard helps you to select an installation medium. Since the VM is created empty, it would otherwise behave just like a real computer with no OS installed. It will do nothing and display an error message that no bootable OS was found.

For this reason, the wizard helps you to select a medium to install an OS from.

- If you have physical CD or DVD media from which you want to install your guest OS, such as a Windows installation CD or DVD, put the media into your host's CD or DVD drive.

  In the wizard's drop-down list of installation media, select **Host Drive** with the correct drive letter. In the case of a Linux host, choose a device file. This will allow your VM to access the media in your host drive, and you can proceed to install from there.

- If you have downloaded installation media from the Internet in the form of an ISO image file such as with a Linux distribution, you would normally burn this file to an empty CD or DVD and proceed as described above. With Oracle VM VirtualBox however, you can skip this step and mount the ISO file directly. Oracle VM VirtualBox will then present this file as a CD or DVD-ROM drive to the virtual machine, much like it does with virtual hard disk images.

  In this case, the wizard's drop-down list contains a list of installation media that were previously used with Oracle VM VirtualBox.

  If your medium is not in the list, especially if you are using Oracle VM VirtualBox for the first time, click the small folder icon next to the drop-down list to display a standard file dialog. Here you can pick an image file on your host disks.

After completing the choices in the wizard, you will be able to install your OS.

1.9.2. Capturing and Releasing Keyboard and Mouse

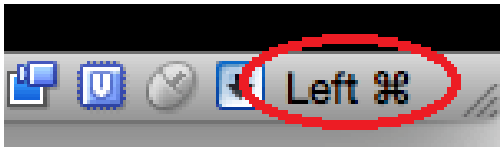
Oracle VM VirtualBox provides a virtual USB tablet device to new virtual machines through which mouse events are communicated to the guest OS. If you are running a modern guest OS that can handle such devices, mouse support may work out of the box without the mouse being *captured* as described below. See Section 3.5.1, “Motherboard Tab”.

Otherwise, if the virtual machine only sees standard PS/2 mouse and keyboard devices, since the OS in the virtual machine does not know that it is not running on a real computer, it expects to have exclusive control over your keyboard and mouse. But unless you are running the VM in full screen mode, your VM needs to share keyboard and mouse with other applications and possibly other VMs on your host.

After installing a guest OS and before you install the Guest Additions, described later, either your VM or the rest of your computer can "own" the keyboard and the mouse. Both cannot own the keyboard and mouse at the same time. You will see a *second* mouse pointer which is always confined to the limits of the VM window. You activate the VM by clicking inside it.

To return ownership of keyboard and mouse to your host OS, Oracle VM VirtualBox reserves a special key on your keyboard: the *Host key*. By default, this is the *right Ctrl key* on your keyboard. On a Mac host, the default Host key is the left Command key. You can change this default in the Oracle VM VirtualBox Global Settings. See Section 1.16, “Global Settings”. The current setting for the Host key is always displayed at the bottom right of your VM window.

Figure 1.7. Host Key Setting on the Virtual Machine Task Bar



This means the following:

- Your **keyboard** is owned by the VM if the VM window on your host desktop has the keyboard focus. If you have many windows open in your guest OS, the window that has the focus in your VM is used. This means that if you want to enter text within your VM, click on the title bar of your VM window first.

  To release keyboard ownership, press the Host key. As explained above, this is typically the right Ctrl key.

  Note that while the VM owns the keyboard, some key sequences, such as Alt-Tab, will no longer be seen by the host, but will go to the guest instead. After you press the Host key to reenable the host keyboard, all key presses will go through the host again, so that sequences such as Alt-Tab will no longer reach the guest. For technical reasons it may not be possible for the VM to get all keyboard input even when it does own the keyboard. Examples of this are the Ctrl-Alt-Del sequence on Windows hosts or single keys grabbed by other applications on X11 hosts like the GNOME desktop's "Control key highlights mouse pointer" functionality.

- Your **mouse** is owned by the VM only after you have clicked in the VM window. The host mouse pointer will disappear, and your mouse will drive the guest's pointer instead of your normal mouse pointer.

  Note that mouse ownership is independent of that of the keyboard. Even after you have clicked on a titlebar to be able to enter text into the VM window, your mouse is not necessarily owned by the VM yet.

  To release ownership of your mouse by the VM, press the Host key.

As this behavior can be inconvenient, Oracle VM VirtualBox provides a set of tools and device drivers for guest systems called the Oracle VM VirtualBox Guest Additions which make VM keyboard and mouse operation a lot more seamless. Most importantly, the Additions will get rid of the second "guest" mouse pointer and make your host mouse pointer work directly in the guest. See Chapter 4, *Guest Additions*.

1.9.3. Typing Special Characters

OSes expect certain key combinations to initiate certain procedures. Some of these key combinations may be difficult to enter into a virtual machine, as there are three candidates as to who receives keyboard input: the host OS, Oracle VM VirtualBox, or the guest OS. Which of these three receives keypresses depends on a number of factors, including the key itself.

- Host OSes reserve certain key combinations for themselves. For example, it is impossible to enter the **Ctrl+Alt+Delete** combination if you want to reboot the guest OS in your virtual machine, because this key combination is usually hard-wired into the host OS, both Windows and Linux intercept this, and pressing this key combination will therefore reboot your *host*.

On Linux and Oracle Solaris hosts, which use the X Window System, the key combination **Ctrl+Alt+Backspace** normally resets the X server and restarts the entire graphical user interface. As the X server intercepts this combination, pressing it will usually restart your *host* graphical user interface and kill all running programs, including Oracle VM VirtualBox, in the process.

On Linux hosts supporting virtual terminals, the key combination **Ctrl+Alt+Fx**, where Fx is one of the function keys from F1 to F12, normally enables you to switch between virtual terminals. As with Ctrl+Alt+Delete, these combinations are intercepted by the host OS and therefore always switch terminals on the *host*.

If, instead, you want to send these key combinations to the *guest* OS in the virtual machine, you will need to use one of the following methods:

  - Use the items in the **Input**, **Keyboard** menu of the virtual machine window. This menu includes the settings **Insert Ctrl+Alt+Delete** and **Ctrl+Alt+Backspace**. The latter will only have an effect with Linux or Oracle Solaris guests, however.

