

Chapter 2. Installation Details

Table of Contents

[2.1. Installing on Windows Hosts](#)

[2.1.1. Prerequisites](#)

[2.1.2. Performing the Installation](#)

[2.1.3. Uninstallation](#)

[2.1.4. Unattended Installation](#)

[2.1.5. Public Properties](#)

[2.2. Installing on macOS Hosts](#)

[2.2.1. Performing the Installation](#)

[2.2.2. Uninstallation](#)

[2.2.3. Unattended Installation](#)

[2.3. Installing on Linux Hosts](#)

[2.3.1. Prerequisites](#)

[2.3.2. The Oracle VM VirtualBox Kernel Modules](#)

[2.3.3. Performing the Installation](#)

[2.3.4. The vboxusers Group](#)

[2.3.5. Starting Oracle VM VirtualBox on Linux](#)

[2.4. Installing on Oracle Solaris Hosts](#)

[2.4.1. Performing the Installation](#)

[2.4.2. The vboxuser Group](#)

[2.4.3. Starting Oracle VM VirtualBox on Oracle Solaris](#)

[2.4.4. Uninstallation](#)

[2.4.5. Unattended Installation](#)

[2.4.6. Configuring a Zone for Running Oracle VM VirtualBox](#)

[2.5. Installing an Extension Pack](#)

[2.5.1. The Extension Pack Manager](#)

As installation of Oracle VM VirtualBox varies depending on your host operating system, the following sections provide installation instructions for Windows, macOS, Linux, and Oracle Solaris.

2.1. Installing on Windows Hosts

2.1.1. Prerequisites

For the various versions of Windows that are supported as host operating systems, please refer to [Section 1.4, "Supported Host Operating Systems"](#).

In addition, Windows Installer must be present on your system. This should be the case for all supported Windows platforms.

2.1.2. Performing the Installation

The Oracle VM VirtualBox installation can be started in either of the following ways:

- By double-clicking on the executable file.
- By entering the following command:

```
VirtualBox-<version>-<revision>-Win.exe -extract
```

This will extract the installer into a temporary directory, along with the .MSI file. Run the following

command to perform the installation:

```
msiexec /i VirtualBox-<version>-<revision>-Win.msi
```

Using either way displays the installation **Welcome** dialog and enables you to choose where to install Oracle VM VirtualBox, and which components to install. In addition to the Oracle VM VirtualBox application, the following components are available:

- **USB support.** This package contains special drivers for your Windows host that Oracle VM VirtualBox requires to fully support USB devices inside your virtual machines.
- **Networking.** This package contains extra networking drivers for your Windows host that Oracle VM VirtualBox needs to support Bridged Networking. This enables your VM's virtual network cards to be accessed from other machines on your physical network.
- **Python support.** This package contains Python scripting support for the Oracle VM VirtualBox API, see [Chapter 11, Oracle VM VirtualBox Programming Interfaces](#). For this to work, an already working Windows Python installation on the system is required.

See, for example: <http://www.python.org/download/windows/>.

Note

Python version at least 2.6 is required. Python 3 is also supported.

Depending on your Windows configuration, you may see warnings about unsigned drivers, or similar. Click **Continue** for these warnings, as otherwise Oracle VM VirtualBox might not function correctly after installation.

The installer will create an Oracle VM VirtualBox group in the Windows **Start** menu, which enables you to launch the application and access its documentation.

With standard settings, Oracle VM VirtualBox will be installed for all users on the local system. If this is not wanted, you must invoke the installer by first extracting as follows:

```
VirtualBox.exe -extract
```

Then, run either of the following commands on the extracted .MSI file. This will install Oracle VM VirtualBox only for the current user.

```
VirtualBox.exe -msiparams ALLUSERS=2
```

```
msiexec /i VirtualBox-<version>-Win.msi ALLUSERS=2
```

If you do not want to install all features of Oracle VM VirtualBox, you can set the optional `ADDLOCAL` parameter to explicitly name the features to be installed. The following features are available:

VBoxApplication

Main binaries of Oracle VM VirtualBox.

