

## Chapter 3. Configuring Virtual Machines

### Table of Contents

#### [3.1. Supported Guest Operating Systems](#)

##### [3.1.1. Mac OS X Guests](#)

##### [3.1.2. 64-bit Guests](#)

#### [3.2. Unattended Guest Installation](#)

##### [3.2.1. Using VBoxManage Commands for Unattended Guest Installation](#)

#### [3.3. Emulated Hardware](#)

#### [3.4. General Settings](#)

##### [3.4.1. Basic Tab](#)

##### [3.4.2. Advanced Tab](#)

##### [3.4.3. Description Tab](#)

##### [3.4.4. Disk Encryption Tab](#)

#### [3.5. System Settings](#)

##### [3.5.1. Motherboard Tab](#)

##### [3.5.2. Processor Tab](#)

##### [3.5.3. Acceleration Tab](#)

#### [3.6. Display Settings](#)

##### [3.6.1. Screen Tab](#)

##### [3.6.2. Remote Display Tab](#)

##### [3.6.3. Recording Tab](#)

#### [3.7. Storage Settings](#)

#### [3.8. Audio Settings](#)

#### [3.9. Network Settings](#)

#### [3.10. Serial Ports](#)

#### [3.11. USB Support](#)

##### [3.11.1. USB Settings](#)

##### [3.11.2. Implementation Notes for Windows and Linux Hosts](#)

#### [3.12. Shared Folders](#)

#### [3.13. User Interface](#)

#### [3.14. Alternative Firmware \(EFI\)](#)

##### [3.14.1. Video Modes in EFI](#)

##### [3.14.2. Specifying Boot Arguments](#)

This chapter provides detailed steps for configuring an Oracle VM VirtualBox virtual machine (VM). For an introduction to Oracle VM VirtualBox and steps to get your first virtual machine running, see [Chapter 1, First Steps](#).

You have considerable latitude when deciding what virtual hardware to provide to the guest. Use virtual hardware to communicate with the host system or with other guests. For example, you can use virtual hardware in the following ways:

- Have Oracle VM VirtualBox present an ISO CD-ROM image to a guest system as if it were a physical CD-ROM.
- Provide a guest system access to the physical network through its virtual network card.
- Provide the host system, other guests, and computers on the Internet access to the guest system.

## 3.1. Supported Guest Operating Systems

Because Oracle VM VirtualBox is designed to provide a generic virtualization environment for x86 systems, it can run guest operating systems (OSes) of any kind.

The following guest OS platforms are supported:

- **Platforms With Full Support.** These guest OS platforms qualify for Oracle Premier Support. See [Table 3.1, “Guest Operating Systems With Full Support”](#).
- **Platforms With Limited Support.** These legacy guest OS platforms can be used with Oracle VM VirtualBox, but only qualify for *best effort* support. Therefore, resolution of customer issues is not guaranteed. See [Table 3.2, “Legacy Guest Operating Systems With Limited Support”](#).

**Table 3.1. Guest Operating Systems With Full Support**

Operating System	Comments
Windows 11 (64-bit)	Insider preview builds are not supported
Windows 10 (32-bit and 64-bit)	Insider preview builds are not supported
Windows 8 and 8.1 (32-bit and 64-bit)	
Windows Server 2019 (64-bit)	
Windows Server 2016 (64-bit)	
Windows Server 2012 and 2012 R2 (64-bit)	
Solaris 11 (32-bit and 64-bit)	
Solaris 10 8/11 Update 10 and later (32-bit and 64-bit)	
Oracle Linux 8 (64-bit)	Includes Red Hat Enterprise Linux 8, CentOS 8
Oracle Linux 7 (64-bit)	Includes Red Hat Enterprise Linux 7, CentOS 7
Oracle Linux 6 (32-bit and 64-bit)	Includes Red Hat Enterprise Linux 6, CentOS 6
Ubuntu 16.04 LTS (Xenial Xerus) (32-bit and 64-bit)	
Ubuntu 18.04 LTS (Bionic Beaver) (64-bit)	
Ubuntu 20.04 LTS (Focal Fossa) (64-bit)	
SUSE Linux Enterprise Server 15 (64-bit)	
SUSE Linux Enterprise Server 12 (64-bit)	

**Table 3.2. Legacy Guest Operating Systems With Limited Support**

Operating System	Comments
Windows 7 (32-bit and 64-bit)	
Windows Vista SP2 and later (32-bit and 64-bit)	

Operating System	Comments
Windows XP (32-bit)	
Windows Vista (32-bit)	
Windows Server 2008 and 2008 R2 (32-bit and 64-bit)	
Windows Server 2003 (32-bit and 64-bit)	
Oracle Linux 5 (32-bit and 64-bit)	Includes Red Hat Enterprise Linux 5, CentOS 5
Ubuntu 14.04.5 LTS (Trusty Tahr) (32-bit and 64-bit)	
OS/2 Warp 4.5	

### 3.1.1. Mac OS X Guests

Oracle VM VirtualBox enables you to install and execute unmodified versions of Mac OS X guests on supported host hardware. Note that this feature is experimental and thus unsupported.

Oracle VM VirtualBox is the first product to provide the modern PC architecture expected by OS X without requiring any of the modifications used by competing virtualization solutions. For example, some competing solutions perform modifications to the Mac OS X install DVDs, such as a different boot loader and replaced files.

Be aware of the following important issues before you attempt to install a Mac OS X guest:

- Mac OS X is commercial, licensed software and contains **both license and technical restrictions** that limit its use to certain hardware and usage scenarios. You must understand and comply with these restrictions.

In particular, Apple prohibits the installation of most versions of Mac OS X on non-Apple hardware.

These license restrictions are also enforced on a technical level. Mac OS X verifies that it is running on Apple hardware. Most DVDs that accompany Apple hardware check for the exact model. These restrictions are *not* circumvented by Oracle VM VirtualBox and continue to apply.

- Only **CPUs** that are known and tested by Apple are supported. As a result, if your Intel CPU is newer than the Mac OS X build, or if you have a non-Intel CPU, you will likely encounter a panic during bootup with an "Unsupported CPU" exception.

Ensure that you use the Mac OS X DVD that comes with your Apple hardware.

- The Mac OS X installer expects the hard disk to be *partitioned*. So, the installer will not offer a partition selection to you. Before you can install the software successfully, start the Disk Utility from the Tools menu and partition the hard disk. Close the Disk Utility and proceed with the installation.
- In addition, Mac OS X support in Oracle VM VirtualBox is an experimental feature. See [Chapter 14, Known Limitations](#).