    This menu also includes an option for inserting the Host key combination.

  - Use special key combinations with the Host key, normally the right Control key. Oracle VM VirtualBox will then translate these key combinations for the virtual machine:

    - **Host key + Del** to send Ctrl+Alt+Del to reboot the guest.

    - **Host key + Backspace** to send Ctrl+Alt+Backspace to restart the graphical user interface of a Linux or Oracle Solaris guest.

    - **Host key + Function key**. For example, to simulate Ctrl+Alt+Fx to switch between virtual terminals in a Linux guest.

- For some other keyboard combinations such as **Alt-Tab** to switch between open windows, Oracle VM VirtualBox enables you to configure whether these combinations will affect the host or the guest, if a virtual machine currently has the focus. This is a global setting for all virtual machines and can be found under **File**, **Preferences**, **Input**.

1.9.4. Changing Removable Media

While a virtual machine is running, you can change removable media in the **Devices** menu of the VM's window. Here you can select in detail what Oracle VM VirtualBox presents to your VM as a CD, DVD, or floppy drive.

The settings are the same as those available for the VM in the **Settings** dialog of the Oracle VM VirtualBox main window. But as the **Settings** dialog is disabled while the VM is in the Running or Saved state, the **Devices** menu saves you from having to shut down and restart the VM every time you want to change media.

Using the **Devices** menu, you can attach the host drive to the guest or select a floppy or DVD image, as described in Section 3.7, “Storage Settings”.

The **Devices** menu also includes an option for creating a virtual ISO (VISO) from selected files on the host.

1.9.5. Resizing the Machine's Window

You can resize the virtual machine's window when it is running. In that case, one of the following things will happen:

1. If you have **scaled mode** enabled, then the virtual machine's screen will be scaled to the size of the window. This can be useful if you have many machines running and want to have a look at one of them while it is running in the background. Alternatively, it might be useful to enlarge a window if the VM's output screen is very small, for example because you are running an old OS in it.

   To enable scaled mode, press **Host key + C**, or select **Scaled Mode** from the **View** menu in the VM window. To leave scaled mode, press **Host key + C** again.

The aspect ratio of the guest screen is preserved when resizing the window. To ignore the aspect ratio, press **Shift** during the resize operation.

See Chapter 14, *Known Limitations* for additional remarks.

2. If you have the Guest Additions installed and they support automatic **resizing**, the Guest Additions will automatically adjust the screen resolution of the guest OS. For example, if you are running a Windows guest with a resolution of 1024x768 pixels and you then resize the VM window to make it 100 pixels wider, the Guest Additions will change the Windows display resolution to 1124x768.

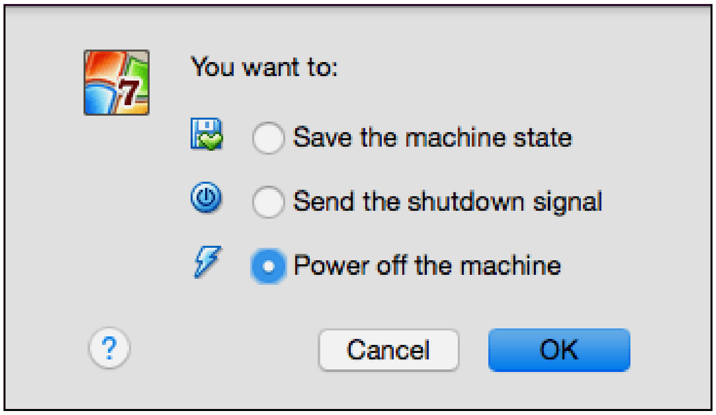
   See Chapter 4, *Guest Additions*.

3. Otherwise, if the window is bigger than the VM's screen, the screen will be centered. If it is smaller, then scroll bars will be added to the machine window.

1.9.6. Saving the State of the Machine

When you click on the **Close** button of your virtual machine window, at the top right of the window, just like you would close any other window on your system, Oracle VM VirtualBox asks you whether you want to save or power off the VM. As a shortcut, you can also press **Host key + Q**.

Figure 1.8. Closing Down a Virtual Machine



You want to:

○ Save the machine state

○ Send the shutdown signal

● Power off the machine

The difference between the three options is crucial. They mean the following:

- **Save the machine state:** With this option, Oracle VM VirtualBox *freezes* the virtual machine by completely saving its state to your local disk.

  When you start the VM again later, you will find that the VM continues exactly where it was left off. All your programs will still be open, and your computer resumes operation. Saving the state of a virtual machine is thus in some ways similar to suspending a laptop computer by closing its lid.

- **Send the shutdown signal.** This will send an ACPI shutdown signal to the virtual machine, which has the same effect as if you had pressed the power button on a real computer. So long as the VM is running a fairly modern OS, this should trigger a proper shutdown mechanism from within the VM.

- **Power off the machine:** With this option, Oracle VM VirtualBox also stops running the virtual machine, but *without* saving its state.

  > **Warning**
  >
  > This is equivalent to pulling the power plug on a real computer without shutting it down properly. If you start the machine again after powering it off, your OS will have to reboot completely and may begin a lengthy check of its virtual system disks. As a result, this should not normally be done, since it can potentially cause data loss or an inconsistent state of the guest system on disk.

  As an exception, if your virtual machine has any snapshots, see Section 1.11, "Snapshots", you can use this option to quickly **restore the current snapshot** of the virtual machine. In that case, powering off the machine will not disrupt its state, but any changes made since that snapshot was taken will be lost.