Note

This feature must not be absent, since it contains the minimum set of files to have working Oracle VM VirtualBox installation.

VBoxUSB

USB support.

VBoxNetwork

All networking support. This includes the VBoxNetworkFlt and VBoxNetworkAdp features.

VBoxNetworkFlt

Bridged networking support.

VBoxNetworkAdp

Host-only networking support

VBoxPython

Python support

For example, to only install USB support along with the main binaries, run either of the following commands:

```
VirtualBox.exe -msiparams ADDLOCAL=VBoxApplication,VBoxUSB
```

```
msiexec /i VirtualBox-<version>-Win.msi ADDLOCAL=VBoxApplication,VBoxUSB
```

The user is able to choose between NDIS5 and NDIS6 host network filter drivers during the installation. This is done using a command line parameter, `NETWORKTYPE`. The NDIS6 driver is the default for most supported Windows hosts. For some legacy Windows versions, the installer will automatically select the NDIS5 driver and this cannot be changed.

You can force an install of the legacy NDIS5 host network filter driver by specifying `NETWORKTYPE=NDIS5`. For example, to install the NDIS5 driver on Windows 7 use either of the following commands:

```
VirtualBox.exe -msiparams NETWORKTYPE=NDIS5
```

```
msiexec /i VirtualBox-<version>-Win;.msi NETWORKTYPE=NDIS5
```

2.1.3. Uninstallation

As Oracle VM VirtualBox uses the standard Microsoft Windows installer, Oracle VM VirtualBox can be safely uninstalled at any time. Click the program entry in the **Add/Remove Programs** list in the Windows Control Panel.

2.1.4. Unattended Installation

Unattended installations can be performed using the standard MSI support.

2.1.5. Public Properties

Public properties can be specified with the MSI API, to control additional behavior and features of the Windows host installer. Use either of the following commands:

```
VirtualBox.exe -msiparams NAME=VALUE [...]
```

```
msiexec /i VirtualBox-<version>-Win.msi NAME=VALUE [...]
```

The following public properties are available.

- **VBOX_INSTALLDESKTOPSHORTCUT**

Specifies whether or not an Oracle VM VirtualBox icon on the desktop should be created.

Set to 1 to enable, 0 to disable. Default is 1.

- **VBOX_INSTALLQUICKLAUNCHSHORTCUT**

Specifies whether or not an Oracle VM VirtualBox icon in the Quick Launch Bar should be created.

Set to 1 to enable, 0 to disable. Default is 1.

- **VBOX_REGISTERFILEEXTENSIONS**

Specifies whether or not the file extensions .vbox, .vbox-extpack, .ovf, .ova, .vdi, .vmdk, .vhd and .vdd should be associated with Oracle VM VirtualBox. Files of these types then will be opened with Oracle VM VirtualBox.

Set to 1 to enable, 0 to disable. Default is 1.

- **VBOX_START**

Specifies whether to start Oracle VM VirtualBox right after successful installation.

Set to 1 to enable, 0 to disable. Default is 1.

2.2. Installing on macOS Hosts

2.2.1. Performing the Installation

For macOS hosts, Oracle VM VirtualBox ships in a `dmg` disk image file. Perform the following steps to install on a macOS host:

1. Double-click on the `dmg` file, to mount the contents.
2. A window opens, prompting you to double-click on the `VirtualBox.pkg` installer file displayed in that window.
3. This starts the installer, which enables you to select where to install Oracle VM VirtualBox.
4. An Oracle VM VirtualBox icon is added to the `Applications` folder in the Finder.

2.2.2. Uninstallation

To uninstall Oracle VM VirtualBox, open the disk image `dmg` file and double-click on the uninstall icon shown.

2.2.3. Unattended Installation

To perform a non-interactive installation of Oracle VM VirtualBox you can use the command line version of the installer application.