### 3.1.2. 64-bit Guests

#### Warning

Be sure to enable **I/O APIC** for virtual machines that you intend to use in 64-bit mode. This is

especially true for 64-bit Windows VMs. See [Section 3.5.1, “Motherboard Tab”](#). For 64-bit Windows guests, ensure that the VM uses the **Intel networking device** because there is no 64-bit driver support for the AMD PCNet card. See [Section 6.1, “Virtual Networking Hardware”](#).

If you use the **Create VM** wizard of VirtualBox Manager, Oracle VM VirtualBox automatically uses the correct settings for each selected 64-bit OS type. See [Section 1.8, “Creating Your First Virtual Machine”](#).

## 3.2. Unattended Guest Installation

Oracle VM VirtualBox can install a guest OS automatically. You only need to provide the installation medium and a few other parameters, such as the name of the default user.

You can perform an unattended guest installation in the following ways:

- **Use the Create Virtual Machine wizard.** An optional step in the wizard enables you to configure unattended installation. You can specify the default user credentials for the guest OS and also whether to install the Guest Additions automatically. See [Section 1.8, “Creating Your First Virtual Machine”](#).

During this step, Oracle VM VirtualBox scans the installation medium and changes certain parameters to ensure a seamless installation as a guest running on Oracle VM VirtualBox.

- **Use the VBoxManage commands.** [Section 3.2.1, “Using VBoxManage Commands for Unattended Guest Installation”](#) describes how to perform an unattended guest installation for an Oracle Linux guest.

When you first start a VM that has been configured for unattended installation, the guest OS installation is performed automatically.

The installation operation changes the boot device order to boot the virtual hard disk first and then the virtual DVD drive. If the virtual hard disk is empty prior to the automatic installation, the VM boots from the virtual DVD drive and begins the installation.

If the virtual hard disk contains a bootable OS, the installation operation exits. In this case, change the boot device order manually by pressing F12 during the BIOS splash screen.

### 3.2.1. Using VBoxManage Commands for Unattended Guest Installation

The following example shows how to perform an unattended guest installation for an Oracle Linux VM. The example uses various **VBoxManage** commands to prepare the guest VM. The **VBoxManage unattended install** command is then used to install and configure the guest OS.

#### 1. Create the virtual machine.

```
# VM="ol7-autoinstall"
# VBoxManage list ostypes
# VBoxManage createvm --name $VM --ostype "Oracle_64" --register
```

Note the following:

- The `$VM` variable represents the name of the VM.
- The **VBoxManage list ostypes** command lists the guest OSes supported by Oracle VM VirtualBox, including the name used for each OS in the **VBoxManage** commands.
- A 64-bit Oracle Linux 7 VM is created and registered with Oracle VM VirtualBox.
- The VM has a unique UUID.

- An XML settings file is generated.

## 2. Create a virtual hard disk and storage devices for the VM.

```
# VBoxManage createhd --filename /VirtualBox/$VM/$VM.vdi --size 32768
# VBoxManage storagectl $VM --name "SATA Controller" --add sata --controller IntelAHCI
# VBoxManage storageattach $VM --storagectl "SATA Controller" --port 0 --device 0 \
--type hdd --medium /VirtualBox/$VM/$VM.vdi
# VBoxManage storagectl $VM --name "IDE Controller" --add ide
# VBoxManage storageattach $VM --storagectl "IDE Controller" --port 0 --device 0 \
--type dvddrive --medium /u01/Software/OL/OracleLinux-R7-U6-Server-x86_64-dvd.iso
```

The previous commands do the following:

- Create a 32768 MB virtual hard disk.
- Create a SATA storage controller and attach the virtual hard disk.
- Create an IDE storage controller for a virtual DVD drive and attach an Oracle Linux installation ISO.

## 3. (Optional) Configure some settings for the VM.

```
# VBoxManage modifyvm $VM --ioapic on
# VBoxManage modifyvm $VM --boot1 dvd --boot2 disk --boot3 none --boot4 none
# VBoxManage modifyvm $VM --memory 8192 --vram 128
```

The previous commands do the following:

- Enable I/O APIC for the motherboard of the VM.
- Configure the boot device order for the VM.
- Allocate 8192 MB of RAM and 128 MB of video RAM to the VM.

## 4. Perform an unattended install of the OS.

```
# VBoxManage unattended install $VM \
--iso=/u01/Software/OL/OracleLinux-R7-U6-Server-x86_64-dvd.iso \
--user=login --full-user-name=name --password password \
--install-additions --time-zone=CET
```

The previous command does the following:

- Specifies an Oracle Linux ISO as the installation ISO.
- Specifies a login name, full name, and login password for a default user on the guest OS.  
Note that the specified password is also used for the root user account on the guest.
- Installs the Guest Additions on the VM.
- Sets the time zone for the guest OS to Central European Time (CET).

## 5. Start the virtual machine.

This step completes the unattended installation process.

```
# VBoxManage startvm $VM --type headless
```

The VM starts in headless mode, which means that the VirtualBox Manager window does not open.

## 6. (Optional) Update the guest OS to use the latest Oracle Linux packages.

On the guest VM, run the following command:

```
# yum update
```

### 3.3. Emulated Hardware

Oracle VM VirtualBox virtualizes nearly all hardware of the host. Depending on a VM's configuration, the guest will see the following virtual hardware:

- **Input devices.** Oracle VM VirtualBox can emulate a standard PS/2 keyboard and mouse. These devices are supported by most guest OSes.

In addition, Oracle VM VirtualBox can provide virtual USB input devices to avoid having to capture mouse and keyboard, as described in [Section 1.9.2, “Capturing and Releasing Keyboard and Mouse”](#).

- **Graphics.** The default Oracle VM VirtualBox graphics device for Windows guests is an SVGA device. For Linux guests, the default graphics device emulates a VMware SVGA graphics device. See [Section 3.6.1, “Screen Tab”](#).

For legacy guest OSes, a VGA-compatible graphics device is available.

- **Storage.** Oracle VM VirtualBox emulates the most common types of hard disk controllers. See [Section 5.1, “Hard Disk Controllers”](#). Whereas supporting only one of these controllers would be enough for Oracle VM VirtualBox by itself, this multitude of storage adapters is required for compatibility with other hypervisors. Windows is very selective about its boot devices, and migrating VMs between hypervisors is very difficult or impossible if the storage controllers are different.

- **Networking.** See [Section 6.1, “Virtual Networking Hardware”](#).

