Installing the Chesapeake Bay Watershed Model on a PC from Scratch (Using Fedora 12)

By: John Paine

(With substantial help from Rob Burgholzer, Gary Shenk, Heidi Moltz, Jan Ducnuigeen and Al Thompson)

17 March 2010

Introduction

This document chronicles the process I went through to install the Chesapeake Bay Watershed Model (CBWM) Phase 5.2 on a PC, in February and March 2010. It has not been checked by anyone, and may contain really stupid errors in application or judgment, and for that reason I am writing it in the first person singular (apologies in advance—I *really* dislike the first person singular). Nonetheless, this is the document that could have saved me many hours of trial-and-error discovery if it had been available at the start of my undertaking.

A warning up front—depending upon your skills with hardware and software installation and configuration, and your ability to use Linux, this process could take you quite a long time to sort out. If you are an impatient person or have an "executive mindset" you may be better off quitting now. On the other hand, if you decide to roll up your sleeves and go to work the journey can be full of discovery.

The first issue to address is that CBWM has been developed in the Linux/Unix world. Bye, bye comfort zone for those of us who are anchored in Microsoft Windows.

Initially I tried to install Linux (Ubuntu and Fedora) on a computer that was just not up to the task—an old Pentium 4 desktop machine that has a PCI bus. I mention this because it took several days of wading through Internet forums and doing countless trial-and-error boots and patches to finally figure out that the current Linux Kernel does not support add-in video cards on a PCI bus (leaving untold millions with nothing but frustration when they try Linux). The work-around for this problem is to yank the PCI video card out of the computer, and use the (less powerful) on-board video display adapter. After some difficulty I was able to install Linux in a dual-boot mode with Windows XP SP3. Problem was the video was draggy, and left a lot to be desired.

If you have a decent computer, Linux installation can be done successfully in a couple of hours. Ultimately, I ran out and bought a new desktop computer running Windows 7. It has a 64-bit AMD Athalon II quad-core processor, 6 GB of RAM, and a 1-TB hard drive. Installing Linux on the new machine was fairly straightforward.

Installing Linux on a PC

There are several Linux platforms for the PC. Two viable options are Ubuntu (purportedly the most popular and widely supported) and Fedora (based on Red Hat Linux). These two operating system have common Linux kernels and software applications, and large followings. They are both appealing, well supported with huge user communities, and have thousands of very fine open-source or freeware programs available. They both use the Gnome2 desktop, so the look-and-feel is pretty much the same.

There are several discussion blogs on the Internet about differences between the two, but it appears there are more similarities than differences. Both are fine.

I started with Ubuntu 9.10, then started over with Fedora 12. If I was going to go through this process again, knowing what I know now, I would install Ubuntu to dual boot with Windows using the WUBI installer, because it came with more software installed right from the start, and it seemed to run faster than Fedora 12—no proof for that last statement, just an observation (that could easily be wrong).

Both Ubuntu and Fedora allow you to download an ISO file from within Windows, process it to a CD burner, reboot using the new CD, and try their system without making any permanent changes to your hard drive (this is called the “Live CD” approach). They also both have multiple alternate installation processes, such as installing from flash drives, from directly inside Windows, from network installations, etc.

