

# White Paper

## Whitepaper

### *How to Create a Windows Server Virtual Laboratory Environment Using VirtualBox*

March 2011

"If everyone is thinking alike, then somebody isn't thinking." *Gen. George S. Patton, Jr.*

Copyright © 2011 Michael Wright. All rights reserved.  
email: [mike@goshenpass.com](mailto:mike@goshenpass.com) web: [www.goshenpass.com](http://www.goshenpass.com)



## Table of Contents

---

Background .....	1
Introduction .....	1
Pre-Requisites .....	2
Defining and Configuring the Lab Infrastructure .....	2
Creating a Baseline Server .....	4
Virtual Networking Overview .....	10
Cloning a Baseline Server .....	12
Copyright & Disclaimer .....	16
About the Author .....	16

## Background

---

With the ever increasing speed at which new technologies are being released today, it is becoming increasingly difficult for support engineers to maintain their proficiency at supporting these technologies. Besides formal education, the next method most often used by support engineers is on-the-job training. However, this type of training can often times have adverse effects in production environments. Therefore the support engineer **NEEDS** a way to learn *effectively*. Reading alone will not provide everything needed. Vendor conferences and presentations will not fill the void either. So what is the support engineer to do? It is this question that reminds me of the ancient Chinese proverb that says,

***Tell me, I'll forget. Show me, I'll remember. Involve me, I'll understand.***

This paper is meant to provide a means to *involve* you and to allow you to *learn by doing* by giving you the steps to create your very own Virtual Computing laboratory. Even though this paper focuses primarily on creating a **Citrix XenApp infrastructure**, the methods shown here can be used for numerous scenarios besides just Citrix.

## Introduction

---

In order to create a Virtual Computing laboratory, you will need a few essentials – specifically you will need hardware, operating system software and a platform on which to create your virtual environment. I have selected Oracle VirtualBox as the platform for this paper. Other products can be used such as VMWare, XenServer, MS Virtual PC and MS Virtual Server – but I have found that VirtualBox is the quickest, easiest and most cost efficient way to create a virtual computing laboratory. Oracle VirtualBox is an open source package for creating virtualized servers. Presently, VirtualBox runs on Windows, Linux, Macintosh and OpenSolaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Server 2008, Vista, Windows 7), DOS/Windows 3.x, Linux (2.4 and 2.6), Solaris and OpenSolaris, and OpenBSD. VirtualBox supports 64-bit guest operating systems, even on 32-bit host operating systems, provided that the following conditions are met:

- ❖ You need a 64-bit processor with hardware virtualization support.
- ❖ You must enable hardware virtualization for the particular VM for which you want 64-bit support; software virtualization is not supported for 64-bit VMs.
- ❖ If you want to use 64-bit guest support on a 32-bit host operating system, you must also select a 64-bit operating system for the particular VM. Since supporting 64 bits on 32-bit hosts incurs additional overhead, VirtualBox only enables this support upon explicit request.
- ❖ On 64-bit hosts (which typically come with hardware virtualization support), 64-bit guest operating systems are always supported regardless of settings, so you can simply install a 64-bit operating system in the guest.

You can very easily create systems that will run almost anything. I have used the configuration in this paper to create virtual labs for information assurance, penetration testing, teaching

network security courses and learning Citrix XenApp. Your only limitations are your CPU(s) and total memory. I was able to create a virtual domain utilizing only two laptops – each laptop had 4GBs RAM and each had a dual-core processor – AND – I did not have to re-configure my laptops.



**NOTE:** This paper will not cover how to configure VirtualBox virtual machines to act as a router, email server, certificate server or how to completely isolate your virtualized environment – *this is only a basic lab configuration* – and should NOT be used for in-depth security penetration testing and/or risk analysis using network security tools.

I suggest that you download the latest VirtualBox manual and learn about the capabilities of using VirtualBox. You can use and configure several different hardware configurations. You can use this package to even read and use other virtual images (like .VDI, .VMDK, .VHD, .HDD). Many vendors now provide evaluation virtual appliances that can be used for testing and training. In many cases, VirtualBox can read and use these vendor supplied virtual machines. But why do I like VirtualBox? It's free and it's simple.

## Pre-Requisites

---

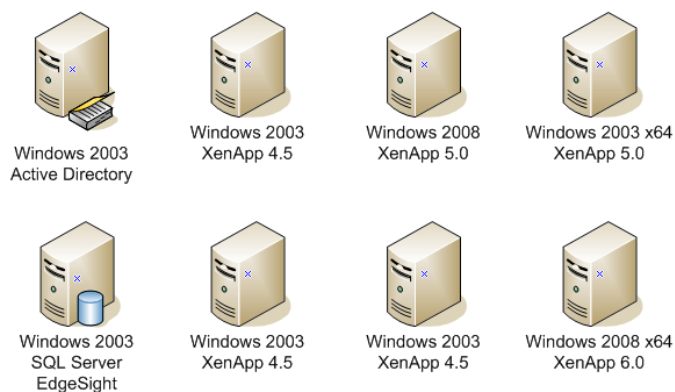
At a minimum, you will need at least one laptop with a decent processor, at least 80GB of storage space and at least 2GB of RAM. This will allow you to create at least three servers that can emulate a Citrix environment. My configuration utilizes two networked laptops (each with dual core processors and 4GB RAM each). My configuration allows me to easily create and use an eight server domain environment that has every XenApp product installed. You will also need installation media (and in some cases license keys). Evaluation keys are typically available for almost all products.