Mount the `dmg` disk image file, as described in the installation procedure, or use the following command line:

```
hdiutil attach /path/to/VirtualBox-xyz.dmg
```

Open a terminal session and run the following command:

```
sudo installer -pkg /Volumes/VirtualBox/VirtualBox.pkg -target /Volumes/Macintosh\ HD
```

2.3. Installing on Linux Hosts

2.3.1. Prerequisites

For the various versions of Linux that are supported as host operating systems, see [Section 1.4, “Supported Host Operating Systems”](#).

You may need to install the following packages on your Linux system before starting the installation. Some

systems will do this for you automatically when you install Oracle VM VirtualBox.

- Qt 5.3.2 or later. Qt 5.6.2 or later is recommended.
- SDL 1.2.7 or later. This graphics library is typically called `libsdl` or similar.

Note

These packages are only required if you want to run the Oracle VM VirtualBox graphical user interfaces. In particular, **VirtualBox**, the graphical VirtualBox Manager, requires both Qt and SDL. If you only want to run **VBoxHeadless**, neither Qt nor SDL are required.

2.3.2. The Oracle VM VirtualBox Kernel Modules

In order to run other operating systems in virtual machines alongside your main operating system, Oracle VM VirtualBox needs to integrate very tightly with your system. To do this it installs a driver module called **vboxdrv** into the system kernel. The kernel is the part of the operating system which controls your processor and physical hardware. Without this kernel module, you can still use VirtualBox Manager to configure virtual machines, but they will not start.

Network drivers called **vboxnetflt** and **vboxnetadp** are also installed. They enable virtual machines to make more use of your computer's network capabilities and are needed for any virtual machine networking beyond the basic NAT mode.

Since distributing driver modules separately from the kernel is not something which Linux supports well, the Oracle VM VirtualBox install process creates the modules on the system where they will be used. This means that you may need to install some software packages from the distribution which are needed for the build process. Required packages may include the following:

- GNU compiler (GCC)
- GNU Make (make)
- Kernel header files

Also ensure that all system updates have been installed and that your system is running the most up-to-date kernel for the distribution.

Note

The running kernel and the kernel header files must be updated to matching versions.

The following list includes some details of the required files for some common distributions. Start by finding the version name of your kernel, using the command **uname -r** in a terminal. The list assumes that you have not changed too much from the original installation, in particular that you have not installed a different kernel type.

- With Debian and Ubuntu-based distributions, you must install the correct version of the `linux-headers`, usually whichever of `linux-headers-generic`, `linux-headers-amd64`, `linux-headers-i686` or `linux-headers-i686-pae` best matches the kernel version name. Also, the `linux-kbuild` package if it exists. Basic Ubuntu releases should have the correct packages installed by default.
- On Fedora, Red Hat, Oracle Linux and many other RPM-based systems, the kernel version sometimes has a code of letters or a word close to the end of the version name. For example "uek" for the Oracle Unbreakable Enterprise Kernel or "default" or "desktop" for the standard kernels. In this case, the package name is `kernel-uek-devel` or equivalent. If there is no such code, it is usually `kernel-devel`.
- On some SUSE and openSUSE Linux versions, you may need to install the `kernel-source` and `kernel-syms`

packages.

If you suspect that something has gone wrong with module installation, check that your system is set up as described above and try running the following command, as root:

```
rcvboxdrv setup
```

2.3.2.1. Kernel Modules and UEFI Secure Boot

If you are running on a system using UEFI (Unified Extensible Firmware Interface) Secure Boot, you may need to sign the following kernel modules before you can load them:

- **vboxdrv**
- **vboxnetadp**
- **vboxnetflt**
- **vboxpci**

See your system documentation for details of the kernel module signing process.

2.3.3. Performing the Installation

Oracle VM VirtualBox is available in a number of package formats native to various common Linux distributions. See [Section 1.4, “Supported Host Operating Systems”](#). In addition, there is an alternative generic installer (.run) which you can use on supported Linux distributions.

