

Chapter 4. Guest Additions

Table of Contents

- [4.1. Introduction to Guest Additions](#)
- [4.2. Installing and Maintaining Guest Additions](#)
 - [4.2.1. Guest Additions for Windows](#)
 - [4.2.2. Guest Additions for Linux](#)
 - [4.2.3. Guest Additions for Oracle Solaris](#)
 - [4.2.4. Guest Additions for OS/2](#)
- [4.3. Shared Folders](#)
 - [4.3.1. Manual Mounting](#)
 - [4.3.2. Automatic Mounting](#)
- [4.4. Drag and Drop](#)
 - [4.4.1. Supported Formats](#)
 - [4.4.2. Known Limitations](#)
- [4.5. Hardware-Accelerated Graphics](#)
 - [4.5.1. Hardware 3D Acceleration \(OpenGL and Direct3D 8/9\)](#)
 - [4.5.2. Hardware 2D Video Acceleration for Windows Guests](#)
- [4.6. Seamless Windows](#)
- [4.7. Guest Properties](#)
 - [4.7.1. Using Guest Properties to Wait on VM Events](#)
- [4.8. Guest Control File Manager](#)
 - [4.8.1. Using the Guest Control File Manager](#)
- [4.9. Guest Control of Applications](#)
- [4.10. Memory Overcommitment](#)
 - [4.10.1. Memory Ballooning](#)
 - [4.10.2. Page Fusion](#)
- [4.11. Controlling Virtual Monitor Topology](#)
 - [4.11.1. X11/Wayland Desktop Environments](#)

The previous chapter covered getting started with Oracle VM VirtualBox and installing operating systems in a virtual machine. For any serious and interactive use, the Oracle VM VirtualBox Guest Additions will make your life much easier by providing closer integration between host and guest and improving the interactive performance of guest systems. This chapter describes the Guest Additions in detail.

4.1. Introduction to Guest Additions

As mentioned in [Section 1.2, “Some Terminology”](#), the Guest Additions are designed to be installed *inside* a virtual machine after the guest operating system has been installed. They consist of device drivers and system applications that optimize the guest operating system for better performance and usability. See [Section 3.1, “Supported Guest Operating Systems”](#) for details on what guest operating systems are fully supported with Guest Additions by Oracle VM VirtualBox.

The Oracle VM VirtualBox Guest Additions for all supported guest operating systems are provided as a single CD-ROM image file which is called `VBoxGuestAdditions.iso`. This image file is located in the installation directory of Oracle VM VirtualBox. To install the Guest Additions for a particular VM, you mount this ISO file in your VM as a virtual CD-ROM and install from there.

The Guest Additions offer the following features:

- **Mouse pointer integration.** To overcome the limitations for mouse support described in [Section 1.9.2, “Capturing and Releasing Keyboard and Mouse”](#), this feature provides you with seamless mouse support.

You will only have one mouse pointer and pressing the Host key is no longer required to *free* the mouse from being captured by the guest OS. To make this work, a special mouse driver is installed in the guest that communicates with the physical mouse driver on your host and moves the guest mouse pointer accordingly.

- **Shared folders.** These provide an easy way to exchange files between the host and the guest. Much like ordinary Windows network shares, you can tell Oracle VM VirtualBox to treat a certain host directory as a shared folder, and Oracle VM VirtualBox will make it available to the guest operating system as a network share, irrespective of whether the guest actually has a network. See [Section 4.3, “Shared Folders”](#).
- **Better video support.** While the virtual graphics card which Oracle VM VirtualBox emulates for any guest operating system provides all the basic features, the custom video drivers that are installed with the Guest Additions provide you with extra high and non-standard video modes, as well as accelerated video performance.

In addition, with Windows, Linux, and Oracle Solaris guests, you can resize the virtual machine's window if the Guest Additions are installed. The video resolution in the guest will be automatically adjusted, as if you had manually entered an arbitrary resolution in the guest's **Display** settings. See [Section 1.9.5, “Resizing the Machine's Window”](#).

If the Guest Additions are installed, 3D graphics and 2D video for guest applications can be accelerated. See [Section 4.5, “Hardware-Accelerated Graphics”](#).

- **Seamless windows.** With this feature, the individual windows that are displayed on the desktop of the virtual machine can be mapped on the host's desktop, as if the underlying application was actually running on the host. See [Section 4.6, “Seamless Windows”](#).
- **Generic host/guest communication channels.** The Guest Additions enable you to control and monitor guest execution. The *guest properties* provide a generic string-based mechanism to exchange data bits between a guest and a host, some of which have special meanings for controlling and monitoring the guest. See [Section 4.7, “Guest Properties”](#).

Additionally, applications can be started in a guest from the host. See [Section 4.9, “Guest Control of Applications”](#).

- **Time synchronization.** With the Guest Additions installed, Oracle VM VirtualBox can ensure that the guest's system time is better synchronized with that of the host.

For various reasons, the time in the guest might run at a slightly different rate than the time on the host. The host could be receiving updates through NTP and its own time might not run linearly. A VM could also be paused, which stops the flow of time in the guest for a shorter or longer period of time. When the wall clock time between the guest and host only differs slightly, the time synchronization service attempts to gradually and smoothly adjust the guest time in small increments to either catch up or lose time. When the difference is too great, for example if a VM paused for hours or restored from saved state, the guest time is changed immediately, without a gradual adjustment.

The Guest Additions will resynchronize the time regularly. See [Section 9.11.3, “Tuning the Guest Additions Time Synchronization Parameters”](#) for how to configure the parameters of the time synchronization mechanism.

- **Shared clipboard.** With the Guest Additions installed, the clipboard of the guest operating system can optionally be shared with your host operating system. See [Section 3.4, “General Settings”](#).
- **Automated logins.** Also called credentials passing. See [Section 9.1, “Automated Guest Logins”](#).

Each version of Oracle VM VirtualBox, even minor releases, ship with their own version of the Guest Additions.

While the interfaces through which the Oracle VM VirtualBox core communicates with the Guest Additions are kept stable so that Guest Additions already installed in a VM should continue to work when Oracle VM VirtualBox is upgraded on the host, for best results, it is recommended to keep the Guest Additions at the same version.

The Windows and Linux Guest Additions therefore check automatically whether they have to be updated. If the host is running a newer Oracle VM VirtualBox version than the Guest Additions, a notification with further instructions is displayed in the guest.

To disable this update check for the Guest Additions of a given virtual machine, set the value of its `/VirtualBox/GuestAdd/CheckHostVersion` guest property to 0. See [Section 4.7, “Guest Properties”](#).

4.2. Installing and Maintaining Guest Additions

Guest Additions are available for virtual machines running Windows, Linux, Oracle Solaris, or OS/2. The following sections describe the specifics of each variant in detail.

4.2.1. Guest Additions for Windows

The Oracle VM VirtualBox Windows Guest Additions are designed to be installed in a virtual machine running a Windows operating system. The following versions of Windows guests are supported:

- Microsoft Windows NT 4.0 (any service pack)
- Microsoft Windows 2000 (any service pack)
- Microsoft Windows XP (any service pack)
- Microsoft Windows Server 2003 (any service pack)
- Microsoft Windows Server 2008
- Microsoft Windows Vista (all editions)
- Microsoft Windows 7 (all editions)
- Microsoft Windows 8 (all editions)
- Microsoft Windows 10 RTM build 10240
- Microsoft Windows Server 2012

4.2.1.1. Installing the Windows Guest Additions

In the **Devices** menu in the virtual machine's menu bar, Oracle VM VirtualBox has a menu item **Insert Guest Additions CD Image**, which mounts the Guest Additions ISO file inside your virtual machine. A Windows guest should then automatically start the Guest Additions installer, which installs the Guest Additions on your Windows guest.