## Defining and Configuring the Lab Infrastructure

---

Before you begin, you need to define what you want to accomplish. So – **MAKE YOUR PLAN!**

Identify what you need – it may be helpful to create a network diagram. Below is a basic virtual lab and one that I typically use for most of my Citrix work.



Download the latest version of VirtualBox from <http://www.virtualbox.org/> and install it. Just follow the defaults for the installation.



**NOTE:** You will need to be able to allocate at least 10GB of storage space per server and have at least 256M RAM per server.

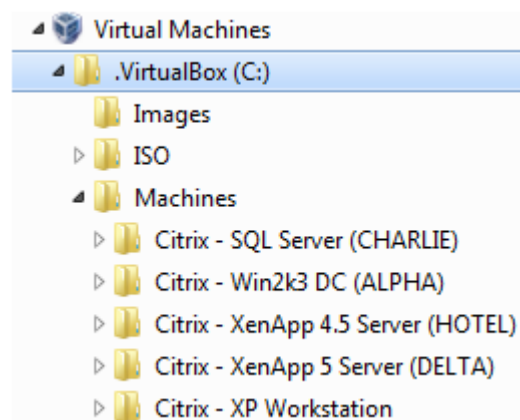
In my “network diagram” the most common server is a Windows 2003 server. In this case I used Window 2003 Enterprise. This is the first server I created, and I used this image as a base from which I could easily clone more servers.

Before you start creating servers, you must have either the installation media for the servers you want to install or have downloaded the vendor evaluation server files. If you are going to create your own servers from scratch, I recommend creating .iso images from the installation CDs. I use *ImageBurn* to create my .iso images. *ImageBurn* is an open source product that can be downloaded at <http://www.imgburn.com/> You do not HAVE to create .iso images, but if you do not, you must continually swap CDs out. Since I’m lazy, and I also build images quite a bit, I like using the .iso images. So, if you are going to utilize .iso images, you should have them available in a central location as described below.



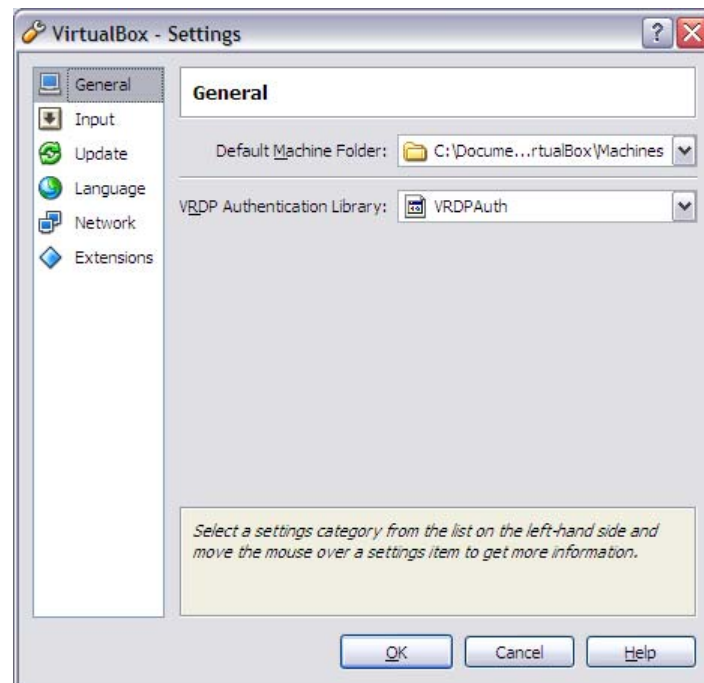
**NOTE:** There are many pre-configured virtual machines now provided by vendors. VirtualBox can typically read these disk files (even though they may say you need to use it with VMWare or MS Virtual Server). For example, you can download a [Windows Server 2008 Enterprise Edition x86 \(Full Install\) VHD](#) for use. The use of these pre-configured virtual appliances is not discussed in this version of the whitepaper.

The last thing you should consider is organization of your files. My setup is like:



My root folder is **.VirtualBox** with the subfolders **Images**, **ISO** and **Machines**. I store all hard disk images in the **Images** folder. I store CD\DVD images in the **ISO** folder and store all my virtual machine configurations in the **Machines** folder (with each Virtual Machine having its own subfolder). You do not need to create the individual folders for your lab servers, but creating the folders **Images**, **ISO** and **Machines** will make things easier later. You should

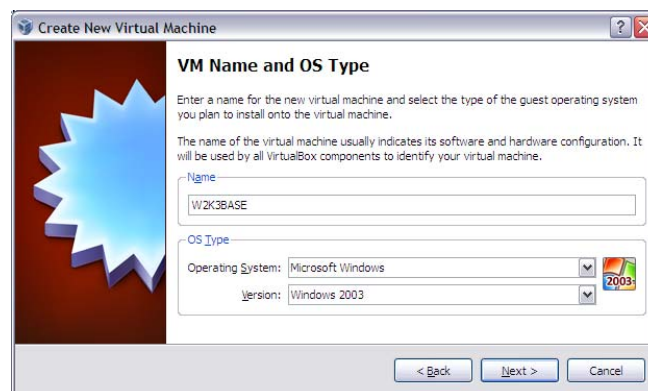
configure VirtualBox so that it automatically selects the Machines folder as your Default. From the file menu, select **File | Preferences**. Set your **Default Machine Folder**. Now for the fun!



