

**Oracle® VM VirtualBox
Installation Instructions for Windows 7
and
Linux Virtual Machine Creation
Targeting Avnet Development Boards**

**Version 1.4
April 2017**

Installing Oracle VM VirtualBox on Windows 7

This document shows how to install VirtualBox and the Extension Pack to enable the use of a Linux virtual machine for the cross build platform.

General Instruction:

Install Oracle VM VirtualBox using the official VirtualBox installer. For legal distribution reasons, the VirtualBox installation executable cannot be included with any public Avnet materials. To obtain a free legal copy of the Oracle VM VirtualBox and the Extension Pack, please download from the VirtualBox website:

<https://www.virtualbox.org/wiki/Downloads>

The version downloaded may differ from the version shown in this documentation (5.1.2). Be sure to read the VirtualBox EULA to ensure you do not violate the *Personal Use and Evaluation License* (PUEL). You may also wish to consult the *VirtualBox Licensing Frequently Asked Questions* for a quick overview of the intent of the license agreements:

https://www.virtualbox.org/wiki/Licensing_FAQ


Step-by-Step Instructions:

1. To obtain a free legal copy of Oracle VM VirtualBox, download the installer from this website:

<https://www.virtualbox.org/wiki/Downloads>

The version downloaded may differ from the version shown in this documentation (5.1.18). You also need to download the Extension Pack. Make sure the Extension Pack you download is the same version as your VirtualBox installer.

2. Launch the VirtualBox installer from Windows Explorer by double-clicking the self-extracting executable. Allow the installer to make changes to your computer, if so prompted.

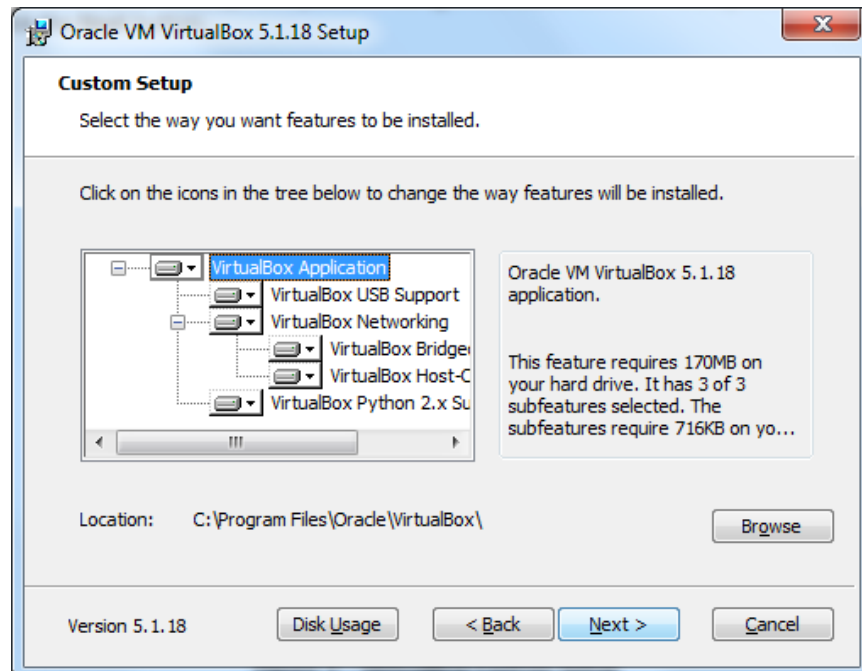
Name	Date modified	Type	Size
 VirtualBox-5.1.18-114002-Win.exe	3/28/2017 7:45 AM	Application	120,421 KB

VirtualBox Installer for Windows

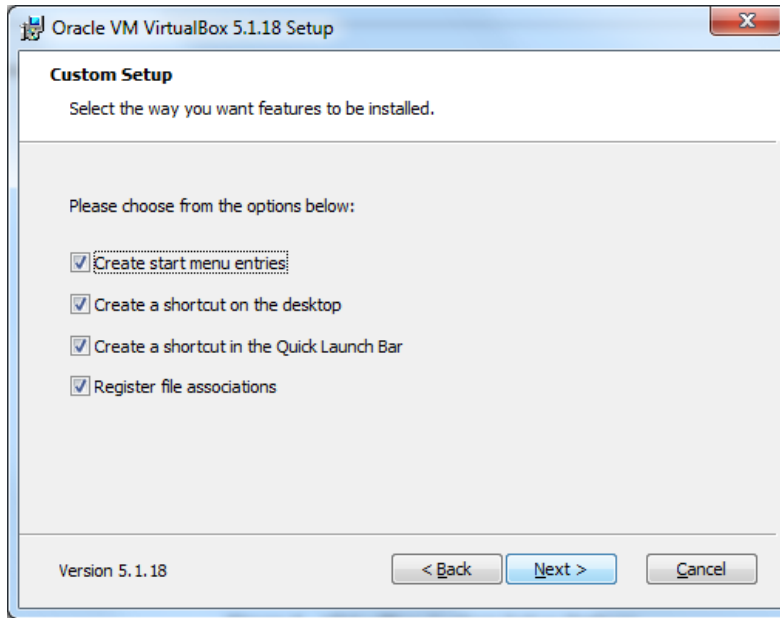
- Once the VirtualBox installation wizard appears, click the **Next** button.



- You may accept all the installation defaults, although you may wish to change the installation location on your development platform using the **Browse** button. If the options are acceptable, click the **Next** button.



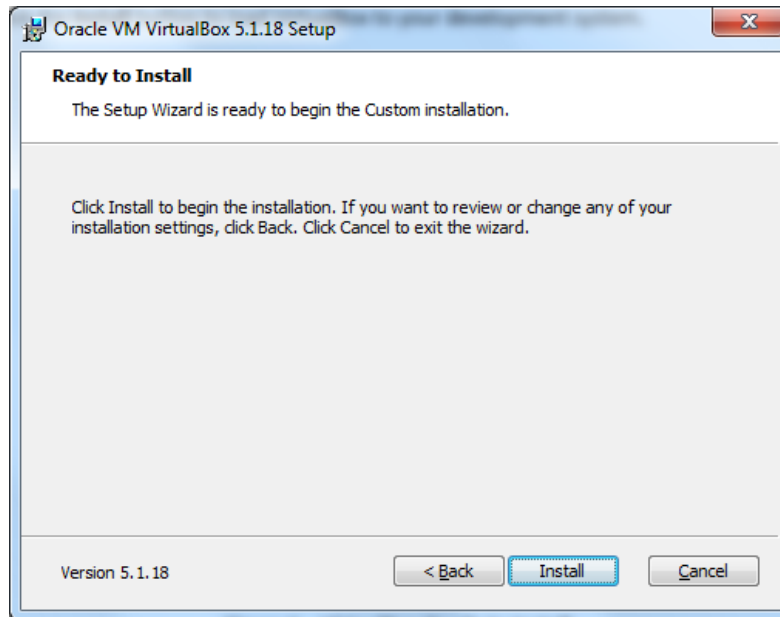
- You may again accept the default options and click the **Next** button.



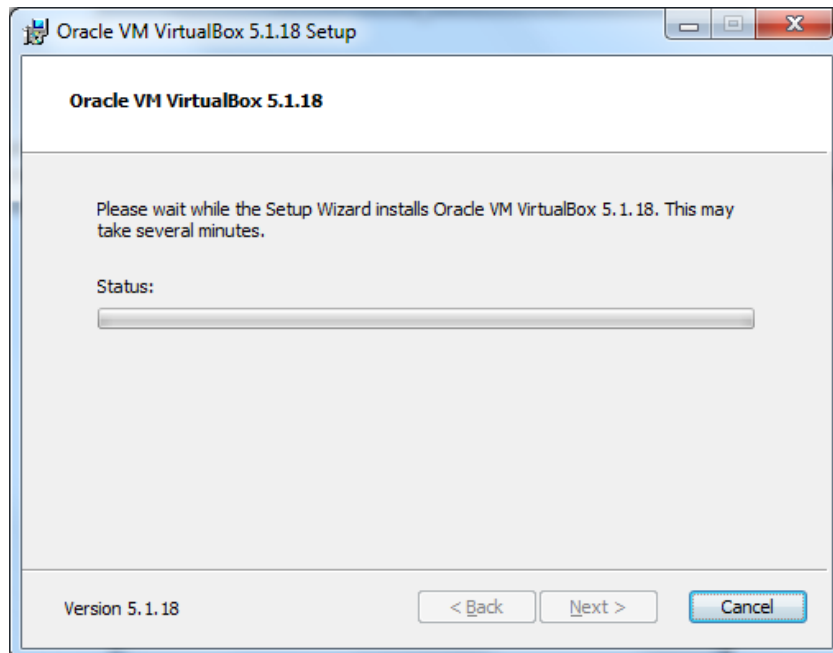
6. Click the **Yes** button to continue with the installation wizard.



- Click the **Install** button to load VirtualBox to your development system.



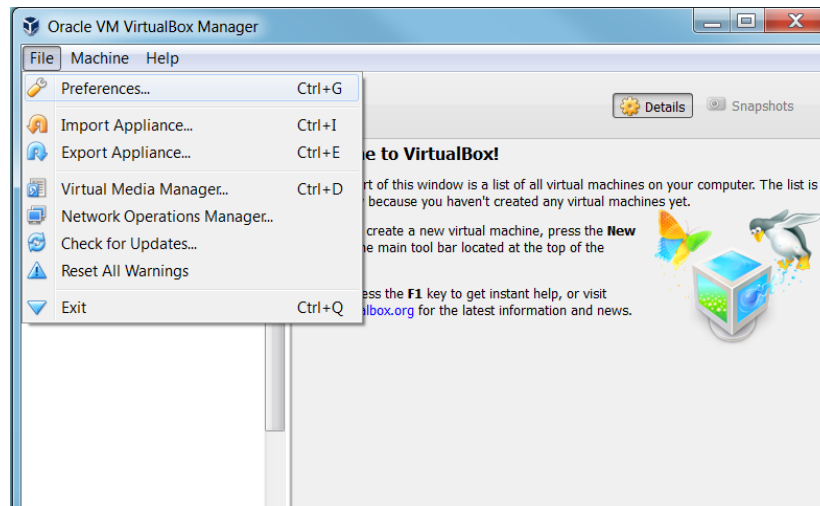
- During the installation you may receive prompts to authorize installation of various components. If prompted, allow the installer to make changes to your system, including installation of the USB interface and Network adapters.



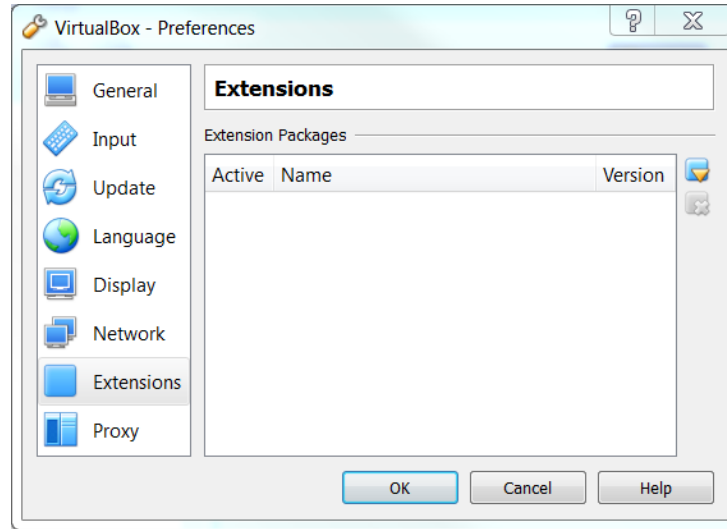
- Click the **Finish** button to complete the installation. Leave the checkbox enabled so VirtualBox will start after the installer finishes.




- Once VirtualBox starts (you can also start it from the Desktop shortcut, or the Windows Start button), the Extension Pack must be added. From the main menu, select **File > Preferences**.



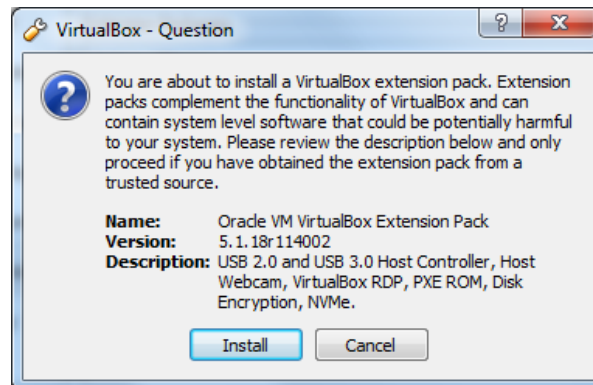
11. Select **Extensions**. Right-click in the *Extension Packages* whitespace box, and select **Add Package**.