- **USB.** Oracle VM VirtualBox emulates these types of USB host controllers: xHCI, EHCI, and OHCI. While xHCI handles all USB transfer speeds, some legacy guest OSes may not support xHCI. Note that for some legacy Windows guests, third party drivers must be installed for xHCI support.

Legacy guest OSes typically support OHCI and EHCI. These two controllers are needed because OHCI only handles USB low-speed and full-speed devices (both USB 1.x and 2.0), while EHCI only handles high-speed devices (USB 2.0 only).

The emulated USB controllers do not communicate directly with devices on the host. Instead they communicate with a virtual USB layer which abstracts the USB protocol and enables the use of remote USB devices.

- **Audio.** See [Section 3.8, “Audio Settings”](#).

### 3.4. General Settings

In the **Settings** window, under **General**, you can configure the most fundamental aspects of the virtual machine such as memory and essential hardware. The following tabs are available.

#### 3.4.1. Basic Tab

In the **Basic** tab of the **General** settings category, you can find these settings:

- **Name:** The name of the the VM, as shown in the list of VMs in the main VirtualBox Manager window. Using this name, Oracle VM VirtualBox also saves the VM's configuration files. If you change the name, Oracle VM VirtualBox renames these files as well. As a result, you can only use characters which are allowed for file names on your host OS.

Note that internally, Oracle VM VirtualBox uses unique identifiers (UUIDs) to identify virtual machines. You can display these using the **VBoxManage** commands.

- **Type:** The type of the guest OS for the VM. This is the same setting that is specified in the **New Virtual Machine** wizard. See [Section 1.8, “Creating Your First Virtual Machine”](#).

Whereas the default settings of a newly created VM depend on the selected OS type, changing the type later has no effect on VM settings. This value is purely informational and decorative.

- **Version:** The version of the guest OS for the VM. This is the same setting that is specified in the **New Virtual Machine** wizard. See [Section 1.8, “Creating Your First Virtual Machine”](#).

### 3.4.2. Advanced Tab

The following settings are available in the **Advanced** tab:

- **Snapshot Folder:** By default, Oracle VM VirtualBox saves snapshot data together with your other Oracle VM VirtualBox configuration data. See [Section 10.1, “Where Oracle VM VirtualBox Stores its Files”](#). With this setting, you can specify any other folder for each VM.
- **Shared Clipboard:** You can select here whether the clipboard of the guest OS should be shared with that of your host. If you select **Bidirectional**, then Oracle VM VirtualBox will always make sure that both clipboards contain the same data. If you select **Host to Guest** or **Guest to Host**, then Oracle VM VirtualBox will only ever copy clipboard data in one direction.

Clipboard sharing requires that the Oracle VM VirtualBox Guest Additions be installed. In such a case, this setting has no effect. See [Chapter 4, Guest Additions](#).

For security reasons, the shared clipboard is disabled by default. This setting can be changed at any time using the **Shared Clipboard** menu item in the **Devices** menu of the virtual machine.

- **Drag and Drop:** This setting enables support for drag and drop. Select an object, such as a file, from the host or guest and directly copy or open it on the guest or host. Multiple drag and drop modes for a VM enable restricting of access in either direction.

For drag and drop to work the Guest Additions need to be installed on the guest.

#### Note

Drag and drop is disabled by default. This setting can be changed at any time using the **Drag and Drop** menu item in the **Devices** menu of the virtual machine.

See [Section 4.4, “Drag and Drop”](#).

### 3.4.3. Description Tab

On the **Description** tab you can enter a description for your virtual machine. This has no effect on the functionality of the machine, but you may find this space useful to note down things such as the configuration of a virtual machine and the software that has been installed into it.

To insert a line break into the **Description** text field, press Shift+Enter.

### 3.4.4. Disk Encryption Tab

The **Disk Encryption** tab enables you to encrypt disks that are attached to the virtual machine.

To enable disk encryption, select the **Enable Disk Encryption** check box.

Settings are available to configure the cipher used for encryption and the encryption password.

**Note**

All files related to the virtual machine except disk images are stored unencrypted. To encrypt these files, use the **VBoxManage encryptvm** command as described in [Section 9.22, “Encryption of VMs”](#).

## 3.5. System Settings

The **System** category groups various settings that are related to the basic hardware that is presented to the virtual machine.

**Note**

As the activation mechanism of Microsoft Windows is sensitive to hardware changes, if you are changing hardware settings for a Windows guest, some of these changes may trigger a request for another activation with Microsoft.

The following tabs are available.

### 3.5.1. Motherboard Tab

On the **Motherboard** tab, you can configure virtual hardware that would normally be on the motherboard of a real computer.

- **Base Memory:** Sets the amount of RAM that is allocated and given to the VM when it is running. The specified amount of memory will be requested from the host OS, so it must be available or made available as free memory on the host when attempting to start the VM and will not be available to the host while the VM is running. This is the same setting that was specified in the **New Virtual Machine** wizard, as described in [Section 1.8, “Creating Your First Virtual Machine”](#).

Generally, it is possible to change the memory size after installing the guest OS. But you must not reduce the memory to an amount where the OS would no longer boot.

- **Boot Order:** Determines the order in which the guest OS will attempt to boot from the various virtual boot devices. Analogous to a real PC's BIOS setting, Oracle VM VirtualBox can tell a guest OS to start from the virtual floppy, the virtual CD/DVD drive, the virtual hard drive (each of these as defined by the other VM settings), the network, or none of these.

If you select **Network**, the VM will attempt to boot from a network using the PXE mechanism. This needs to be configured in detail on the command line. See [Section 8.10, “VBoxManage modifyvm”](#).

- **Chipset:** You can select which chipset will be presented to the virtual machine. PIIX3 is the default chipset for most guests. For some guest OSes such as Mac OS X, the PIIX3 chipset is not well supported. As a result, Oracle VM VirtualBox supports an emulation of the ICH9 chipset, which supports PCI express, three PCI buses, PCI-to-PCI bridges and Message Signaled Interrupts (MSI). This enables modern OSes to address more PCI devices and no longer requires IRQ sharing. Using the ICH9 chipset it is also possible to configure up to 36 network cards, compared to a maximum of eight network adapters with PIIX3. Note that ICH9 support is experimental and not recommended for guest OSes which do not require it.
- **TPM:** Enables support for a Trusted Platform Module (TPM) security processor. Choose from the supported TPM versions.
- **Pointing Device:** The default virtual pointing device for some guest OSes is the traditional PS/2 mouse. If set to **USB Tablet**, Oracle VM VirtualBox reports to the virtual machine that a USB tablet device is present and communicates mouse events to the virtual machine through this device. Another setting is **USB Multi-**



**Touch Tablet**, which is suitable for guests running Windows 8 or later.

Using the virtual USB tablet has the advantage that movements are reported in absolute coordinates, instead of as relative position changes. This enables Oracle VM VirtualBox to translate mouse events over the VM window into tablet events without having to "capture" the mouse in the guest as described in [Section 1.9.2, "Capturing and Releasing Keyboard and Mouse"](#). This makes using the VM less tedious even if Guest Additions are not installed.