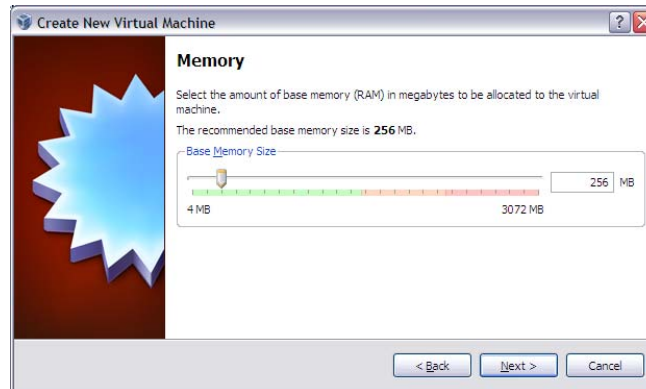
## Creating a Baseline Server

The whole reason behind a virtual machine laboratory is so that you can quickly create and destroy servers and not have to worry about the consequences. The key to this process is having a baseline server from which you can clone other servers. Now that you have your .iso images loaded or your installation media ready, you will create your baseline image.

- 1) Click on the **New** button to start the New Virtual Machine Wizard. Click **Next**. You will need to enter a Name for your virtual machine. I kept things simple and just entered **W2K3BASE**. This will automatically create a subfolder under your default machine folder with the name choose. You will also select the Operating System and Version (in this case Microsoft Windows and Windows 2003). Click **Next**.



- 2) You will need to set the VM's memory – I typically start with 256MB (this can be adjusted later). Remember, this memory will be taken away from your physical memory **ONLY** while the VM is running. Click **Next**.



- 3) Now you will create your boot hard disk. For almost all of my VM's, I only use a single hard disk. In many cases this does not exactly mirror environments I support, but for lab purposes a single hard drive is typically adequate. Make sure **Boot Hard Disk** is checked and select **Create new hard disk**. Click **Next**.



- 4) This will automatically start the *New Virtual Disk Wizard*. Click **Next**. There are two types of storage: *Dynamic* and *Fixed*. I always select **Dynamically Expanding Storage**. Click **Next**.

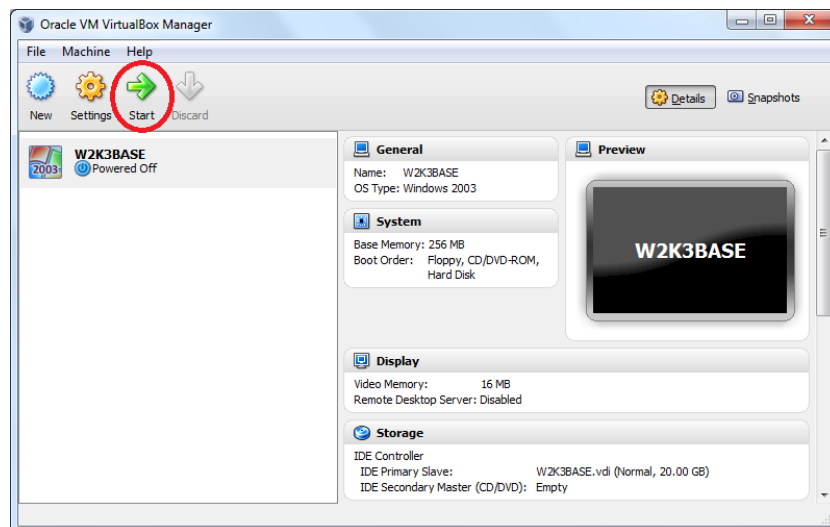




- 5) For the disk size, I typically set it to 20.00GB. Since it is dynamically sized, it will not necessarily utilize all 20GB and will grow as needed. If you have set your default machine folder, the disk image will be stored in the subfolder created in step 1. You may also select a different location where you want the image to be saved by clicking on the folder icon (circled in red). Click **Next** then click **Finish**.

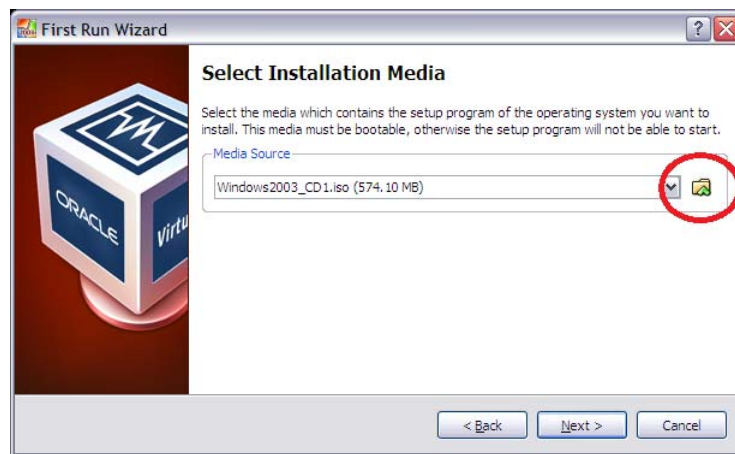


- 6) That completes the creation of the hard disk wizard and also completes the initial configuration of the Virtual Machine. There are some other “tweaking” type settings I will cover later, but for your baseline image, that’s all you need.
- 7) Now, you need to power on your new server. You should see your new server in VirtualBox. To start your server, highlight the server and click the **Start** button.



