Physical to Virtual for Legacy Windows

This is an overview of how to create a digital clone of a PC running a legacy 32-bit version of Microsoft Windows, then deploy and run the clone in a virtual computing environment.

All the tools used are free, with no license fees, and almost all are open source.

You can use them, too... just like we do. Most importantly, crafting solutions with free tools makes them fully portable and serviceable by anyone with sufficient technical skill, so there's never any vendor lock-in.

Questions? We're happy to help... visit us at https://xviewsolutions.com

Start by preparing the computer you want to digitally clone.

These Microsoft IDE (Integrated Drive Electronics) storage device driver files must be present for the QEMU virtual machine to be able to successfully boot the digital clone.

```
C:\WINDOWS\system32\drivers\atapi.sys
C:\WINDOWS\system32\drivers\intelide.sys
C:\WINDOWS\system32\drivers\pciide.sys
C:\WINDOWS\system32\drivers\pciidex.sys
```

If the driver files are not already present, retrieve them from an installation CD or DVD of the legacy Windows version installed in the PC, or find and download the files from the Internet.

Retrieve a copy of Disk2vhd from Microsoft Sysinternals. If the current version available for download does not work with the legacy Windows version installed in your PC, search the Internet for an earlier version of Disk2vhd that will work for you. The Wayback Machine is a good place to look.

Microsoft Sysinternals https://learn.microsoft.com/en-us/sysinternals
Wayback Machine https://web.archive.org

The destination for the VHD (Virtual Hard Disk) file you are going to create should be a removable storage device or a Windows network share mapped to a drive letter.
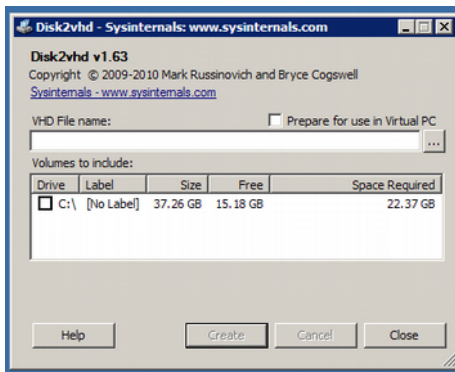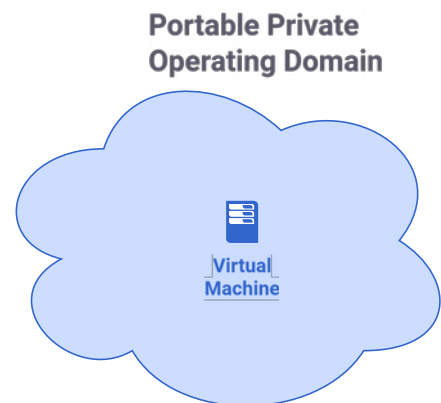

Run Disk2vhd as an Administrator, select the disk volumes to be copied, set the drive path and name of the VHD file, then click Create.

Creating the clone may take many hours and should not be interrupted. You may want to perform the task overnight.

Ultimately, the VHD file you create needs to end up in a virtual computing environment or some other host environment running a GNU/Linux operating system.

When working in the host environment we'll use some of the virt-tools collection – specifically libguestfs, libvirt and QEMU – to create, configure and launch a virtual machine.

Virt Tools https://virt-tools.org
libguestfs https://libguestfs.org
libvirt https://libvirt.org
QEMU https://www.qemu.org

You can install the tools using the package manager (e.g. RPM or APT) of the host environment's GNU/Linux distribution.

The /dev/fuse and /dev/kvm character devices must be present in the GNU/Linux host environment and your user account must have read and write permissions for those devices.

```
crw-rw-rw-  1 root root 10, 229 Jul 26 15:11 /dev/fuse
crw-rw-rw-+ 1 root kvm  10, 232 Jul 29 00:04 /dev/kvm
```

Open a bash shell in the host environment and convert the VHD file to RAW format – which is suitable for use with QEMU and easily exportable to other virtual disk formats – using the qemu-img utility.

```
Command > qemu-img convert -f vpc -O raw /path/to/file.vhd /path/to/file.img
```

Obtain information about the virtual disks, partitions and file systems contained in the RAW image file with the virt-filesystems utility.

```
Command > virt-filesystems -a /path/to/file.img --all --long --human-readable --uuid
```

Verify the required IDE storage device drivers are present in the RAW image file using the following bash shell commands.

```
for each in atapi.sys intelide.sys pciide.sys pciidex.sys; do
  virt-ls -a /path/to/file.img /WINDOWS/system32/drivers | grep ${each}
done

atapi.sys
intelide.sys
pciide.sys
pciidex.sys
```

Obtain the Windows registry change file that needs to be applied to the RAW image so that Windows will use the required IDE disk drivers.

Free download at https://noc1.org/public/qemu/share/windows/mergeide.reg

Use the virt-win-reg utility to search the Windows registry in the RAW image file to find the control set that Windows will use when it starts under the QEMU machine emulator.

```
Command > virt-win-reg /path/to/file.img "HKEY_LOCAL_MACHINE\SYSTEM\Select"

[HKEY_LOCAL_MACHINE\SYSTEM\Select]
"Current"=dword:00000001
"Default"=dword:00000001
"Failed"=dword:00000000
"LastKnownGood"=dword:00000002
```

In this example the current and default is control set one; yours may differ.

If it does, edit the mergeide.reg file and change every occurence of ControlSet001 in the file to whatever control set Windows is using in your RAW image. For example ControlSet002 or ControlSet003.