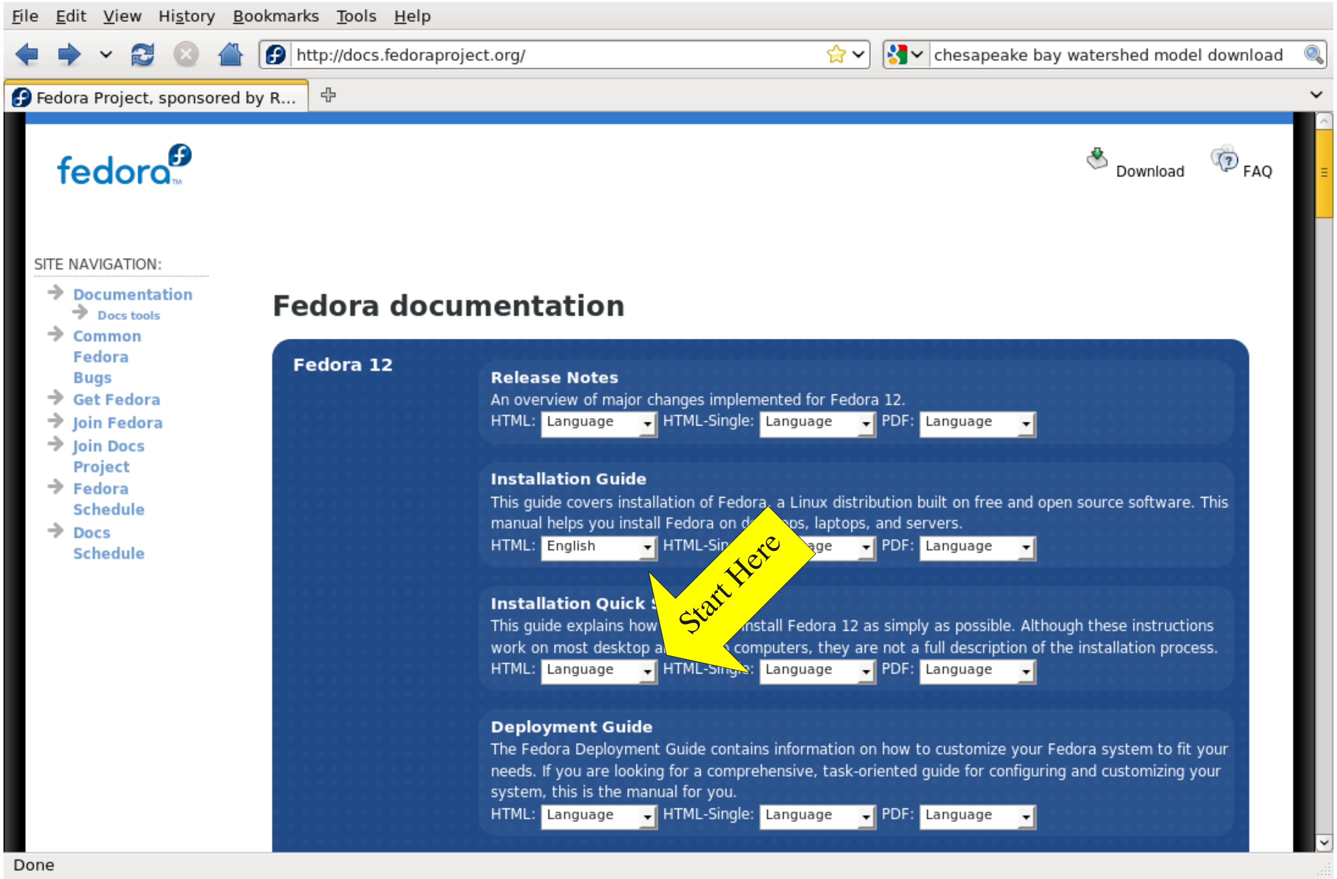
Fedora Installation

There are lots of installation options for Fedora. You can go to YouTube and watch videos, or read any of a number of articles on how to install Fedora on the Internet. One word of caution: there are a lot of well-meaning people who post answers to questions on blogs and forums—some of which may be correct, and others that will surely lead you down blind alleys. Take a little time and do your homework before trying the first ‘answer’ you come across. I really, *really* mean it.

Before proceeding, back up your entire existing system.

Go to the Fedora 12 documents site at http://docs.fedoraproject.org/ and select your language (as in ‘English’), then read the *Quick Installation Guide*. When you get to “Step 3. Download the Fedora live CD image file” click on the link shown for the ISO image file and start the download to your computer. This download is a single ISO file that you will process to a CD. Depending on the speed of your Internet connection, this could take a while. Without belaboring the point, if you have a 64-bit computer, you should be downloading a 64-bit version of Fedora (the approach described above should automate the selection for you).

When the Fedora Live CD ISO file download has completed, use a program such as Windows 7 Disk Image Burner to process the ISO file and create a bootable CD (make sure you check the box to “Verify disc after burning" so that you start with a good CD). You cannot simply copy or burn the ISO file to a CD—it has to be processed. If you are confused, see the documentation in the Fedora Quick Installation Guide. If you're using Windows XP, there are lots of freeware ISO burners available.



Fedora documentation

Fedora 12

Release Notes

An overview of major changes implemented for Fedora 12.