- 8) As soon as the server starts, you will get the **First Run Wizard**, click **Next**. The next option is where you will select your installation media (either an .iso, .dmg, or .cdr image OR a CD/DVD loaded on the host). Select the path by clicking the folder icon circled in red below.





9) Once you select a CD image, click **Next**.



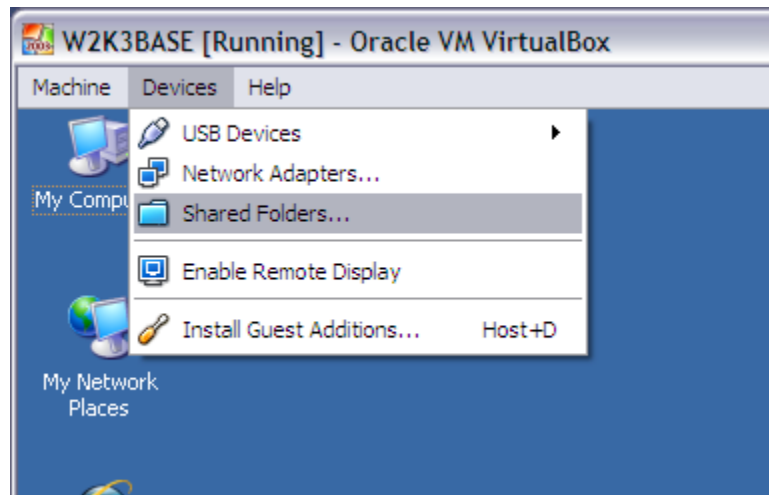
10) The media will be temporarily mounted on your virtual machine and the machine will start execution after you click **Finish**.



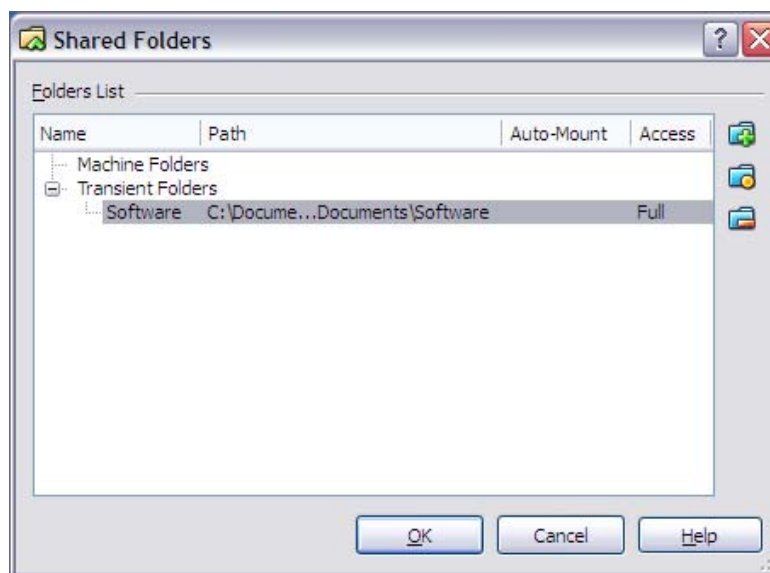
**NOTE:** VirtualBox reserves a special key on your keyboard for itself: the "**host key**". By default, this is the *right Control key* on your keyboard. You can change this default in the VirtualBox Global Settings. In any case, the current setting for the host key is always displayed *at the bottom right of your VM window*, should you have forgotten about it. To send CTRL-ALT-DEL you would hit Host + Delete. These shortcuts are also available from the VM VirtualBox file menu, select **Machine** for a drop down list of commands.

11) The server will boot from the installation media and begin your operating system installation. Proceed with the server installation. I typically install *IIS* and *Terminal Services* during the install. You can accept default settings for everything, because you will be changing this later. As for a computer name, you should continue to use something simple, like **W2K3BASE**. Continue with your Windows installation until it is complete and asks you to reboot. *When you do reboot the machine, the .iso image will still be mounted and visible to the OS – so DO NOT boot from the CD image again, or you will run setup again!* You will need to remember the password you use for the default Administrator account.

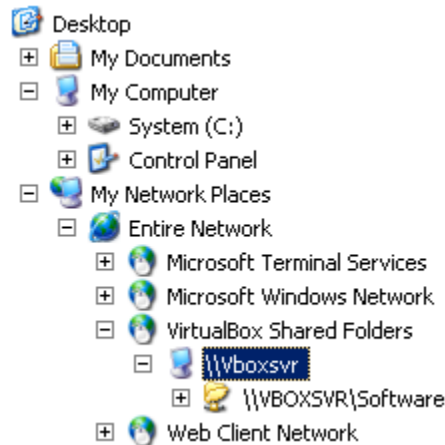
- 12) The default VirtualBox network configuration for a new machine is NAT. The IP address assignment and NAT translation is automatically handled by VirtualBox. Since you did not change any network settings during the Windows install, Windows should obtain an IP address from VirtualBox like 10.0.2.15. The only issue you may run into is your firewall program blocking outbound access from VirtualBox (which is easily fixed).
- 13) You can either download patches from the Internet or copy files from your host computer to your virtual machine. To copy files from your computer you must set up *Shared Folders*. From the VM VirtualBox file menu, select **Devices | Shared Folders**.



