

## Outpost on Linux

Bill Steele WA5WVP

March 2013

### Installation of Ubuntu, WineHQ, and Outpost

This document describes several tasks, with the eventual goal of producing a computer system running the Ubuntu Linux 12.04 operating system, the WineHQ 1.4 “compatibility layer” (that permits running many Microsoft Windows® applications on Linux and other POSIX-compliant systems), and the [Outpost 2.8c12 BETA](#) Windows-based packet message client. It also discusses the potentially difficult problem of dealing with the Linux serial ports from Wine HQ and Outpost.

[This covers both built-in serial ports and USB to Serial adapters.](#)

### Installing Ubuntu Linux

These instructions assume that you have an appropriate bootable CD containing Ubuntu Linux and a suitable computer system. I used Ubuntu 12.04, but there are more recent versions available. In general, these instructions should work for any of them. There are instructions on the Internet<sup>1</sup> that I originally followed. From those instructions I wrote the steps for myself to keep me from making the same mistakes multiple times. I have edited them with the hope that someone else might also be able to understand them. It was my first attempt at Linux and took at least 6 or more complete reloads to get all three programs (Linux, Wine HQ, and Outpost 2.6c029) working together.

1. Insert the CD in the drive on your PC. If the PC is not yet running, you might not be able to do this yet. Instead, just power up the PC to get to the point where pressing the appropriate button on the drive will open it, then insert the CD, and reboot the PC – or press the reset button if your machine has such a feature.
2. Restart the computer if it was previously running.
3. Press the **F12**<sup>2</sup> key on the keyboard during startup to get to the BIOS menu that gives you a chance to alter the “boot sequence,” if necessary, to boot from the CD instead of the hard disk drive. If your system is already appropriately configured, you don’t need to do anything but continue with the booting sequence. Otherwise, alter the boot sequence, and then tell the BIOS to save the modified configuration and continue.
4. Your system should now boot from the Ubuntu CD.
5. The “welcome” screen asks you to choose between “Run From CD” or “Install.” Choose “Install,” as you cannot make the needed changes to the CD.
6. The installation procedure next checks the space available on your hard disk drive and checks to see if you are connected to the Internet. If you are connected then you can check “Install updates while loading” and, if you like, “Third party Software.”
7. The installation procedure next asks you to identify your “Installation Type.” The simplest choice is to choose “Erase and use all of the Hard Drive for Ubuntu”. This assumes you will only be using this computer for Ubuntu. Another option is to install in a different partition.

---

<sup>1</sup> For example, see <http://www.ubuntu.com/download/help/install-desktop-latest>

<sup>2</sup> Your system may use something other than **F12** to access the BIOS. Refer to your system’s documentation for the appropriate key. Alternately, just try other traditional keys, like F3, DEL, etc.

8. I selected “Erase Ubuntu 12.XX<sup>3</sup> and reinstall.” You may see a different message if the system already contained a previous operating system. I did not partition the hard drive.
9. Next the installation procedure asks you to indicate the local time zone – essentially asking “Where are you?” This screen does not seem to be very user friendly. It starts up on Chicago (Central Time Zone). You can use the mouse and click on a different area/time zone. Be very precise or you will end up in another country. I tried typing in the city but you have to know the cities in the list. That just did not work.
10. Next you specify your “keyboard layout.” You can test your keyboard here. Pick the one you have. If it works then click on “Continue.”
11. The “Who are you?” section is important. There are several steps to complete here.
  - a. Enter your name, as usual, in the box for “Your name.”
  - b. In the “Your computer’s name” box, select and delete any text that might be automatically configured by the installer. In its place, enter the name you’d like to use to reference this machine on your local area network.
  - c. In the “Pick a Username” box enter a suitable Linux username. Normally this should be something you can easily remember, like your first name or last name. It should not contain any upper case characters.
  - d. In the “Choose a Password” box enter the password you wish to use. I found that a complicated password was very difficult to repeat. Also it appears that if you TAB to the next then there may be a TAB included in your password. So use the mouse to move to the “Confirm your Password” After several failures I came up with a password that met the requirements for “GOOD” by using numbers, capitals, lower case, and punctuation that I could easily remember and type in.
  - e. There are three check boxes next. I recommend that you use box for “Log in automatically” due to all the password problems I had as a new person to Ubuntu Linux. I decided that what I had on the Ubuntu computer was not of any value to anyone. You may want to do differently, if you have more experience.
12. Now the installation procedure continues for some time without the need for additional input. Screens will scroll by that illustrate the functions and options of Ubuntu. It took a long time.
13. Eventually (assuming all goes well) you’ll see “Installation Complete” and “Restart Now” messages.
14. Next you’ll see the message “Please remove installation media from CD tray and press ENTER”. The CD tray should have opened so that the CD can be removed. Press “ENTER”
15. The system should then reboot from disk, and the Ubuntu desktop should be visible.