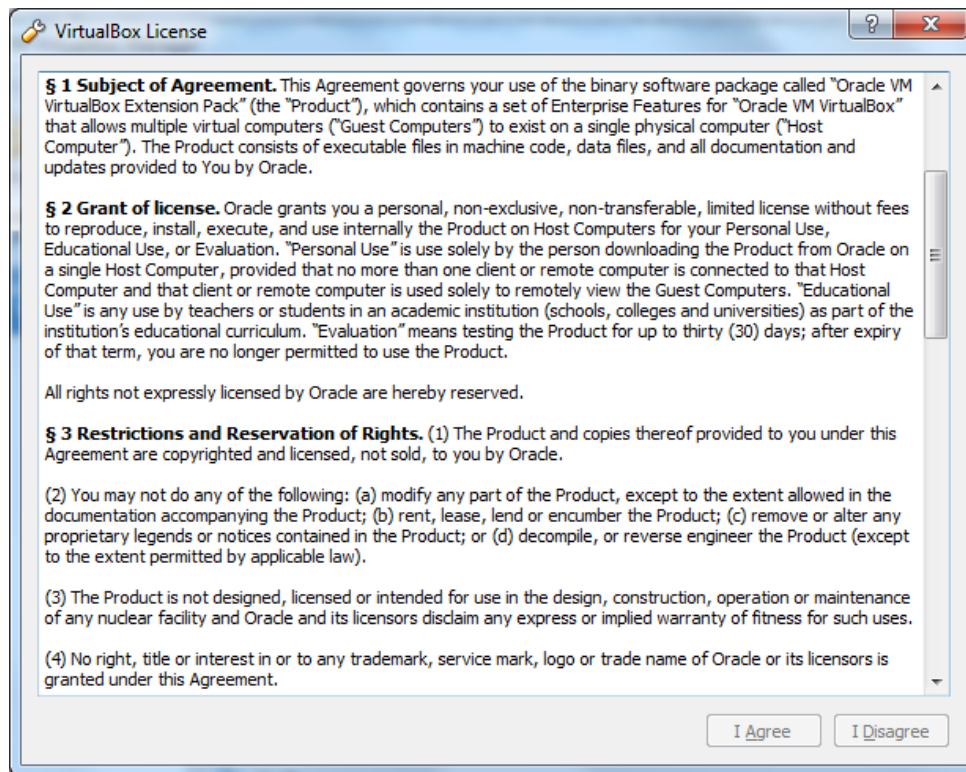
12. Browse to the location where you downloaded the VirtualBox Extension Pack compatible with your VirtualBox version. Select the Extension Pack and click the **Open** button.

Name	Date modified	Type	Size
 Oracle_VM_VirtualBox_Extension_Pack-5.1.18-114002.vbox-extpack	3/28/2017 7:44 AM	VirtualBox Extensi...	19,101 KB

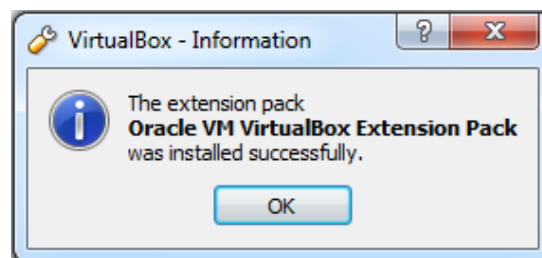
13. Click the **Install** button to add the VirtualBox Extension Pack.



14. Read the VirtualBox Extension Pack PUEL License to ensure you will not be in violation of the Oracle definition of Personal Use. See the *VirtualBox Licensing Frequently Asked Questions* for additional details. If you can accept the license conditions, scroll to the bottom of the agreement text box and click the **I Agree** button¹. If prompted, allow the installer to make changes to your development system.

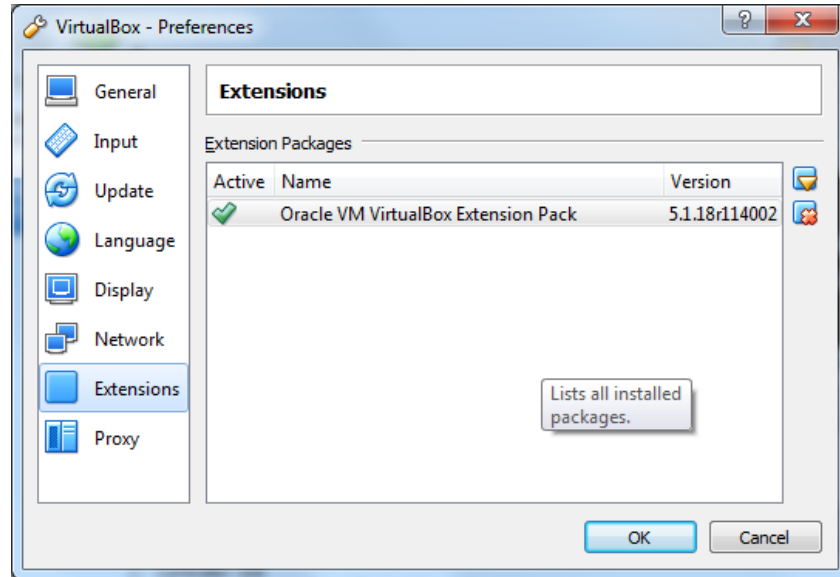


15. Click the **OK** button to complete the installation.



¹ If you must disagree, the installation will be terminated. You should either purchase a commercial license or uninstall VirtualBox from your host computer.

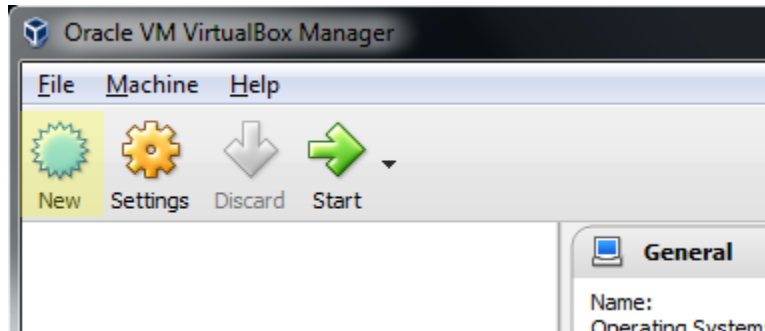
16. Click the **OK** button to return to VirtualBox.



This completes the installation of VirtualBox on your host development system. VirtualBox is now ready to accept a new Virtual Machine.

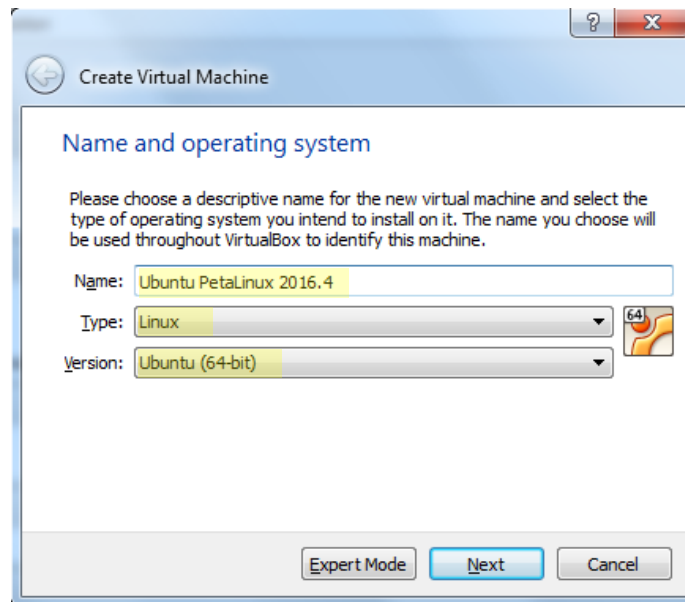
Create a New Virtual Machine

1. Launch Oracle VM VirtualBox Manager and click the **New** icon at the upper left.

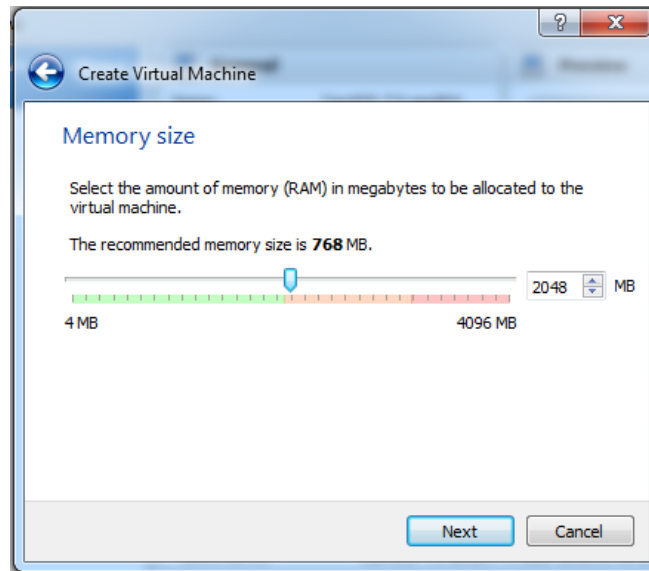


2. Select a descriptive name for the VM. Set the *Type* to **Linux** and the *Version* to one that corresponds to the OS you wish to install.
 - a. For Ubuntu, choose **Ubuntu (64-bit)**.

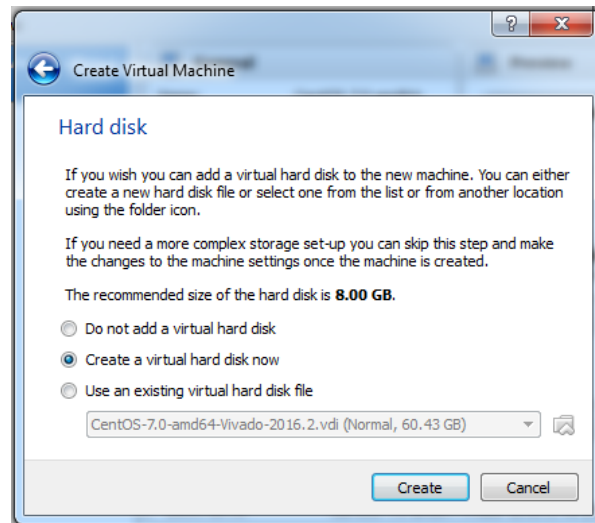
Click the **Next** button.



3. Select the amount of memory to be allocated to the Virtual Machine. Allocating more memory² will improve the VM performance, but you must leave sufficient memory available for your host system for all other concurrent processes. For a host system with 16 GB of RAM, a value of **2048** MB is recommended for the Virtual Machine. You may wish to experiment with this value to optimize your performance as larger density target devices have higher memory requirements. Once you have selected a memory size, click the **Next** button.



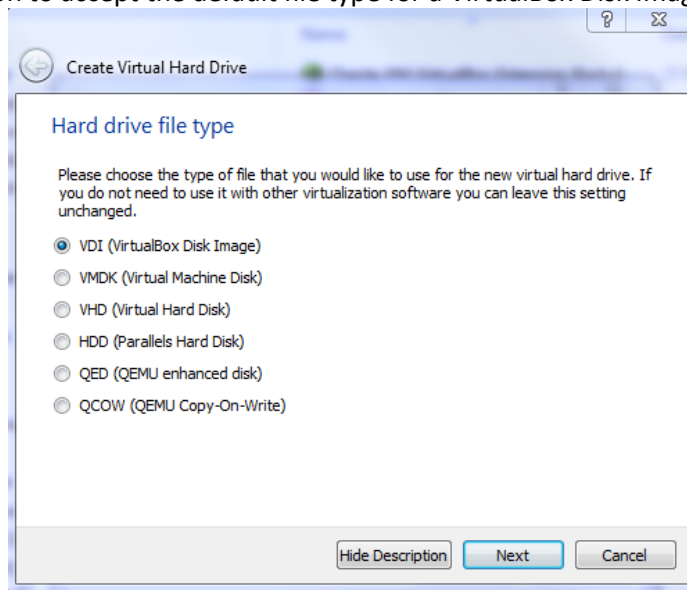
4. Click the **Create**³ button to allocate a virtual hard drive now.



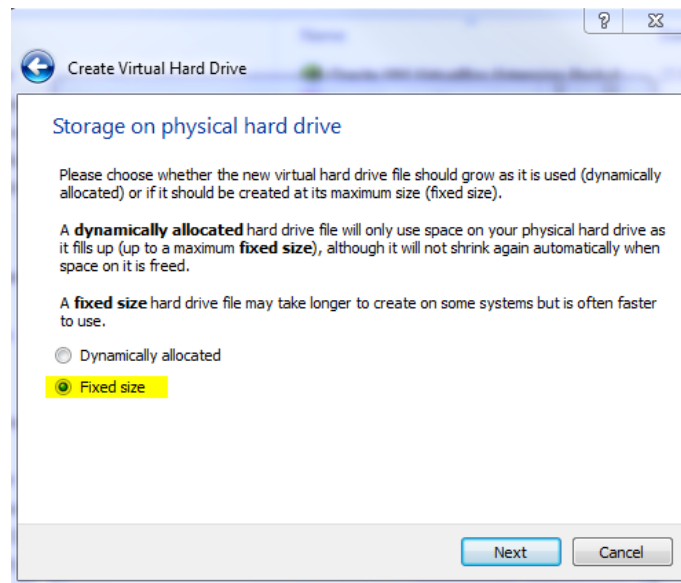
² You may need more memory if you intend to run Vivado with large Xilinx devices.

³ If you are importing an existing Virtual Machine, click the "Use an existing..." button.

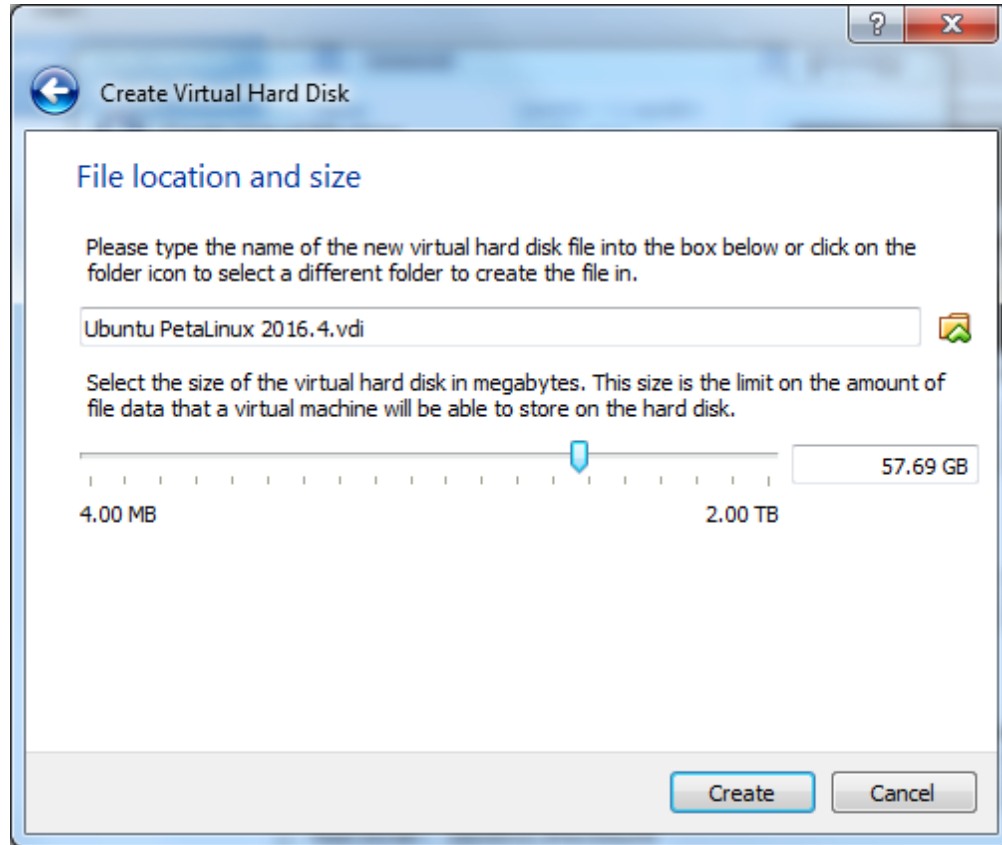
Click the **Next** button to accept the default file type for a VirtualBox Disk Image.



