



# Virtualizing a RedHawk™ Linux® 5.1 Guest on a RedHawk 5.1 Host

Jason Baietto  
Chief Systems Architect  
Concurrent Computer Corporation



## ABSTRACT

**With a fixed capital equipment requirement and a need for additional real-time Linux environments, Concurrent was recently approached by a prominent aerospace customer for a recommendation to virtualize RedHawk Linux. This paper discusses the journey from research to recommendation of Concurrent's suggested RedHawk Linux 5.1 virtualization solution.**

## INTRODUCTION

The Concurrent evaluation team's task was to research and recommend a working method for virtualizing a RedHawk 5.1 guest on a host system running RedHawk 5.1. We reviewed several virtualization solutions that are currently available (VMware™, Xen™, QEMU™, Virtual PC™, VirtualBox™, KVM, etc.) and determined that Sun's VirtualBox OSE ("Open Source Edition") appeared to be the best option due to the fact that it is nearly as mature from an interface perspective as VMware and yet it is 100% GPL software. As RedHawk runs a derivative distribution kernel based on the latest kernels from *kernel.org*, a solution that is 100% open source is definitely a requirement for Concurrent.

## OVERVIEW

The first step was to get RedHawk 5.1 working as a guest under Sun's pre-built VirtualBox binaries for Fedora Core 8. These binaries have the advantage of not requiring compilation, however they contain license restrictions that prevent them from being part of a commercial product without a Sun partnership. Thus, this version of VirtualBox was only used for the initial investigation stages.

After spending some time with the VirtualBox "End User Documentation" we were able to get RedHawk 5.1 installed and properly working as a guest under a Fedora Core 8 host. For the details of how to properly install and configure the guest, see the "Install RedHawk 5.1 as a Guest" section of this document.

The second step was to build VirtualBox on a RedHawk 5.1 system so that the same version of RedHawk 5.1 could run as both the host and guest at the same time. This was definitely more difficult due to the many dependencies that VirtualBox has on various tools, however eventually we found the right recipe, which is covered in the "Install VirtualBox on RedHawk 5.1" section of this document.

## INSTALL VIRTUALBOX ON REDHAWK 5.1

The first step was to download the Open Source Edition of VirtualBox from Sun's *virtualbox.org* website. The version downloaded for this investigation was:

```
VirtualBox-1.6.4-OSE.tar.bz2
```

Unpacking this package results in the creation of a *VirtualBox-1.6.4* directory that is filled with all the source code for the product. In addition, it has key programs to assist in building the software. The primary one of interest at this point is called *configure*. Running *configure* will tell you which software dependencies you are currently missing on the host system in order to build the VirtualBox software.

VirtualBox has some dependencies that are beyond what is supplied by a stock RedHawk 5.1 full installation. At the time of the writing of this paper, the *configure* script returned the following results, which will be discussed and resolved in the following sections one issue at a time.

### Get IASL Dependency

The first time you run *configure* you will see it abort with the following error:

```
Checking for iasl:
```

```
** iasl (variable IASL) not found!
```

The IASL package is an assembler from Intel. The *iasl* compiles ASL (ACPI Source Language) into AML (ACPI Machine Language), which is suitable for inclusion as a DSDT in system firmware. It also can disassemble AML, for debugging purposes.

To resolve this for RedHawk 5.1, download and install the following RPM (found via *rpmfind.net* but it is available elsewhere as well):

```
iasl-20061109-3.fc8.i386.rpm
```

This RPM was built for Fedora Core 8 but installs and works just fine on a RedHawk 5.1 system as well.

**Note: All paths used in these examples are for example only and will almost certainly be different depending on where you unpack the software.**

## Disable PulseAudio

Once the IASL issue is resolved, the next configure issue encountered is with PulseAudio:

Checking for PulseAudio:

```
pulse not found at -lpulse or pulse
headers not found
```

Check the file `/home/jason/virt/VirtualBox-1.6.4/configure.log` for detailed error information.