### ***After the Installation...***

The first thing I did was to go to “Brightness and Lock”. You can change the time before Ubuntu turns on the screen saver, or turn it off completely if you wish. There is also a check box that determines if you will be required to enter your password when the system wakes up from the suspend state. I recommend turning off the screen saver and unchecking the “require password box” until you get everything set up.

---

<sup>3</sup> This identifies the version of Ubuntu – or perhaps some other system – that might have previously been installed (perhaps incompletely) on your system.

## Installing WineHQ 1.4

1. If you haven't already done so, install Ubuntu Linux.
2. Open the "Ubuntu Software Center" (it looks like a paper bag with handles and something in the bag on the left side of the screen). If you position the mouse pointer over it, the "hover text" (the message that sometimes appears after leaving the mouse pointer over an icon for a short time) should indicate "Ubuntu Software Center". Double click very quickly to open it.
3. Move your mouse pointer to the very top black line above the Ubuntu Software Center screen you just opened. You should see these choices: File Edit View Help . Click on Edit .
4. You should see "Software Sources" at the bottom of the drop-down list. Click on "Software Sources".
5. There are five tabs on the "Software Sources" screen. They are "Ubuntu Software", "Other Software", "Updates", "Authentication", and "Statics". Click to open the "Other Software" tab.
6. Click on "Add".
7. Enter the following into the APT line box. (From the WineHQ /Ubuntu installation instructions)  
**ppa:ubuntu-wine/ppa**
8. Click on "Add Source". You will be asked to "Authenticate". This is one of the many places that the password you entered during installation will be used. Type in your password, press the enter key, and then close the "Software Sources" screen.
9. Close the "Ubuntu Software Center".
10. Open "Firefox Web Browser".
11. Search for WineHQ Ubuntu
12. Find the listing "WineHQ – Installing the latest Wine on Ubuntu" and go to it.
13. You should be on the Wine for Ubuntu instructions, scroll down to "Installing Wine" Click on the link to install the stable version of Wine 1.4
14. Launch Application screen should appear. Click OK.
15. Ubuntu Software Center should open with Wine Windows Program Loader. Click on Install.
16. You will need to "Authenticate" again, enter password. Click on "'Authenticate'".
17. You will be asked "Do you accept the EULA license terms?" If you do not accept the program will not install.
18. Time Passes as WineHQ installs. (about 20 Minutes)
19. The installation completes with a white on green check mark followed by an installation date.
20. Close the screens.
21. Double click on the wine glass on the left-hand side of the screen. It should show Wine Windows Program Loader. This will activate and update WINEHQ.
22. Restart Ubuntu.

## Installing Outpost 2.6c029

1. If you haven't already done so, install Ubuntu Linux and WineHQ.
2. Double Click on "Firefox Web Browser". The icon will pulsate and finally open the Browser.
3. Type in the Goggle search "Outpost Packet Message Manager" Click on Search.
4. Click on "Outpost Packet Message Manager homepage".
5. Click on "downloads" – Download opesetup2.6c029 [*jimo*: or the latest package available]
6. Save file and close all programs.
7. Click on Home Folder Icon on left up side of (Ubuntu) screen.
8. Go to Downloads (file "opesetup2.6c029.exe" should be there).
9. **"Right"** click on Outpost Icon.
10. Click on "Open with Wine Windows Program Loader".
11. Follow Outpost setup instructions. Be sure to click the check box for "Create Desktop Icon" when that screen appears. Normally I recommend taking the defaults for Outpost.
12. After installation is complete then the first screen is "Station Identification".
13. The User Call Sign and User Name are needed. The rest can be updated as need arises.
14. You can uncheck the check box for show this screen on startup. You can edit the information in the "OUTPOST" under setup – identification later if needed.
15. You may find that "OUTPOST" cannot access the com port<sup>4</sup>, if so proceed to "Setting up Ubuntu for Com Port" in this document. This is for Com Port one and assumes you have a built in com port. There were other instructions on the "Internet" for using a USB Com Port. The following may still apply also.

**Note:** A "Power Blip/Outage" or "Shutting the Computer down normally without closing Outpost first" may result in loss of connection between the on screen shortcut with WineHQ and the Outpost program. Outpost is still there but the shortcut will not work. Re-install Outpost to re-establish the connection, being sure to check box for installing the Shortcut. All of your setup/ history in Outpost should still be there.

Closing Outpost first before shutting the Computer down should prevent it. An UPS should protect from Power Blips/Outages.