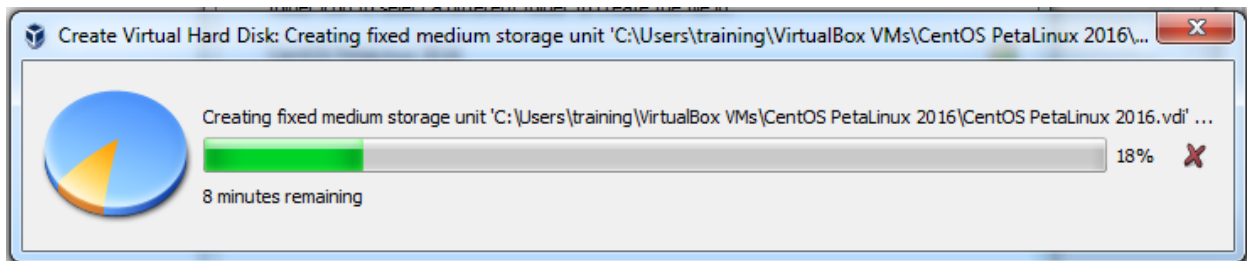
5. Select **Fixed Size** for the physical storage on your host hard drive. This will improve overall performance of the Virtual Machine. Click the **Next** button.



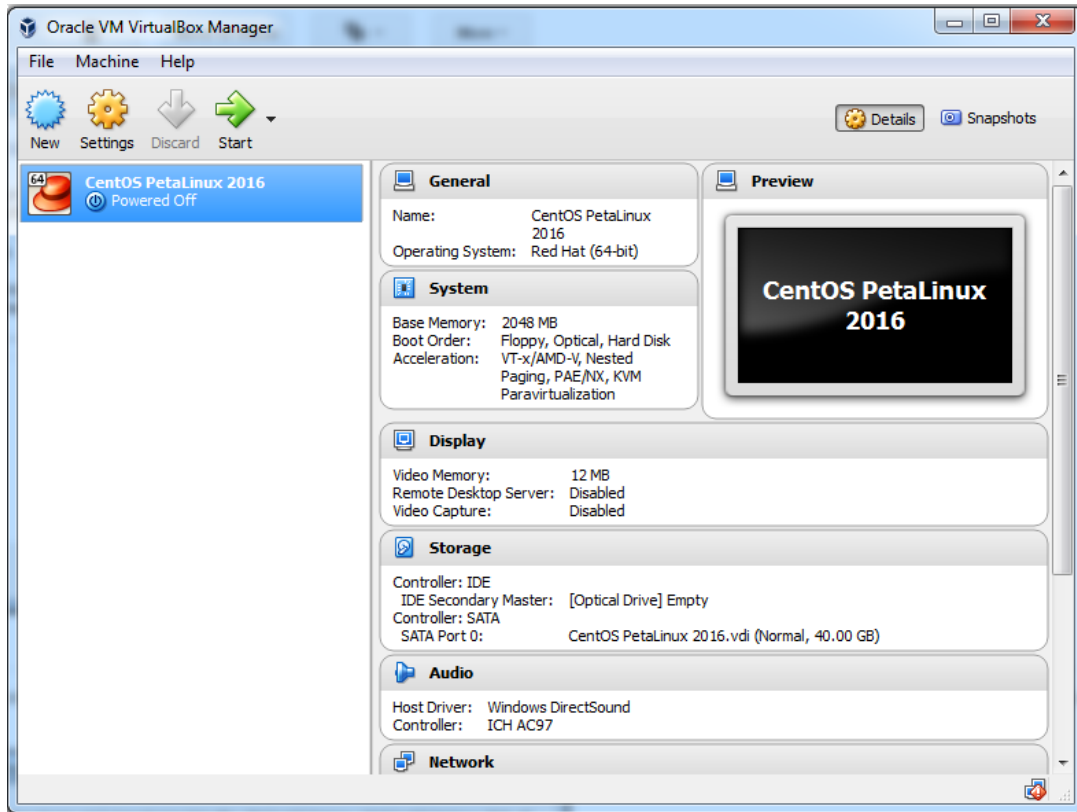
6. Select the name and location for the Virtual Machine within your host file system. It is recommended that the default size be set at **57.69 GB** to accommodate the Xilinx tools. If this space is unavailable, 50.0 GB is sufficient if you plan to install only the Software Development Kit and not Vivado. Click the **Create** button.



The Virtual Hard Disk may take a few minutes to create and initialize on your host file system.



7. Once the Virtual Disk completes, your VM is ready to accept an operating system.



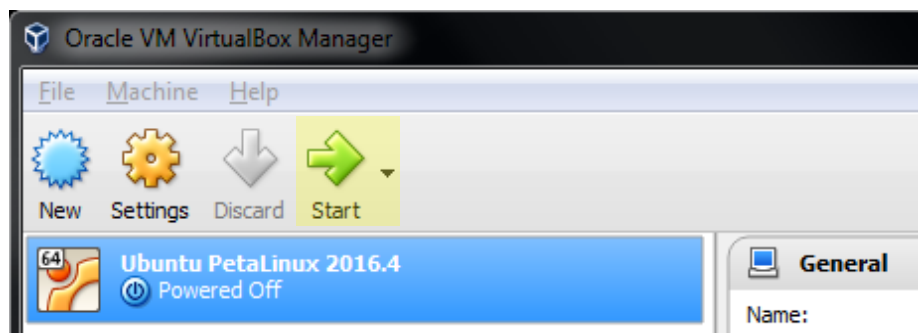
Install the Ubuntu 16.04 Operating System


To perform the steps in this section, you will need to download a bootable OS image in .iso format to your host system. The most recent Ubuntu images can be downloaded from:

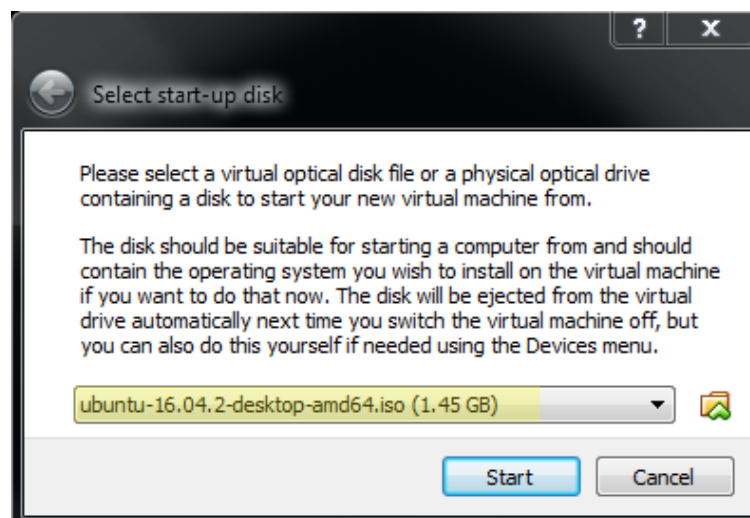
<http://www.ubuntu.com/download/desktop>

Repeat the steps outlined in **Create a New Virtual Machine**, entering Ubuntu as the **Name** of the VM. Once the Virtual Disk completes, your VM is ready to accept an operating system.

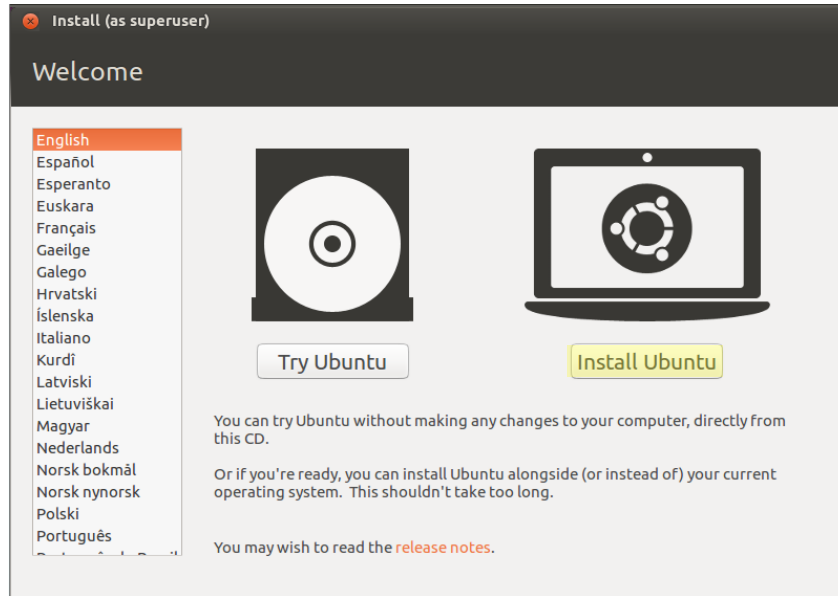
1. Launch VirtualBox (if necessary) and select the VM you wish to start in the left-hand panel. Click the **Start** button to execute the VM.



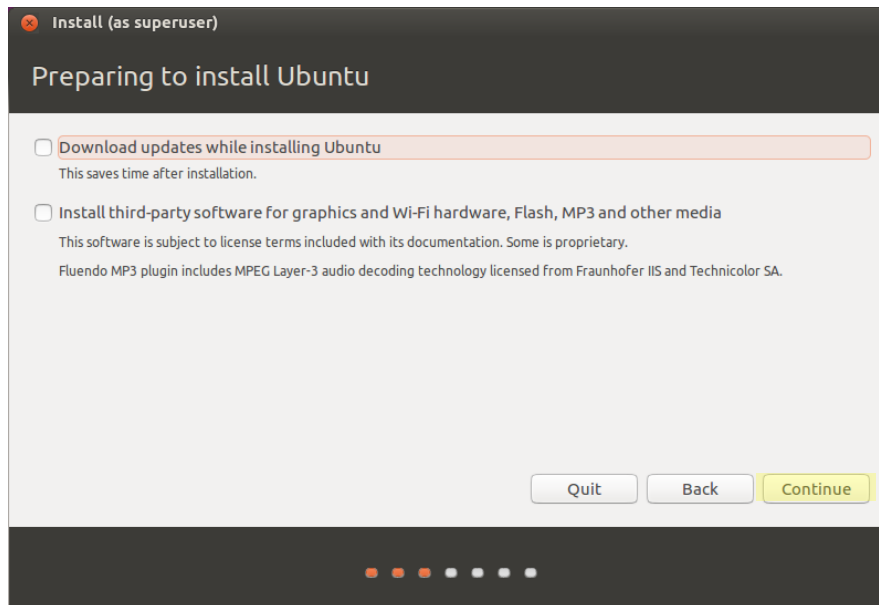
2. Select the **Browse**  icon to locate the .iso image for the OS you wish to install on your Virtual Machine. Click the **Start** button to begin.



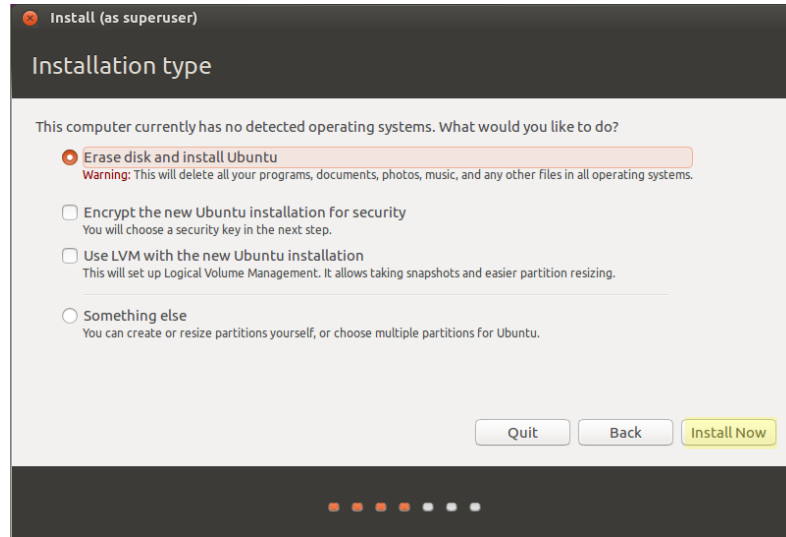
- When the Install Welcome screen appears, select English and click the **Install Ubuntu** button.



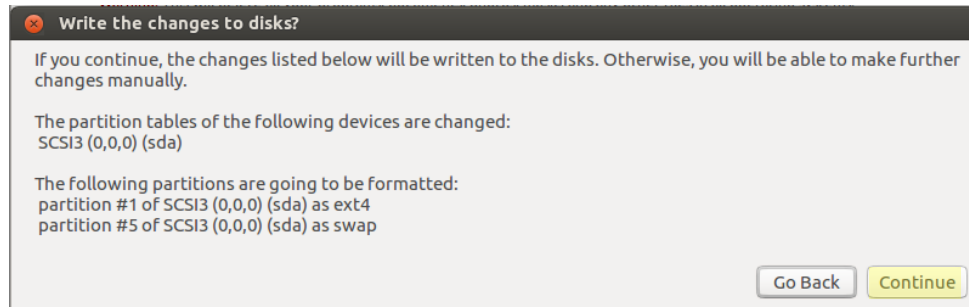
- The installer shows requirements for installation. Click the **Continue** button.