For other guest operating systems, or if automatic start of software on a CD is disabled, you need to do a manual start of the installer.

Note

For the basic Direct3D acceleration to work in a Windows guest, you have to install the WDDM video driver available for Windows Vista or later.

For Windows 8 and later, only the WDDM Direct3D video driver is available. For basic Direct3D acceleration to work in Windows XP guests, you have to install the Guest Additions in Safe

Mode. See [Chapter 14, Known Limitations](#) for details.

If you prefer to mount the Guest Additions manually, you can perform the following steps:

1. Start the virtual machine in which you have installed Windows.
2. Select **Optical Drives** from the **Devices** menu in the virtual machine's menu bar and then **Choose/Create a Disk Image**. This displays the Virtual Media Manager, described in [Section 5.3, "The Virtual Media Manager"](#).
3. In the Virtual Media Manager, click **Add** and browse your host file system for the `VBoxGuestAdditions.iso` file.
 - On a Windows host, this file is in the Oracle VM VirtualBox installation directory, usually in `C:\Program files\Oracle\VirtualBox`.
 - On macOS hosts, this file is in the application bundle of Oracle VM VirtualBox. Right-click on the Oracle VM VirtualBox icon in Finder and choose **Show Package Contents**. The file is located in the `Contents/MacOS` folder.
 - On a Linux host, this file is in the `additions` folder where you installed Oracle VM VirtualBox, usually `/opt/VirtualBox/`.
 - On Oracle Solaris hosts, this file is in the `additions` folder where you installed Oracle VM VirtualBox, usually `/opt/VirtualBox`.
4. In the Virtual Media Manager, select the ISO file and click the **Add** button. This mounts the ISO file and presents it to your Windows guest as a CD-ROM.

Unless you have the Autostart feature disabled in your Windows guest, Windows will now autostart the Oracle VM VirtualBox Guest Additions installation program from the Additions ISO. If the Autostart feature has been turned off, choose `VBoxWindowsAdditions.exe` from the CD/DVD drive inside the guest to start the installer.

The installer will add several device drivers to the Windows driver database and then invoke the hardware detection wizard.

Depending on your configuration, it might display warnings that the drivers are not digitally signed. You must confirm these in order to continue the installation and properly install the Additions.

After installation, reboot your guest operating system to activate the Additions.

4.2.1.2. Updating the Windows Guest Additions

Windows Guest Additions can be updated by running the installation program again. This replaces the previous Additions drivers with updated versions.

Alternatively, you can also open the Windows Device Manager and select **Update Driver...** for the following devices:

- Oracle VM VirtualBox Graphics Adapter
- Oracle VM VirtualBox System Device

For each, choose the option to provide your own driver, click **Have Disk** and navigate to the CD-ROM drive with the Guest Additions.

4.2.1.3. Unattended Installation of the Windows Guest Additions

You can configure unattended installation of the Oracle VM VirtualBox Guest Additions when you create a new

VM using the **Create Virtual Machine** wizard. Select the **Guest Additions** check box on the **Unattended Guest OS Install** page of the wizard.

Guest Additions are installed automatically, following completion of the guest OS installation.

Installing Code Signing Certificates

To avoid popups when performing an unattended installation of the Oracle VM VirtualBox Guest Additions, the code signing certificates used to sign the drivers needs to be installed in the correct certificate stores on the guest operating system. Failure to do this will cause a typical Windows installation to display multiple dialogs asking whether you want to install a particular driver.

Note

On some legacy Windows versions, such as Windows 2000 and Windows XP, the user intervention popups mentioned above are always displayed, even after importing the Oracle certificates.

Installing the code signing certificates on a Windows guest can be done automatically. Use the `VBoxCertUtil.exe` utility from the `cert` folder on the Guest Additions installation CD.

Use the following steps:

1. Log in as Administrator on the guest.
2. Mount the Oracle VM VirtualBox Guest Additions .ISO.
3. Open a command line window on the guest and change to the `cert` folder on the Oracle VM VirtualBox Guest Additions CD.
4. Run the following command:

```
VBoxCertUtil.exe add-trusted-publisher vbox*.cer --root vbox*.cer
```

This command installs the certificates to the certificate store. When installing the same certificate more than once, an appropriate error will be displayed.

To allow for completely unattended guest installations, you can specify a command line parameter to the install launcher:

```
VBoxWindowsAdditions.exe /S
```

This automatically installs the right files and drivers for the corresponding platform, either 32-bit or 64-bit.

Note

By default on an unattended installation on a Vista or Windows 7 guest, there will be the XPDM graphics driver installed. This graphics driver does not support Windows Aero / Direct3D on the guest. Instead, the WDDM graphics driver needs to be installed. To select this driver by default, add the command line parameter `/with_wddm` when invoking the Windows Guest Additions installer. This is only required for Vista and Windows 7.

Note

For Windows Aero to run correctly on a guest, the guest's VRAM size needs to be configured to at least 128 MB.

For more options regarding unattended guest installations, consult the command line help by using the command:

```
VBoxWindowsAdditions.exe /?
```

4.2.1.4. Manual File Extraction

If you would like to install the files and drivers manually, you can extract the files from the Windows Guest Additions setup as follows:

```
VBoxWindowsAdditions.exe /extract
```

To explicitly extract the Windows Guest Additions for another platform than the current running one, such as 64-bit files on a 32-bit system, you must use the appropriate platform installer. Use `VBoxWindowsAdditions-x86.exe` or `VBoxWindowsAdditions-amd64.exe` with the `/extract` parameter.

4.2.2. Guest Additions for Linux

Like the Windows Guest Additions, the Oracle VM VirtualBox Guest Additions for Linux are a set of device drivers and system applications which may be installed in the guest operating system.

The following Linux distributions are officially supported:

- Oracle Linux as of version 5, including UEK kernels
- Fedora as of Fedora Core 4
- Red Hat Enterprise Linux as of version 3
- SUSE and openSUSE Linux as of version 9
- Ubuntu as of version 5.10

Many other distributions are known to work with the Guest Additions.

The version of the Linux kernel supplied by default in SUSE and openSUSE 10.2, Ubuntu 6.10 (all versions) and Ubuntu 6.06 (server edition) contains a bug which can cause it to crash during startup when it is run in a virtual machine. The Guest Additions work in those distributions.

Note that some Linux distributions already come with all or part of the Oracle VM VirtualBox Guest Additions. You may choose to keep the distribution's version of the Guest Additions but these are often not up to date and limited in functionality, so we recommend replacing them with the Guest Additions that come with Oracle VM VirtualBox. The Oracle VM VirtualBox Linux Guest Additions installer tries to detect an existing installation and replace them but depending on how the distribution integrates the Guest Additions, this may require some manual interaction. It is highly recommended to take a snapshot of the virtual machine before replacing preinstalled Guest Additions.