- **Enable I/O APIC:** Advanced Programmable Interrupt Controllers (APICs) are an x86 hardware feature that have replaced Programmable Interrupt Controllers (PICs). With an I/O APIC, OSeS can use more than 16 interrupt requests (IRQs) and therefore avoid IRQ sharing for improved reliability.

#### Note

Enabling the I/O APIC is *required*, especially for 64-bit Windows guest OSeS. It is also required if you want to use more than one virtual CPU in a virtual machine.

However, software support for I/O APICs has been unreliable with some OSeS other than Windows. Also, the use of an I/O APIC slightly increases the overhead of virtualization and therefore slows down the guest OS a little.

#### Warning

All Windows OSeS install different kernels, depending on whether an I/O APIC is available. As with ACPI, the I/O APIC therefore *must not be turned off after installation* of a Windows guest OS. Turning it on after installation will have no effect however.

- **Hardware Clock in UTC Time:** If selected, Oracle VM VirtualBox will report the system time in UTC format to the guest instead of the local (host) time. This affects how the virtual real-time clock (RTC) operates and may be useful for UNIX-like guest OSeS, which typically expect the hardware clock to be set to UTC.
- **Enable EFI:** Enables Extensible Firmware Interface (EFI), which replaces the legacy BIOS and may be useful for certain advanced use cases. See [Section 3.14, "Alternative Firmware \(EFI\)"](#).
- **Enable Secure Boot:** Enables Secure Boot, to provide a secure environment for starting the guest OS.

In addition, you can turn off the **Advanced Configuration and Power Interface (ACPI)** which Oracle VM VirtualBox presents to the guest OS by default.

ACPI is the current industry standard to allow OSeS to recognize hardware, configure motherboards and other devices and manage power. As most computers contain this feature and Windows and Linux support ACPI, it is also enabled by default in Oracle VM VirtualBox. ACPI can only be turned off using the command line. See [Section 8.10, "VBoxManage modifyvm"](#).

#### Warning

All Windows OSeS install different kernels, depending on whether ACPI is available. This means that ACPI *must not be turned off* after installation of a Windows guest OS. However, turning it on after installation will have no effect.

### 3.5.2. Processor Tab

On the **Processor** tab, you can configure settings for the CPU used by the virtual machine.

- **Processor(s):** Sets the number of virtual CPU cores the guest OSeS can see. Oracle VM VirtualBox supports symmetrical multiprocessing (SMP) and can present up to 32 virtual CPU cores to each virtual machine.

You should not configure virtual machines to use more CPU cores than are available physically. This includes real cores, with no hyperthreads.

- **Execution Cap:** Configures the CPU execution cap. This limits the amount of time a host CPU spends to emulate a virtual CPU. The default setting is 100%, meaning that there is no limitation. A setting of 50% implies a single virtual CPU can use up to 50% of a single host CPU. Note that limiting the execution time of the virtual CPUs may cause guest timing problems.

A warning is displayed at the bottom of the Processor tab if an Execution Cap setting is made that may affect system performance.

- **Enable PAE/NX:** Determines whether the PAE and NX capabilities of the host CPU will be exposed to the virtual machine.

PAE stands for Physical Address Extension. Normally, if enabled and supported by the OS, then even a 32-bit x86 CPU can access more than 4 GB of RAM. This is made possible by adding another 4 bits to memory addresses, so that with 36 bits, up to 64 GB can be addressed. Some OSes, such as Ubuntu Server, require PAE support from the CPU and cannot be run in a virtual machine without it.

- **Enable Nested VT-x/AMD-V:** Enables nested virtualization, with passthrough of hardware virtualization functions to the guest VM.

With virtual machines running modern server OSes, Oracle VM VirtualBox also supports CPU hot-plugging. For details, see [Section 9.4, “CPU Hot-Plugging”](#).

### 3.5.3. Acceleration Tab

On this tab, you can configure Oracle VM VirtualBox to use hardware virtualization extensions that your host CPU supports.

- **Paravirtualization Interface:** Oracle VM VirtualBox provides paravirtualization interfaces to improve time-keeping accuracy and performance of guest OSes. The options available are documented under the `--paravirt-provider` option in [Section 8.10, “VBoxManage modifyvm”](#). For further details on the paravirtualization providers, see [Section 10.5, “Paravirtualization Providers”](#).
- **Hardware Virtualization:** You can configure hardware virtualization features for each virtual machine.
  - **Enable Nested Paging:** If the host CPU supports the nested paging (AMD-V) or EPT (Intel VT-x) features, then you can expect a significant performance increase by enabling nested paging in addition to hardware virtualization. For technical details, see [Section 10.6, “Nested Paging and VPIDs”](#). For Intel EPT security recommendations, see [Section 13.4.1, “CVE-2018-3646”](#).

Advanced users may be interested in technical details about hardware virtualization. See [Section 10.3, “Hardware Virtualization”](#).

In most cases, the default settings on the **Acceleration** tab will work well. Oracle VM VirtualBox selects sensible defaults, depending on the OS that you selected when you created the virtual machine. In certain situations, however, you may want to change the preconfigured defaults.

## 3.6. Display Settings

The following tabs are available for configuring the display for a virtual machine.

### 3.6.1. Screen Tab

- **Video Memory:** Sets the size of the memory provided by the virtual graphics card available to the guest, in

MB. As with the main memory, the specified amount will be allocated from the host's resident memory. Based on the amount of video memory, higher resolutions and color depths may be available.

VirtualBox Manager will show a warning if the amount of video memory is too small to be able to switch the VM into full screen mode. The minimum value depends on the number of virtual monitors, the screen resolution and the color depth of the host display as well as on the use of *3D acceleration* and *2D video acceleration*. A rough estimate is  $(color\ depth / 8) \times vertical\ pixels \times horizontal\ pixels \times number\ of\ screens = number\ of\ bytes$ . Extra memory may be required if display acceleration is used.