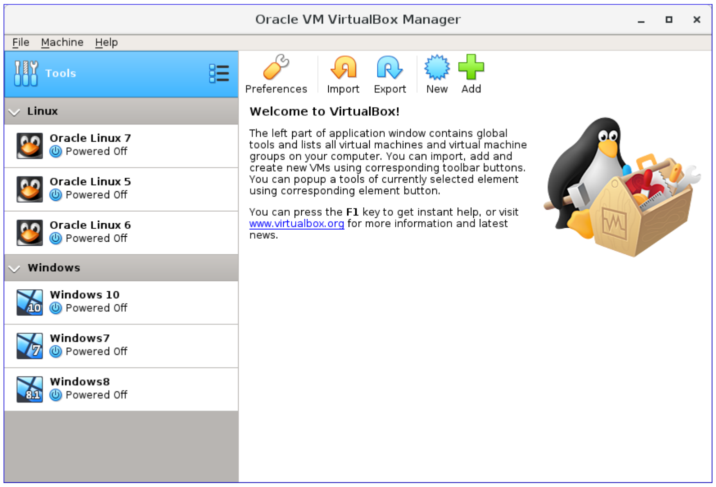
The **Discard** button in the VirtualBox Manager window discards a virtual machine's saved state. This has the same effect as powering it off, and the same warnings apply.

1.10. Using VM Groups

VM groups enable the user to create ad hoc groups of VMs, and to manage and perform functions on them collectively, as well as individually.

The following figure shows VM groups displayed in VirtualBox Manager.

Figure 1.9. Groups of Virtual Machines



Welcome to VirtualBox!

The left part of application window contains global tools and lists all virtual machines and virtual machine groups on your computer. You can import, add and create new VMs using corresponding toolbar buttons. You can popup a tools of currently selected element using corresponding element button.

You can press the **F1** key to get instant help, or visit www.virtualbox.org for more information and latest news.

The following features are available for groups:

- Create a group using the VirtualBox Manager. Do one of the following:
    - Drag one VM on top of another VM.
    - Select multiple VMs and select **Group** from the right-click menu.
- Create and manage a group using the command line. Do one of the following:
    - Create a group and assign a VM. For example:

      ```
      VBoxManage modifyvm "vm01" --groups "/TestGroup"
      ```

      This command creates a group "TestGroup" and attaches the VM "vm01" to that group.
    - Detach a VM from the group, and delete the group if empty. For example:

      ```
      VBoxManage modifyvm "vm01" --groups ""
      ```

      This command detaches all groups from the VM "vm01" and deletes the empty group.
- Create multiple groups. For example:

  ```
  VBoxManage modifyvm "vm01" --groups "/TestGroup,/TestGroup2"
  ```

  This command creates the groups "TestGroup" and "TestGroup2", if they do not exist, and attaches the VM "vm01" to both of them.
- Create nested groups, having a group hierarchy. For example:

  ```
  VBoxManage modifyvm "vm01" --groups "/TestGroup/TestGroup2"
  ```

  This command attaches the VM "vm01" to the subgroup "TestGroup2" of the "TestGroup" group.

- The following is a summary of group commands: Start, Pause, Reset, Close (save state, send shutdown signal, poweroff), Discard Saved State, Show in File System, Sort.

1.11. Snapshots

With snapshots, you can save a particular state of a virtual machine for later use. At any later time, you can revert to that state, even though you may have changed the VM considerably since then. A snapshot of a virtual machine is thus similar to a machine in Saved state, but there can be many of them, and these saved states are preserved.

To see the snapshots of a virtual machine, click on the machine name in VirtualBox Manager. Then click the **List** icon next to the machine name, and select **Snapshots**. Until you take a snapshot of the machine, the list of snapshots will be empty except for the **Current State** item, which represents the "now" point in the lifetime of the virtual machine.

1.11.1. Taking, Restoring, and Deleting Snapshots

There are three operations related to snapshots, as follows:

1. **Take a snapshot**. This makes a copy of the machine's current state, to which you can go back at any given time later.

    - If your VM is running, select **Take Snapshot** from the **Machine** pull-down menu of the VM window.

    - If your VM is in either the Saved or the Powered Off state, as displayed next to the VM name in the Oracle VM VirtualBox main window, click the **List** icon next to the machine name and select **Snapshots**. The snapshots window is shown. Do one of the following:

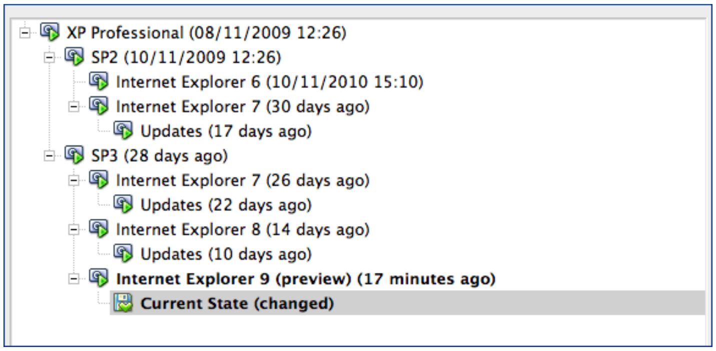
        - Click the **Take** icon.

        - Right-click on the **Current State** item in the list and select **Take**.

    In either case, a window is displayed prompting you for a snapshot name. This name is purely for reference purposes to help you remember the state of the snapshot. For example, a useful name would be "Fresh installation from scratch, no Guest Additions", or "Service Pack 3 just installed". You can also add a longer text in the **Description** field.

    Your new snapshot will then appear in the snapshots list. Underneath your new snapshot, you will see an item called **Current State**, signifying that the current state of your VM is a variation based on the snapshot you took earlier. If you later take another snapshot, you will see that they are displayed in sequence, and that each subsequent snapshot is derived from an earlier one.

    Figure 1.10. Snapshots List For a Virtual Machine

    ```
    XP Professional (08/11/2009 12:26)
      SP2 (10/11/2009 12:26)
        Internet Explorer 6 (10/11/2010 15:10)
        Internet Explorer 7 (30 days ago)
          Updates (17 days ago)
      SP3 (28 days ago)
        Internet Explorer 7 (26 days ago)
          Updates (22 days ago)
        Internet Explorer 8 (14 days ago)
          Updates (10 days ago)
        Internet Explorer 9 (preview) (17 minutes ago)
          Current State (changed)
    ```

    Oracle VM VirtualBox imposes no limits on the number of snapshots you can take. The only practical limitation is disk space on your host. Each snapshot stores the state of the virtual machine and thus occupies some disk space. See [Section 1.11.2, "Snapshot Contents"](#) for details on what is stored in a snapshot.