4.2.2.1. Installing the Linux Guest Additions

The Oracle VM VirtualBox Guest Additions for Linux are provided on the same virtual CD-ROM file as the Guest Additions for Windows. See [Section 4.2.1.1, “Installing the Windows Guest Additions”](#). They also come with an installation program that guides you through the setup process. However, due to the significant differences between Linux distributions, installation may be slightly more complex when compared to Windows.

Installation generally involves the following steps:

1. Before installing the Guest Additions, you prepare your guest system for building external kernel modules. This works as described in [Section 2.3.2, “The Oracle VM VirtualBox Kernel Modules”](#), except that this step must be performed in your Linux *guest* instead of on a Linux host system.

If you suspect that something has gone wrong, check that your guest is set up correctly and run the following command as root:

```
rcvboxadd setup
```

2. Insert the `VBoxGuestAdditions.iso` CD file into your Linux guest's virtual CD-ROM drive, as described for a Windows guest in [Section 4.2.1.1, "Installing the Windows Guest Additions"](#).
3. Change to the directory where your CD-ROM drive is mounted and run the following command as root:

```
sh ./VBoxLinuxAdditions.run
```

4.2.2.2. Unattended Installation of the Linux Guest Additions

You can configure unattended installation of the Oracle VM VirtualBox Guest Additions when you create a new VM using the **Create Virtual Machine** wizard. Select the **Guest Additions** check box on the **Unattended Guest OS Install** page of the wizard.

Guest Additions are installed automatically, following completion of the guest OS installation.

4.2.2.3. Graphics and Mouse Integration

In Linux and Oracle Solaris guests, Oracle VM VirtualBox graphics and mouse integration goes through the X Window System. Oracle VM VirtualBox can use the X.Org variant of the system, or XFree86 version 4.3 which is identical to the first X.Org release. During the installation process, the X.Org display server will be set up to use the graphics and mouse drivers which come with the Guest Additions.

After installing the Guest Additions into a fresh installation of a supported Linux distribution or Oracle Solaris system, many unsupported systems will work correctly too, the guest's graphics mode will change to fit the size of the Oracle VM VirtualBox window on the host when it is resized. You can also ask the guest system to switch to a particular resolution by sending a video mode hint using the **VBoxManage** tool.

Multiple guest monitors are supported in guests using the X.Org server version 1.3, which is part of release 7.3 of the X Window System version 11, or a later version. The layout of the guest screens can be adjusted as needed using the tools which come with the guest operating system.

If you want to understand more about the details of how the X.Org drivers are set up, in particular if you wish to use them in a setting which our installer does not handle correctly, see [Section 9.3.2, "Guest Graphics and Mouse Driver Setup in Depth"](#).

Starting from Oracle VM VirtualBox 7, Linux guest screen resize functionality for guests running VMSVGA graphics configuration has been changed. Since then, this functionality consists of a standalone daemon called `VBoxDRMClient` and its Desktop Environment helper counterpart.

`VBoxDRMClient` runs as a root process and is a bridge between the host and the guest's `vmwgfx` driver. This means that `VBoxDRMClient` listens to screen resize hints from the host and forwards them to the `vmwgfx` driver. This enables guest screen resize functionality to be available before the user has performed a graphical login.

In order to perform Desktop Environment specific actions, such as setting the primary screen in a multimonitor setup, a Desktop Environment helper is used. Once the user has performed a graphical login operation, the helper daemon starts with user session scope and attempts to connect to `VBoxDRMClient` using an IPC connection. When `VBoxDRMClient` has received a corresponding command from the host, it is forwarded to the helper daemon over IPC and the action is then performed.

By default, `VBoxDRMClient` allows any process to connect to its IPC socket. This can be restricted by using the following steps:

1. The Guest Additions Linux installer creates a `vboxdrmpc` user group. A corresponding user needs to be added to this group.
2. You must set the `DRMIpcRestricted` guest property, as follows:

```
VBoxManage guestproperty set "VM name" /VirtualBox/GuestAdd/DRMIpcRestricted 1 \
--flags RONLYGUEST
```

It is important to set only the `RONLYGUEST` flag for the property, so that it cannot be changed from inside the guest.

Note

Both steps are required. If one of them is missing, all processes will have access to the IPC socket.

Restricted mode can be disabled by unsetting the guest property, as follows:

```
VBoxManage guestproperty unset "VM name" /VirtualBox/GuestAdd/DRMIpcRestricted
```

4.2.2.4. Updating the Linux Guest Additions

The Guest Additions can simply be updated by going through the installation procedure again with an updated CD-ROM image. This will replace the drivers with updated versions. You should reboot after updating the Guest Additions.

4.2.2.5. Uninstalling the Linux Guest Additions

If you have a version of the Guest Additions installed on your virtual machine and wish to remove it without installing new ones, you can do so by inserting the Guest Additions CD image into the virtual CD-ROM drive as described above. Then run the installer for the current Guest Additions with the `uninstall` parameter from the path that the CD image is mounted on in the guest, as follows:

```
sh ./VBoxLinuxAdditions.run uninstall
```

While this will normally work without issues, you may need to do some manual cleanup of the guest in some cases, especially of the `XFree86Config` or `xorg.conf` file. In particular, if the Additions version installed or the guest operating system were very old, or if you made your own changes to the Guest Additions setup after you installed them.

You can uninstall the Additions as follows:

```
/opt/VBoxGuestAdditions-version/uninstall.sh
```

Replace `/opt/VBoxGuestAdditions-version` with the correct Guest Additions installation directory.

4.2.3. Guest Additions for Oracle Solaris

Like the Windows Guest Additions, the Oracle VM VirtualBox Guest Additions for Oracle Solaris take the form of a set of device drivers and system applications which may be installed in the guest operating system.

The following Oracle Solaris distributions are officially supported:

- Oracle Solaris 11, including Oracle Solaris 11 Express
- Oracle Solaris 10 4/08 and later

Other distributions may work if they are based on comparable software releases.

4.2.3.1. Installing the Oracle Solaris Guest Additions

The Oracle VM VirtualBox Guest Additions for Oracle Solaris are provided on the same ISO CD-ROM as the Additions for Windows and Linux. They come with an installation program that guides you through the setup process.

Installation involves the following steps:

1. Mount the `VBoxGuestAdditions.iso` file as your Oracle Solaris guest's virtual CD-ROM drive, exactly the same way as described for a Windows guest in [Section 4.2.1.1, "Installing the Windows Guest Additions"](#).

If the CD-ROM drive on the guest does not get mounted, as seen with some versions of Oracle Solaris 10, run the following command as root:

```
svcadm restart volfs
```

2. Change to the directory where your CD-ROM drive is mounted and run the following command as root:

```
pkgadd -G -d ./VBoxSolarisAdditions.pkg
```

3. Choose **1** and confirm installation of the Guest Additions package. After the installation is complete, log out and log in to X server on your guest, to activate the X11 Guest Additions.

4.2.3.2. Unattended Installation of the Oracle Solaris Guest Additions

You can configure unattended installation of the Oracle VM VirtualBox Guest Additions when you create a new VM using the **Create Virtual Machine** wizard. Select the **Guest Additions** check box on the **Unattended Guest OS Install** page of the wizard.

Guest Additions are installed automatically, following completion of the guest OS installation.

4.2.3.3. Uninstalling the Oracle Solaris Guest Additions

The Oracle Solaris Guest Additions can be safely removed by removing the package from the guest. Open a root terminal session and run the following command:

```
pkgrm SUNWvboxguest
```

4.2.3.4. Updating the Oracle Solaris Guest Additions

The Guest Additions should be updated by first uninstalling the existing Guest Additions and then installing the new ones. Attempting to install new Guest Additions without removing the existing ones is not possible.