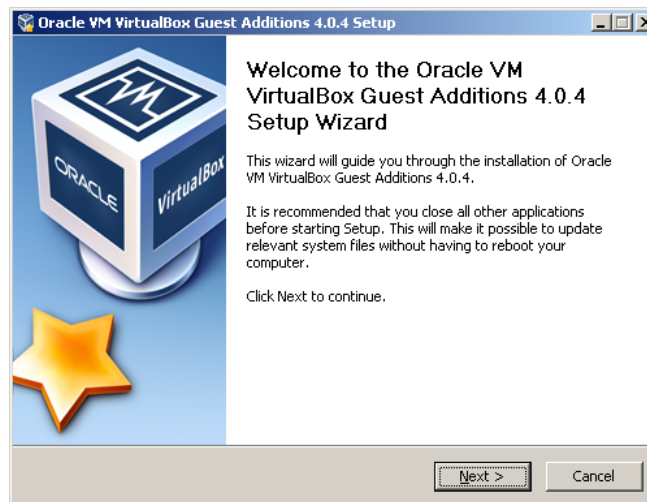
- 14) Choose a folder on your host computer from which you want to copy files – this folder *does not* need to be shared on your host computer. *Machine Folders* are configured prior to the virtual machine being turned on where *Transient Folders* can be configured while the virtual machine is running. Machine Folders are persistent where Transient Folders are not. In this case, I have a folder on my host under my “My Documents” folder called software, so I have set that up as a Transient folder.



- 15) When you open Windows explorer in your virtual machine, you can access files directly on your host computer by expanding My Network Places and going to the *VirtualBox Shared Folders* node like below. You will see the name you have given your folder under the `\\Vboxsvr` node.



- 16) Verify Internet connectivity and install all the latest Microsoft patches (I keep my lab servers completely patched so that I can see if a patch creates a problem).
- 17) Once you have installed all hotfixes, service packs, etc. – you may want to copy some system tools or scripts to you may want to use – I have some custom scripts that I utilize, but primarily – I use the software utilities from *SysInternals*, the *Windows Support tools* and the *Windows Resource kit*.
- 18) The last item you need to install is the VirtualBox Guest Additions. The setup files are included with VirtualBox and can be executed from within your virtual machine by selecting **Device | Install Guest Additions** from the VM VirtualBox file menu. If you do not have autorun disabled, the setup wizard will start automatically (VirtualBox mounts the setup files as a CD in your virtual machine) – otherwise, run the setup.exe file from CD in the virtual machine.



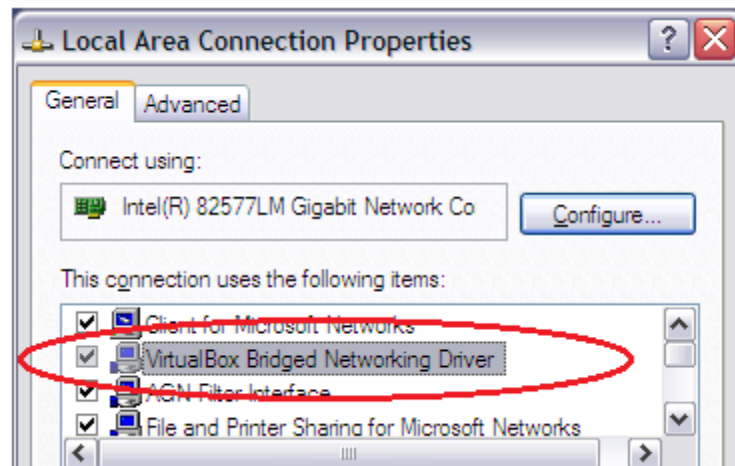
- 19) Shutdown the virtual machine. Move the .vdi file (in my case **W2K3BASE.VDI**) to the **IMAGES** folder. Change the properties of the .vdi file to read-only. You want to do this so that you do not accidentally associate your baseline image to a virtual machine.
- 20) Remove the Baseline Virtual Machine from the VM VirtualBox Manager. Select the virtual machine, right click and select **REMOVE**.

## Virtual Networking Overview

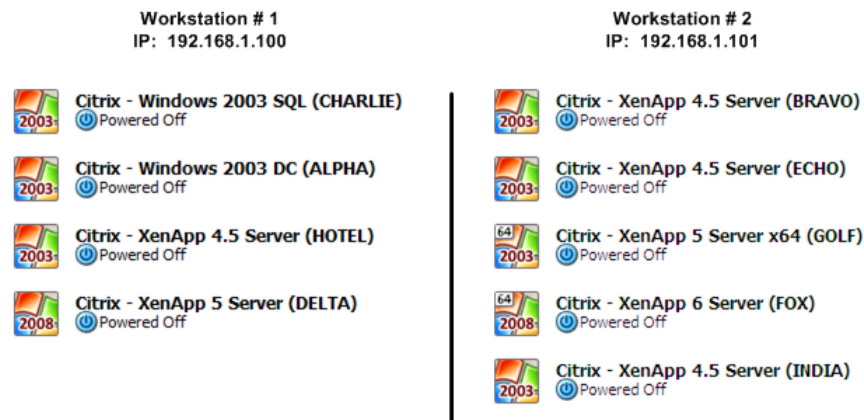
---

Before you clone your baseline image, you should understand the networking models available in VirtualBox. From the VirtualBox manual, there are five options:

- ❖ **Not attached** In this mode, VirtualBox reports to the guest that a network card is present, but that there is no connection – as if no Ethernet cable was plugged into the card.
- ❖ **Network Address Translation (NAT)** Allows you to browse the Web, download files and view e-mail inside the guest. NAT configured VMs on the same host can communicate to each other.
- ❖ **Bridged networking** This is for more advanced networking needs such as network simulations and running servers in a guest. When enabled, VirtualBox sets up an additional, software-based network interface on the host to which the virtual machine is connected (shown below). Bridged configured VMs on different hosts can communicate to each other as long as the hosts are in the same network.