2. **Restore a snapshot**. In the list of snapshots, right-click on any snapshot you have taken and select **Restore**. By restoring a snapshot, you go back or forward in time. The current state of the machine is lost, and the machine is restored to the exact state it was in when the snapshot was taken.

   > **Note**
   >
   > Restoring a snapshot will affect the virtual hard drives that are connected to your VM, as the entire state of the virtual hard drive will be reverted as well. This means also that all files that have been created since the snapshot and all other file changes *will be lost.* In order to prevent such data loss while still making use of the snapshot feature, it is possible to add a second hard drive in *write-through* mode using the **VBoxManage** interface and use it to store your data. As write-through hard drives are *not* included in snapshots, they remain unaltered when a machine is reverted. See Section 5.4, “Special Image Write Modes”.

   To avoid losing the current state when restoring a snapshot, you can create a new snapshot before the restore operation.

   By restoring an earlier snapshot and taking more snapshots from there, it is even possible to create a kind of alternate reality and to switch between these different histories of the virtual machine. This can result in a whole tree of virtual machine snapshots, as shown in the screenshot above.

3. **Delete a snapshot**. This does not affect the state of the virtual machine, but only releases the files on disk that Oracle VM VirtualBox used to store the snapshot data, thus freeing disk space. To delete a snapshot, right-click on the snapshot name in the snapshots tree and select **Delete**. Snapshots can be deleted even while a machine is running.

   > **Note**
   >
   > Whereas taking and restoring snapshots are fairly quick operations, deleting a snapshot can take a considerable amount of time since large amounts of data may need to be copied between several disk image files. Temporary disk files may also need large amounts of disk space while the operation is in progress.

   There are some situations which cannot be handled while a VM is running, and you will get an appropriate message that you need to perform this snapshot deletion when the VM is shut down.

1.11.2. Snapshot Contents

Think of a snapshot as a point in time that you have preserved. More formally, a snapshot consists of the following:

- The snapshot contains a complete copy of the VM settings, including the hardware configuration, so that when you restore a snapshot, the VM settings are restored as well. For example, if you changed the hard disk configuration or the VM's system settings, that change is undone when you restore the snapshot.

  The copy of the settings is stored in the machine configuration, an XML text file, and thus occupies very little space.

- The complete state of all the virtual disks attached to the machine is preserved. Going back to a snapshot means that all changes that had been made to the machine's disks, file by file and bit by bit, will be undone as well. Files that were since created will disappear, files that were deleted will be restored, changes to files will be reverted.

  Strictly speaking, this is only true for virtual hard disks in "normal" mode. You can configure disks to behave differently with snapshots, see Section 5.4, “Special Image Write Modes”. In technical terms, it is not the virtual disk itself that is restored when a snapshot is restored. Instead, when a snapshot is taken, Oracle VM VirtualBox creates differencing images which contain only the changes since the snapshot were taken. When the snapshot is restored, Oracle VM VirtualBox throws away that differencing image, thus going back to the previous state. This is both faster and uses less disk space. For the details, which can be complex, see Section 5.5, “Differencing Images”.

Creating the differencing image as such does not occupy much space on the host disk initially, since the differencing image will initially be empty and grow dynamically later with each write operation to the disk. The longer you use the machine after having created the snapshot, however, the more the differencing image will grow in size.

- If you took a snapshot while the machine was running, the memory state of the machine is also saved in the snapshot. This is in the same way that memory can be saved when you close a VM window. When you restore such a snapshot, execution resumes at exactly the point when the snapshot was taken.

  The memory state file can be as large as the memory size of the virtual machine and will therefore occupy quite some disk space as well.

1.12. Virtual Machine Configuration

When you select a virtual machine from the list in the VirtualBox Manager window, you will see a summary of that machine's settings on the right.

Clicking on **Settings** displays a window, where you can configure many of the properties of the selected VM. But be careful when changing VM settings. It is possible to change all VM settings after installing a guest OS, but certain changes might prevent a guest OS from functioning correctly if done after installation.

> **Note**
>
> The **Settings** button is disabled while a VM is either in the Running or Saved state. This is because the **Settings** dialog enables you to change fundamental characteristics of the virtual machine that is created for your guest OS. For example, the guest OS may not perform well if half of its memory is taken away. As a result, if the **Settings** button is disabled, shut down the current VM first.

Oracle VM VirtualBox provides a wide range of parameters that can be changed for a virtual machine. The various settings that can be changed in the **Settings** window are described in detail in Chapter 3, *Configuring Virtual Machines*. Even more parameters are available when using the **VBoxManage** command line interface. See Chapter 8, *VBoxManage*.

1.13. Removing and Moving Virtual Machines

You can remove a VM from Oracle VM VirtualBox or move the VM and its associated files, such as disk images, to another location on the host.

- **Removing a VM.** To remove a VM, right-click on the VM in the VirtualBox Manager's machine list and select **Remove**.

  The confirmation dialog enables you to specify whether to only remove the VM from the list of machines or to remove the files associated with the VM.

  Note that the **Remove** menu item is disabled while a VM is running.

- **Moving a VM.** To move a VM to a new location on the host, right-click on the VM in the VirtualBox Manager's machine list and select **Move**.

  The file dialog prompts you to specify a new location for the VM.

  When you move a VM, Oracle VM VirtualBox configuration files are updated automatically to use the new location on the host.

  Note that the **Move** menu item is disabled while a VM is running.

  You can also use the **VBoxManage movevm** command to move a VM. See Section 8.10, “VBoxManage movevm”.