- **Monitor Count:** With this setting, Oracle VM VirtualBox can provide more than one virtual monitor to a virtual machine. If a guest OS supports multiple attached monitors, Oracle VM VirtualBox can pretend that multiple virtual monitors are present. Up to eight such virtual monitors are supported.

The output of the multiple monitors are displayed on the host in multiple VM windows which are running side by side. However, in full screen and seamless mode, they use the available physical monitors attached to the host. As a result, for full screen and seamless modes to work with multiple monitors, you will need at least as many physical monitors as you have virtual monitors configured, or Oracle VM VirtualBox will report an error.

You can configure the relationship between guest and host monitors using the **View** menu by pressing Host key + Home when you are in full screen or seamless mode.

See also [Chapter 14, Known Limitations](#).

- **Scale Factor:** Enables scaling of the display size. For multiple monitor displays, you can set the scale factor for individual monitors, or globally for all of the monitors. Use the slider to select a scaling factor up to 200%.

You can set a default scale factor for all VMs. Use the **Display** tab in the Preferences window.

- **Graphics Controller:** Specifies the graphics adapter type used by the guest VM. Note that you must install the Guest Additions on the guest VM to specify the VBoxSVGA or VMSVGA graphics controller. The following options are available:
  - **VBoxSVGA:** The default graphics controller for new VMs that use Windows 7 or later.  
  
This graphics controller improves performance and 3D support when compared to the legacy VBoxVGA option.
  - **VBoxVGA:** Use this graphics controller for legacy guest OSes. This is the default graphics controller for Windows versions before Windows 7 and for Oracle Solaris.  
  
3D acceleration is not supported for this graphics controller.
  - **VMSVGA:** Use this graphics controller to emulate a VMware SVGA graphics device. This is the default graphics controller for Linux guests.
  - **None:** Does not emulate a graphics adapter type.
- **Enable 3D Acceleration:** If a virtual machine has Guest Additions installed, you can select here whether the guest should support accelerated 3D graphics. See [Section 4.5.1, "Hardware 3D Acceleration \(OpenGL and Direct3D 8/9\)"](#).

### 3.6.2. Remote Display Tab

On the **Remote Display** tab, if the VirtualBox Remote Display Extension (VRDE) is installed, you can enable the VRDP server that is built into Oracle VM VirtualBox. This enables you to connect to the console of the virtual

machine remotely with any standard RDP viewer, such as **mstsc.exe** that comes with Microsoft Windows. On Linux and Oracle Solaris systems you can use the standard open source **rdesktop** program. These features are described in [Section 7.1, “Remote Display \(VRDP Support\)”](#).

- **Enable Server:** Select this check box and configure settings for the remote display connection.

### 3.6.3. Recording Tab

On the **Recording** tab you can enable video and audio recording for a virtual machine and change related settings. Note that these features can be enabled and disabled while a VM is running.

- **Enable Recording:** Select this check box and select a **Recording Mode** option.
- **Recording Mode:** You can choose to record video, audio, or both video and audio.

Some settings on the **Recording** tab may be grayed out, depending on the **Recording Mode** setting.

- **File Path:** The file where the recording is saved.
- **Frame Size:** The video resolution of the recorded video, in pixels. The drop-down list enables you to select from common frame sizes.
- **Frame Rate:** Use the slider to set the maximum number of video frames per second (FPS) to record. Frames that have a higher frequency are skipped. Increasing this value reduces the number of skipped frames and increases the file size.
- **Video Quality:** Use the slider to set the the bit rate of the video in kilobits per second. Increasing this value improves the appearance of the video at the cost of an increased file size.
- **Audio Quality:** Use the slider to set the quality of the audio recording. Increasing this value improves the audio quality at the cost of an increased file size.
- **Screens:** For a multiple monitor display, you can select which screens to record video from.

As you adjust the video and audio recording settings, the approximate output file size for a five minute video is shown.

## 3.7. Storage Settings

The **Storage** category in the VM settings enables you to connect virtual hard disk, CD/DVD, and floppy images and drives to your virtual machine.

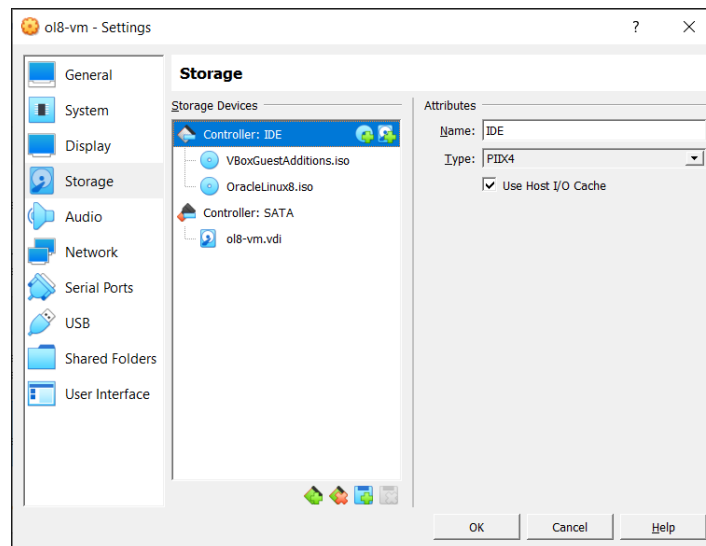
In a real computer, so-called *storage controllers* connect physical disk drives to the rest of the computer. Similarly, Oracle VM VirtualBox presents virtual storage controllers to a virtual machine. Under each controller, the virtual devices, such as hard disks, CD/DVD or floppy drives, attached to the controller are shown.

### Note

This section gives a quick introduction to the Oracle VM VirtualBox storage settings. See [Chapter 5, \*Virtual Storage\*](#) for a full description of the available storage settings in Oracle VM VirtualBox.

If you have used the **Create Virtual Machine** wizard to create a machine, you will normally see something like the following:

**Figure 3.1. Storage Settings for a Virtual Machine**



Depending on the guest OS type that you selected when you created the VM, a new VM includes the following storage devices:

- **IDE controller.** A virtual CD/DVD drive is attached to device 0 on the secondary channel of the IDE controller.
- **SATA controller.** This is a modern type of storage controller for higher hard disk data throughput, to which the virtual hard disks are attached. Initially you will normally have one such virtual disk, but as shown in the previous screenshot, you can have more than one. Each is represented by a disk image file, such as a VDI file in this example.