4.2.4. Guest Additions for OS/2

Oracle VM VirtualBox also ships with a set of drivers that improve running OS/2 in a virtual machine. Due to restrictions of OS/2 itself, this variant of the Guest Additions has a limited feature set. See [Chapter 14, *Known Limitations*](#) for details.

The OS/2 Guest Additions are provided on the same ISO CD-ROM as those for the other platforms. Mount the ISO in OS/2 as described previously. The OS/2 Guest Additions are located in the directory `\OS2`.

We do not provide an automatic installer at this time. See the `readme.txt` file in the CD-ROM directory, which describes how to install the OS/2 Guest Additions manually.

4.3. Shared Folders

With the *shared folders* feature of Oracle VM VirtualBox, you can access files of your host system from within the guest system. This is similar to how you would use network shares in Windows networks, except that shared folders do not require networking, only the Guest Additions. Shared folders are supported with Windows 2000 or later, Linux, and Oracle Solaris guests. Oracle VM VirtualBox includes experimental support for Mac OS X and OS/2 guests.

Shared folders physically reside on the *host* and are then shared with the guest, which uses a special file system driver in the Guest Additions to talk to the host. For Windows guests, shared folders are implemented as a pseudo-network redirector. For Linux and Oracle Solaris guests, the Guest Additions provide a virtual file system.

To share a host folder with a virtual machine in Oracle VM VirtualBox, you must specify the path of the folder and choose a *share name* that the guest can use to access the shared folder. This happens on the host. In the guest you can then use the share name to connect to it and access files.

There are several ways in which shared folders can be set up for a virtual machine:

- In the window of a running VM, you select **Shared Folders** from the **Devices** menu, or click on the folder icon on the status bar in the bottom right corner.
- If a VM is not currently running, you can configure shared folders in the virtual machine's **Settings** window.
- From the command line, you can create shared folders using **VBoxManage**, as follows:

```
VBoxManage sharedfolder add "VM name" --name "sharename" --hostpath "C:\test"
```

See [Section 8.42, "VBoxManage sharedfolder"](#).

There are two types of shares:

- Permanent shares, that are saved with the VM settings.
- Transient shares, that are added at runtime and disappear when the VM is powered off. These can be created using a check box in VirtualBox Manager, or by using the `--transient` option of the **VBoxManage sharedfolder add** command.

Shared folders can either be read-write or read-only. This means that the guest is either allowed to both read and write, or just read files on the host. By default, shared folders are read-write. Read-only folders can be created using a check box in the VirtualBox Manager, or with the `--readonly` option of the **VBoxManage sharedfolder add** command.

Oracle VM VirtualBox shared folders also support symbolic links, also called *symlinks*, under the following conditions:

- The host operating system must support symlinks. For example, a macOS, Linux, or Oracle Solaris host is required.
- Currently only Linux and Oracle Solaris Guest Additions support symlinks.
- For security reasons the guest OS is not allowed to create symlinks by default. If you trust the guest OS to not abuse the functionality, you can enable creation of symlinks for a shared folder as follows:

```
VBoxManage setextradata "VM name" VBoxInternal2/SharedFoldersEnableSymlinksCreate/sharename 1
```

4.3.1. Manual Mounting

You can mount the shared folder from inside a VM, in the same way as you would mount an ordinary network share:

- In a Windows guest, shared folders are browseable and therefore visible in Windows Explorer. To attach the host's shared folder to your Windows guest, open Windows Explorer and look for the folder in **My Networking Places, Entire Network, Oracle VM VirtualBox Shared Folders**. By right-clicking on a shared folder and selecting **Map Network Drive** from the menu that pops up, you can assign a drive letter to that shared folder.

Alternatively, on the Windows command line, use the following command:

```
net use x: \\vboxsvr\sharename
```

While `vboxsvr` is a fixed name, note that `vboxsrv` would also work, replace `x:` with the drive letter that you want to use for the share, and `sharename` with the share name specified with **VBoxManage**.

- In a Linux guest, use the following command:

```
mount -t vboxsf [-o OPTIONS] sharename mountpoint
```

To mount a shared folder during boot, add the following entry to `/etc/fstab`:

```
sharename mountpoint vboxsf defaults 0 0
```

- In a Oracle Solaris guest, use the following command:

```
mount -F vboxfs [-o OPTIONS] sharename mountpoint
```

Replace `sharename`, use a lowercase string, with the share name specified with **VBoxManage** or VirtualBox Manager. Replace `mountpoint` with the path where you want the share to be mounted on the guest, such as `/mnt/share`. The usual mount rules apply. For example, create this directory first if it does not exist yet.

Here is an example of mounting the shared folder for the user jack on Oracle Solaris:

```
$ id
uid=5000(jack) gid=1(other)
$ mkdir /export/home/jack/mount
$ pfexec mount -F vboxfs -o uid=5000,gid=1 jackshare /export/home/jack/mount
$ cd ~/mount
$ ls
sharedfile1.mp3 sharedfile2.txt
$
```

Beyond the standard options supplied by the **mount** command, the following are available:

```
iocharset CHARSET
```

This option sets the character set used for I/O operations. Note that on Linux guests, if the `iocharset` option is not specified, then the Guest Additions driver will attempt to use the character set specified by the `CONFIG_NLS_DEFAULT` kernel option. If this option is not set either, then UTF-8 is used.

```
convertcp CHARSET
```

This option specifies the character set used for the shared folder name. This is UTF-8 by default.

The generic mount options, documented in the **mount** manual page, apply also. Especially useful are the options `uid`, `gid` and `mode`, as they can allow access by normal users in read/write mode, depending on the settings, even if root has mounted the filesystem.

- In an OS/2 guest, use the **VBoxControl** command to manage shared folders. For example:

```
VBoxControl sharedfolder use D: MyShareName
VBoxControl sharedfolder unuse D:
VBoxControl sharedfolder list
```

As with Windows guests, shared folders can also be accessed via UNC using `\\VBoxSF\`, `\\VBoxSrv\` or `\\VBoxSrv\` as the server name and the shared folder name as *sharename*.

4.3.2. Automatic Mounting

Oracle VM VirtualBox provides the option to mount shared folders automatically. When automatic mounting is enabled for a shared folder, the Guest Additions service will mount it for you automatically. For Windows or OS/2, a preferred drive letter can also be specified. For Linux or Oracle Solaris, a mount point directory can also be specified.

If a drive letter or mount point is not specified, or is in use already, an alternative location is found by the Guest Additions service. The service searches for an alternative location depending on the guest OS, as follows:

- **Windows and OS/2 guests.** Search for a free drive letter, starting at `z:`. If all drive letters are assigned, the folder is not mounted.
- **Linux and Oracle Solaris guests.** Folders are mounted under the `/media` directory. The folder name is normalized (no spaces, slashes or colons) and is prefixed with `sf_`.

For example, if you have a shared folder called `myfiles`, it will appear as `/media/sf_myfiles` in the guest.

The guest properties `/VirtualBox/GuestAdd/SharedFolders/MountDir` and the more generic `/VirtualBox/GuestAdd/SharedFolders/MountPrefix` can be used to override the automatic mount directory and prefix. See [Section 4.7, “Guest Properties”](#).