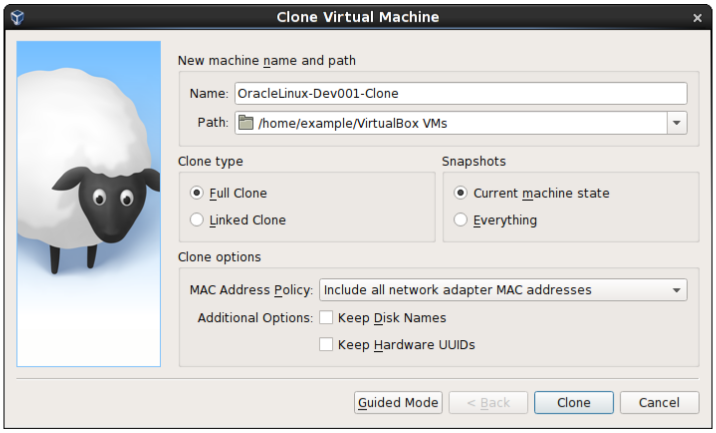
For information about removing or moving a disk image file from Oracle VM VirtualBox, see Section 5.3, “The Virtual Media Manager”.

1.14. Cloning Virtual Machines

You can create a full copy or a linked copy of an existing VM. This copy is called a *clone*. You might use a cloned VM to experiment with a VM configuration, to test different guest OS levels, or to back up a VM.

The **Clone Virtual Machine** wizard guides you through the cloning process.

Figure 1.11. The Clone Virtual Machine Wizard



Clone Virtual Machine

New machine name and path

Name: OracleLinux-Dev001-Clone

Path: /home/example/VirtualBox VMs

Clone type

● Full Clone

○ Linked Clone

Snapshots

● Current machine state

○ Everything

Clone options

MAC Address Policy: Include all network adapter MAC addresses

Additional Options: □ Keep Disk Names

□ Keep Hardware UUIDs

Start the wizard by clicking **Clone** in the right-click menu of the VirtualBox Manager's machine list or in the **Snapshots** view of the selected VM.

Specify a new **Name** for the clone. You can choose a **Path** for the cloned virtual machine, otherwise Oracle VM VirtualBox uses the default machines folder.

The **Clone Type** option specifies whether to create a clone linked to the source VM or to create a fully independent clone:

- **Full Clone:** Copies all dependent disk images to the new VM folder. A full clone can operate fully without the source VM.

- **Linked Clone:** Creates new differencing disk images based on the source VM disk images. If you select the current state of the source VM as the clone point, Oracle VM VirtualBox creates a new snapshot.

The **Snapshots** option specifies whether to create a clone of the current machine state only or of everything.

- **Everything:** Clones the current machine state and all its snapshots.

- **Current Machine State and All Children:**. Clones a VM snapshot and all its child snapshots.

The following clone options are available:

- **MAC Address Policy:** Specifies how to retain network card MAC addresses when cloning the VM.

  For example, the **Generate New MAC Addresses For All Network Adapters** value assigns a new MAC address to each network card during cloning. This is the default setting. This is the best option when both the source VM and the cloned VM must operate on the same network. Other values enable you to retain the existing MAC addresses in the cloned VM.

- **Keep Disk Names:** Retains the disk image names when cloning the VM.

- **Keep Hardware UUIDs:** Retains the hardware universally unique identifiers (UUIDs) when cloning the VM.

The duration of the clone operation depends on the size and number of attached disk images. In addition, the clone operation saves all the differencing disk images of a snapshot.

Note that the **Clone** menu item is disabled while a machine is running.

You can also use the **VBoxManage clonevm** command to clone a VM. See Section 8.9, "VBoxManage clonevm".

1.15. Importing and Exporting Virtual Machines

Oracle VM VirtualBox can import and export virtual machines in the following formats:

- **Open Virtualization Format (OVF).** This is the industry-standard format. See Section 1.15.1, "About the OVF Format".
- **Cloud service formats.** Export to cloud services such as Oracle Cloud Infrastructure is supported. Import is not supported. See Section 1.15.4, "Exporting an Appliance to Oracle Cloud Infrastructure".

1.15.1. About the OVF Format

OVF is a cross-platform standard supported by many virtualization products which enables the creation of ready-made virtual machines that can then be imported into a hypervisor such as Oracle VM VirtualBox. Oracle VM VirtualBox makes OVF import and export easy to do, using the VirtualBox Manager window or the command-line interface.

Using OVF enables packaging of *virtual appliances*. These are disk images, together with configuration settings that can be distributed easily. This way one can offer complete ready-to-use software packages, including OSes with applications, that need no configuration or installation except for importing into Oracle VM VirtualBox.

> **Note**
>
> The OVF standard is complex, and support in Oracle VM VirtualBox is an ongoing process. In particular, no guarantee is made that Oracle VM VirtualBox supports all appliances created by other virtualization software. For a list of known limitations, see Chapter 14, *Known Limitations*.

Appliances in OVF format can appear in the following variants:

- They can come in several files, as one or several disk images, typically in the widely-used VMDK format. See Section 5.2, "Disk Image Files (VDI, VMDK, VHD, HDD)". They also include a textual description file in an XML dialect with an `.ovf` extension. These files must then reside in the same directory for Oracle VM VirtualBox to be able to import them.
- Alternatively, the above files can be packed together into a single archive file, typically with an `.ova` extension. Such archive files use a variant of the TAR archive format and can therefore be unpacked outside of Oracle VM VirtualBox with any utility that can unpack standard TAR files.

> **Note**
>
> OVF cannot describe snapshots that were taken for a virtual machine. As a result, when you export a virtual machine that has snapshots, only the current state of the machine will be exported. The disk images in the export will have a *flattened* state identical to the current state of the virtual machine.

1.15.2. Importing an Appliance in OVF Format

The following steps show how to import an appliance in OVF format.