Modify the Windows registry in the RAW image using the virt-win-reg utility.

```
Command >  virt-win-reg --merge /path/to/file.img /path/to/mergeide.reg
```

Modifying the registry will make Windows use the required IDE disk device drivers when it starts in the QEMU machine emulator instead of whatever disk device drivers it was configured to use in the original physical PC.

At this point you should be able to start the virtual machine and manage it from the command line using the qemu-kvm application. Here we are starting a virtual machine with one gigabyte of random access memory, a US English keyboard, a USB mouse and a VGA graphical display.
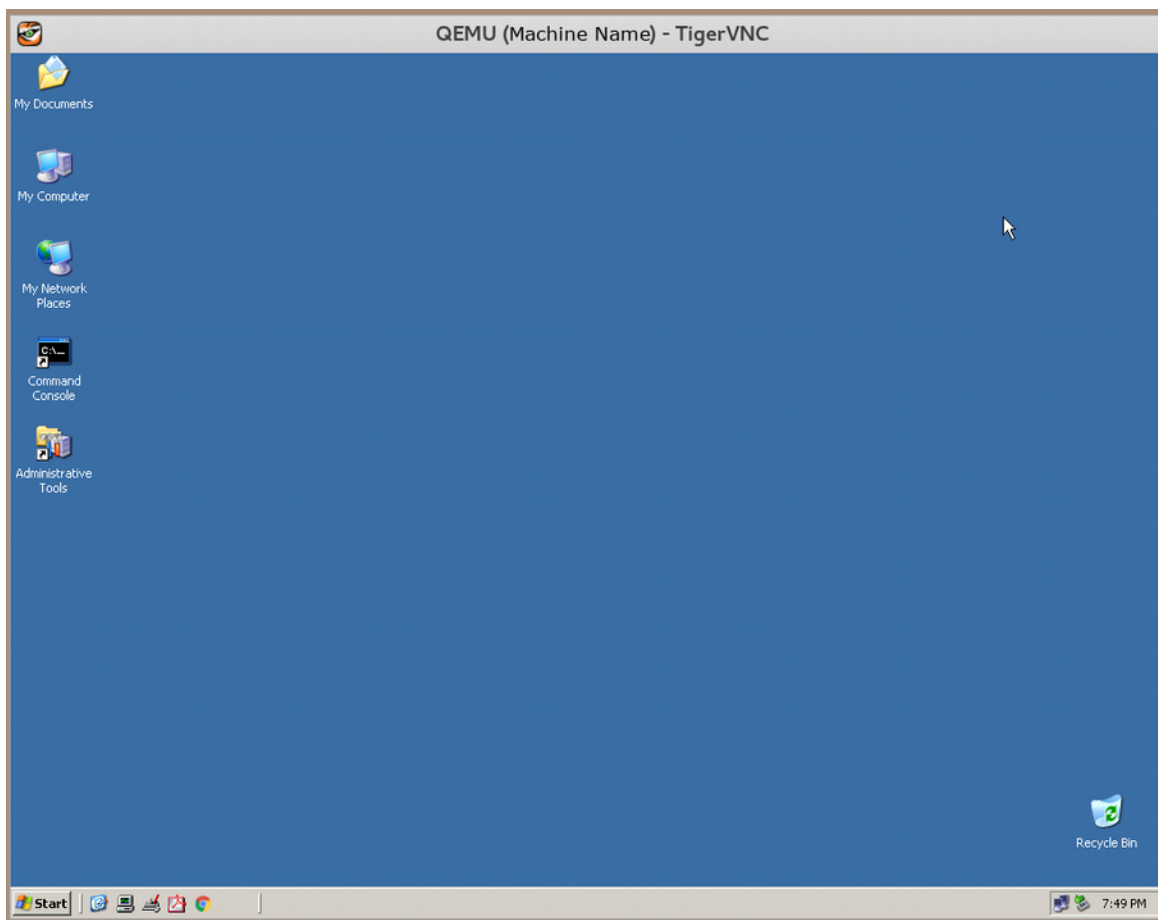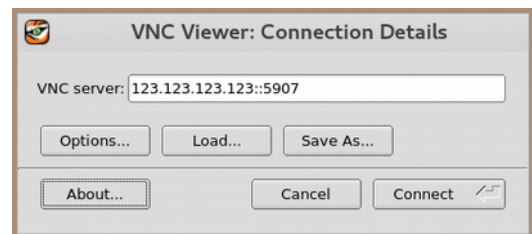
```
Command > /path/to/qemu-kvm -name 'computer name' -enable-kvm -vga std \
          -k en-us -vnc 0.0.0.0:7 -m 1024M --usbdevice tablet -monitor stdio \
          -drive format=raw,file=path/to/file.img
```

After the virtual machine starts you should find yourself at the QEMU monitor prompt.

```
QEMU 2.12.0 monitor - type 'help' for more information
(qemu) ▮
```

In this example we made the virtual machine accessible on port 5907 of the GNU/Linux host's network address using the VNC (virtual network computing) protocol. Start a VNC viewer program and point it there to connect to the virtual machine's Windows desktop. Log in as you would on the original Windows PC.

TigerVNC https://tigervnc.org
TightVNC https://www.tightvnc.com
UltraVNC https://uvnc.com
VNC Viewer https://vncviewer.org

From the QEMU monitor prompt you may safely power off the virtual machine using the system_powerdown command.

```
QEMU 2.12.0 monitor - type 'help' for more information
(qemu) system_powerdown
```

Questions? We're happy to help... visit us at https://xviewsolutions.com