Access to an automatically mounted shared folder is granted to everyone in a Windows guest, including the guest user. For Linux and Oracle Solaris guests, access is restricted to members of the group `vboxsf` and the `root` user.

4.4. Drag and Drop

Oracle VM VirtualBox enables you to drag and drop content from the host to the guest, and vice versa. For this to work the latest version of the Guest Additions must be installed on the guest.

Drag and drop transparently allows copying or opening files, directories, and even certain clipboard formats from one end to the other. For example, from the host to the guest or from the guest to the host. You then can perform drag and drop operations between the host and a VM, as it would be a native drag and drop operation on the host OS.

At the moment drag and drop is implemented for Windows-based and X-Windows-based systems, both on the host and guest side. As X-Windows supports many different drag and drop protocols only the most common one, XDND, is supported for now. Applications using other protocols, such as Motif or OffiX, will not be recognized by Oracle VM VirtualBox.

In the context of using drag and drop, the origin of the data is called the *source*. That is, where the actual data comes from and is specified. The *destination* specifies where the data from the source should go to. Transferring data from the source to the destination can be done in various ways, such as copying, moving, or linking.

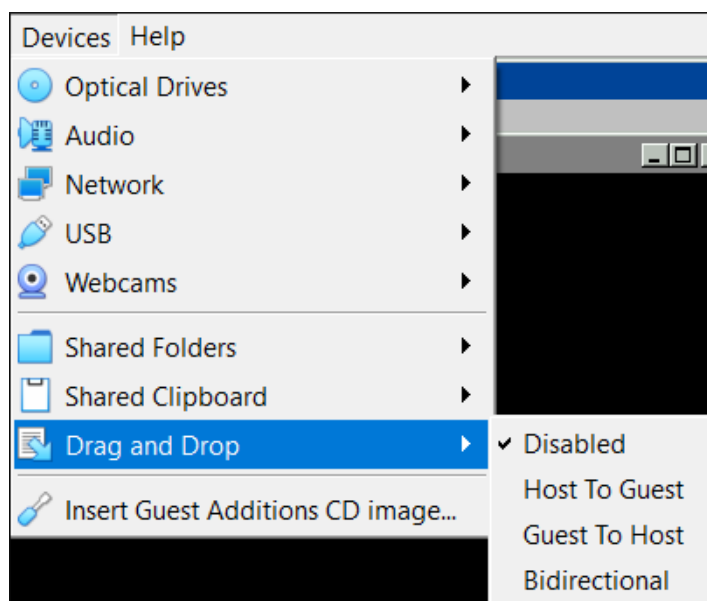
Note

At the moment only copying of data is supported. Moving or linking is not yet implemented.

When transferring data from the host to the guest OS, the host in this case is the source, whereas the guest OS is the destination. However, when transferring data from the guest OS to the host, the guest OS this time became the source and the host is the destination.

For security reasons drag and drop can be configured at runtime on a per-VM basis either using the **Drag and Drop** menu item in the **Devices** menu of the virtual machine, as shown below, or the **VBoxManage** command.

Figure 4.1. Drag and Drop Menu Options



The following drag and drop modes are available:

- **Disabled.** Disables the drag and drop feature entirely. This is the default when creating a new VM.
- **Host To Guest.** Enables drag and drop operations from the host to the guest only.
- **Guest To Host.** Enables drag and drop operations from the guest to the host only.
- **Bidirectional.** Enables drag and drop operations in both directions: from the host to the guest, and from the guest to the host.

Note

Drag and drop support depends on the frontend being used. At the moment, only the VirtualBox Manager frontend provides this functionality.

To use the **VBoxManage** command to control the current drag and drop mode, see [Chapter 8, VBoxManage](#). The **modifyvm** and **controlvm** commands enable setting of a VM's current drag and drop mode from the command line.

4.4.1. Supported Formats

As Oracle VM VirtualBox can run on a variety of host operating systems and also supports a wide range of guests, certain data formats must be translated after transfer. This is so that the destination operating system, which receives the data, is able to handle them in an appropriate manner.

Note

When dragging files no data conversion is done in any way. For example, when transferring a file from a Linux guest to a Windows host the Linux-specific line endings are not converted to Windows line endings.

The following formats are handled by the Oracle VM VirtualBox drag and drop service:

- **Plain text:** From applications such as text editors, internet browsers and terminal windows.
- **Files:** From file managers such as Windows Explorer, Nautilus, and Finder.
- **Directories:** For directories, the same formats apply as for files.

4.4.2. Known Limitations

The following limitations are known for drag and drop:

On Windows hosts, dragging and dropping content between UAC-elevated (User Account Control) programs and non-UAC-elevated programs is not allowed. If you start Oracle VM VirtualBox with Administrator privileges then drag and drop will not work with Windows Explorer, which runs with regular user privileges by default.

On Linux hosts and guests, programs can query for drag and drop data while the drag operation is still in progress. For example, on LXDE using the PCManFM file manager. This currently is not supported. As a workaround, a different file manager, such as Nautilus, can be used instead.

4.5. Hardware-Accelerated Graphics

4.5.1. Hardware 3D Acceleration (OpenGL and Direct3D 8/9)

The Oracle VM VirtualBox Guest Additions contain experimental hardware 3D support for Windows, Linux, and Oracle Solaris guests.

With this feature, if an application inside your virtual machine uses 3D features through the OpenGL or Direct3D 8/9 programming interfaces, instead of emulating them in software, which would be slow, Oracle VM VirtualBox will attempt to use your host's 3D hardware. This works for all supported host platforms, provided that your host operating system can make use of your accelerated 3D hardware in the first place.

The 3D acceleration feature currently has the following preconditions:

- It is only available for certain Windows, Linux, and Oracle Solaris guests. In particular:
 - 3D acceleration with Windows guests requires Windows 2000 or later. Apart from on Windows 2000 guests, both OpenGL and Direct3D 8/9 are supported on an experimental basis.
 - OpenGL on Linux requires kernel 2.6.27 or later, as well as X.org server version 1.5 or later. Ubuntu 10.10 and Fedora 14 have been tested and confirmed as working.
 - OpenGL on Oracle Solaris guests requires X.org server version 1.5 or later.
- The Guest Additions must be installed.

Note

For the basic Direct3D acceleration to work in a Windows Guest, Oracle VM VirtualBox needs to replace Windows system files in the virtual machine. As a result, the Guest Additions installation program offers Direct3D acceleration as an option that must be explicitly enabled. Also, you must install the Guest Additions in Safe Mode. This does *not* apply to the WDDM Direct3D video driver available for Windows Vista and later. See [Chapter 14, Known Limitations](#) for details.

- Because 3D support is still experimental at this time, it is disabled by default and must be *manually enabled* in the VM settings. See [Section 3.6, "Display Settings"](#).

Note

Untrusted guest systems should not be allowed to use the 3D acceleration features of Oracle VM VirtualBox, just as untrusted host software should not be allowed to use 3D acceleration. Drivers for 3D hardware are generally too complex to be made properly secure and any software which is allowed to access them may be able to compromise the operating system running them. In addition, enabling 3D acceleration gives the guest direct access to a large body of additional program code in the Oracle VM VirtualBox host process which it might conceivably be able to use to crash the virtual machine.