1. Double-click on the OVF or OVA file.

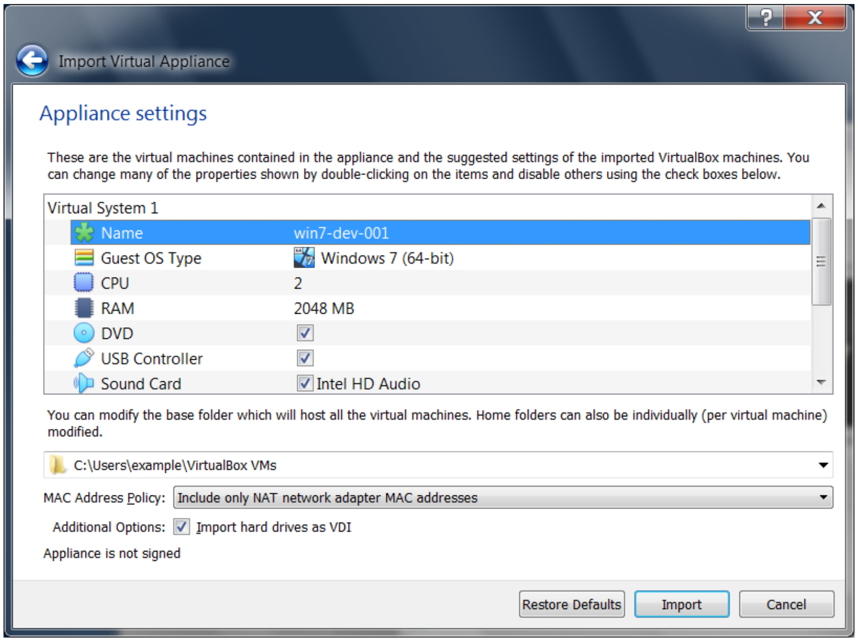
   Oracle VM VirtualBox creates file type associations automatically for any OVF and OVA files on your host OS.

2. Select **File**, **Import Appliance** from the VirtualBox Manager window.

   From the file dialog, go to the file with either the `.ovf` or the `.ova` file extension.

Click **Import** to open the **Appliance Settings** screen.

Figure 1.12. Appliance Settings Screen for Import Appliance



Appliance settings

These are the virtual machines contained in the appliance and the suggested settings of the imported VirtualBox machines. You can change many of the properties shown by double-clicking on the items and disable others using the check boxes below.

Virtual System 1

Name	win7-dev-001
Guest OS Type	Windows 7 (64-bit)
CPU	2
RAM	2048 MB
DVD	☑
USB Controller	☑
Sound Card	☑ Intel HD Audio

You can modify the base folder which will host all the virtual machines. Home folders can also be individually (per virtual machine) modified.

C:\Users\example\VirtualBox VMs

MAC Address Policy: Include only NAT network adapter MAC addresses

Additional Options: ☑ Import hard drives as VDI

Appliance is not signed

This screen shows the VMs described in the OVF or OVA file and enables you to change the VM settings.

By default, membership of VM groups is preserved on import for VMs that were initially exported from Oracle VM VirtualBox. You can change this behavior by using the **Primary Group** setting for the VM.

The following global settings apply to all of the VMs that you import:

- **Base Folder:** Specifies the directory on the host in which to store the imported VMs.

  If an appliance has multiple VMs, you can specify a different directory for each VM by editing the **Base Folder** setting for the VM.

- **MAC Address Policy:** Reinitializes the MAC addresses of network cards in your VMs prior to import, by default. You can override the default behavior and preserve the MAC addresses on import.

- **Import Hard Drives as VDI:** Imports hard drives in the VDI format rather that in the defalut VMDK format.

3. Click **Import** to import the appliance.

   Oracle VM VirtualBox copies the disk images and creates local VMs with the settings described on the **Appliance Settings** screen. The imported VMs are shown in the list of VMs in VirtualBox Manager.

   Because disk images are large, the VMDK images that are included with virtual appliances are shipped in a compressed format that cannot be used directly by VMs. So, the images are first unpacked and copied, which might take several minutes.

You can use the **VBoxManage import** command to import an appliance. See Section 8.11, “VBoxManage import”.

1.15.3. Exporting an Appliance in OVF Format

The following steps show how to export an appliance in OVF format.

1. Select **File**, **Export Appliance** to open the **Export Virtual Appliance** wizard.

   From the initial window, you can combine several VMs into an OVF appliance.

Select one or more VMs to export, and click **Next**.

2. The **Appliance Settings** screen enables you to select the following settings:

    - **Format:** Selects the **Open Virtualization Format** value for the output files.

      The **Oracle Cloud Infrastructure** value exports export to Oracle Cloud Infrastructure. See Section 1.15.4, "Exporting an Appliance to Oracle Cloud Infrastructure".

    - **File:** Selects the location in which to store the exported files.

    - **MAC Address Policy:** Specifies whether to retain or reassign network card MAC addresses on export.

    - **Write Manifest File:** Enables you to include a manifest file in the exported archive file.

    - **Include ISO Image Files:** Enables you to include ISO image files in the exported archive file.

3. Click **Next** to show the **Virtual System Settings** screen.

   You can edit settings for the virtual appliance. For example, you can change the name of the virtual appliance or add product information, such as vendor details or license text.

   Double-click the appropriate field to change its value.

4. Click **Export** to begin the export process. Note that this operation might take several minutes.

You can use the **VBoxManage export** command to export an appliance. See Section 8.12, "VBoxManage export".

1.15.4. Exporting an Appliance to Oracle Cloud Infrastructure

Oracle VM VirtualBox supports the export of VMs to an Oracle Cloud Infrastructure service.

Before you can export a VM to Oracle Cloud Infrastructure, ensure that you perform the following configuration steps:

- Generate an API signing key pair that is used for API requests to Oracle Cloud Infrastructure.

    - The key pair is usually installed in the `.oci` folder in your home directory. For example, `~/.oci` on a Linux system.

    - Upload the public key of the key pair to the cloud service.

  For step-by-step instructions for creating and uploading an API signing key for Oracle Cloud Infrastructure, see:

  https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm#How

- Create a profile for your cloud account.

  The cloud profile contains resource identifiers for your cloud account, such as your user OCID, and the fingerprint for your public key. You can create a cloud profile in the following ways:

    - Automatically by using the **Cloud Profile Manager**. See Section 1.15.5, "The Cloud Profile Manager".

    - Manually by creating an `oci_config` file in your Oracle VM VirtualBox global configuration directory. For example, this is `$HOME/.config/VirtualBox/oci_config` on a Linux host.