- ❖ **Internal networking** This can be used to create a different kind of software-based network which is visible to selected virtual machines, but not to applications running on the host or to the outside world.
- ❖ **Host-only networking** This can be used to create a network containing the host and a set of virtual machines, without the need for the host's physical network interface. Instead, a virtual network interface (similar to a loopback interface) is created on the host, providing connectivity among virtual machines and the host.



If you have a multiple host configured virtual lab where two or more workstations are running VirtualBox and have virtual machines configured (as shown in the figure above), your easiest network configuration option is to use *Bridged networking*. This mode has the following advantages:

- ❖ Allows virtual machines from different hosts to communicate to each other as long as the hosts are on the same network.
- ❖ Allows access to the virtual machines from the host and other virtual machines via RDP, http, etc.

This paper only addresses the *Bridged networking* mode. Other modes can be used, but additional configuration is required in order to achieve the same results as what is presented here. I also recommend that you utilize static IP addresses (which must be available on your host network.) For my Citrix XenApp laboratory, I use the following configuration:

Host Name	IP Address	Function
Alpha	192.168.1.50	2003 - Domain Controller, TS License Server, Certificate Server
Bravo	192.168.1.51	2003 - XenApp 4.5
Charlie	192.168.1.52	2003 - SQL Server 2005, EdgeSight Console
Delta	192.168.1.53	2008 - XenApp 5.0
Echo	192.168.1.54	2003 - XenApp 4.5, Password Manager
Fox	192.168.1.55	2008x64 – XenApp 6.0
Golf	192.168.1.56	2003x64 – XenApp 5.0
Hotel	192.168.1.57	2003 – XenApp 4.5
India	192.168.1.58	2003 – XenApp 4.5

## Cloning a Baseline Server

---

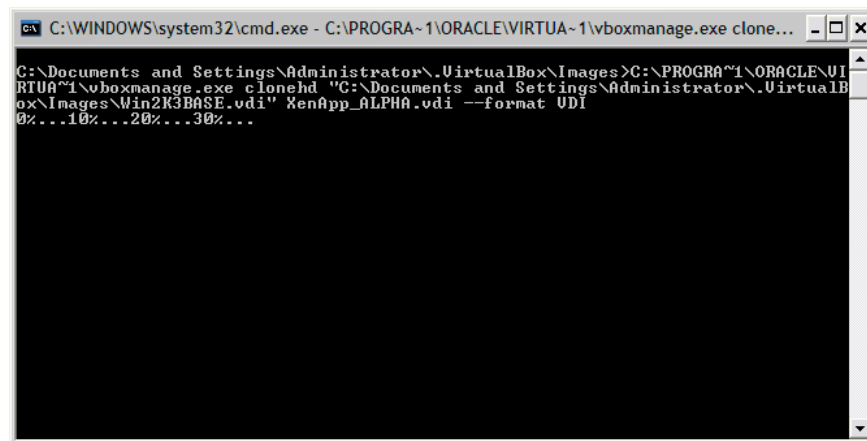
Typically, in a production environment, you must generalize any image that will be cloned by using the Microsoft System Preparation (SysPrep) utility. For the purposes of our lab, we do not need to that.

To make a clone of your BASELINE server, you must use the VirtualBox command line utility **vboxmanage.exe**. The **vboxmanage.exe** file can be found in the “Program Files\oracle\virtualbox” folder. The command line looks like this:

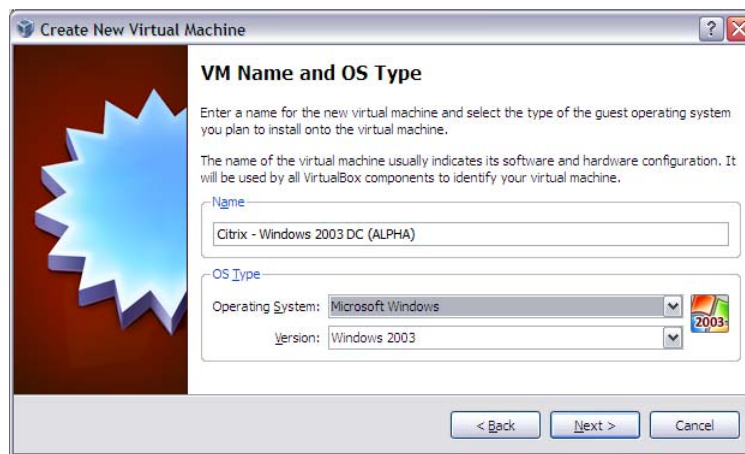
```
vboxmanage.exe clonehd BASELINE.VDI NEWSERVER.VDI --format VDI
```

Where BASELINE.VDI is your baseline image server and NEWSERVER.VDI will be your new server image. The image name you give your cloned .vdi file should be something representative of the new server; either function or server name (i.e. I use the phonetic alphabet for all my server names; ALPHA, BRAVO, CHARLIE, DELTA, etc. – this format is effective and simple.) In this case, I have used **XenApp\_ALPHA.VDI** as the image name.