To enable Aero theme support, the Oracle VM VirtualBox WDDM video driver must be installed, which is available with the Guest Additions installation. The WDDM driver is not installed by default for Vista and Windows 7 guests and must be *manually selected* in the Guest Additions installer by clicking **No** in the **Would You Like to Install Basic Direct3D Support** dialog displayed when the Direct3D feature is selected.

The Aero theme is not enabled by default on Windows. See your Windows platform documentation for details of how to enable the Aero theme.

Technically, Oracle VM VirtualBox implements 3D acceleration by installing an additional hardware 3D driver inside the guest when the Guest Additions are installed. This driver acts as a hardware 3D driver and reports to the guest operating system that the virtual hardware is capable of 3D hardware acceleration. When an application in the guest then requests hardware acceleration through the OpenGL or Direct3D programming interfaces, these are sent to the host through a special communication tunnel implemented by Oracle VM VirtualBox. The *host* then performs the requested 3D operation using the host's programming interfaces.

4.5.2. Hardware 2D Video Acceleration for Windows Guests

The Oracle VM VirtualBox Guest Additions contain experimental hardware 2D video acceleration support for Windows guests.

With this feature, if an application such as a video player inside your Windows VM uses 2D video overlays to play a movie clip, then Oracle VM VirtualBox will attempt to use your host's video acceleration hardware instead of performing overlay stretching and color conversion in software, which would be slow. This currently works for Windows, Linux and macOS host platforms, provided that your host operating system can make use of 2D video acceleration in the first place.

Hardware 2D video acceleration currently has the following preconditions:

- Only available for Windows guests, running Windows XP or later.
- Guest Additions must be installed.
- Because 2D support is still experimental at this time, it is disabled by default and must be *manually enabled* in the VM settings. See [Section 3.6, "Display Settings"](#).

Technically, Oracle VM VirtualBox implements this by exposing video overlay DirectDraw capabilities in the Guest Additions video driver. The driver sends all overlay commands to the host through a special communication tunnel implemented by Oracle VM VirtualBox. On the host side, OpenGL is then used to implement color space transformation and scaling.

4.6. Seamless Windows

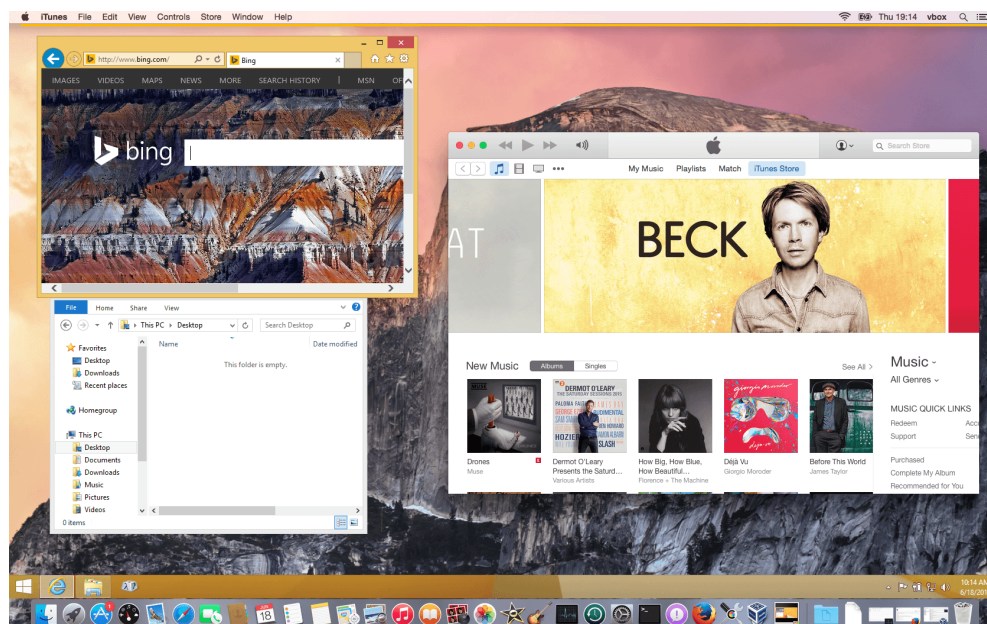
With the *seamless windows* feature of Oracle VM VirtualBox, you can have the windows that are displayed within a virtual machine appear side by side next to the windows of your host. This feature is supported for the following guest operating systems, provided that the Guest Additions are installed:

- Windows guests.

- Supported Linux or Oracle Solaris guests running the X Window System.

After seamless windows are enabled, Oracle VM VirtualBox suppresses the display of the desktop background of your guest, allowing you to run the windows of your guest operating system seamlessly next to the windows of your host.

Figure 4.2. Seamless Windows on a Host Desktop



To enable seamless mode, after starting the virtual machine, press the **Host key + L**. The Host key is normally the right control key. This will enlarge the size of the VM's display to the size of your host screen and mask out the guest operating system's background. To disable seamless windows and go back to the normal VM display, press the Host key + L again.

4.7. Guest Properties

Oracle VM VirtualBox enables requests of some properties from a running guest, provided that the Oracle VM VirtualBox Guest Additions are installed and the VM is running. This provides the following advantages:

- A number of predefined VM characteristics are automatically maintained by Oracle VM VirtualBox and can be retrieved on the host. For example, to monitor VM performance and statistics.
- Arbitrary string data can be exchanged between guest and host. This works in both directions.

To accomplish this, Oracle VM VirtualBox establishes a private communication channel between the Oracle VM VirtualBox Guest Additions and the host, and software on both sides can use this channel to exchange string data for arbitrary purposes. Guest properties are simply string keys to which a value is attached. They can be set, or written to, by either the host and the guest. They can also be read from both sides.

In addition to establishing the general mechanism of reading and writing values, a set of predefined guest properties is automatically maintained by the Oracle VM VirtualBox Guest Additions to allow for retrieving interesting guest data such as the guest's exact operating system and service pack level, the installed version of the Guest Additions, users that are currently logged into the guest OS, network statistics and more. These predefined properties are all prefixed with `/VirtualBox/` and organized into a hierarchical tree of keys.

Some of this runtime information is shown when you select **Session Information Dialog** from a virtual machine's **Machine** menu.