If you created your VM with an older version of Oracle VM VirtualBox, the default storage layout may differ. You might then only have an IDE controller to which both the CD/DVD drive and the hard disks have been attached. This might also apply if you selected an older OS type when you created the VM. Since older OSes do not support SATA without additional drivers, Oracle VM VirtualBox will make sure that no such devices are present initially. See [Section 5.1, “Hard Disk Controllers”](#).

Oracle VM VirtualBox also provides a *floppy controller*. You cannot add devices other than floppy drives to this controller. Virtual floppy drives, like virtual CD/DVD drives, can be connected to either a host floppy drive, if you have one, or a disk image, which in this case must be in RAW format.

You can modify these media attachments freely. For example, if you wish to copy some files from another virtual disk that you created, you can connect that disk as a second hard disk, as in the above screenshot. You could also add a second virtual CD/DVD drive, or change where these items are attached. The following options are available:

- To **add another virtual hard disk, or a CD/DVD or floppy drive**, select the storage controller to which it should be added (such as IDE, SATA, SCSI, SAS, floppy controller) and then click the **Add Disk** button below the tree. You can then either select **Optical Drive** or **Hard Disk**. If you clicked on a floppy controller, you can add a floppy drive instead. Alternatively, right-click on the storage controller and select a menu item there.

A dialog is displayed, enabling you to select an existing disk image file or to create a new disk image file. Depending on the type of disk image, the dialog is called **Hard Disk Selector**, **Optical Disk Selector**, or **Floppy Disk Selector**.

See [Section 5.2, “Disk Image Files \(VDI, VMDK, VHD, HDD\)”](#) for information on the image file types that are supported by Oracle VM VirtualBox.

For virtual CD/DVD drives, the image files will typically be in the standard ISO format instead. Most

commonly, you will select this option when installing an OS from an ISO file that you have obtained from the Internet. For example, most Linux distributions are available in this way.

Depending on the type of disk image, you can set the following **Attributes** for the disk image in the right part of the Storage settings page:

- The **device slot** of the controller that the virtual disk is connected to. IDE controllers have four slots: primary device 0, primary device 1, secondary device 0, and secondary device 1. By contrast, SATA and SCSI controllers offer you up to 30 slots for attaching virtual devices.
  - **Solid-state Drive** presents a virtual disk to the guest as a solid-state device.
  - **Hot-pluggable** presents a virtual disk to the guest as a hot-pluggable device.
  - For virtual CD/DVD drives, you can select **Live CD/DVD**. This means that the virtual optical disk is not removed from when the guest system ejects it.
- To **remove an attachment**, either select it and click on the **Remove** icon at the bottom, or right-click on it and select the menu item.

Removable media, such as CD/DVDs and floppies, can be changed while the guest is running. Since the **Settings** window is not available at that time, you can also access these settings from the **Devices** menu of your virtual machine window.

## 3.8. Audio Settings

The **Audio** section in a virtual machine's **Settings** window determines whether the VM will detect a connected sound card, and if the audio output should be played on the host system.

To enable audio for a guest, select the **Enable Audio** check box. The following settings are available:

- **Host Audio Driver:** The audio driver that Oracle VM VirtualBox uses on the host.  
  
The **Default** option is enabled by default for all new VMs. This option selects the best audio driver for the host platform automatically. This enables you to move VMs between different platforms without having to change the audio driver.  
  
On a Linux host, depending on your host configuration, you can select between the OSS, ALSA, or the PulseAudio subsystem. On newer Linux distributions, the PulseAudio subsystem is preferred.  
  
Only OSS is supported on Oracle Solaris hosts. The Oracle Solaris Audio audio backend is no longer supported on Oracle Solaris hosts.
- **Audio Controller:** You can choose between the emulation of an Intel AC'97 controller, an Intel HD Audio controller, or a SoundBlaster 16 card.
- **Enable Audio Output:** Enables audio output only for the VM.
- **Enable Audio Input:** Enables audio input only for the VM.

## 3.9. Network Settings

The **Network** section in a virtual machine's **Settings** window enables you to configure how Oracle VM VirtualBox presents virtual network cards to your VM, and how they operate.

When you first create a virtual machine, Oracle VM VirtualBox by default enables one virtual network card and selects the Network Address Translation (NAT) mode for it. This way the guest can connect to the outside world

using the host's networking and the outside world can connect to services on the guest which you choose to make visible outside of the virtual machine.

This default setup is good for the majority of Oracle VM VirtualBox users. However, Oracle VM VirtualBox is extremely flexible in how it can virtualize networking. It supports many virtual network cards per virtual machine. The first four virtual network cards can be configured in detail in VirtualBox Manager. Additional network cards can be configured using the **VBoxManage** command.

Many networking options are available. See [Chapter 6, Virtual Networking](#) for more information.

## 3.10. Serial Ports

Oracle VM VirtualBox supports the use of virtual serial ports in a virtual machine.

Ever since the original IBM PC, personal computers have been equipped with one or two serial ports, also called COM ports by DOS and Windows. Serial ports were commonly used with modems, and some computer mice used to be connected to serial ports before USB became commonplace.

While serial ports are no longer as common as they used to be, there are still some important uses left for them. For example, serial ports can be used to set up a primitive network over a null-modem cable, in case Ethernet is not available. Also, serial ports are indispensable for system programmers needing to do kernel debugging, since kernel debugging software usually interacts with developers over a serial port. With virtual serial ports, system programmers can do kernel debugging on a virtual machine instead of needing a real computer to connect to.

If a virtual serial port is enabled, the guest OS sees a standard 16550A compatible UART device. Other UART types can be configured using the **VBoxManage modifyvm** command. Both receiving and transmitting data is supported. How this virtual serial port is then connected to the host is configurable, and the details depend on your host OS.

You can use either the Settings tabs or the **VBoxManage** command to set up virtual serial ports. For the latter, see [Section 8.10, "VBoxManage modifyvm"](#) for information on the `--uart`, `--uart-mode` and `--uart-type` options.

You can configure up to four virtual serial ports per virtual machine. For each device, you must set the following:

1. **Port Number:** This determines the serial port that the virtual machine should see. For best results, use the traditional values as follows:

- COM1: I/O base 0x3F8, IRQ 4
- COM2: I/O base 0x2F8, IRQ 3
- COM3: I/O base 0x3E8, IRQ 4
- COM4: I/O base 0x2E8, IRQ 3

You can also configure a user-defined serial port. Enter an I/O base address and interrupt (IRQ).