2.3.3.1. Installing Oracle VM VirtualBox from a Debian or Ubuntu Package

Download the appropriate package for your distribution. The following example assumes that you are installing to a 64-bit Ubuntu Xenial system. Use **dpkg** to install the Debian package, as follows:

```
sudo dpkg -i virtualbox-version-number_Ubuntu_xenial_amd64.deb
```

The installer will also try to build kernel modules suitable for the current running kernel. If the build process is not successful you will be shown a warning and the package will be left unconfigured. Look at `/var/log/vbox-install.log` to find out why the compilation failed. You may have to install the appropriate Linux kernel headers, see [Section 2.3.2, “The Oracle VM VirtualBox Kernel Modules”](#). After correcting any problems, run the following command:

```
sudo rcvboxdrv setup
```

This will start a second attempt to build the module.

If a suitable kernel module was found in the package or the module was successfully built, the installation script will attempt to load that module. If this fails, please see [Section 12.7.1, “Linux Kernel Module Refuses to Load”](#) for further information.

Once Oracle VM VirtualBox has been successfully installed and configured, you can start it by clicking **VirtualBox** in your **Start** menu or from the command line. See [Section 2.3.5, “Starting Oracle VM VirtualBox on Linux”](#).

2.3.3.2. Using the Alternative Generic Installer (VirtualBox.run)

The alternative generic installer performs the following steps:

- Unpacks the application files to the target directory `/opt/VirtualBox/`, which cannot be changed.

- Builds and installs the Oracle VM VirtualBox kernel modules: **vboxdrv**, **vboxnetflt**, and **vboxnetadp**.
- Creates `/sbin/rcvboxdrv`, an init script to start the Oracle VM VirtualBox kernel module.
- Creates a new system group called `vboxusers`.
- Creates symbolic links in `/usr/bin` to a shell script `/opt/VirtualBox/VBox` which does some sanity checks and dispatches to the actual executables: **VirtualBox**, **VBoxVRDP**, **VBoxHeadless** and **VBoxManage**.
- Creates `/etc/udev/rules.d/60-vboxdrv.rules`, a description file for udev, if that is present, which makes the USB devices accessible to all users in the `vboxusers` group.
- Writes the installation directory to `/etc/vbox/vbox.cfg`.

The installer must be executed as root with either `install` or `uninstall` as the first parameter. For example:

```
sudo ./VirtualBox.run install
```

Or if you do not have the **sudo** command available, run the following as root instead:

```
./VirtualBox.run install
```

Add every user who needs to access USB devices from a VirtualBox guests to the group `vboxusers`. Either use the OS user management tools or run the following command as root:

```
sudo usermod -a -G vboxusers username
```

Note

The **usermod** command of some older Linux distributions does not support the `-a` option, which adds the user to the given group without affecting membership of other groups. In this case, find out the current group memberships with the **groups** command and add all these groups in a comma-separated list to the command line after the `-G` option. For example:
usermod -G group1,group2,vboxusers username.

2.3.3.3. Performing a Manual Installation

If you cannot use the shell script installer described in [Section 2.3.3.2, “Using the Alternative Generic Installer \(VirtualBox.run\)”](#), you can perform a manual installation. Run the installer as follows:

```
./VirtualBox.run --keep --noexec
```

This will unpack all the files needed for installation in the directory `install` under the current directory. The Oracle VM VirtualBox application files are contained in `VirtualBox.tar.bz2` which you can unpack to any directory on your system. For example:

```
sudo mkdir /opt/VirtualBox
sudo tar jxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
```

To run the same example as root, use the following commands:

```
mkdir /opt/VirtualBox
tar jxf ./install/VirtualBox.tar.bz2 -C /opt/VirtualBox
```

The sources for Oracle VM VirtualBox's kernel module are provided in the `src` directory. To build the module, change to the directory and use the following command:

```
make
```

If everything builds correctly, run the following command to install the module to the appropriate module directory:

```
sudo make install
```

In case you do not have `sudo`, switch the user account to root and run the following command:

```
make install
```

The Oracle VM VirtualBox kernel module needs a device node to operate. The above **make** command will tell you how to create the device node, depending on your Linux system. The procedure is slightly different for a classical Linux setup with a `/dev` directory, a system with the now deprecated **devfs** and a modern Linux system with **udev**.