---

<sup>4</sup> The term "com port" here refers to a serial port. The term "com port" is potentially ambiguous, especially in an environment where TCP/IP network port numbers are being referenced.

## Setting up Ubuntu for hardware Com Ports

You will need to use a terminal program to edit a file in order to provide access to the com port. The steps described below are appropriate to achieve this task.

1. If you already have a terminal program installed on your Ubuntu system, skip ahead to step 5. Otherwise continue with step 2 to begin the installation of a terminal program.
2. Open the “Ubuntu Software Center.
3. Go to “Accessories”, then to Tilda or Sakura, and click install.
4. Enter your password to “Authenticate” your right to perform the install.
5. Open the terminal program by clicking on its shortcut icon (which should be on the left side of the screen). A window should open for that terminal program.<sup>5</sup>
6. Type

```
sudo xedit /etc/rc.local
```

and press the enter key. You will be required to enter the superuser<sup>6</sup> password at this point, since the file `/etc/rc.local` is write-protected for everyone except the superuser.

A window with the title “`rc.local`” should appear. The contents of the window represent the lines in the text file named `/etc/rc.local` as displayed by the `xedit` text editor<sup>7</sup>.

7. The file `/etc/rc.local` is one of several script files (which contain command lines) that is executed by the Ubuntu system each time it booted. Each of the lines in the `rc.local` file that starts with `#` is a comment, and is completely ignored by the system. What we’re going to do now is add a line to this script to explicitly change the permissions associated with the serial ports on the system. In particular, we’re going to change them so ordinary application programs can directly read and write the serial ports. The line to be added can be placed almost anywhere in the file. The most logical places to insert the new line would be before the first existing line in the file.
8. The initial suggested modification to `/etc/rc.local` is to add a line like this:

```
chmod 660 /dev/ttyS0
```

This changes the privileges associated with the serial port defined by the file `/dev/ttyS0` so everyone (that is, every executing program) on the system can read and write the serial port. If you know that this is the particular serial port to which your TNC is attached, then you can ignore the next suggestion.

---

<sup>5</sup> A terminal program is roughly the Linux equivalent of a “DOS box” when you’re using Microsoft Windows. More precisely, it is a terminal emulator connected to a shell program (command interpreter) like `bash`.

<sup>6</sup> There’s a lot going on here that could use explanation. Linux, like Windows and many other operating systems, prohibits “ordinary” users from manipulating files critical to the system’s proper operation. To modify one of these files, you must be, or temporarily assume the role of, the “superuser,” the user who does have permission to manipulate (e.g. edit) these files. Prefixing a command with the word `sudo` is a technique for indicating you want to “do” a command as `super user`.

<sup>7</sup> Actually, almost any text editor program can be used to edit the file. So if you’re familiar with `vi` (or `vim`) or `emacs` or some other editor, it can be used in place of `xedit`. Do not, however, use a word processor program, because they will not (necessarily) preserve the structure of the text file. That is, files with types of “.doc” and “.docx” in Windows are not like the “flat ASCII file” we’re dealing with here.

But it's not always obvious which of the `/dev/tty*` files is associated with which physical serial port. Thus my recommendation is to add a line like this:

```
chmod 666 /dev/ttys[0123]
```

This changes the privileges for the four serial port `/dev/ttyS0` through `/dev/ttyS3`, so it shouldn't matter which physical serial port is used. Of course, you could also add four separate lines similar to the first line shown above, one for each of the serial ports.

9. Since lines beginning with `#` are treated as comments, you may find it useful to add a line something like this immediately after the line you just added"

```
# The previous command allows open use of serial ports
```

10. The final line in the `/etc/rc.local` file is the command "`exit 0`", which just indicates that the script is terminating normally.

11. Save the changes to the file and exit the editor program using the appropriate editor command(s).

12. It wouldn't hurt to review the contents of the file after you've edited it. Type the command

```
cat /etc/rc.local
```

and verify it looks correct. If you spot any mistakes, edit it again (see step 5) to correct the problem(s).

13. You will need to reboot the system for the changes to take effect.

## Setting up Ubuntu for USB-to-Serial Com Ports

1. Add your user account to the "dialout" group is done with:

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

This only needs to be done once for each user that you want to allow access to the ttys. The result is stored in `/etc/group`. No editing of `/etc/rc.local` is needed.