2. **Port Mode:** What the virtual port is connected to. For each virtual serial port, you have the following options:

- **Disconnected:** The guest will see the device, but it will behave as if no cable had been connected to it.
- **Host Device:** Connects the virtual serial port to a physical serial port on your host. On a Windows host, this will be a name like COM1. On Linux or Oracle Solaris hosts, it will be a device node like `/dev/ttyS0`. Oracle VM VirtualBox will then simply redirect all data received from and sent to the virtual serial port to the physical device.



- **Host Pipe:** Configure Oracle VM VirtualBox to connect the virtual serial port to a software pipe on the host. This depends on your host OS, as follows:
  - On a Windows host, data will be sent and received through a named pipe. The pipe name must be in the format `\\.\pipe\name` where *name* should identify the virtual machine but may be freely chosen.
  - On a Mac OS, Linux, or Oracle Solaris host, a local domain socket is used instead. The socket filename must be chosen such that the user running Oracle VM VirtualBox has sufficient privileges to create and write to it. The `/tmp` directory is often a good candidate.

On Linux there are various tools which can connect to a local domain socket or create one in server mode. The most flexible tool is **socat** and is available as part of many distributions.

In this case, you can configure whether Oracle VM VirtualBox should create the named pipe, or the local domain socket non-Windows hosts, itself or whether Oracle VM VirtualBox should assume that the pipe or socket exists already. With the **VBoxManage** command-line options, this is referred to as server mode or client mode, respectively.

For a direct connection between two virtual machines, corresponding to a null-modem cable, simply configure one VM to create a pipe or socket and another to attach to it.

- **Raw File:** Send the virtual serial port output to a file. This option is very useful for capturing diagnostic output from a guest. Any file may be used for this purpose, as long as the user running Oracle VM VirtualBox has sufficient privileges to create and write to the file.
- **TCP:** Useful for forwarding serial traffic over TCP/IP, acting as a server, or it can act as a TCP client connecting to other servers. This option enables a remote machine to directly connect to the guest's serial port using TCP.
  - **TCP Server:** Deselect the **Connect to Existing Pipe/Socket** check box and specify the port number in the **Path/Address** field. This is typically 23 or 2023. Note that on UNIX-like systems you will have to use a port a number greater than 1024 for regular users.

The client can use software such as **PuTTY** or the **telnet** command line tool to access the TCP Server.

- **TCP Client:** To create a virtual null-modem cable over the Internet or LAN, the other side can connect using TCP by specifying *hostname:port* in the **Path/Address** field. The TCP socket will act in client mode if you select the **Connect to Existing Pipe/Socket** check box.

Up to four serial ports can be configured per virtual machine, but you can pick any port numbers out of the above. However, serial ports cannot reliably share interrupts. If both ports are to be used at the same time, they must use different interrupt levels, for example COM1 and COM2, but not COM1 and COM3.

## 3.11. USB Support

### 3.11.1. USB Settings

The **USB** section in a virtual machine's **Settings** window enables you to configure Oracle VM VirtualBox's sophisticated USB support.

Oracle VM VirtualBox can enable virtual machines to access the USB devices on your host directly. To achieve this, Oracle VM VirtualBox presents the guest OS with a virtual USB controller. As soon as the guest system starts using a USB device, it will appear as unavailable on the host.



**Note**

- Be careful with USB devices that are currently in use on the host. For example, if you allow your guest to connect to your USB hard disk that is currently mounted on the host, when the guest is activated, it will be disconnected from the host without a proper shutdown. This may cause data loss.
- Oracle Solaris hosts have a few known limitations regarding USB support. See [Chapter 14, Known Limitations](#).

In addition to allowing a guest access to your local USB devices, Oracle VM VirtualBox even enables your guests to connect to remote USB devices by use of the VirtualBox Remote Desktop Extension (VRDE). See [Section 7.1.4, “Remote USB”](#).

To enable USB for a VM, select the **Enable USB Controller** check box. The following settings are available:

- **USB Controller:** Selects a controller with the specified level of USB support, as follows:
  - OHCI for USB 1.1
  - EHCI for USB 2.0. This also enables OHCI.
  - xHCI for USB 3.0. This supports all USB speeds.
- **USB Device Filters:** When USB support is enabled for a VM, you can determine in detail which devices will be automatically attached to the guest. For this, you can create filters by specifying certain properties of the USB device. USB devices with a matching filter will be automatically passed to the guest once they are attached to the host. USB devices without a matching filter can be passed manually to the guest, for example by using the **Devices, USB** menu.

Clicking on the + button to the right of the **USB Device Filters** window creates a new filter. You can give the filter a name, for later reference, and specify the filter criteria. The more criteria you specify, the more precisely devices will be selected. For instance, if you specify only a vendor ID of 046d, all devices produced by Logitech will be available to the guest. If you fill in all fields, on the other hand, the filter will only apply to a particular device model from a particular vendor, and not even to other devices of the same type with a different revision and serial number.

In detail, the following criteria are available:

- **Vendor and Product ID.** With USB, each vendor of USB products carries an identification number that is unique world-wide, called the *vendor ID*. Similarly, each line of products is assigned a *product ID* number. Both numbers are commonly written in hexadecimal, and a colon separates the vendor from the product ID. For example, 046d:c016 stands for Logitech as a vendor, and the M-UV69a Optical Wheel Mouse product.

Alternatively, you can also specify **Manufacturer** and **Product** by name.

To list all the USB devices that are connected to your host machine with their respective vendor IDs and product IDs, use the following command:

```
VBoxManage list usbhost
```

On Windows, you can also see all USB devices that are attached to your system in the Device Manager. On Linux, you can use the **lsusb** command.

- **Serial Number.** While vendor ID and product ID are quite specific to identify USB devices, if you have two identical devices of the same brand and product line, you will also need their serial

numbers to filter them out correctly.

- **Remote.** This setting specifies whether the device will be local only, remote only, such as over VRDP, or either.

On a Windows host, you will need to unplug and reconnect a USB device to use it after creating a filter for it.

As an example, you could create a new USB filter and specify a vendor ID of 046d for Logitech, Inc, a manufacturer index of 1, and "not remote". Then any USB devices on the host system produced by Logitech, Inc with a manufacturer index of 1 will be visible to the guest system.

Several filters can select a single device. For example, a filter which selects all Logitech devices, and one which selects a particular webcam.

You can deactivate filters without deleting them by deselecting the check box next to the filter name.