HTML: Language HTML-Single: Language PDF: Language

Installation Guide

This guide covers installation of Fedora, a Linux distribution built on free and open source software. This manual helps you install Fedora on d ps, laptops, and servers.

HTML: English HTML-Sin age PDF: Language

Installation Quick S

This guide explains how install Fedora 12 as simply as possible. Although these instructions work on most desktop a computers, they are not a full description of the installation process.

HTML: Language HTML-Single: Language PDF: Language

Deployment Guide

The Fedora Deployment Guide contains information on how to customize your Fedora system to fit your needs. If you are looking for a comprehensive, task-oriented guide for configuring and customizing your system, this is the manual for you.

HTML: Language HTML-Single: Language PDF: Language

Figure 1. Fedora Documentation Page

I decided to configure my new PC to dual-boot Windows 7 and Linux, choosing which operating system to use when the computer is booted. Dual-boot setups work well (having now gone through the process several times).

In terms of computer preparation, all you have really have to do (after making your backup) is ensure you have a lot of storage space on your hard drive. My system had a 1-TB hard drive, formatted entirely in a single Windows NTFS partition when I started. You do NOT need to re-partition your hard drive ahead of time—the Fedora 12 installation process will do that for you.

Put your newly created Fedora Live CD in your computer, and reboot.

Choose "Live System User" when prompted, and double-click "Install to Hard Drive."

Follow the prompts, and accept "localhost.localdomain" as the host name. Pick your time zone. Enter a password for you as a user. This will be your log-on password for your user profile in Linux. (There will be a separate super user password that you will enter later.)

When given the option during the Fedora installation process, you want to "Shrink Current System" and check the box to "Review and modify partitioning layout." What you will be doing is shrinking the Windows NTFS partition that you started with to make room for installation of Linux (i.e. Fedora 12) on a new partition. I decided to allow 200 GB for Windows 7, and 800 GB for Linux because CBWM files can take a LOT of storage.

When you "Write changes to disk" the Fedora Installation Program will re-size and reconfigure your hard drive. The Installation Program should then display two boot loader labels: "Fedora" and "Other"—edit "Other" to say "Windows 7" (or XP or whatever). Press the Next button. Then wait.

When the "Congratulations" screen appears, remove the Fedora Live CD from your CD drive, then press the Close button, then click on "Live System User" in the upper right corner of the screen, then log out.

Click the power button icon in the bottom right corner of the screen to "Shut Down." You computer will turn off.

Power on (with an empty CD drive), and watch for the Grub boot loader prompt. You need to be a little quick because it will default-boot into Windows if you fail to press a key to enter the Grub boot loader within a few seconds.

Select the Linux Kernel to boot, and off you go. Follow the prompts. You should end up with a user profile (for example "John") and will have to give another password—this time to be used as a Linux "super user."

Not in Kansas Anymore

When you finally land, you will be in the Gnome2 desktop. Play around, customize the look and feel. You should soon be prompted to install updates (there could be hundreds of them). Install the updates. Use the Firefox Internet browser on the top panel to browse the Internet for answers to any specific questions or problems you may have. You will probably want to install graphics drivers for your system. Google for how-to information. Generally, you should install software either through the package installer that comes with Fedora (from the System/Administration/Add-Remove Software menu) or through the terminal (Applications/System Tools/Terminal) using yum or rpm. The reason is that updates will be automated if you stick to these methods. If you fly out and install something from some other source (i.e. bypassing yum, rpm, or the package installer) you could run into problems later on.

Gnome2 has some fancy desktop effects, including putting four desktop environments on opposing faces of a three-dimensional cube for rapid switching among four desktops. Windows wobble and snap and do other cool things that make you feel like you're using a modern operating system. Play around and get these capabilities working. Set your background to some cool image (from the Internet or your family photos). Go exploring.

One software suite I would recommend at this point is Open Office. You can install it through the Fedora package installer. The programs are basically replacements for Microsoft Office programs (Word, Excel, PowerPoint, Access). The word processor, OpenOffice.org Writer, is really easy to use and should come naturally to experienced Word users.