A more flexible way to use this channel is with the **VBoxManage guestproperty** command. See [Section 8.43, “VBoxManage guestproperty”](#). For example, to have *all* the available guest properties for a given running VM listed with their respective values, use this command:

```
$ VBoxManage guestproperty enumerate "Windows Vista III"
VirtualBox Command Line Management Interface Version version-number
Copyright (C) 2005-2023 Oracle and/or its affiliates

Name: /VirtualBox/GuestInfo/OS/Product, value: Windows Vista Business Edition,
    timestamp: 1229098278843087000, flags:
Name: /VirtualBox/GuestInfo/OS/Release, value: 6.0.6001,
    timestamp: 1229098278950553000, flags:
Name: /VirtualBox/GuestInfo/OS/ServicePack, value: 1,
    timestamp: 1229098279122627000, flags:
Name: /VirtualBox/GuestAdd/InstallDir,
    value: C:/Program Files/Oracle/VirtualBox
    Guest Additions, timestamp: 1229098279269739000, flags:
Name: /VirtualBox/GuestAdd/Revision, value: 40720,
    timestamp: 1229098279345664000, flags:
Name: /VirtualBox/GuestAdd/Version, value: version-number,
    timestamp: 1229098279479515000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxControl.exe, value: version-number40720,
    timestamp: 1229098279651731000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxHook.dll, value: version-number40720,
    timestamp: 1229098279804835000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxDisp.dll, value: version-number40720,
    timestamp: 1229098279880611000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxMRXNP.dll, value: version-number40720,
    timestamp: 1229098279882618000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxService.exe, value: version-number40720,
    timestamp: 1229098279883195000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxTray.exe, value: version-number40720,
    timestamp: 1229098279885027000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxGuest.sys, value: version-number40720,
    timestamp: 1229098279886838000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxMouse.sys, value: version-number40720,
    timestamp: 1229098279890600000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxSF.sys, value: version-number40720,
    timestamp: 1229098279893056000, flags:
Name: /VirtualBox/GuestAdd/Components/VBoxVideo.sys, value: version-number40720,
    timestamp: 1229098279895767000, flags:
Name: /VirtualBox/GuestInfo/OS/LoggedInUsers, value: 1,
    timestamp: 1229099826317660000, flags:
Name: /VirtualBox/GuestInfo/OS/NoLoggedInUsers, value: false,
    timestamp: 1229098455580553000, flags:
Name: /VirtualBox/GuestInfo/Net/Count, value: 1,
    timestamp: 1229099826299785000, flags:
Name: /VirtualBox/HostInfo/GUI/LanguageID, value: C,
    timestamp: 1229098151272771000, flags:
Name: /VirtualBox/GuestInfo/Net/0/V4/IP, value: 192.168.2.102,
    timestamp: 1229099826300088000, flags:
Name: /VirtualBox/GuestInfo/Net/0/V4/Broadcast, value: 255.255.255.255,
    timestamp: 1229099826300220000, flags:
Name: /VirtualBox/GuestInfo/Net/0/V4/Netmask, value: 255.255.255.0,
    timestamp: 1229099826300350000, flags:
Name: /VirtualBox/GuestInfo/Net/0/Status, value: Up,
    timestamp: 1229099826300524000, flags:
Name: /VirtualBox/GuestInfo/OS/LoggedInUsersList, value: username,
    timestamp: 1229099826317386000, flags:
```

To query the value of a single property, use the **get** subcommand as follows:

```
$ VBoxManage guestproperty get "Windows Vista III" "/VirtualBox/GuestInfo/OS/Product"
VirtualBox Command Line Management Interface Version version-number
```

Copyright (C) 2005-2023 Oracle and/or its affiliates

Value: Windows Vista Business Edition

To add or change guest properties from the guest, use the tool **VBoxControl**. This tool is included in the Guest Additions. When started from a Linux guest, this tool requires root privileges for security reasons.

```
$ sudo VBoxControl guestproperty enumerate
VirtualBox Guest Additions Command Line Management Interface Version version-number
Copyright (C) 2005-2023 Oracle and/or its affiliates
```

```
Name: /VirtualBox/GuestInfo/OS/Release, value: 2.6.28-18-generic,
      timestamp: 1265813265835667000, flags: <NULL>
Name: /VirtualBox/GuestInfo/OS/Version, value: #59-Ubuntu SMP Thu Jan 28 01:23:03 UTC 2010,
      timestamp: 1265813265836305000, flags: <NULL>
...
```

For more complex needs, you can use the Oracle VM VirtualBox programming interfaces. See [Chapter 11, Oracle VM VirtualBox Programming Interfaces](#).

4.7.1. Using Guest Properties to Wait on VM Events

The properties `/VirtualBox/HostInfo/VBoxVer`, `/VirtualBox/HostInfo/VBoxVerExt` or `/VirtualBox/HostInfo/VBoxRev` can be waited on to detect that the VM state was restored from saved state or snapshot:

```
$ VBoxControl guestproperty wait /VirtualBox/HostInfo/VBoxVer
```

Similarly the `/VirtualBox/HostInfo/ResumeCounter` can be used to detect that a VM was resumed from the paused state or saved state.

4.8. Guest Control File Manager

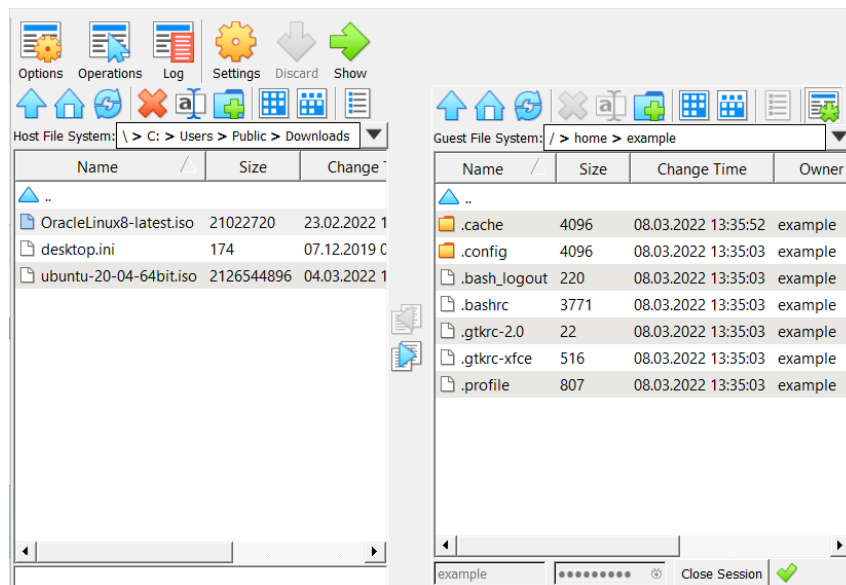
The Guest Control File Manager is a feature of the Guest Additions that enables easy copying and moving of files between a guest and the host system. Other file management operations provide support to create new folders and to rename or delete files.

This feature is useful when the VM window of a guest is not visible. For example, when the guest is running in headless mode.

Note

To use the Guest Control File Manager, the guest must be running. For powered-off guests, it is disabled automatically.

Figure 4.3. Guest Control File Manager



The Guest Control File Manager works by mounting the host file system. Guest users must authenticate and create a guest session before they can transfer files.

4.8.1. Using the Guest Control File Manager

The following steps describe how to use the Guest Control File Manager.

1. Open the Guest Control File Manager. Do either of the following:
 - In the guest VM, select **Machine, File Manager**.
 - In VirtualBox Manager, click on the machine name. Click **File Manager** in the machine tools menu for the VM.

The left pane shows the files on the host system.

2. Create a guest session.

At the bottom of the Guest Control File Manager, enter authentication credentials for a user on the guest system.

Click **Create Session**.

The contents of the guest VM file system appears in the right pane of the Guest Control File Manager.

3. Transfer files between the guest and the host system by using the move and copy file transfer icons.

You can copy and move files from the guest to the host system or from the host system to the guest.

4. Close the Guest Control File Manager.

Click **Close** to end the guest session.

4.9. Guest Control of Applications

The Guest Additions enable starting of applications inside a guest VM from the host system. This feature can be used to automate deployment of software within the guest.

For this to work, the application needs to be installed on the guest. No additional software needs to be installed

on the host. Additionally, text mode output to stdout and stderr can be shown on the host for further processing. There are options to specify user credentials and a timeout value, in milliseconds, to limit the time the application is able to run.

The Guest Additions for Windows allow for automatic updating. This applies for already installed Guest Additions versions. Also, copying files from host to the guest as well as remotely creating guest directories is available.