On certain Linux distributions, you might experience difficulties building the module. You will have to analyze the error messages from the build system to diagnose the cause of the problems. In general, make sure that the correct Linux kernel sources are used for the build process.

Note that the `/dev/vboxdrv` kernel module device node must be owned by `root:root` and must be read/writable only for the user.

Next, you install the system initialization script for the kernel module and activate the initialization script using the right method for your distribution, as follows:

```
cp /opt/VirtualBox/vboxdrv.sh /sbin/rcvboxdrv
```

This example assumes you installed Oracle VM VirtualBox to the `/opt/VirtualBox` directory.

Create a configuration file for Oracle VM VirtualBox, as follows:

```
mkdir /etc/vbox
echo INSTALL_DIR=/opt/VirtualBox > /etc/vbox/vbox.cfg
```

Create the following symbolic links:

```
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VirtualBox
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxManage
ln -sf /opt/VirtualBox/VBox.sh /usr/bin/VBoxHeadless
```

2.3.3.4. Updating and Uninstalling Oracle VM VirtualBox

Before updating or uninstalling Oracle VM VirtualBox, you must terminate any virtual machines which are currently running and exit the Oracle VM VirtualBox or VBoxSVC applications. To update Oracle VM VirtualBox, simply run the installer of the updated version. To uninstall Oracle VM VirtualBox, run the installer as follows:

```
sudo ./VirtualBox.run uninstall
```

As root, you can use the following command:

```
./VirtualBox.run uninstall
```

You can uninstall the `.run` package as follows:

```
/opt/VirtualBox/uninstall.sh
```

To manually uninstall Oracle VM VirtualBox, perform the manual installation steps in reverse order.

2.3.3.5. Automatic Installation of Debian Packages

The Debian packages will request some user feedback when installed for the first time. The `debconf` system is used to perform this task. To prevent any user interaction during installation, default values can be defined. A file `vboxconf` can contain the following `debconf` settings:


```
virtualbox virtualbox/module-compilation-allowed boolean true
virtualbox virtualbox/delete-old-modules boolean true
```

The first line enables compilation of the vboxdrv kernel module if no module was found for the current kernel. The second line enables the package to delete any old vboxdrv kernel modules compiled by previous installations.

These default settings can be applied prior to the installation of the Oracle VM VirtualBox Debian package, as follows:

```
debconf-set-selections vboxconf
```

In addition there are some common configuration options that can be set prior to the installation. See [Section 2.3.3.7, “Automatic Installation Options”](#).

2.3.3.6. Automatic Installation of RPM Packages

The RPM format does not provide a configuration system comparable to the debconf system. See [Section 2.3.3.7, “Automatic Installation Options”](#) for how to set some common installation options provided by Oracle VM VirtualBox.

2.3.3.7. Automatic Installation Options

To configure the installation process for .deb and .rpm packages, you can create a response file named `/etc/default/virtualbox`. The automatic generation of the udev rule can be prevented with the following setting:

```
INSTALL_NO_UDEV=1
```

The creation of the group vboxusers can be prevented as follows:

```
INSTALL_NO_GROUP=1
```

If the following line is specified, the package installer will not try to build the **vboxdrv** kernel module if no module fitting the current kernel was found.

```
INSTALL_NO_VBOXDRV=1
```

2.3.4. The vboxusers Group

The Linux installers create the system user group `vboxusers` during installation. Any system user who is going to use USB devices from Oracle VM VirtualBox guests must be a member of that group. A user can be made a member of the group `vboxusers` either by using the desktop user and group tools, or with the following command:

```
sudo usermod -a -G vboxusers username
```

2.3.5. Starting Oracle VM VirtualBox on Linux

The easiest way to start an Oracle VM VirtualBox program is by running the program of your choice (**VirtualBox**, **VBoxManage**, or **VBoxHeadless**) from a terminal. These are symbolic links to **VBox.sh** that start the required program for you.