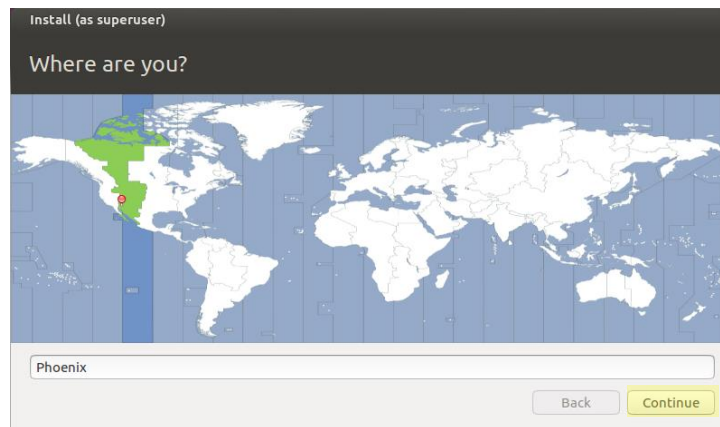
- The installer displays various installation types. The default displays as **Erase disk and install Ubuntu**. Click the **Install Now** button.



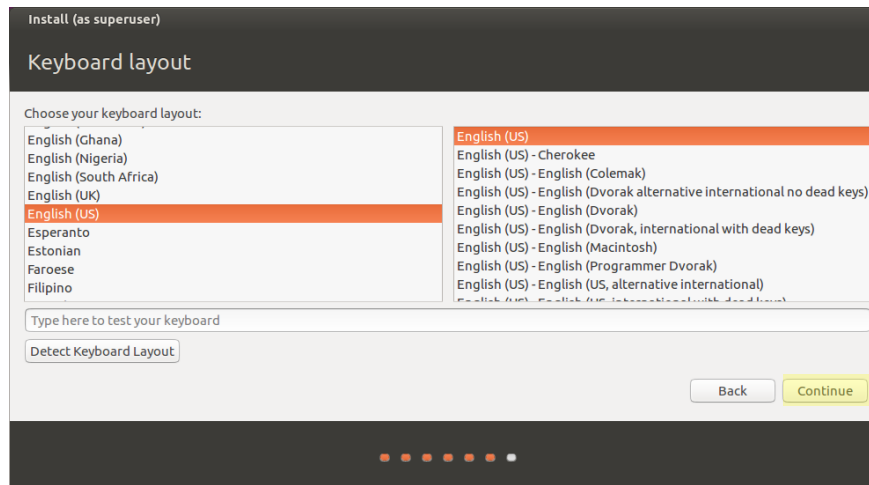
As this is a new installation, we want all changes written to the disks. Click the **Continue** button.



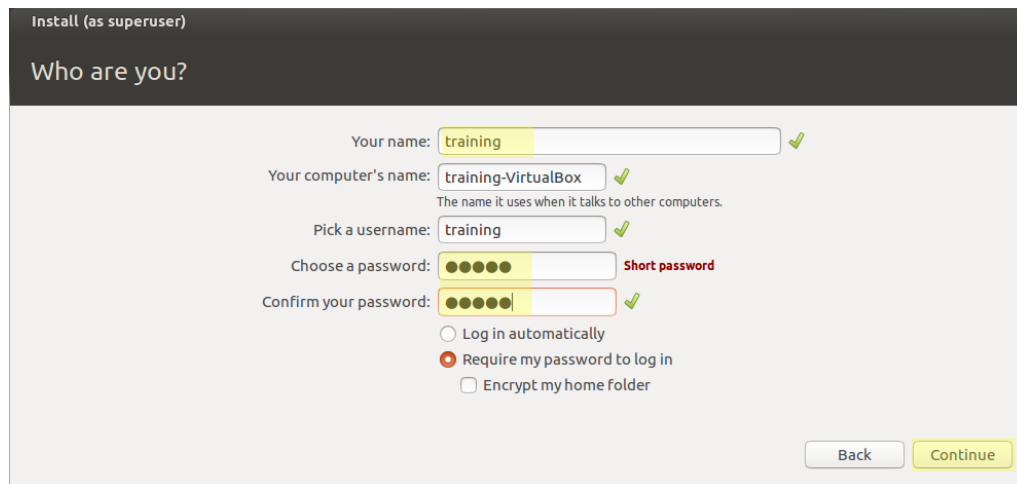
- Select your time zone and click the **Continue** button.



7. Select your preferred keyboard layout option. The default displays as English (US). Click the **Continue** button.



8. Enter the primary user name for the Virtual Machine. The system will auto-populate the computer name and username. Enter and confirm a password. Click the **Continue** button.



9. The installation displays a Welcome graphic and proceeds with installation. When the installation is complete, a screen appears asking you to restart. Click the **Restart Now** button.

If the Restart appears to “freeze”, you can force a reboot manually:

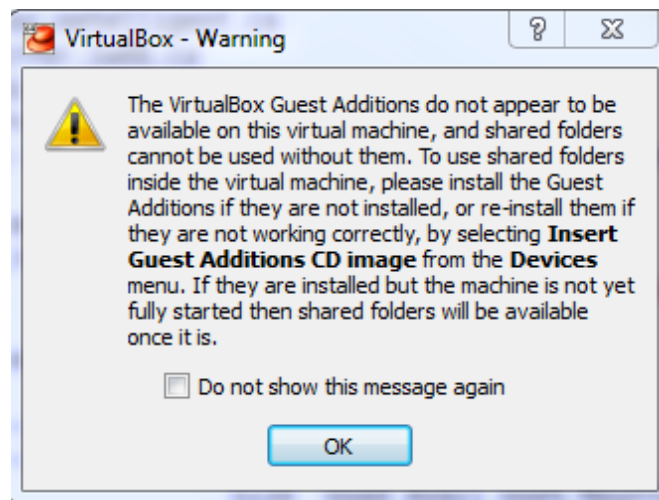
- a. From the main VirtualBox menu, select **File | Close**.
- b. In the *Close Virtual Machine* dialog, select **Power off the machine** and click the **OK** button.
- c. In the Oracle VM VirtualBox Manager, select your Virtual Machine and click the **Start** button.

VirtualBox Installation on Ubuntu

Installing VirtualBox on a supported Linux system should be straightforward, but depending on the precise configuration of your development system, there could be some challenges to create an optimal environment. This section outlines a few of the common issues that may be encountered.

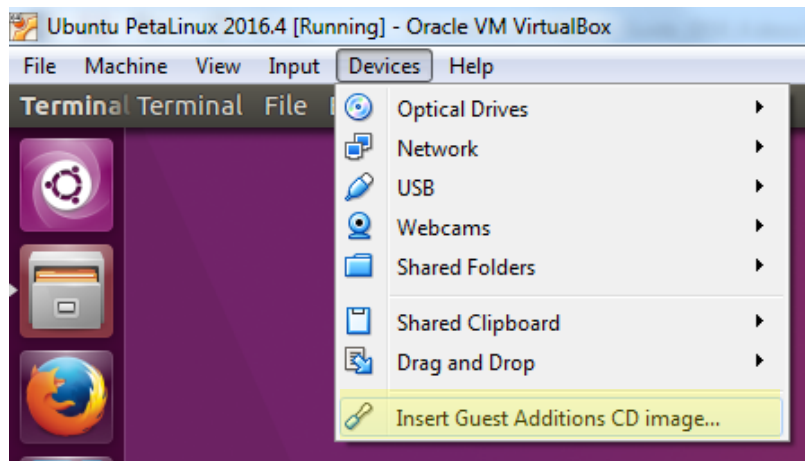
VirtualBox Guest Additions Installation (File Sharing)

1. The use of shared folders allows for easy transfer of files between the host and guest systems. To use the shared file facility of VirtualBox, you must install the Guest Additions. If you attempt to use the shared folder facility without the Additions, you will receive the following error message.

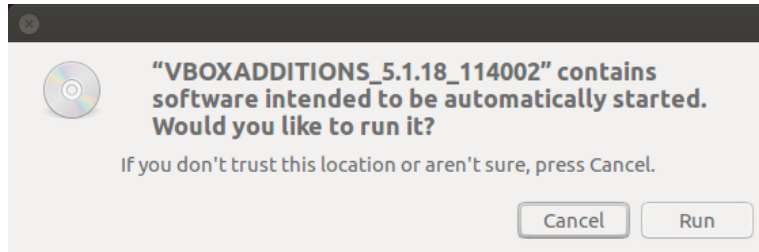


After Guest Additions are installed, you can move the cursor between the Virtual Machine and the host OS without having to use the *Right Ctrl* key to recapture the cursor in the host.

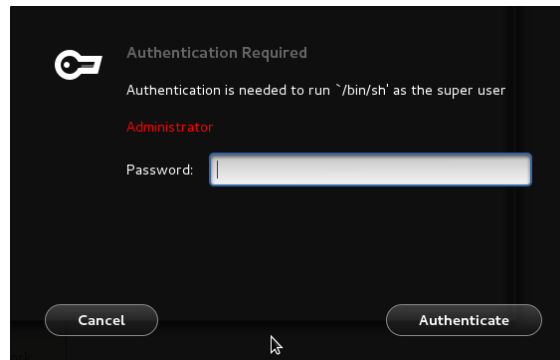
- a. From the **Devices** menu, select **Insert Guest Additions CD image...**



- b. Click the **Run** button to execute the installation.



- c. Enter the root password and click the **Authenticate** button.



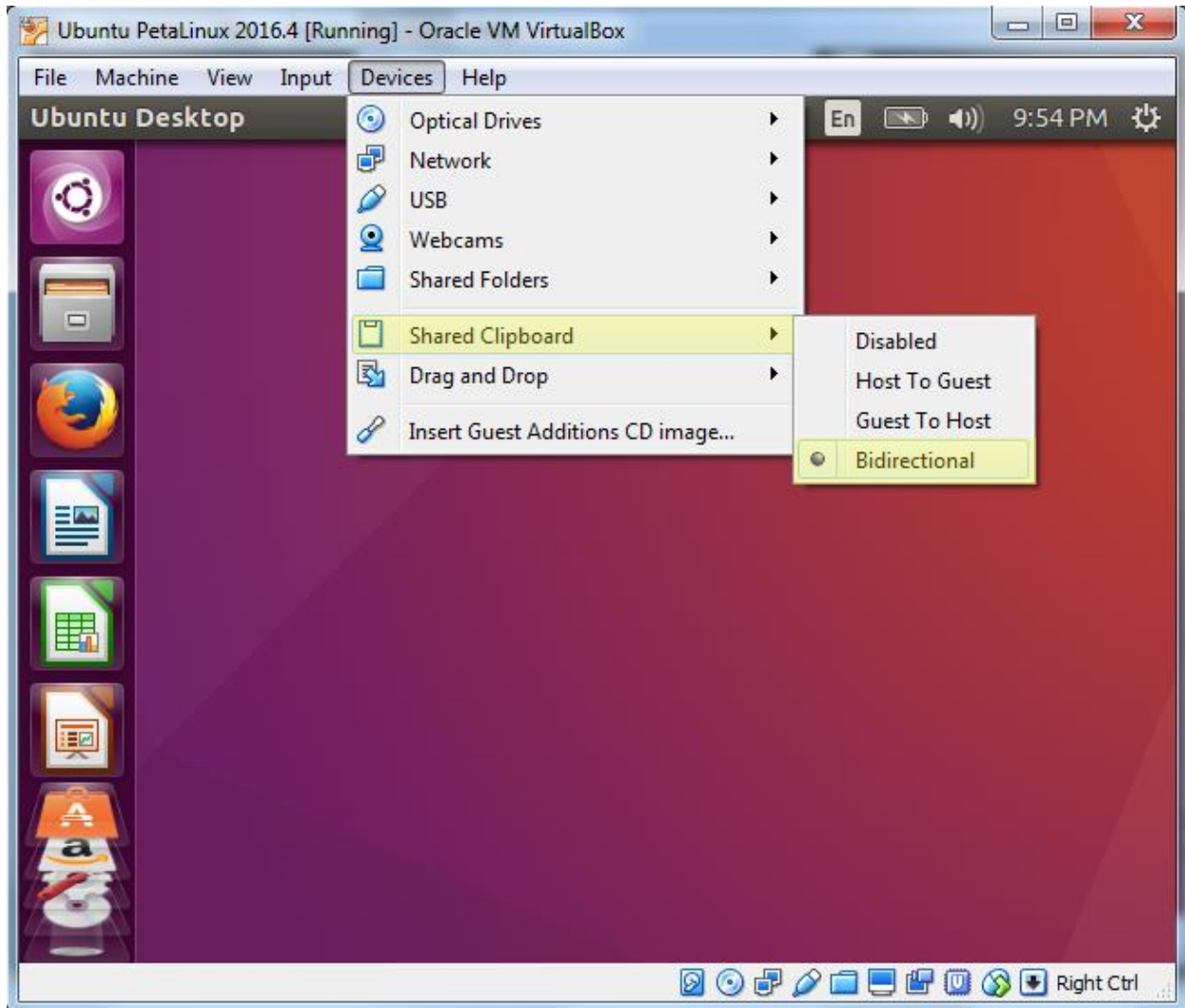
- d. The Guest Additions should install and verify with no failures. Press the **Enter** key to close the installation window. Be sure to restart Linux to ensure that the Guest Additions is started properly before moving on to a later section of this guide.

```
Terminal
Verifying archive integrity... All good.
Uncompressing VirtualBox 5.1.18 Guest Additions for Linux.....
VirtualBox Guest Additions installer
Copying additional installer modules ...
Installing additional modules ...
vboxadd.sh: Building Guest Additions kernel modules.
```

VirtualBox Shared Clipboard

Once the Guest Additions have been installed, you can enable the Shared Clipboard which will make it easier to copy and paste text strings from tutorial documents from the host system to the guest systems. This is very useful for later tutorials where it is desirable to copy command prompt instructions verbatim from the tutorial guide document directly into the command prompt of the guest system.