A word of caution and encouragement—spend some time learning Fedora and Linux before you plow ahead. There are lots of resources on the web. Bookmark the good ones you find. I can recommend:

- http://www.computerhope.com/unix.htm (for Linux commands—scroll down!)
- http://lowfatlinux.com/linux-execute-script.html (to learn shell scripting basics)
- https://lists.sourceforge.net/lists/listinfo/npsource-cbpmodel (lifeline for CBWM modelers)

I recommend you use the resources above to get competent with the following Linux commands and concepts: cd, chmod, cp, ls, make, path, using the terminal, and bash scripting. Those who spent time in DOS should have a leg up at this point.

Also, spend some time getting handy with Nautilus (the file browser that appears when you select anywhere from the Places menu on the Gnome2 desktop).

Installing a Fortran Compiler and Other Required Software

CBWM is coded in Fortran 77, and requires lots of shell scripts to execute properly. Although some distributions (including Phase 5.2) of CBWM come with binary executable files, these only work if you have a computer that has a similar architecture and compatible operating system with the program author's computer. In my case, I had to eliminate the binary executables and compile all of the Fortran code to create new binary executable files. In future distributions the binary files may be eliminated so that if you want to run CBWM you *must* compile the source code.

If you are interested in working with or understanding the CBWM source code, see the GFortran Wiki at http://gcc.gnu.org/wiki/GFortran to find out more about Fortran. The home page for the GFortran project can be found at http://gcc.gnu.org/fortran/ and the documentation is available at http://gcc.gnu.org/onlinedocs/ (make sure you get the GNU Fortran Manual that matches your version).

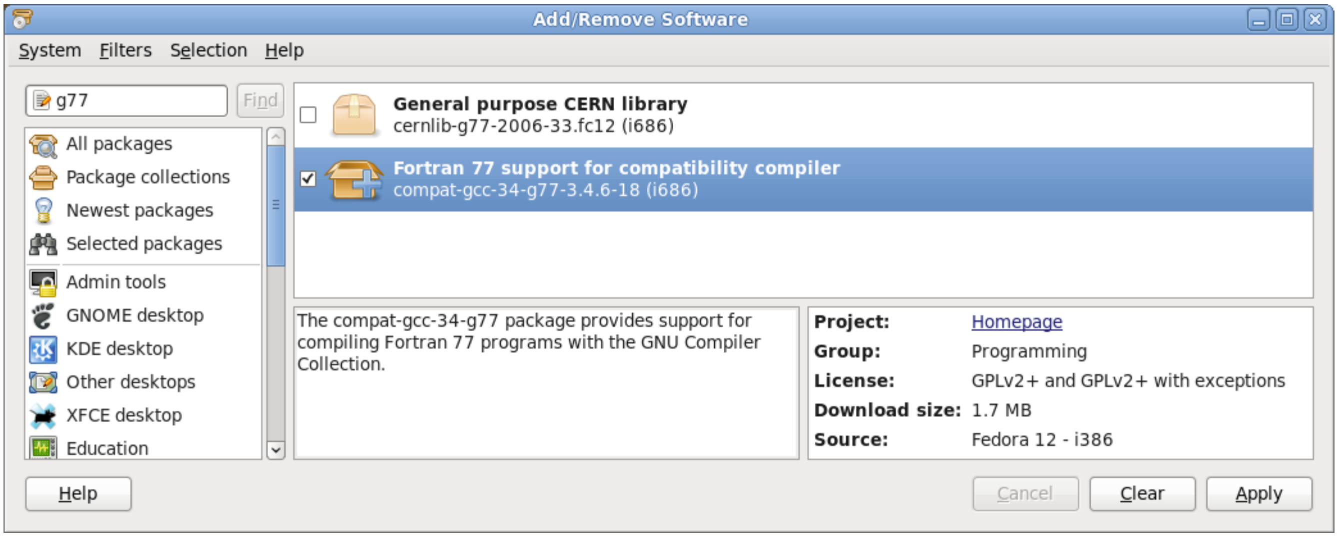
If you have a nice, wide monitor you can save a tree by not printing this type of document. Select the pdf version of the manual, and Fedora will open a very snazzy reader that makes everything electronically searchable. Just for fun, go to the reader's File menu and save a copy to your desktop.

Here's some good news. You only need to install three software packages under Fedora to run CBWM, they're all free, and it only takes two installation processes to execute—I think. "I think" because I did a lot of playing around before proceeding, and some of the other software packages I installed may have installed packages that are required to compile and run CBWM.