The following detailed instructions should only be of interest if you wish to execute Oracle VM VirtualBox without installing it first. You should start by compiling the **vboxdrv** kernel module and inserting it into the Linux kernel. Oracle VM VirtualBox consists of a service daemon, **VBoxSVC**, and several application programs. The daemon is automatically started if necessary. All Oracle VM VirtualBox applications will communicate with the daemon through UNIX local domain sockets. There can be multiple daemon instances under different user accounts and applications can only communicate with the daemon running under the user account as the application. The local domain socket resides in a subdirectory of your system's directory for temporary files called `.vbox-`

<username>-ipc. In case of communication problems or server startup problems, you may try to remove this directory.

All Oracle VM VirtualBox applications (**VirtualBox**, **VBoxManage**, and **VBoxHeadless**) require the Oracle VM VirtualBox directory to be in the library path, as follows:

```
LD_LIBRARY_PATH=. ./VBoxManage showvminfo "Windows XP"
```

2.4. Installing on Oracle Solaris Hosts

For the specific versions of Oracle Solaris that are supported as host operating systems, see [Section 1.4, “Supported Host Operating Systems”](#).

If you have a previously installed instance of Oracle VM VirtualBox on your Oracle Solaris host, please uninstall it first before installing a new instance. See [Section 2.4.4, “Uninstallation”](#) for uninstall instructions.

2.4.1. Performing the Installation

Oracle VM VirtualBox is available as a standard Oracle Solaris package. Download the Oracle VM VirtualBox SunOS package, which includes the 64-bit version of Oracle VM VirtualBox. *The installation must be performed as root and from the global zone.* This is because the Oracle VM VirtualBox installer loads kernel drivers, which cannot be done from non-global zones. To verify which zone you are currently in, execute the **zonename** command.

To start installation, run the following commands:

```
gunzip -cd VirtualBox-version-number-SunOS.tar.gz | tar xvf -
```

The Oracle VM VirtualBox kernel package is integrated into the main package. Install the Oracle VM VirtualBox package as follows:

```
pkgadd -d VirtualBox-version-number-SunOS.pkg
```

The installer will then prompt you to enter the package you wish to install. Choose **1** or **all** and proceed. Next the installer will ask you if you want to allow the postinstall script to be executed. Choose **y** and proceed, as it is essential to execute this script which installs the Oracle VM VirtualBox kernel module. Following this confirmation the installer will install Oracle VM VirtualBox and execute the postinstall setup script.

Once the postinstall script has been executed your installation is now complete. You may now safely delete the uncompressed package and `autoresponse` files from your system. Oracle VM VirtualBox is installed in `/opt/VirtualBox`.

Note

If you need to use Oracle VM VirtualBox from non-global zones, see [Section 2.4.6, “Configuring a Zone for Running Oracle VM VirtualBox”](#).

2.4.2. The vboxuser Group

The installer creates the system user group `vboxuser` during installation for Oracle Solaris hosts that support the USB features required by Oracle VM VirtualBox. Any system user who is going to use USB devices from Oracle VM VirtualBox guests must be a member of this group. A user can be made a member of this group either by using the desktop user and group tools or by running the following command as root:

```
usermod -G vboxuser username
```

Note that adding an active user to the `vboxuser` group will require the user to log out and then log in again. This

should be done manually after successful installation of the package.

2.4.3. Starting Oracle VM VirtualBox on Oracle Solaris

The easiest way to start an Oracle VM VirtualBox program is by running the program of your choice (**VirtualBox**, **VBoxManage**, or **VBoxHeadless**) from a terminal. These are symbolic links to **VBox.sh** that start the required program for you.

Alternatively, you can directly invoke the required programs from `/opt/VirtualBox`. Using the links provided is easier as you do not have to enter the full path.

You can configure some elements of the **VirtualBox** Qt GUI, such as fonts and colours, by running **VBoxQtconfig** from the terminal.