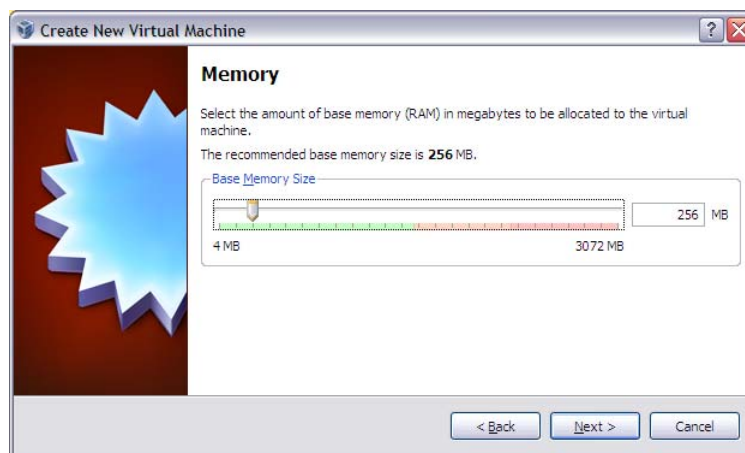
- 1) Open a command prompt and enter the command and parameters (you will need to use the full path to **vboxmanage.exe**). Once you hit enter, the server image will begin cloning and display a progress indicator similar to the one shown below.



- 2) It may take several minutes for the cloning process to complete. Once the process is complete, you will need to create a new server following the steps similar to the ones you used in **Creating a Baseline Server** above.
- 3) Click on the **New** button to start the New Virtual Machine Wizard. Click **Next**. You will need to enter a Name for your virtual machine. This is where you should give your Virtual Machine a descriptive name. Whatever name you provide here will display in the VirtualBox console and will also create a subfolder under the default directory for your virtual machines. In the sample below, I have entered **Citrix – Windows 2003 DC (ALPHA)**. This is *NOT* the actual computer name, just the description of your virtual machine.



- 4) You will need to set the VM's memory – I typically start with 256MB (this can be adjusted later). Remember, this memory will be taken away from your physical memory ONLY while the VM is running. Click **Next**.

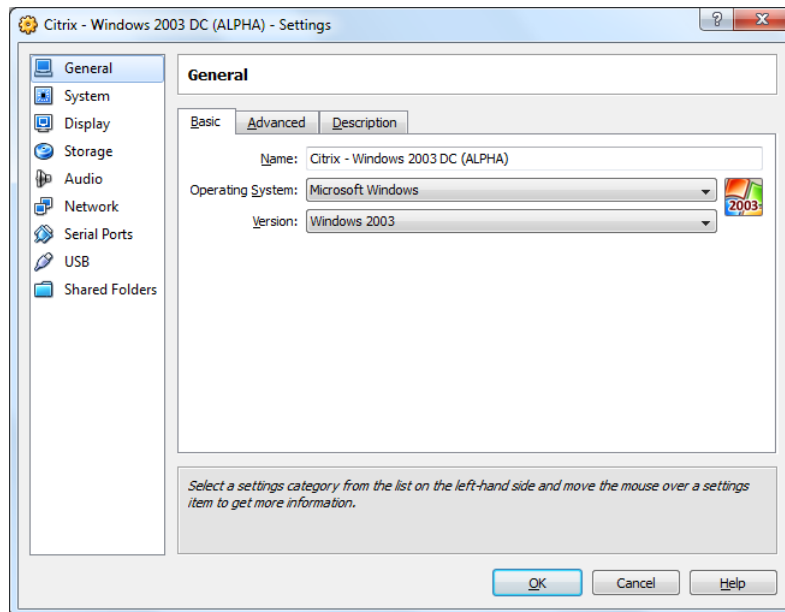


- 5) Now you will need to associate the hard disk (or image) that you cloned in step you're your virtual machine. If the hard disk image appears in the drop down list, you can either select it – otherwise, you will need to select the images folder where the .vdi image is stored and manually select it. Click **Next**. Then click **Finish**.

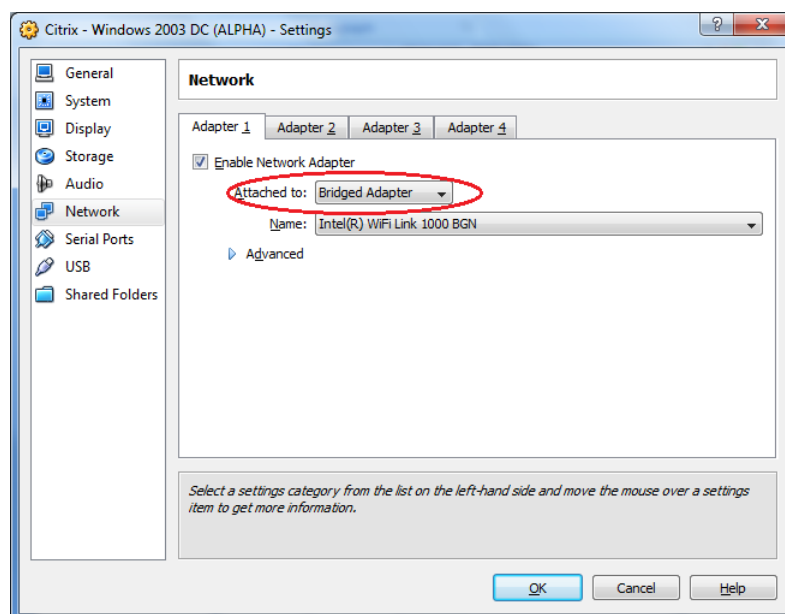




- 6) Your new cloned virtual machine is now created and you must make some changes so that you can use the machine in your environment. Highlight your VM and click the **Settings** button. This screen (shown below) allows you to change the hardware configuration of your virtual machine.