Just in case you have been reading older CBWM documentation, you do not need to install the Korn Shell—because the Bourne Again Shell includes the Bash scripting language with Korn and C-Shell integrated in an upwardly compatible manner. When all is said and done, you need the Bourne Again

Shell, GCC Compiler Collection, and a Fortran 77 compiler.

Go to System/Administration/Add-Remove Software, and search for “g77” without the quotes, select “Fortran 77 support for compatibility compiler” and click the Apply button. Accept prompts to install any additional dependent packages.



□ **General purpose CERN library**
cernlib-g77-2006-33.fc12 (i686)

☑ **Fortran 77 support for compatibility compiler**
compat-gcc-34-g77-3.4.6-18 (i686)

The compat-gcc-34-g77 package provides support for compiling Fortran 77 programs with the GNU Compiler Collection.

Project: Homepage
Group: Programming
License: GPLv2+ and GPLv2+ with exceptions
Download size: 1.7 MB
Source: Fedora 12 - i386

Figure 2. Fortran Compiler Installation Using Fedora Package Manager

Repeat this process to install the Bourne Again Shell.

When you’re finished, it might be a good idea to run System/Administration/Software Update.

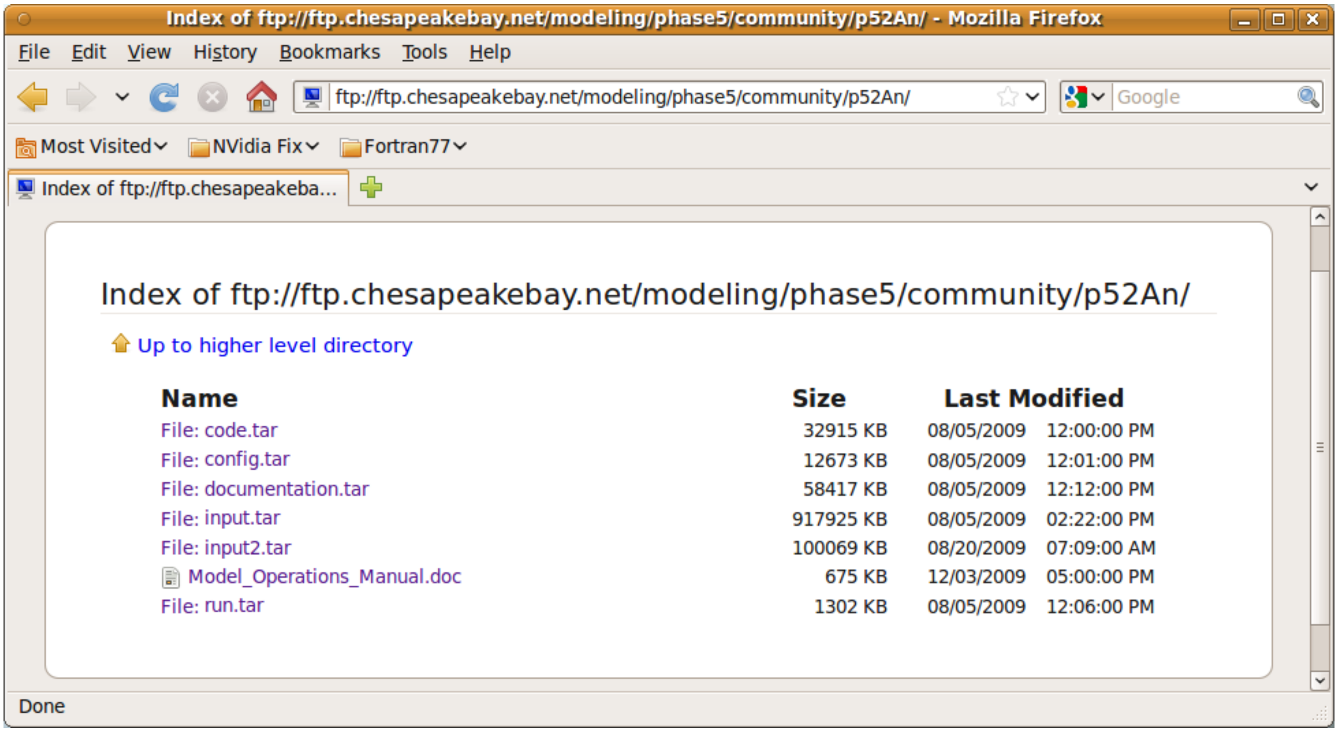
Maybe you can read the Model Operations Manual (or other documentation that comes with CBWM) and just do what it says to compile and run the model. But if you're a Linux neophyte like me, one of the first things you'll need to know is that a 'csh' file is an executable script file, much like an old DOS Batch file. You can edit a csh file with a text editor. Csh stands for “C-Shell.” as in the C programming language. The CBWM developers have kindly distributed a lot of script files, without which those who come behind would be very, very lost. Script files are used to compile the code, run different program scenarios, and perform lots of other batch or script operations.