2. Plug in the adapter
3. From the terminal window, run "dmesg" and look for the name that is assigned. It will be `/dev/ttyUSB0`, `/dev/ttyUSB1`, etc., depending on the order it was plugged-in and/or discovered by the OS. If you have a single USB adapter, then it's always going to be `/dev/ttyUSB0`.
4. If you have more than one adapter, they may be discovered by the OS in a different order each time the machine boots (or at least a different order than what you originally intended) and, if/when that happens, they can/will switch names. This means that when you link the Wine com1 port to the `/dev/ttyUSB#` port, it may work now, and then stop working after a reboot because that port is now named something else by the OS.
5. To eliminate that problem, configure an udev rule to create a symbolic link based on specific physical characteristics of the particular adapter. This is most easily done with adapters which use the FTDI chipset because they each have a unique serial number. However, it may be possible to do with the prolific chipset adapters as well.<sup>8</sup> For example,

---

<sup>8</sup> This is the bible on writing udev rules: [http://www.reactivated.net/writing\\_udev\\_rules.html](http://www.reactivated.net/writing_udev_rules.html)

- a. Plug in the usb-to-serial adapter.
  - b. Run **dmesg** to see what name it was assigned. For this example, let's assume it's `/dev/ttyUSB3`.
  - c. Run "**sudo udevadm info --name=/dev/ttyUSB3 --attribute-walk**" (this is the Ubuntu 12.04 equivalent of the `udevinfo` command in the above document).
  - d. Use the displayed values to create a udev rule. I won't attempt to recreate all of the advice in the above website<sup>9</sup>. Read that first.
6. Now, let's say you want to use a udev rule to create a symlink in Linux as `/dev/ttyWineCom1`. First, create the text file: `/etc/udev/rules.d/usb-serial.rules` that contains the following:
- ```
SUBSYSTEM=="tty", SUBSYSTEMS=="usb", ATTRS{product}=="FT232R USB
UART", ATTRS{manufacturer}=="FTDI", ATTRS{serial}=="ABC123DEF",
SYMLINK+="ttyWineCom1"
```
- (of course, substitute the actual values shown in the output of "`udevadm info`")
7. Save the file. Then unplug and replug in the adapter.
  8. Now, **ls -l /dev/ttyUSB\***  
You'll see `/dev/ttyWineCom1` linked to whatever actual `/dev/ttyUSB#` name the OS assigned. Now, if I always refer to it as `/dev/ttyWineCom1`, then it doesn't matter if you unplug the adapters, and then replug them in a different order. The `/dev/ttyWineCom1` symlink will always point to whichever `/dev/ttyUSB#` name the OS assigns to the device which has the attributes specified in the udev rule.

## Setting Up Wine Com Ports

1. Open a terminal window after restarting the computer.
  2. In the terminal window type `$ cd .wine/dosdevices`
  3. You should now be in the "`.wine/dosdevices$`" directory
  4. Create a symbolic link between the Linux `ttyS0` and Windows naming convention for com ports. By typing "`sudo ln -s /dev/ttyS0 com1`". Then enter. You would be asked for your password.
  5. Duplicate the above step to link `ttyS1` to `com2` and `ttyS2` to `com3`.  
**ln -s /dev/ttyS1 com2**  
**ln -s /dev/ttyS2 com3**
- Type **ls** and you should see the links you have created **com1 com2 com3**
6. Following the above USB-to-Serial example, use  
**"ln -s /dev/ttyWineCom1 com1"**

You Are now linking the Wine port called "com1" to the linux port called "`/dev/ttyWineCom1`" which, in turn, is linked to whatever `/dev/ttyUSB...` name happens to be assigned to the usb adapter by the OS on the next boot.

---

<sup>9</sup> This is the bible on writing udev rules: [http://www.reactivated.net/writing\\_udev\\_rules.html](http://www.reactivated.net/writing_udev_rules.html)

**Close the Terminal window and run “Ipserial”.**

I found my com port by using Ipserial connecting the TNC one com port at a time, essentially changing Ipserial com port from 1 through 3 until I established communication with my TNC. I went through all three of my board physical com ports trying each in Ipserial before I found that the back panel com port was com2. This is by trial and error trying each physical com port location while setting Ipserial as Com1 then Com2 then Com3 to see if you can establish communication with your TNC. If not then go to the next physical location and repeat the process. It assumes you know your TNC's baud rate and that it is functioning.

I found some information in WineHQ instructions “**Other Things to Configure**” “**4.3.1 Serial and Parallel Ports**”.

I got the idea of opening access to the serial ports in `/etc/rc.local` from WA5AOI. I would also like to thank Stanley A. Wileman, WA5AOI for editing this to correct and improve readability.

**Terminology**

1. `sudo` - super user run
2. `xedit` - edit
3. `rc.local` – file / Script name to be edited
4. `#` at the beginning of a line indicates that it is text.
5. `Chmod` - change mode
6. `666` – allows computer global access.
7. `Ls` – list
8. `ln` - link