2.4.4. Uninstallation

Uninstallation of Oracle VM VirtualBox on Oracle Solaris requires root permissions. To perform the uninstallation, start a root terminal session and run the following command:

```
pkgrm SUNWvbox
```

After confirmation, this will remove Oracle VM VirtualBox from your system.

2.4.5. Unattended Installation

To perform a non-interactive installation of Oracle VM VirtualBox there is a response file named `autoresponse`. The installer uses this for responses to inputs, rather than prompting the user.

Extract the `tar.gz` package as described in [Section 2.4.1, “Performing the Installation”](#). Then open a root terminal session and run the following command:

```
pkgadd -d VirtualBox-version-number-SunOS-x86 -n -a autoresponse SUNWvbox
```

To perform a non-interactive uninstallation, open a root terminal session and run the following command:

```
pkgrm -n -a /opt/VirtualBox/autoresponse SUNWvbox
```

2.4.6. Configuring a Zone for Running Oracle VM VirtualBox

Assuming that Oracle VM VirtualBox has already been installed into your zone, you need to give the zone access to Oracle VM VirtualBox's device node. This is done by performing the following steps. Start a root terminal and run the following command:

```
zonecfg -z vboxzone
```

Replace `vboxzone` with the name of the zone where you intend to run Oracle VM VirtualBox.

Use **zonecfg** to add the `device` resource and `match` properties to the zone, as follows:

```
zonecfg:vboxzone>add device
zonecfg:vboxzone:device>set match=/dev/vboxdrv
zonecfg:vboxzone:device>end
zonecfg:vboxzone>add device
zonecfg:vboxzone:device>set match=/dev/vboxdrvu
zonecfg:vboxzone:device>end
zonecfg:vboxzone>exit
```

On Oracle Solaris 11 or later, you may also add a device for `/dev/vboxusbmon`, similar to that shown above.

If you are not using sparse root zones, you will need to loopback mount `/opt/VirtualBox` from the global zone into the non-global zone at the same path. This is specified below using the `dir` attribute and the `special` attribute. For example:

```
zonecfg:vboxzone>add fs
zonecfg:vboxzone:device>set dir=/opt/VirtualBox
zonecfg:vboxzone:device>set special=/opt/VirtualBox
zonecfg:vboxzone:device>set type=lofs
zonecfg:vboxzone:device>end
zonecfg:vboxzone>exit
```

Reboot the zone using **zoneadm** and you should be able to run Oracle VM VirtualBox from within the configured zone.

2.5. Installing an Extension Pack

Extension packs provide extra functionality to the Oracle VM VirtualBox base package, such as extended USB device support and cloud integration features. See [Section 1.5, “Installing Oracle VM VirtualBox and Extension Packs”](#).

To install an Oracle VM VirtualBox extension pack, do the following:

1. Double-click on the extension package file name.

Oracle VM VirtualBox extension packs have a `.vbox-extpack` file name extension.

2. Follow the on-screen instructions to install the extension pack.

You can also use the Extension Pack Manager tool to install an extension pack. See [Section 2.5.1, “The Extension Pack Manager”](#).

2.5.1. The Extension Pack Manager

Extension packs can be installed and managed using the **Extension Pack Manager** tool in VirtualBox Manager.

The Extension Pack Manager lists the extension packs that are currently installed on the host, and enables you to install and uninstall extension packs.

To display the Extension Pack Manager, go to the global **Tools** menu and click **Extensions**. The Extension Pack Manager is shown.

To install an extension pack using the Extension Pack Manager, click **Install** and select an extension package file. The extension pack is installed on the host and listed in Extension Pack Manager.

To uninstall an extension pack with the Extension Pack Manager, do the following:

1. Select the extension pack in the Extension Pack Manager window and click **Uninstall**.
2. Click **Remove** in the prompt dialog.

The extension pack is uninstalled from the host and removed from the Extension Pack Manager.

Alternatively, you can use the **VBoxManage** command line to install and manage Oracle VM VirtualBox extension packs. See [Section 8.52, “VBoxManage extpack”](#).