A little background: csh is a command language interpreter incorporating a history mechanism, job control facilities, interactive file name and user name completion, and a C-like syntax. It is used both as an interactive login shell and a shell script command processor.

Downloading the CBWM Files

Finally, something easy! But before we get to that, you need to know that there are a lot of path adjustments in CBWM scripts. If you want to minimize path headaches, create a folder such as “CBWM” in your home folder. Copy all of your tar files (see below) into the CBWM folder and do all your extractions to that CBWM folder.

As of March 2010, browse to ftp://ftp.chesapeakebay.net/modeling/phase5/community/p52An/ to download the CBWM Phase 5.2 files. Just use the Firefox browser once you're in Fedora to download the files. After you save all the tar files to the same folder (e.g. the CBWM folder described above), right-click on the file name and "Extract Here" and everything should be good to go (at this point that means ready to compile).


Index of ftp://ftp.chesapeakebay.net/modeling/phase5/community/p52An/ - Mozilla Firefox
File Edit View History Bookmarks Tools Help
ftp://ftp.chesapeakebay.net/modeling/phase5/community/p52An/
Google
Most Visited
NVidia Fix
Fortran77
Index of ftp://ftp.chesapeakeba...
Done

Index of ftp://ftp.chesapeakebay.net/modeling/phase5/community/p52An/

Up to higher level directory

Name	Size	Last Modified	
File: code.tar	32915 KB	08/05/2009	12:00:00 PM
File: config.tar	12673 KB	08/05/2009	12:01:00 PM
File: documentation.tar	58417 KB	08/05/2009	12:12:00 PM
File: input.tar	917925 KB	08/05/2009	02:22:00 PM
File: input2.tar	100069 KB	08/20/2009	07:09:00 AM
Model_Operations_Manual.doc	675 KB	12/03/2009	05:00:00 PM
File: run.tar	1302 KB	08/05/2009	12:06:00 PM