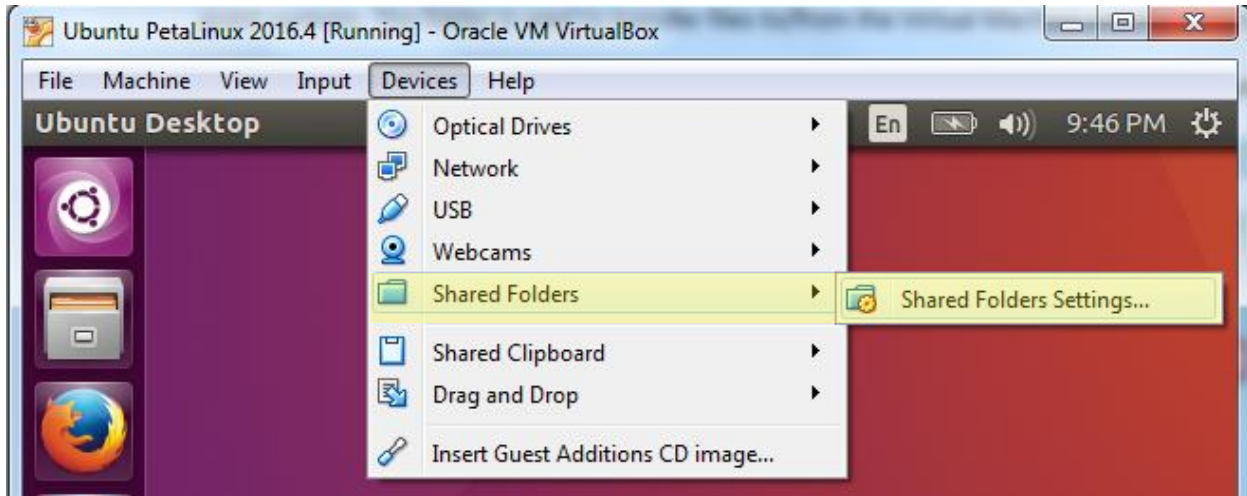
1. From the VirtualBox main menu, select **Devices > Shared Clipboard > Bidirectional**



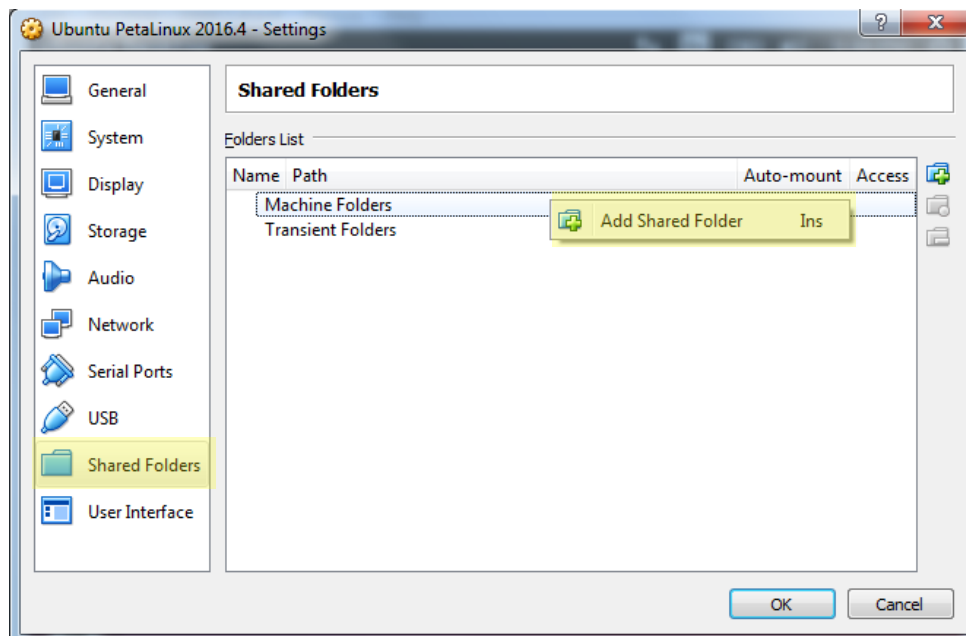
VirtualBox Shared Folders

Once Guest Additions have been installed, you must select a folder to share between the host and guest systems. This folder is used to transfer files to/from the Virtual Machine and the Host system.

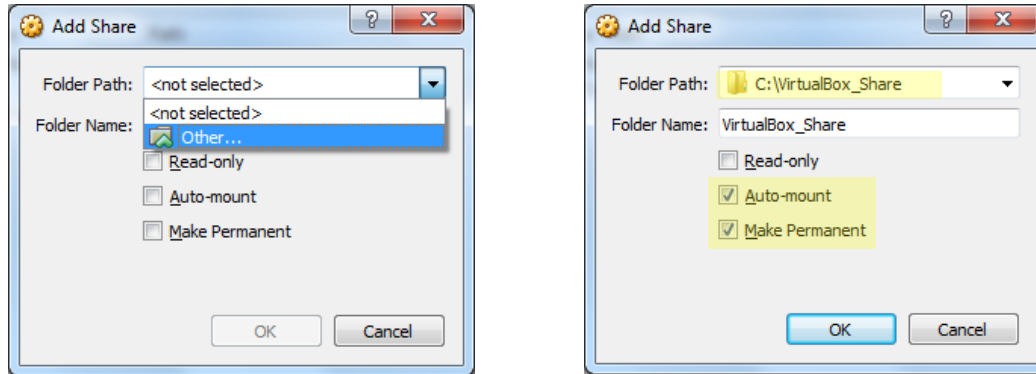
1. From the VirtualBox main menu, select **Devices > Shared Folder > Shared Folders Settings...**



2. Right-click *Machine Folders* and select **Add Shared Folder**.



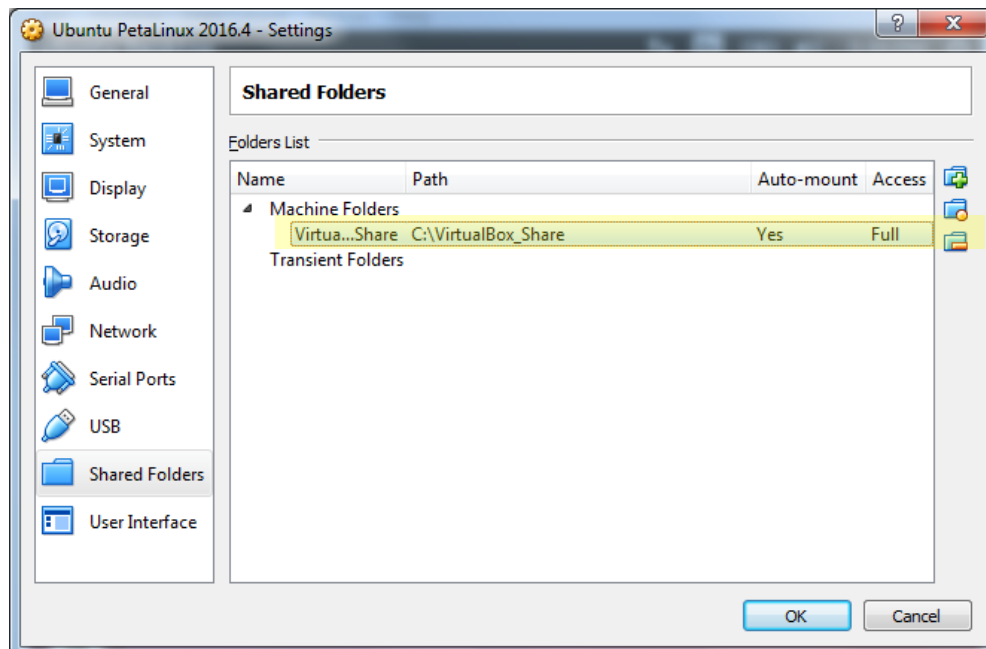
- In the *Folder Path* box, click the dropdown arrow on the right. Select the **Other** entry to open a Windows Explorer pane. Browse to the location in Windows where you want to set up a shared folder and click **Select Folder** in the Explorer pane. Click the checkboxes for **Auto-mount** and **Make Permanent**. Click the **OK** button.



- The location of the shared folder in Windows is shown in the Path column. The folder will Auto-mount and Full Access is allowed. The corresponding folder in the Linux VM is `/media/sf_<Windows Folder Name>`. In the example shown, this corresponds to:

`/media/sf_VirtualBox_Share`

Click the **OK** button to close the panel.



5. Shared folders are only available to user accounts that are also members of the group *vboxsf*. This means the user account created earlier must be added to this group. This can only be done in Ubuntu 16.04 from the command line. To view available groups and members, in a Terminal window enter one or more of the following commands.

This command lists all groups:

```
$ getent group
```

This command lists a specific group named *vboxsf*:

```
$ getent group vboxsf
```

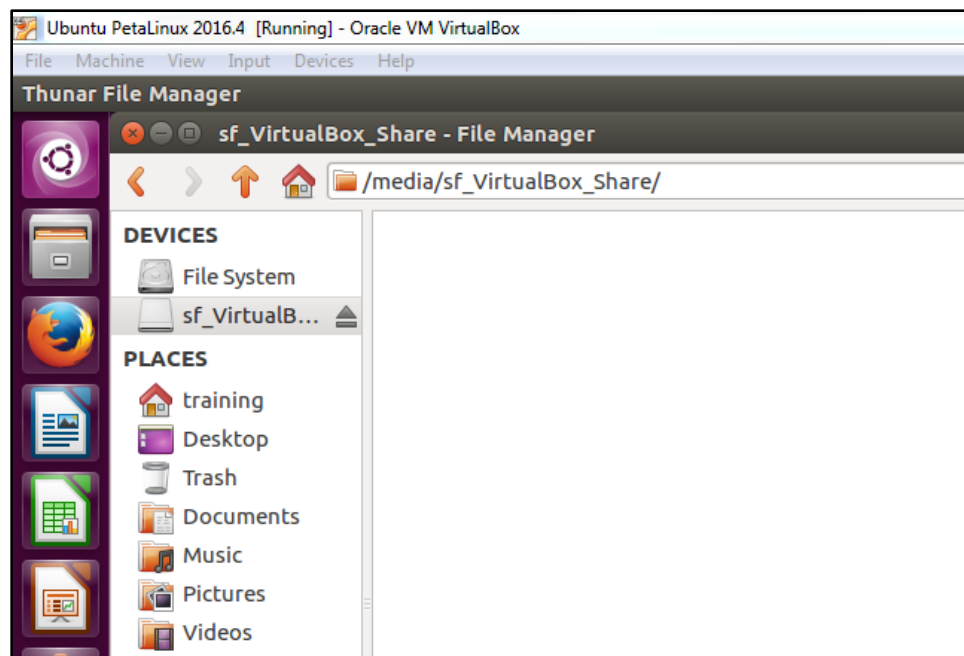
To add an existing user to an existing group, in a Terminal enter the following command:

```
$ sudo usermod -a -G vboxsf <current username>
```

6. Reboot the Virtual Machine.

7. The selected user name will belong to the *vboxsf* group on the next login. To access the shared folder from the Virtual machine, browse to: ***/media/sf_<sharename>***

In this example, the folder in Windows is named *VirtualBox_Share*, so the *<sharename>* in Linux is ***sf_VirtualBox_Share***, automatically mounted in the ***/media*** folder. Any files in this folder are available to the Virtual Machine and the Host OS system.

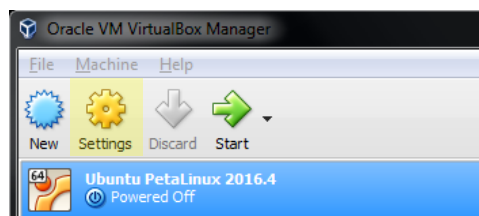


Network Bridging

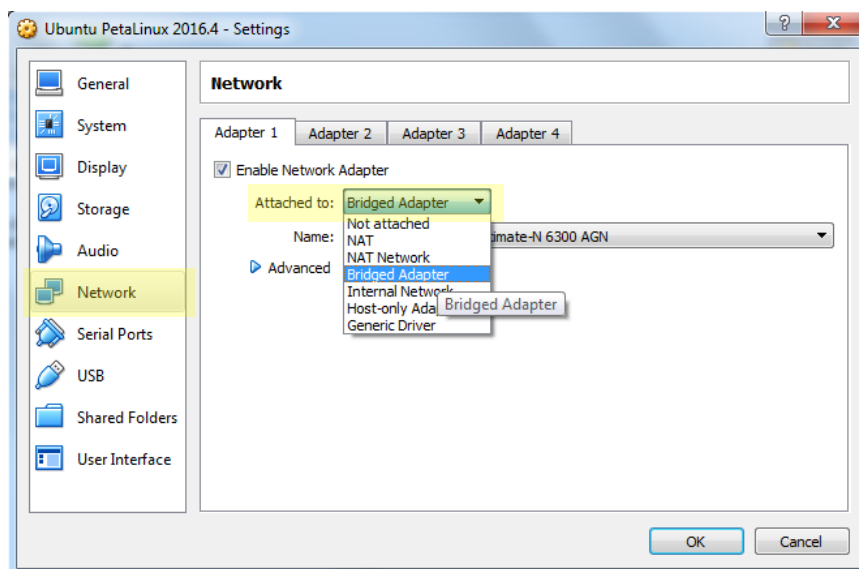
When VirtualBox is installed with its default options, the networking is set up to use Network Address Translation (NAT). This allows your Virtual Machine complete outgoing access to your LAN and/or the Internet, but it assigns an internal IP address that may not be compatible with your LAN and the IP address of your host system.

If you would like your Virtual Machine to accept an address from a local DHCP server, you can change the default network type to Bridged. This will make the Virtual Machine available to any other connected device on the same subnet on your LAN.