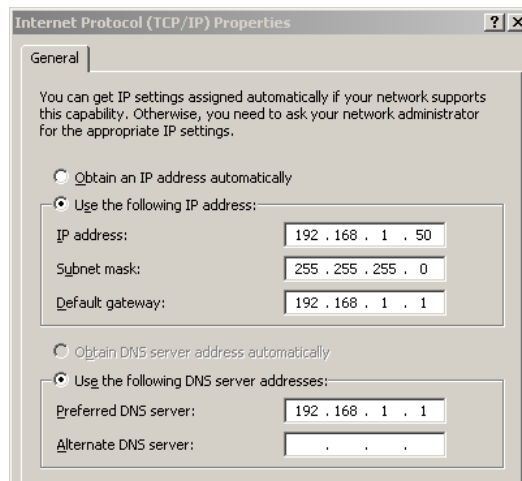


- 7) Since this lab does not require complete isolation from your host, I recommend using the bridged network adapter setting (as described in the previous section) since it is the easiest to configure. Select the **Network** option. Change the **Attached to** setting to **Bridged Adapter**. Ensure that the **Name** is the network card on your host that allows outbound Internet access.



- 8) Click **OK** to save your settings and start your virtual machine.

- 9) Logon to your new virtual machine. First, you will want to assign a static IP address. From the previous sections, I recommended that you create a layout of what you want your lab to look like, and to assign computer names and IP addresses to your virtual machines. Remember, these IP addresses **MUST** be on your local network in order to use virtual machines hosted on multiple hosts and to access the Internet. Following my example, I assigned the static IP address: 192.168.1.50 Mask: 255.255.255.0 Gateway: 192.168.1.1 (Note: The gateway address and DNS server is simply the address of my router). Save your settings and verify that you can access the Internet.



- 10) The last item you will need to do is to rename your server. In this case, I renamed this server to **ALPHA**. Reboot your virtual machine.
- 11) That's it. You have successfully configured a cloned virtual image and can now configure the server for whatever role you need it to perform!



**NOTE:** You can change almost every hardware aspect of your virtual machine. You can use virtual SCSI controllers, change the type of IDE controller, add more network cards – the list is huge. There are several configurations you can utilize. The virtual machines for a network security course I taught were configured to use para-virtualized network cards and the domain controller also served as a router for the virtual network to connect to the Internet. The virtual network was completely isolated from the host in every way – primarily because we were using tools such as BackTrack, Cain & Abel, L0phtcrack and other assorted tools that don't play well with intrusion detection systems and system security suites.

## Copyright & Disclaimer

---

This document is copyright © 2011 Michael Wright. Permission is hereby granted to publish this document subject to the following terms and conditions:

- ❖ Credit shall be provided to the author as the source of the excerpt. Please include the following line somewhere in your forthcoming text: 'Published with the permission of Michael Wright.'
- ❖ Permission is granted to use the excerpts in the document referenced herein for worldwide distribution, in the English language, for this edition and all or any subsequent editions of identical or similar format.
- ❖ The copyright of any third parties must not be infringed as a result of this granted permission, and in the event that it does, the author cannot be held responsible and will not in any way be held liable for any consequences.

In no event shall the author be liable for any found errors contained herein or for any direct, indirect, special, incidental or consequential damages (including lost profit or lost data) whether based on warranty, contract, tort, or any other legal theory in connection with the furnishing, performance, or use of this material. The information contained within this document may be subject to change without any notice. All other product names mentioned herein are trademarks or registered trademarks of their respective owners.

## About the Author

---

**Michael Wright**, CISSP, MCSE, CCEA, CCNP  
Information Security Specialist



Michael is currently a consultant with IBM providing support for Citrix technologies. With more than 20 years of professional experience in the information technology fields, Michael has spent the last eleven years working as both an IT and INFOSEC consultant working on projects for a variety of organizations, including the Pennsylvania state government, Pennsylvania Turnpike Commission, Computer Sciences Corporation, United States Navy and the Defense Information Systems Agency (DISA).

Michael is a co-author of the book, *Securing Citrix XenApp Server in the Enterprise* which was used as one of the supporting resources for establishing the official Department of Defense security guidance for Citrix XenApp Server. In addition, he has provided advice and recommendations regarding best practices for implementing security in Citrix XenApp environments to various Department of Defense organizations.

His professional affiliations include the Information Systems Security Association (ISSA), Institute of Electrical and Electronics Engineers (IEEE) and InfraGard. Michael guest lectures on topics of information security and information assurance and serves as a volunteer for the Boy Scouts of America. A U.S. Marines veteran, Virginia Military Institute alumnus and a magna cum laude graduate from Harrisburg University of Science and Technology, Michael currently resides in Central Pennsylvania with his wife, Linda.