To use these features, use the Oracle VM VirtualBox command line. See [Section 8.44, “VBoxManage guestcontrol”](#).

4.10. Memory Overcommitment

In server environments with many VMs, the Guest Additions can be used to share physical host memory between several VMs. This reduces the total amount of memory in use by the VMs. If memory usage is the limiting factor and CPU resources are still available, this can help with running more VMs on each host.

4.10.1. Memory Ballooning

The Guest Additions can change the amount of host memory that a VM uses, while the machine is running. Because of how this is implemented, this feature is called *memory ballooning*.

Note

- Oracle VM VirtualBox supports memory ballooning only on 64-bit hosts. It is not supported on macOS hosts.
- Memory ballooning does not work well with large pages enabled. To turn off large pages support for a VM, run **VBoxManage modifyvm *vmname* --large-pages off**

Normally, to change the amount of memory allocated to a virtual machine, you have to shut down the virtual machine entirely and modify its settings. With memory ballooning, memory that was allocated for a virtual machine can be given to another virtual machine without having to shut the machine down.

When memory ballooning is requested, the Oracle VM VirtualBox Guest Additions, which run inside the guest, allocate physical memory from the guest operating system on the kernel level and lock this memory down in the guest. This ensures that the guest will not use that memory any longer. No guest applications can allocate it, and the guest kernel will not use it either. Oracle VM VirtualBox can then reuse this memory and give it to another virtual machine.

The memory made available through the ballooning mechanism is only available for reuse by Oracle VM VirtualBox. It is *not* returned as free memory to the host. Requesting balloon memory from a running guest will therefore not increase the amount of free, unallocated memory on the host. Effectively, memory ballooning is therefore a memory overcommitment mechanism for multiple virtual machines while they are running. This can be useful to temporarily start another machine, or in more complicated environments, for sophisticated memory management of many virtual machines that may be running in parallel depending on how memory is used by the guests.

At this time, memory ballooning is only supported through **VBoxManage**. Use the following command to increase or decrease the size of the memory balloon within a running virtual machine that has Guest Additions installed:

```
VBoxManage controlvm "VM name" guestmemoryballoon n
```

where *VM name* is the name or UUID of the virtual machine in question and *n* is the amount of memory to allocate from the guest in megabytes. See [Section 8.20, “VBoxManage controlvm”](#).

You can also set a default balloon that will automatically be requested from the VM every time after it has started up with the following command:

```
VBoxManage modifyvm "VM name" --guest-memory-balloon n
```

By default, no balloon memory is allocated. This is a VM setting, like other **modifyvm** settings, and therefore can only be set while the machine is shut down. See [Section 8.10, “VBoxManage modifyvm”](#).

4.10.2. Page Fusion

Whereas memory ballooning simply reduces the amount of RAM that is available to a VM, Page Fusion works differently. It avoids memory duplication between several similar running VMs.

In a server environment running several similar VMs on the same host, lots of memory pages are identical. For example, if the VMs are using identical operating systems. Oracle VM VirtualBox's Page Fusion technology can efficiently identify these identical memory pages and share them between multiple VMs.

Note

Oracle VM VirtualBox supports Page Fusion only on 64-bit hosts, and it is not supported on macOS hosts. Page Fusion currently works only with Windows 2000 and later guests.

The more similar the VMs on a given host are, the more efficiently Page Fusion can reduce the amount of host memory that is in use. It therefore works best if all VMs on a host run identical operating systems. Instead of having a complete copy of each operating system in each VM, Page Fusion identifies the identical memory pages in use by these operating systems and eliminates the duplicates, sharing host memory between several machines. This is called *deduplication*. If a VM tries to modify a page that has been shared with other VMs, a new page is allocated again for that VM with a copy of the shared page. This is called *copy on write*. All this is fully transparent to the virtual machine.

You may be familiar with this kind of memory overcommitment from other hypervisor products, which call this feature *page sharing* or *same page merging*. However, Page Fusion differs significantly from those other solutions, whose approaches have several drawbacks:

- Traditional hypervisors scan *all* guest memory and compute checksums, also called hashes, for every single memory page. Then, they look for pages with identical hashes and compare the entire content of those pages. If two pages produce the same hash, it is very likely that the pages are identical in content. This process can take rather long, especially if the system is not idling. As a result, the additional memory only becomes available after a significant amount of time, such as hours or sometimes days. Even worse, this kind of page sharing algorithm generally consumes significant CPU resources and increases the virtualization overhead by 10 to 20%.

Page Fusion in Oracle VM VirtualBox uses logic in the Oracle VM VirtualBox Guest Additions to quickly identify memory cells that are most likely identical across VMs. It can therefore achieve most of the possible savings of page sharing almost immediately and with almost no overhead.

- Page Fusion is also much less likely to be confused by identical memory that it will eliminate, just to learn seconds later that the memory will now change and having to perform a highly expensive and often service-disrupting reallocation.

At this time, Page Fusion can only be controlled with **VBoxManage**, and only while a VM is shut down. To enable Page Fusion for a VM, use the following command:

```
VBoxManage modifyvm "VM name" --page-fusion on
```

You can observe Page Fusion operation using some metrics. `RAM/VMM/Shared` shows the total amount of fused pages, whereas the per-VM metric `Guest/RAM/Usage/Shared` will return the amount of fused memory for a given VM.

See [Section 8.46, “VBoxManage metrics”](#) for information on how to query metrics.

Note

Enabling Page Fusion might indirectly increase the chances for malicious guests to successfully attack other VMs running on the same host. See [Section 13.3.4, “Potentially Insecure Operations”](#).

4.11. Controlling Virtual Monitor Topology

4.11.1. X11/Wayland Desktop Environments

The Guest Additions provide services for controlling the guest system's monitor topology. Monitor topology means the resolution of each virtual monitor and its state (disabled/enabled). The resolution of a virtual monitor can be modified from the host side either by resizing the window that hosts the virtual monitor, by using the **View** menu or the **VBoxManage controlvm *vmname* setscreenlayout** command. On guest operating systems with X11/Wayland desktops this is put into effect by either of the following two services:

```
VBoxClient --vmsvga  
VBoxDRMClient
```

The following are some details about guest screen resolution control functionality:

- On X11/Wayland desktops the resizing service is started during desktop session initialization, that is desktop login. On X11 desktops `VBoxClient --vmsvga` handles screen topology through the RandR extension. On Wayland clients `VBoxDRMClient` is used. The decision is made automatically at each desktop session start.
- On 32-bit guest operating systems **VBoxDRMClient** is always used, in order to work around bugs.
- Since the monitor topology control services are initialized during the desktop session start, it is impossible to control the monitor resolution of display managers such as GDM or LightDM. This default behavior can be changed by setting the guest property `/VirtualBox/GuestAdd/DRMResize` of the virtual machine to any value. See [Section 4.7, “Guest Properties”](#) for details of how to update guest properties. When this guest property is set then **VBoxDRMClient** is started during the guest OS boot and stays active all the time, for both the display manager login screen and the desktop session.

4.11.1.1. Known Limitations

VBoxDRMClient is not able to handle arbitrary guest monitor topologies. Specifically, disabling a guest monitor that is not the last one invalidates the monitor topology due to limitations in the `vmwgfx.ko` Linux kernel module. For example, when the guest is configured to have four monitors it is not recommended to disable the second or third monitor.