1. From the main VirtualBox menu in a running Virtual Machine, click on the **Settings** button. If the button is not visible, select **Machine > Settings**.



2. Select the Network entry in the left panel. Select the tab for your NIC (typically **Adapter 1**) and expand the dropdown menu for the *Attached to* field.



3. Select **Bridged Adapter** from the dropdown menu and click the **OK** button to save the changes. Wait a few seconds for your Virtual Machine to request an address from the local DHCP server. Once complete, the VM will now have an address on your local subnet, accessible to all devices on your LAN.

If you do not receive a new IP address after a minute, stop and restart the Ethernet service.

Ubuntu 16.04 Installation

Installing Ubuntu on a supported Virtual Machine should be straightforward, but depending on the precise configuration of your development system, there could be some challenges to create an optimal environment. This section outlines a few of the common issues that may be encountered.

Set a root user password

By default, Ubuntu does not set a password for the root user. You can do this by simply invoking the **sudo passwd** command. You supply your own user password, then set the root user password.

```
sudo passwd  
<Enter user password>  
<Enter new root password>  
<Confirm new root password>
```

```
training@training-VirtualBox:~$ sudo passwd  
[sudo] password for training:  
Enter new UNIX password:  
Retype new UNIX password:  
passwd: password updated successfully
```

From this point forward, you will be able to precede a command with **sudo** to obtain root authority.

Change the default shell to bash for PetaLinux

If you intend to use the PetaLinux tool-chain under Ubuntu, you will find that it requires the bash shell as the default to execute correctly. By default, Ubuntu uses the dash shell, which is an extension of the bash shell with a few additional features and optimized for faster execution. Unfortunately, the dash shell is not compatible with the current PetaLinux tool-chain.

A description for the dash shell and its potential issues can be found here:

<https://wiki.ubuntu.com/DashAsBinSh>

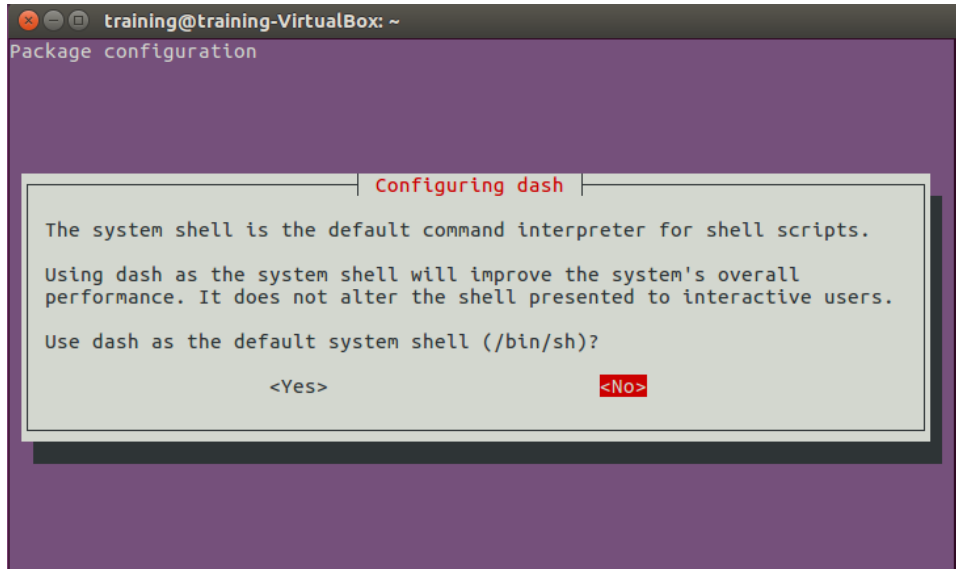
1. Most distributions use `/bin/sh` as a symbolic link to points to the actual default shell. Under your Ubuntu environment, to determine the current shell, enter:

```
$ ls -l /bin/sh
```

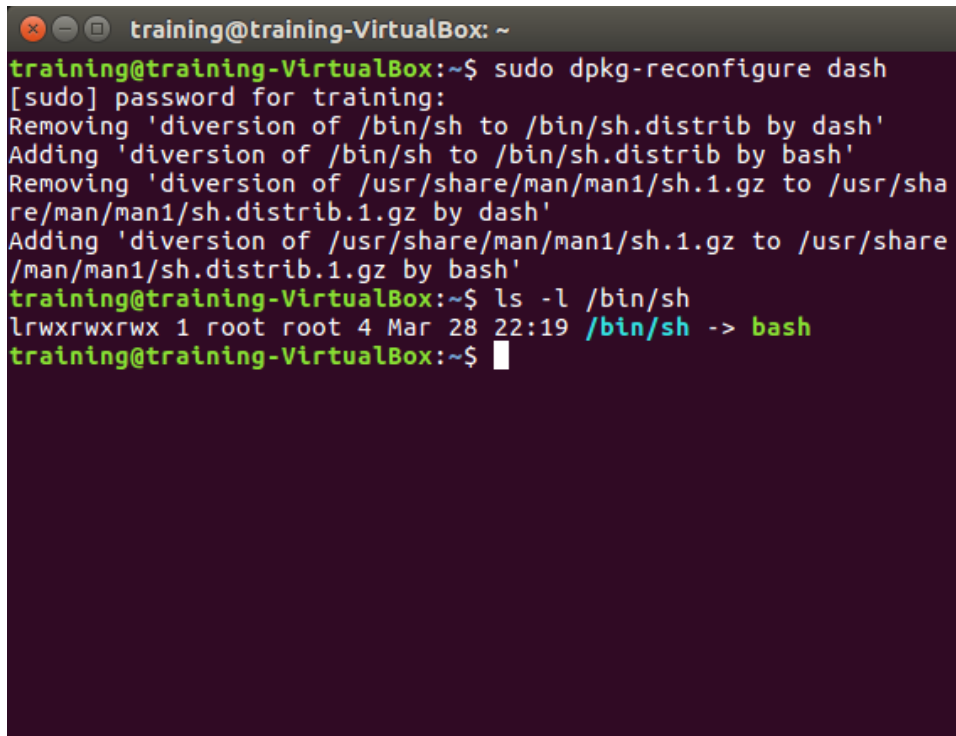
2. To change the default shell for all terminal windows, enter:

```
$ sudo dpkg-reconfigure dash
```

3. Select the option to remove dash as the default shell when prompted.



4. When the change is complete, close all open Terminal windows and open a new Terminal.
5. Verify the default shell is bash using the commands shown previously.



Xilinx Vivado/SDK Installation

Installing Vivado/SDK tools on a supported Linux system should be straightforward, but depending on the precise configuration of your development system, there could be some challenges to create an optimal environment. This section outlines a few of the common issues that may be encountered.

To perform the steps in this section, you need to download a tar.gz image to your host system. The most recent images can be downloaded from: <http://www.xilinx.com/support/download.html>.

NOTE: Once the download is complete, you may want to verify there is an md5sum utility installed by default. Open a terminal window and enter the following:

```
$ md5sum -b <path to your compressed-Vivado-tar.gz file>
```

The valid checksums for the different download archives are available from the Xilinx download site.

Install Vivado in the VirtualBox Linux VM

1. Copy the All-OS version of the compressed Vivado installer from your host system to your VM desktop.



2. Open a terminal window and decompress the installer on your desktop. It will create a new folder automatically in the current directory of your Terminal window.

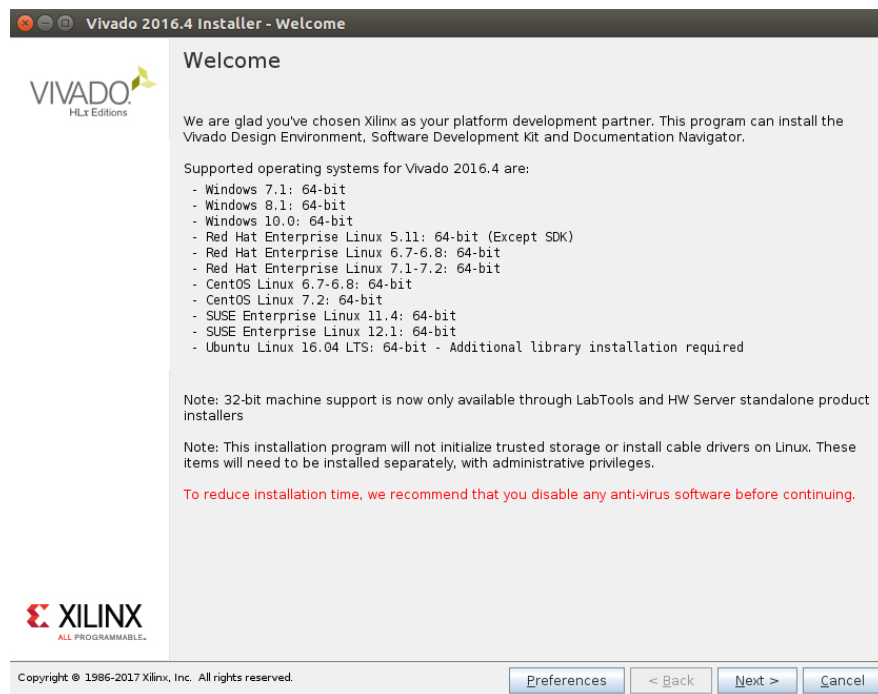
```
$ tar -xvzf ~/Desktop/<Compressed-Vivado-Installer-Name>
```

3. Change into the new folder and execute the installer setup script. You will need root privilege to install into the default directory of `/opt/Xilinx` (preferred).

```
$ sudo ./xsetup
```

```
training@training-VirtualBox: ~/Xilinx_Vivado_SDK_2016.4_1215_1
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/glib-lite.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/unpack.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/tnameserv.exe
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/jsoundds.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/net.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/dcpr.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/sawindbg.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/wsdetect.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/mlib_image.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/jaas_nt.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/klist.exe
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/attach.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/splashscreen.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/JAWTAccessBridge-64.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/keytool.exe
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/jabswitch.exe
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/fontmanager.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/bin/jjfr.dll
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/COPYRIGHT
Xilinx_Vivado_SDK_2016.4_1215_1/tps/win64/jre/THIRDPARTYLICENSEREADME-JAVAFX.txt
Xilinx_Vivado_SDK_2016.4_1215_1/xsetup.exe
training@training-VirtualBox:~$ cd Xilinx_Vivado_SDK_2016.4_1215_1/
training@training-VirtualBox:~/Xilinx_Vivado_SDK_2016.4_1215_1$ sudo ./xsetup
[sudo] password for training:
```

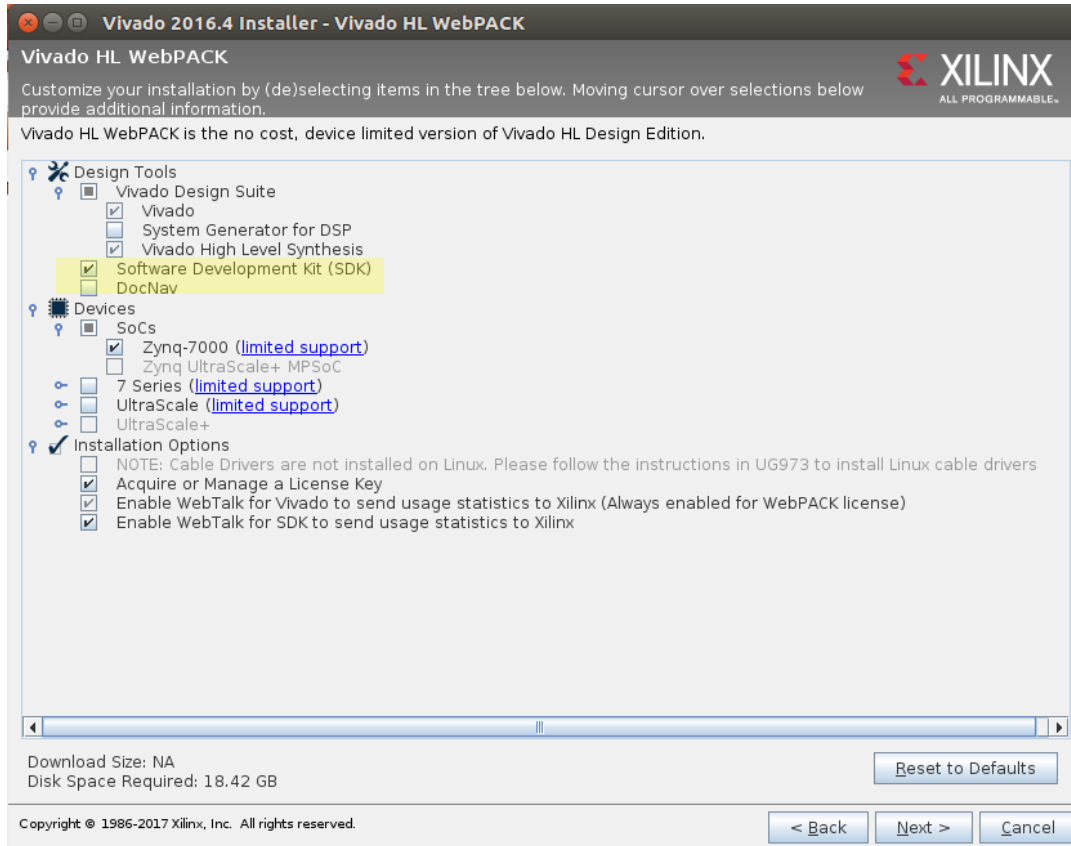
4. The Vivado 2016.4 Installer lists the operating systems officially supported by Xilinx. The tools may run on other Linux distributions and versions, but there will likely be some manual configuration required. Installation on unsupported systems is beyond the scope of these notes.



5. Follow the instructions as shown in the Vivado Installer GUI.

Note: If you also want to install the SDK, select *SDK Development Kit* from the menu below.

 - a. Accept all license agreements.
 - b. Select the Vivado edition (version) you wish to install (or SDK Standalone).
 - c. Select the Devices you need.
 - d. Cable Drivers are no longer installed by this installer under Linux. See instructions in the **Install Missing Cable Drivers** section of this document.