It turns out that RedHawk 5.1 supports the ALSA sound system, so we don't need to have PulseAudio support enabled in the first place. Thus, invoking `configure` as follows will allow the software configuration to complete successfully:

```
./configure --disable-pulse
```

At this point you should now be ready to build VirtualBox from the supplied source code.

## Build VirtualBox Application

Once `configure` has been run successfully, it will tell you how to build all of the necessary pieces of the VirtualBox product.

If you were to follow the suggested rules immediately, you would run the following commands as a normal non-root user:

```
host$ source /home/jason/virt/VirtualBox-1.6.4/env.sh
host$ kmk all
```

However, although this will get quite far, on RedHawk 5.1 this will eventually fail to compile with the following errors:

```
kmk[2]: Entering directory `~/home/jason/virt/VirtualBox
1.6.4/src/apps'
kmk[2]: pass_bldprogs: No such file or directory
kmk[2]: *** No rule to make target pass_bldprogs'. Stop.
kmk[2]: Leaving directory `~/home/jason/virt/VirtualBox-
1.6.4/src/apps'
kmk[1]: *** [pass_bldprogs_before] Error 2
kmk[1]: Leaving directory `~/home/jason/virt/VirtualBox-1.6.4/src'
kmk: *** [pass_bldprogs_before] Error 2
```

Apparently, this is a generic problem for all architectures that will be fixed in the next release of VirtualBox OSE. Until then, the workaround is to do the following two steps:

1. After running `configure --disable-pulse`, invoke `rm -rf src/apps`. This means that the `tunctl` program will no longer be built as part of the normal build process.
2. Thus, get and install the `tunctl` program from a preexisting RPM. We found the following version at [rpmfind.net](http://rpmfind.net) that works with RedHawk 5.1:

```
tunctl-1.4-2.fc8.i386.rpm
```

Now you are ready to do the `source` and `kmk` commands that were described at the end of the `configure` output. The build takes quite a while, but once the `kmk` is completed, you should see the following displayed:

```
kmk[2]: Leaving directory `~/home/jason/virt/VirtualBox-
```

```
1.6.4/src/testcase'
kmk[1]: Leaving directory `~/home/jason/virt/VirtualBox-1.6.4/src'
```

Now you are ready for the next step in the VirtualBox build process.

## Build VirtualBox Kernel Modules

Issue the following commands to build custom versions of the necessary VirtualBox kernel modules needed on the host RedHawk 5.1 system:

```
host$ cd out/linux.x86/release/bin/src
host$ make
host$ sudo make install
host$ sudo depmod -a
host$ sudo modprobe vboxdrv
host$ sudo chmod 777 /dev/vboxdrv
```

With these modules built and installed, you are now ready to run VirtualBox for the first time.

## Testing VirtualBox

Issue the following commands to run VirtualBox on the current RedHawk 5.1 system (the system that will be the VirtualBox host):

```
host$ cd out/linux.x86/release/bin
host$ LD_LIBRARY_PATH=. ./VirtualBox
```

You should be greeted with a Registration Dialog which you should fill in to show your support of VirtualBox OSE to its authors.

## Install VirtualBox

There does not appear to yet be a nice and simple `make install` target for VirtualBox OSE. The easiest way to install VirtualBox is to copy all of the generated shared libraries and binaries to a newly created `/usr/lib/VirtualBox` directory. Do this by issuing the following commands:

```
host$ cd out/linux.x86/release
host$ sudo cp -a bin /usr/lib/VirtualBox
```

Next, create a simple `/usr/bin/VirtualBox` bash script with the following contents:

```
#!/bin/bash

LD_LIBRARY_PATH=/usr/lib/VirtualBox \
    /usr/lib/VirtualBox/VirtualBox
```

Make sure to `chmod 777` this bash script to make it executable. Once this is complete, you should be able to invoke VirtualBox from any location successfully. You will also need to make an executable `/etc/rc.modules` file with the following contents:

```
#!/bin/bash
modprobe vboxdrv
chmod 777 /dev/vboxdrv
```

## Add Users To The vboxusers Group

Installation of VirtualBox on the system requires the creation of a new user group called `vboxusers`. Use the following command to

create this new group:

```
host$ sudo groupadd -f vboxusers
```

For each user that will run VirtualBox, the user must be added to the `vboxusers` group. To do this, issue commands such as the following example command for each user:

```
host$ sudo usermod -a -G vboxusers jason
```

You may need to log out and log back in for this change to take effect.