### 3.11.2. Implementation Notes for Windows and Linux Hosts

On Windows hosts, a kernel mode device driver provides USB proxy support. It implements both a USB monitor, which enables Oracle VM VirtualBox to capture devices when they are plugged in, and a USB device driver to claim USB devices for a particular virtual machine. System reboots are not necessary after installing the driver. Also, you do not need to replug devices for Oracle VM VirtualBox to claim them.

On supported Linux hosts, Oracle VM VirtualBox accesses USB devices through special files in the file system. When Oracle VM VirtualBox is installed, these are made available to all users in the `vboxusers` system group. In order to be able to access USB from guest systems, make sure that you are a member of this group.

## 3.12. Shared Folders

Shared folders enable you to easily exchange data between a virtual machine and your host. This feature requires that the Oracle VM VirtualBox Guest Additions be installed in a virtual machine and is described in detail in [Section 4.3, "Shared Folders"](#).

## 3.13. User Interface

The **User Interface** section enables you to change certain aspects of the user interface of the selected VM.

- **Menu Bar:** This widget enables you to disable a complete menu, by clicking on the menu name to deselect it. Menu entries can be disabled, by deselecting the check box next to the entry. On Windows and Linux hosts, the complete menu bar can be disabled by deselecting the check box on the right.
- **Mini Toolbar:** In full screen or seamless mode, Oracle VM VirtualBox can display a small toolbar that contains some of the items that are normally available from the virtual machine's menu bar. This toolbar reduces itself to a small gray line unless you move the mouse over it. With the toolbar, you can return from full screen or seamless mode, control machine execution, or enable certain devices. If you do not want to see the toolbar, disable the **Show in Full Screen/Seamless** setting.

The **Show at Top of Screen** setting enables you to show the toolbar at the top of the screen, instead of showing it at the bottom.

The Mini Toolbar is not available on macOS hosts.

- **Status Bar:** This widget enables you to disable and reorder icons on the status bar. Deselect the check box of an icon to disable it, or rearrange icons by dragging and dropping the icon. To disable the complete status bar deselect the check box on the left.

## 3.14. Alternative Firmware (EFI)

Oracle VM VirtualBox includes experimental support for the Extensible Firmware Interface (EFI), which is an industry standard intended to replace the legacy BIOS as the primary interface for bootstrapping computers and certain system services later.

By default, Oracle VM VirtualBox uses the BIOS firmware for virtual machines. To use EFI for a given virtual machine, you can enable EFI in the machine's **Settings** window. See [Section 3.5.1, “Motherboard Tab”](#).

Alternatively, use the **VBoxManage** command line interface as follows:

```
VBoxManage modifyvm "VM name" --firmware efi
```

To switch back to using the BIOS:

```
VBoxManage modifyvm "VM name" --firmware bios
```

One notable user of EFI is Apple Mac OS X. More recent Linux versions and Windows releases, starting with Vista, also offer special versions that can be booted using EFI.

Another possible use of EFI in Oracle VM VirtualBox is development and testing of EFI applications, without booting any OS.

Note that the Oracle VM VirtualBox EFI support is experimental and will be enhanced as EFI matures and becomes more widespread. Mac OS X, Linux, and newer Windows guests are known to work fine. Windows 7 guests are unable to boot with the Oracle VM VirtualBox EFI implementation.

### 3.14.1. Video Modes in EFI

EFI provides two distinct video interfaces: GOP (Graphics Output Protocol) and UGA (Universal Graphics Adapter). Modern OSes, such as Mac OS X, generally use GOP, while some older ones still use UGA. Oracle VM VirtualBox provides a configuration option to control the graphics resolution for both interfaces, making the difference mostly irrelevant for users.

The default resolution is 1024x768. To select a graphics resolution for EFI, use the following **VBoxManage** command:

```
VBoxManage setextradata "VM name" VBoxInternal2/EfiGraphicsResolution HxV
```

Determine the horizontal resolution H and the vertical resolution V from the following list of default resolutions:

VGA

640x480, 32bpp, 4:3

SVGA

800x600, 32bpp, 4:3

XGA

1024x768, 32bpp, 4:3

XGA+

1152x864, 32bpp, 4:3

HD

1280x720, 32bpp, 16:9

## WXGA

1280x800, 32bpp, 16:10

## SXGA

1280x1024, 32bpp, 5:4

## SXGA+

1400x1050, 32bpp, 4:3

## WXGA+

1440x900, 32bpp, 16:10

## HD+

1600x900, 32bpp, 16:9

## UXGA

1600x1200, 32bpp, 4:3

## WSXGA+

1680x1050, 32bpp, 16:10

## Full HD

1920x1080, 32bpp, 16:9

## WUXGA

1920x1200, 32bpp, 16:10

## DCI 2K

2048x1080, 32bpp, 19:10

## Full HD+

2160x1440, 32bpp, 3:2

## Unnamed

2304x1440, 32bpp, 16:10

## QHD

2560x1440, 32bpp, 16:9

## WQXGA

2560x1600, 32bpp, 16:10

## QWXGA+

2880x1800, 32bpp, 16:10

## QHD+

3200x1800, 32bpp, 16:9

WQSXGA

3200x2048, 32bpp, 16:10

4K UHD

3840x2160, 32bpp, 16:9

WQUXGA

3840x2400, 32bpp, 16:10

DCI 4K

4096x2160, 32bpp, 19:10

HXGA

4096x3072, 32bpp, 4:3

UHD+

5120x2880, 32bpp, 16:9

WHXGA

5120x3200, 32bpp, 16:10

WHSXGA

6400x4096, 32bpp, 16:10

HUXGA

6400x4800, 32bpp, 4:3

8K UHD2

7680x4320, 32bpp, 16:9

If this list of default resolution does not cover your needs, see [Section 9.6.1, “Custom VESA Resolutions”](#). Note that the color depth value specified in a custom video mode must be specified. Color depths of 8, 16, 24, and 32 are accepted. EFI assumes a color depth of 32 by default.

The EFI default video resolution settings can only be changed when the VM is powered off.

### 3.14.2. Specifying Boot Arguments

It is currently not possible to manipulate EFI variables from within a running guest. For example, setting the `boot-args` variable by running the **nvrnm** tool in a Mac OS X guest will not work. As an alternative method, `VBoxInternal2/EfiBootArgs` extradata can be passed to a VM in order to set the `boot-args` variable. To change the `boot-args` EFI variable, use the following command:

```
VBoxManage setextradata "VM name" VBoxInternal2/EfiBootArgs <value>
```