- e. Select the default directory for installation. Depending on the size of your virtual disk, you may be space limited here. You can delete the compressed installer and the expanded hierarchy to free 12-16 GB of disk space, if necessary.
 - f. If needed for your device or tool selection, obtain and install a license for your tools. If you are planning to use a free WebPACK license for development on your target platform, there is nothing further that is needed and you can close the Vivado License Manger dialog.
6. You may optionally delete the entire folder where you decompressed the installer to free up additional disk space.

Adjust GTK Version Used for Vivado in the VirtualBox Linux VM

It is a known issue that SDK will fail to launch on Ubuntu 16.04 without a workaround and this is addressed in Xilinx Answer Record AR67580:

<https://www.xilinx.com/support/answers/67580.html>

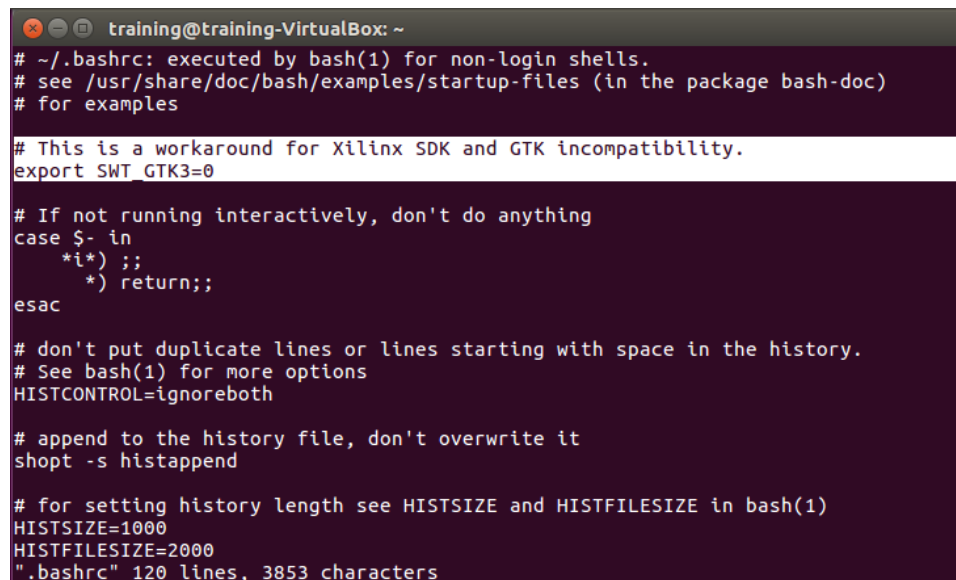
The issue is with the GTK version shipped with Ubuntu 16.04, which has issues with the eclipse. In order to work around this issue, set the environment variable **SWT_GTK3** to 0.

1. To temporarily set this environment variable for the current terminal session, use the following command:

```
$ export SWT_GTK3=0
```

2. To permanently set this environment variable for the all future terminal sessions, insert this comment and export command near the top of the `~/.bashrc` file using your favorite editor:

```
# This is a workaround for Xilinx SDK and GTK incompatibility.  
export SWT_GTK3=0
```



```
training@training-VirtualBox: ~  
# ~/.bashrc: executed by bash(1) for non-login shells.  
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)  
# for examples  
  
# This is a workaround for Xilinx SDK and GTK incompatibility.  
export SWT_GTK3=0  
  
# If not running interactively, don't do anything  
case $- in  
  *i*) ;;  
  *) return;;  
esac  
  
# don't put duplicate lines or lines starting with space in the history.  
# See bash(1) for more options  
HISTCONTROL=ignoreboth  
  
# append to the history file, don't overwrite it  
shopt -s histappend  
  
# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)  
HISTSIZE=1000  
HISTFILESIZE=2000  
".bashrc" 120 lines, 3853 characters
```

3. Save the edits to the `~/.bashrc` file.

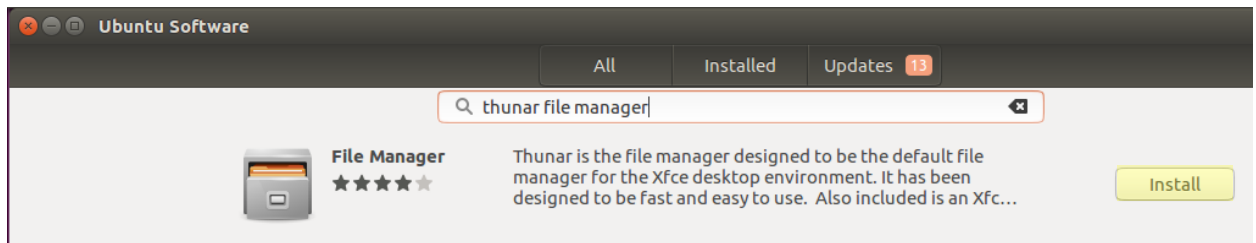
Install Thunar File Manager

The Thunar File Manager is used in some of the Avnet Reference Designs and Tutorial materials since it allows for the folder taxonomy to be expanded in a left pane and shows files and folders in a right pane making it easier to navigate complex folder and file structures within example designs.

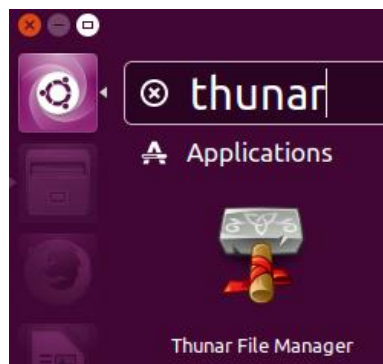
1. Launch the **Ubuntu Software** management application using the Dash App Lens.



2. Within the **Ubuntu Software** management application, search for the **Thunar File Manager** application and click the install button. Enter the superuser password if prompted:



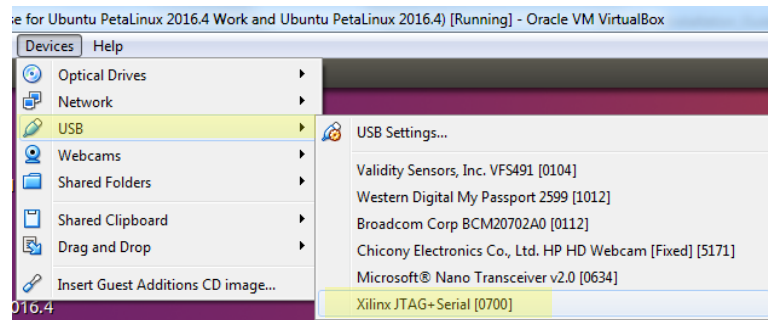
3. Close the Ubuntu Software management application and use the Dash App Lens to launch the newly installed **Thunar File Manager** application to verify that it was installed correctly.



Install GTK Terminal

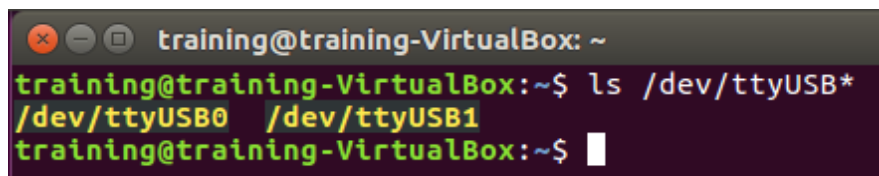
The Ubuntu **Serial port terminal** (or GTK Terminal) application is used in some of the Avnet Reference Designs and Tutorial materials since it allows for simple connection to USB-UART of many development board platforms.

1. Plug in your development board and connect the USB-UART port to the PC so that the USB-UART device is recognized under Windows first. Then locate your USB-UART device under the **Devices**→**USB** selection menu and click on it to remove the device from the host operating system and add it to the guest operating system. In this example the Xilinx JTAG+Serial [0700] device is used as it represents the USB-UART of the Avnet MiniZed board.



2. Once the device is detected and enumerated under Ubuntu, the USB-UART port should be listed under one of the **/dev/ttyUSBx** device entries. Locate the device entry for the USB-UART and make a note of this device for use in a later step.

```
$ ls /dev/ttyUSB*
```



3. To make it easier to launch the terminal app (GtkTerm) without needing to provide the root password each time, open a command window and add the current user to the group for the **/dev/ttyUSBx** devices used for USB-UART:

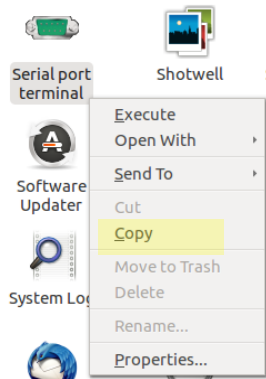
```
$ sudo usermod -a -G dialout <current username>
```

4. Install the gtkterm package:

```
$ sudo apt-get install gtkterm
```

5. **Reboot the Virtual Machine to force the changes to take effect.**

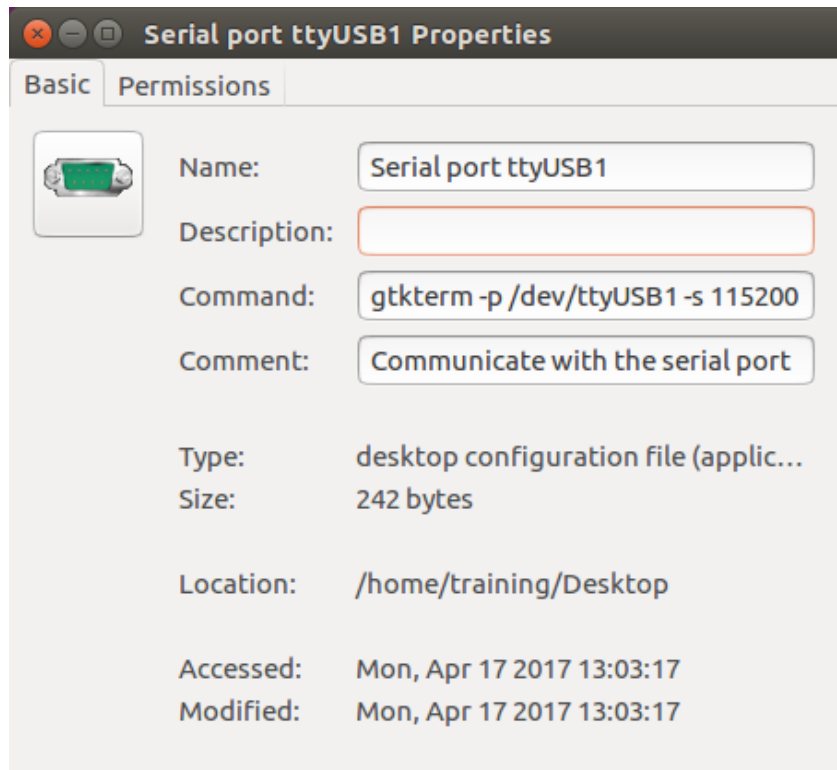
6. Create a Desktop icon by copying and pasting **Serial port terminal (gtkterm)** application from the `/usr/share/applications` folder directly to the `~/Desktop` folder:



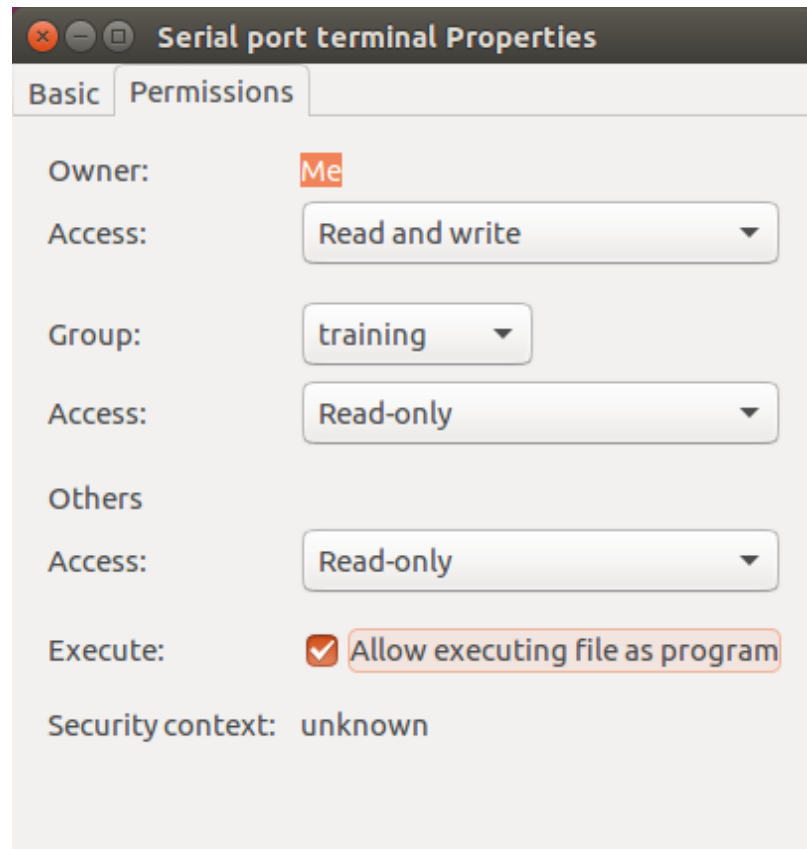
7. Right-click on the new **Serial port terminal (gtkterm)** application Desktop icon and select the **Properties** option.
8. Within the file **Properties** window, set the application attributes to match the USB-UART device attached to the system, in this example the USB-UART is attached to the `/dev/ttyUSB1` device entry:

Name: **Serial port ttyUSB1**

Command: **gtkterm -p /dev/ttyUSB1 -s 115200**



9. Under the **Permissions** tab, make sure that the **Allow executing file as program** option is selected:



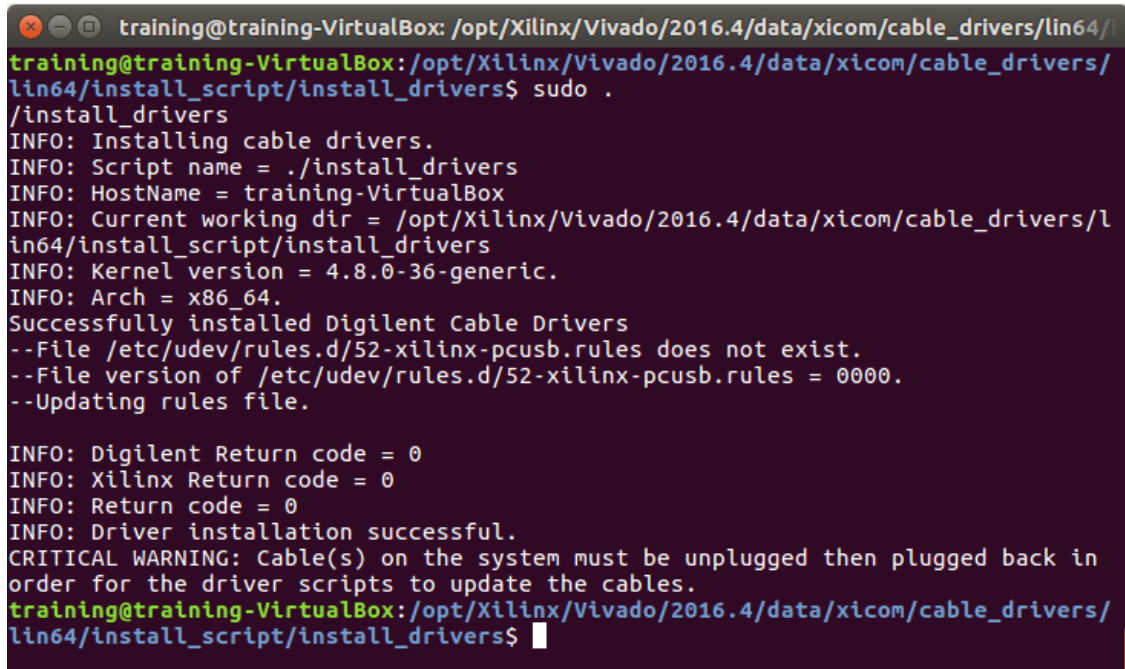
10. Close the Properties window.

Install Missing Cable Drivers

The drivers, which were not included with the install, can be installed manually.

1. Open a Terminal window.
2. At the command prompt, enter:

```
$ cd /opt/Xilinx/Vivado/2016.4/data/xicom/cable_drivers/lin64/ \
install_script/install_drivers/
$ sudo ./install_drivers
```



```
training@training-VirtualBox: /opt/Xilinx/Vivado/2016.4/data/xicom/cable_drivers/lin64/
training@training-VirtualBox: /opt/Xilinx/Vivado/2016.4/data/xicom/cable_drivers/
lin64/install_script/install_drivers$ sudo ./install_drivers
INFO: Installing cable drivers.
INFO: Script name = ./install_drivers
INFO: HostName = training-VirtualBox
INFO: Current working dir = /opt/Xilinx/Vivado/2016.4/data/xicom/cable_drivers/lin64/install_script/install_drivers
INFO: Kernel version = 4.8.0-36-generic.
INFO: Arch = x86_64.
Successfully installed Digilent Cable Drivers
--File /etc/udev/rules.d/52-xilinx-pcusb.rules does not exist.
--File version of /etc/udev/rules.d/52-xilinx-pcusb.rules = 0000.
--Updating rules file.

INFO: Digilent Return code = 0
INFO: Xilinx Return code = 0
INFO: Return code = 0
INFO: Driver installation successful.
CRITICAL WARNING: Cable(s) on the system must be unplugged then plugged back in
order for the driver scripts to update the cables.
training@training-VirtualBox: /opt/Xilinx/Vivado/2016.4/data/xicom/cable_drivers/
lin64/install_script/install_drivers$
```

Vivado and SDK Permission Issues

When you invoke Vivado, you may see the following error message, indicating a permission issue in the directory hierarchy:

```

training@training-VirtualBox: /opt/Xilinx/Vivado/2016.4/data/xicom/cable_drivers/lin64/
CRITICAL WARNING: [Common 17-741] No write access right to the local Tcl store a
t '/home/training/.Xilinx/Vivado/2016.4/XilinxTclStore'. XilinxTclStore is rever
ted to the installation area. If you want to use local Tcl Store, please change
the access right and relaunch Vivado.
ERROR: [Common 17-1257] Failed to create directory '/opt/Xilinx/Vivado/2016.4/tc
lapp'.
start_gui
Error: Failed to save the Vivado user preferences file. Reason: '/home/training/
.Xilinx/Vivado/2016.4/vivado.ini (Permission denied)'
Failed to create the shortcut directory: '/home/training/.Xilinx/Vivado/2016.4/s
hortcuts'
Failed to create the layout directory: '/home/training/.Xilinx/Vivado/2016.4/lay
outs/application'
Failed to create the commands directory: '/home/training/.Xilinx/Vivado/2016.4/c
ommands'
Failed to create the layout directory: '/home/training/.Xilinx/Vivado/2016.4/lay
outs/'
Mar 29, 2017 10:59:04 AM java.util.prefs.FileSystemPreferences$1 run
INFO: Created user preferences directory.
Failed to create directory: /home/training/.profile
Error: Failed to save the Vivado user preferences file. Reason: '/home/training/
.Xilinx/Vivado/2016.4/vivado.ini (Permission denied)'
Error: Failed to save the Vivado user preferences file. Reason: '/home/training/

```

This indicates that the hidden folder named `.Xilinx`, where license and configuration information is stored, is read-only for the invoking username. One method to correct this is to change ownership of the directory hierarchy using root authority.

1. Change to the directory above the `.Xilinx` folder, which in this case is the `/home/<current user>` folder. Enter the following commands:

```

$ cd ~
$ sudo chown <current username> -R .Xilinx
$ sudo chgrp <current username> -R .Xilinx

```

2. Run the Xilinx Vivado and SDK 64-bit environment settings scripts in a terminal window to setup the environment path to the tools install folder so that Vivado and SDK can be located in the next steps.

```

$ source /opt/Xilinx/Vivado/2016.4/settings64.sh
$ source /opt/Xilinx/SDK/2016.4/settings64.sh

```

3. Launch Vivado or the SDK as usual using the fixed permissions on the .Xilinx folder.

To start Vivado from a Terminal window, enter the following command:

```
$ vivado &
```

To start the Xilinx SDK from a terminal window, enter the following command:

```
$ xsdk &
```

Install Missing Desktop Icons

In some cases Vivado and/or SDK desktop icons may be missing. These can be manually added through the use of some launcher shell scripts and by adding some desktop icon entries.

1. Change to the home directory for the current user.

```
$ cd ~
```

2. Using your favorite text editor edit a new shell script named **vivado_launch.sh**

```
$ vi vivado_launch.sh
```

Paste the following text into that script file, save the contents and exit. This creates a script that is capable of invoking the settings script needed for Vivado to launch correctly.

```
#!/bin/bash
. /opt/Xilinx/Vivado/2016.4/settings64.sh
vivado &
```

3. Using your favorite text editor edit a new shell script named **sdk_launch.sh**

```
$ vi sdk_launch.sh
```

Paste the following text into that script file, save the contents and exit. This creates a script that is capable of invoking the settings script needed for Xilinx SDK to launch correctly.

```
#!/bin/bash
# This is a workaround for Xilinx SDK and GTK incompatibility.
export SWT_GTK3=0
. /opt/Xilinx/SDK/2016.4/settings64.sh
xsdk &
```

4. Change permissions of both launcher scripts so that they can be executed.

```
$ chmod u+x vivado_launch.sh
$ chmod u+x sdk_launch.sh
```

5. Change directories to the home Desktop folder of the current user.

```
$ cd ~/Desktop
```


- Using your favorite text editor edit a new file named **Vivado.desktop**

```
$ vi Vivado.desktop
```

Paste the following text into that script file, save the contents and exit. This creates a desktop launcher icon that is capable of invoking the Vivado launcher script created in earlier steps.

```
#!/usr/bin/env xdg-open

[Desktop Entry]
Version=1.0
Type=Application
Terminal=false
Exec=/home/training/vivado_launch.sh
Name=Vivado 2016.4
Comment=Xilinx Vivado Design Suite 2016.4
Icon=/opt/Xilinx/Vivado/2016.4/doc/images/vivado_logo.ico
StartupNotify=true
```

- Using your favorite text editor edit a new file named **SDK.desktop**

```
$ vi SDK.desktop
```

Paste the following text into that script file, save the contents and exit. This creates a desktop launcher icon that is capable of invoking the Xilinx SDK launcher script created in earlier steps.

```
#!/usr/bin/env xdg-open

[Desktop Entry]
Version=1.0
Type=Application
Terminal=false
Exec=/home/training/sdk_launch.sh
Name=Xilinx SDK 2016.4
Comment=Xilinx Software Development Kit 2016.4
Icon=/opt/Xilinx/SDK/2016.4/data/sdk/images/sdk_logo.ico
StartupNotify=true
```

8. Change permissions of both launcher scripts so that they can be executed.

```
$ chmod u+x Vivado.desktop  
$ chmod u+x SDK.desktop
```

Once the execution permission is changed, the icons will populate on the Desktop with the correct graphics.



PetaLinux Installation

Installing PetaLinux tools on a supported Linux system should be straightforward, but depending on the precise configuration of your development system, there could be some challenges to create an optimal environment. This section outlines a few of the common issues that may be encountered. For more information on the installations of PetaLinux tools, please refer to Xilinx User Guide UG1144.

Add tftpboot Folder for PetaLinux

The PetaLinux tools will issue a warning occasionally when a tftpboot folder is not present for built package files to get copied into.

Create a tftpboot folder and assign read/write permissions with the following command:

```
$ sudo mkdir /tftpboot  
$ sudo chmod ugo+rw /tftpboot/
```

Install openssl Libraries for PetaLinux

As of Xilinx 2015.2 release, the PetaLinux tools require the **openssl** libraries on the host system which can be installed under Ubuntu using the following command:

```
$ sudo apt-get --yes install libssl-dev
```

Install Additional System Tools and Library Dependencies for PetaLinux

According to Xilinx User Guide UG1144, PetaLinux 2016.4 tools require some additional system tools and libraries to be installed on the host system. For exact system dependencies, refer to UG1144 document.

If you are using Ubuntu, this can be accomplished in a terminal window with the following command:

```
$ sudo apt-get --yes install tofrodos iproute gawk gcc git-core make  
net-tools libncurses5-dev tftpd zlib1g-dev flex bison lib32z1  
lib32ncurses5 lib32ncursesw5 lib32gomp1 xvfb chrpath socat autoconf  
libtool texinfo gcc-multilib libsdl1.2-dev libgl2.0-dev  
zlib1g:i386
```

Adjusting Permissions of Vivado and PetaLinux Install Folder

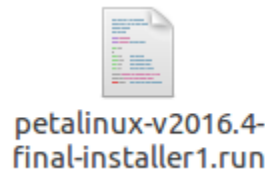
As of 2016.4, PetaLinux tools require the installation as a non-root user so the permissions of these folders under the /opt folder must be adjusted accordingly.

This can be accomplished in a terminal window with the following commands:

```
$ sudo chmod ugo+w /opt/  
$ mkdir /opt/petalinux-v2016.4-final  
$ sudo chmod -R ugo+w /opt/Xilinx  
$ sudo chmod -R ugo+w /opt/petalinux-v2016.4-final
```

Install PetaLinux into /opt/petalinux-v2016.4-final Folder

1. Copy the PetaLinux installer from your host system or Downloads folder to your VM desktop.



2. Open a terminal window and launch the installer while specifying the target install folder.

```
$ cd ~/Desktop
$ ./petalinux-v2016.4-final-installer1.run /opt/petalinux-
v2016.4-final
```

3. The Vivado 2016.4 Installer will take several minutes to verify the integrity of the packed installer and then extract itself.
4. Read the PetaLinux license agreements and, if you can accept the license conditions, press enter to view the license, use keyboard **page up/down** keys to scroll through the agreement, press the **q** key to quit viewing the agreement when finished, and press the **y** key if you accept the license conditions. If prompted, allow the installer to make changes to your development system.

```
training@training-VirtualBox: /media/sf_Xilinx_2016.4
Processing triggers for libc-bin (2.23-0ubuntu5) ...
training@training-VirtualBox: /media/sf_Xilinx_2016.4$ ./petalinux-v2016.4-final-
installer1.run /opt/petalinux-v2016.4-final
INFO: Checking installer checksum...
INFO: Extracting PetaLinux installer...
INFO: Installing PetaLinux...
INFO: Checking PetaLinux installer integrity...
INFO: Extracting Installation files...

LICENSE AGREEMENTS

PetaLinux SDK contains software from a number of sources. Please review
the following licenses and indicate your acceptance of each to continue.

You do not have to accept the licenses, however if you do not then you may
not use PetaLinux SDK.

Use PgUp/PgDn to navigate the license viewer, and press 'q' to close

Press Enter to display the license agreements
Do you accept Xilinx End User License Agreement? [y/N] > y
Do you accept Webtalk Terms and Conditions? [y/N] > y
Do you accept Third Party End User License Agreement? [y/N] > y
INFO: Checking installation environment requirements...
```

5. Be sure to run the `/opt/petalinux-v2016.4-final/settings.sh` script prior to attempting to use the PetaLinux tools for development.

Revision History

Version	Date	Details
1.0	Feb 19, 2015	VirtualBox 4.3, CentOS 6.5, CentOS 7
1.1	Feb 24, 2015	Device tree reverse compilation, Ethernet Adapter Names
1.2	November 2015	Removed CentOS 6.5 support, validated all instructions, and restructured/standardized document.
1.3	August 2016	CentOS 7 support, Ubuntu 16.04, CentOS PetaLinux 2016.2, and Xilinx Vivado/SDK 2016.2
1.4	April 2017	Updated for PetaLinux 2016.4 and Xilinx Vivado/SDK 2016.4 tools, removed instructions and support for CentOS, removed unnecessary sections