    - Manually by creating a `config` file in your Oracle Cloud Infrastructure configuration directory. For example, this is `$HOME/.oci/config` on a Linux host.

      This is the same file that is used by the Oracle Cloud Infrastructure command line interface.

Oracle VM VirtualBox automatically uses the `config` file if no cloud profile file is present in your global configuration directory. Alternatively, you can import this file manually into the Cloud Profile Manager.

For more information about the cloud profile settings used by Oracle Cloud Infrastructure see:

https://docs.cloud.oracle.com/iaas/Content/API/Concepts/sdkconfig.htm

- Ensure that the subnets that are used by source VMs are available in the target compartment on the cloud service.

Perform the following steps to export a VM to Oracle Cloud Infrastructure:

1. Select **File**, **Export Appliance** to open the **Export Virtual Appliance** wizard.

   Select a VM to export and click **Next** to open the **Appliance Settings** screen.

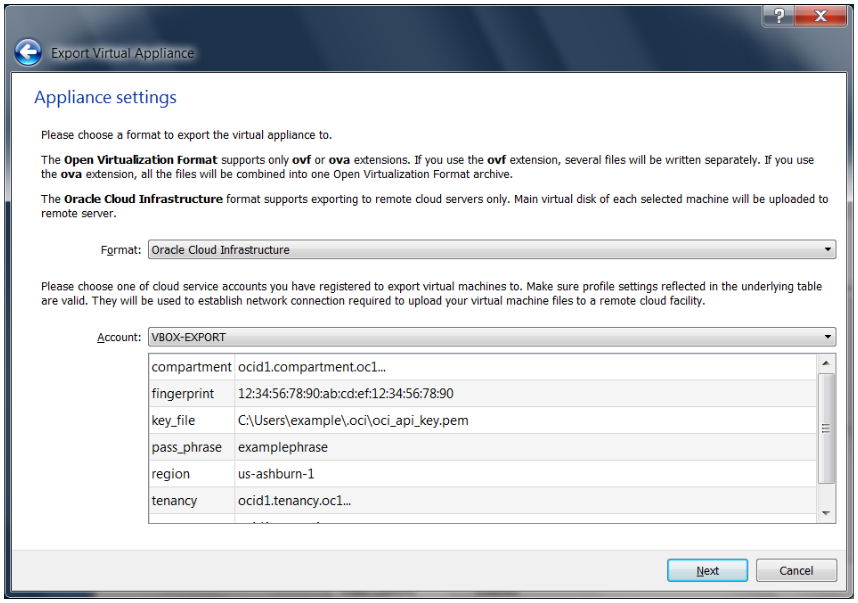
2. From the **Format** drop-down list, select **Oracle Cloud Infrastructure**.

   In the **Account** drop-down list, select your Oracle Cloud Infrastructure account.

   You can set up Oracle Cloud Infrastructure accounts by using the Cloud Profile Manager.

   The list after the **Account** field shows the profile settings for your cloud account.

   Figure 1.13. Appliance Settings Screen, Showing Cloud Profile Settings

   Export Virtual Appliance

   Appliance settings

   Please choose a format to export the virtual appliance to.

   The **Open Virtualization Format** supports only **ovf** or **ova** extensions. If you use the **ovf** extension, several files will be written separately. If you use the **ova** extension, all the files will be combined into one Open Virtualization Format archive.

   The **Oracle Cloud Infrastructure** format supports exporting to remote cloud servers only. Main virtual disk of each selected machine will be uploaded to remote server.

   Format: Oracle Cloud Infrastructure

   Please choose one of cloud service accounts you have registered to export virtual machines to. Make sure profile settings reflected in the underlying table are valid. They will be used to establish network connection required to upload your virtual machine files to a remote cloud facility.

   Account: VBOX-EXPORT

compartment	ocid1.compartment.oc1...
fingerprint	12:34:56:78:90:ab:cd:ef:12:34:56:78:90
key_file	C:\Users\example\.oci\oci_api_key.pem
pass_phrase	examplephrase
region	us-ashburn-1
tenancy	ocid1.tenancy.oc1...

   Click **Next** to make an API request to the Oracle Cloud Infrastructure service and open the **Virtual System Settings** screen.

3. Optionally edit settings used for the virtual machine on Oracle Cloud Infrastructure.

   For example, you can edit the Disk Size and Shape used for the VM instance.

   Click **Export** to export the VMs to the cloud service.

   The VMs are uploaded to Oracle Cloud Infrastructure.

   Instances are created for the uploaded VMs.

   By default, the VM instance is started after upload to Oracle Cloud Infrastructure.

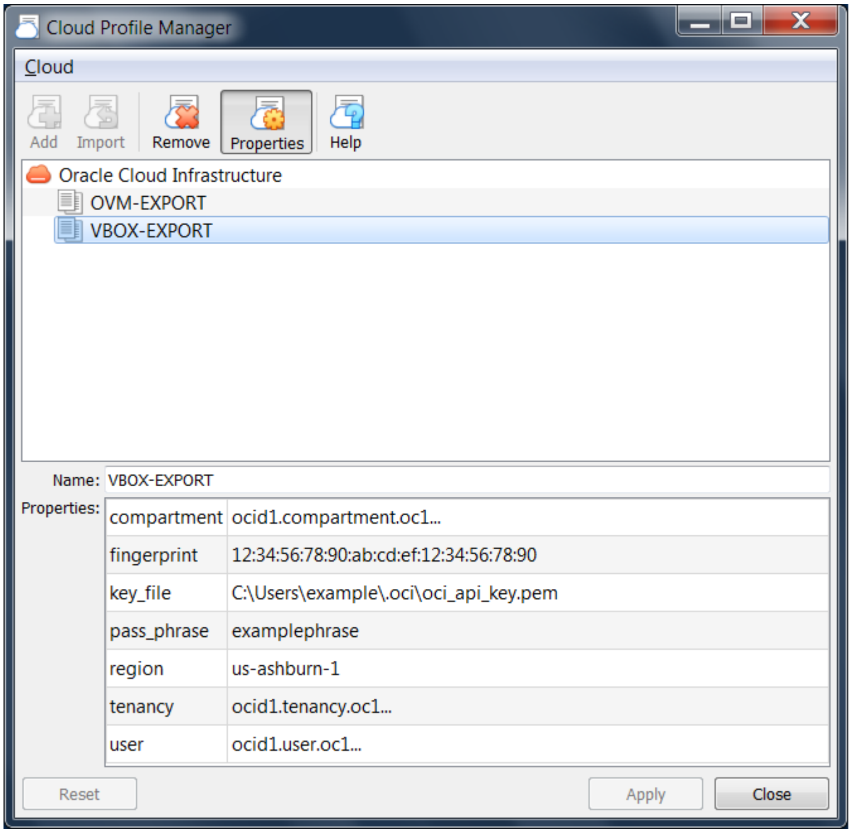
4. Monitor the export process by using the Oracle Cloud Infrastructure Console.