## INSTALL REDHAWK 5.1 AS A GUEST

Once you have registered, you can begin creation of a RedHawk guest operating system. Note that you will need to have the DVD or ISO image of the Red Hat Enterprise Linux 5.1 product and the CDs or ISO images of the RedHawk Linux 5.1 product. If you have misplaced your Red Hat DVD you can download the 5.1 ISO from a Red Hat Network account. Contact [support@ccur.com](mailto:support@ccur.com) if you have misplaced your RedHawk 5.1 CDs.

Run VirtualBox and click "New" to create a new virtual machine. This will start the Virtual Machine creation wizard which will walk you through the process of configuring the virtual machine settings.

When presented with the "VM Name and OS Type" window, use a VM Name like "rh51vbox" and choose "Linux 2.6" for the OS Type. When presented with the base memory size choice, choose 1024MB.

When presented with the "Virtual Hard Disk" window, click "New..." and follow the wizard screens to create a new disk that is a "Dynamically expanding image" of at least 20 GB. Once completed, click "Next" to accept the disk and continue (it should have a name like "rh51vbox.vdi") and then click "Finish".

You are ready to mount the optical disks or disk images and begin a proper RedHawk 5.1 install. Refer to the RedHawk release notes for information on how to install RedHawk.

To mount an optical disk, choose "Settings" from the main window and then select the "CD/DVD-ROM" tab. Enable the "Mount CD/DVD Drive" checkbox and choose the method that works best for you. Choose "Host CD/DVD Drive" if you have optical media inserted in the

host system, or choose "ISO Image File" if you have ISO files on the host system's hard disk.

Once both Red Hat media and RedHawk media are properly installed, reboot to a fresh new RedHawk virtual machine.

You are ready to install the guest operating system tools. These tools are required for clipboard management and improved mouse control. To install them, choose "Install Guest Additions..." under the "Devices" menu. This will map the ISO with the guest tools onto `/dev/cdrom` (note that you may be prompted to download the ISO from the Internet first). You can then perform the following commands as the root user:

```
# mkdir /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
# cd /mnt/cdrom
# ./VboxLinuxAdditions.run
```

Once the script has completed, you should see the following text displayed:

```
Successfully installed the VirtualBox Guest Additions.
You must restart your guest system in order to complete the
installation.
```

Restart the system and you are now ready to experience the full RedHawk virtualization experience on VirtualBox.

## CONCLUSION

Open source virtualization solutions are available and viable for virtualizing RedHawk Linux. Sun's VirtualBox OSE ("Open Source Edition") appears to be the best option due to the fact that it is nearly as mature from an interface perspective as VMware and yet it is 100% GPL software.

Concurrent RedHawk Linux is an industry-standard, real-time version of the open source Linux operating system for Intel® and AMD™-based systems. RedHawk Linux provides the guaranteed performance required in time-critical and hard real-time environments demanded by Aerospace & Defense, Automotive, Energy, Financial Services, Government, Industrial Control, Medical and Telecommunications. RedHawk Linux runs your applications when you want, where you want, and how you want, delivering the highest quality of service.

CONCURRENT A WORLDWIDE LEADER IN REAL-TIME COMPUTING INNOVATION AND SERVICE

Concurrent is a worldwide leader in real-time Linux-based computing technologies with thousands of real-time systems and software deployed globally solving the most demanding time-critical computing challenges.