You can also use the **VBoxManage export** command to export a VM to Oracle Cloud Infrastructure. See Section 8.12.2, “Export to Oracle Cloud Infrastructure”.

1.15.5. The Cloud Profile Manager

The Cloud Profile Manager is a component of Oracle VM VirtualBox that enables you to create, edit, and manage cloud profiles for your cloud service accounts.

Figure 1.14. The Cloud Profile Manager



Name: VBOX-EXPORT

Properties:

compartment	ocid1.compartment.oc1...
fingerprint	12:34:56:78:90:ab:cd:ef:12:34:56:78:90
key_file	C:\Users\example\.oci\oci_api_key.pem
pass_phrase	examplephrase
region	us-ashburn-1
tenancy	ocid1.tenancy.oc1...
user	ocid1.user.oc1...

To open the Cloud Profile Manager select **File**, **Cloud Profile Manager** from the VirtualBox Manager window.

Use the Cloud Profile Manager to create a new cloud profile automatically. Or, create a cloud profile by importing settings from your Oracle Cloud Infrastructure configuration file into the Cloud Profile Manager.

Perform the following steps to create a new cloud profile:

1. Click the **Add** icon and specify a **Name** for the profile.

2. Click **Properties** and specify the following property values for the profile:

    - Compartment OCID
    - Fingerprint of the public key
    - Location of the private key on the client device
    - (Optional) Passphrase for the private key, if the key is encrypted
    - Region OCID
    - Tenancy OCID
    - User OCID

    Some of these are settings for your Oracle Cloud Infrastructure account, which you can view from the Oracle Cloud Infrastructure Console.

3. Click **Apply** to save your changes.

   The cloud profile settings are saved in the `oci_config` file in your Oracle VM VirtualBox global settings directory.

Perform the following steps to import an existing Oracle Cloud Infrastructure configuration file:

1. Ensure that a `config` file is present in your Oracle Cloud Infrastructure configuration directory. For example, this is `$HOME/.oci/config` on a Linux host.

2. Click the **Import** icon to open a dialog that prompts you to import cloud profiles from external files.

   > **Warning**
   >
   > This action overwrites any cloud profiles that are in your Oracle VM VirtualBox global settings directory.

3. Click **Import**.

   Your cloud profile settings are saved to the `oci_config` file in your Oracle VM VirtualBox global settings directory.

4. Click **Properties** to show the cloud profile settings.

   Double-click on the appropriate field to change the value.

5. Click **Apply** to save your changes.

1.16. Global Settings

The **Global Settings** dialog can be displayed using the **File** menu, by clicking the **Preferences** item. This dialog offers a selection of settings, most of which apply to all virtual machines of the current user. The **Extensions** option applies to the entire system.

The following settings are available:

- **General.** Enables the user to specify the default folder/directory for VM files, and the VRDP Authentication Library.

- **Input.** Enables the user to specify the Host key. It identifies the key that toggles whether the cursor is in the focus of the VM or the Host OS windows, see Section 1.9.2, “Capturing and Releasing Keyboard and Mouse”, and which is also used to trigger certain VM actions, see Section 1.9.3, “Typing Special Characters”.

- **Update.** Enables the user to specify various settings for Automatic Updates.

- **Language.** Enables the user to specify the GUI language.

- **Display.** Enables the user to specify the screen resolution, and its width and height. A default scale factor can be specified for all guest screens.

- **Network.** Enables the user to configure the details of Host Only Networks.

- **Extensions.** Enables the user to list and manage the installed extension packages.

- **Proxy.** Enables the user to configure a HTTP Proxy Server.

1.17. Alternative Front-Ends

As briefly mentioned in Section 1.3, “Features Overview”, Oracle VM VirtualBox has a very flexible internal design that enables you to use multiple interfaces to control the same virtual machines. For example, you can start a virtual machine with the VirtualBox Manager window and then stop it from the command line. With

Oracle VM VirtualBox's support for the Remote Desktop Protocol (RDP), you can even run virtual machines remotely on a headless server and have all the graphical output redirected over the network.

The following front-ends are shipped in the standard Oracle VM VirtualBox package:

- **VirtualBox.** This is the VirtualBox Manager, a graphical user interface that uses the Qt toolkit. This interface is described throughout this manual. While this is the simplest and easiest front-end to use, some of the more advanced Oracle VM VirtualBox features are not included.

- **VBoxManage.** A command-line interface for automated and detailed control of every aspect of Oracle VM VirtualBox. See Chapter 8, *VBoxManage*.

- **VBoxHeadless.** A front-end that produces no visible output on the host at all, but can act as a RDP server if the VirtualBox Remote Desktop Extension (VRDE) is installed and enabled for the VM. As opposed to the other graphical interfaces, the headless front-end requires no graphics support. This is useful, for example, if you want to host your virtual machines on a headless Linux server that has no X Window system installed. See Section 7.1.2, “VBoxHeadless, the Remote Desktop Server”.

If the above front-ends still do not satisfy your particular needs, it is possible to create yet another front-end to the complex virtualization engine that is the core of Oracle VM VirtualBox, as the Oracle VM VirtualBox core neatly exposes all of its features in a clean API. See Chapter 11, *Oracle VM VirtualBox Programming Interfaces*.