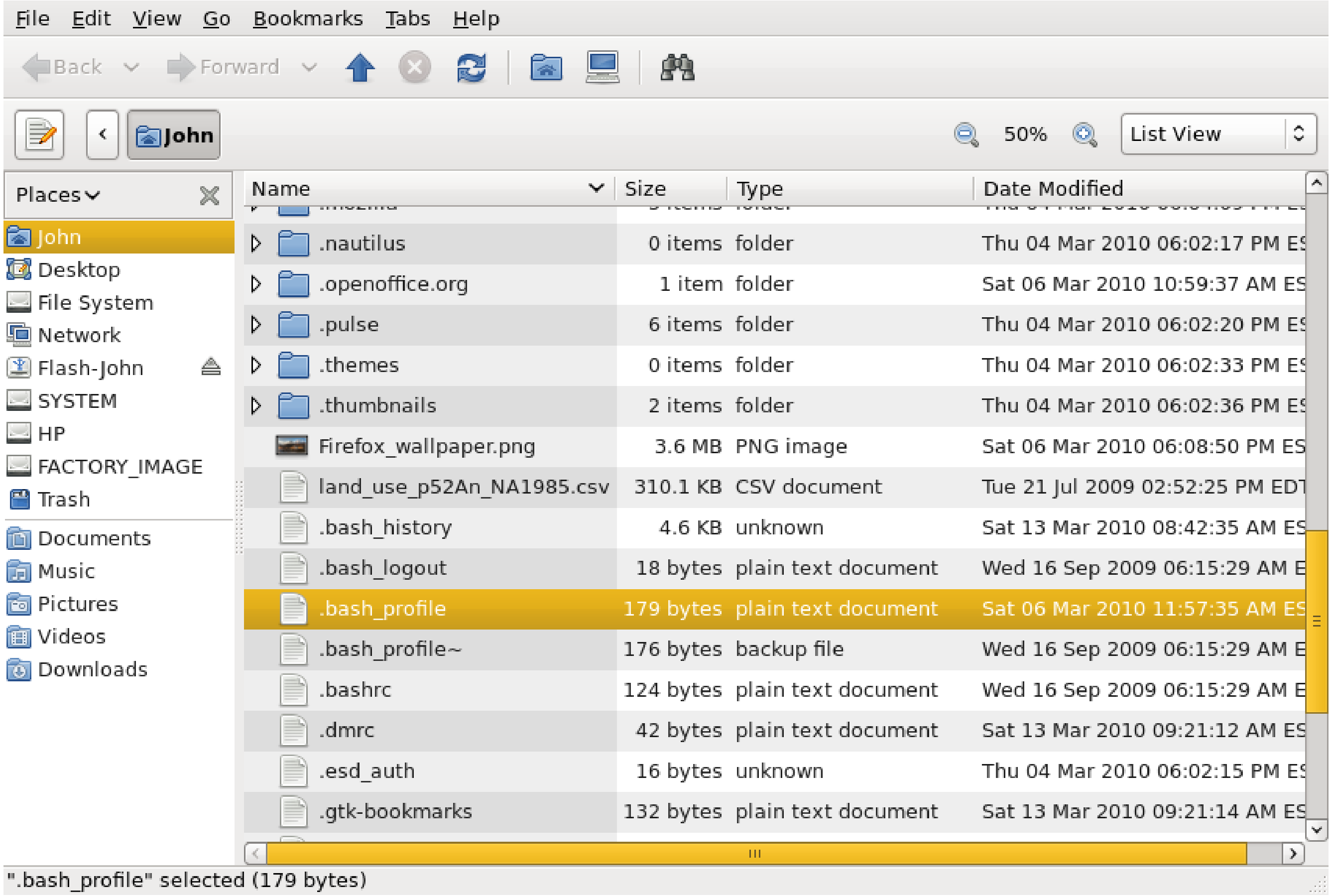
Figure 3. Chesapeake Bay Watershed Model Phase 5.2 Download Site (March 2010)

(If you're nifty with extraction tricks, you may want to extract the Input2.tar file into the Input folder to save a few workarounds later on, but otherwise just keep going.)

Compiling the CBWM Code

At the risk of sounding like I'm taking this way too seriously, this may a good place to stop and get a good night's sleep. When you tackle the following steps you should be clear-headed, comfortable, and isolated from distractions.

The first thing you'll need to do is edit your .bash_profile file to make sure that your environment has a reference to the current directory in your personal executable path. The .bash_profile file is a hidden file in your home/user folder. In my case, I open the File Browser (from the Applications/System Tools menu, make sure the "Show Hidden Files" check box is checked (under the View menu), and double-click on the .bash_profile file to open the gedit text editor, as indicated in Figure 4.



Name	Size	Type	Date Modified
.nautilus	0 items	folder	Thu 04 Mar 2010 06:02:17 PM ES
.openoffice.org	1 item	folder	Sat 06 Mar 2010 10:59:37 AM ES
.pulse	6 items	folder	Thu 04 Mar 2010 06:02:20 PM ES
.themes	0 items	folder	Thu 04 Mar 2010 06:02:33 PM ES
.thumbnails	2 items	folder	Thu 04 Mar 2010 06:02:36 PM ES
Firefox_wallpaper.png	3.6 MB	PNG image	Sat 06 Mar 2010 06:08:50 PM ES
land_use_p52An_NA1985.csv	310.1 KB	CSV document	Tue 21 Jul 2009 02:52:25 PM EDT
.bash_history	4.6 KB	unknown	Sat 13 Mar 2010 08:42:35 AM ES
.bash_logout	18 bytes	plain text document	Wed 16 Sep 2009 06:15:29 AM E
.bash_profile	179 bytes	plain text document	Sat 06 Mar 2010 11:57:35 AM ES
.bash_profile~	176 bytes	backup file	Wed 16 Sep 2009 06:15:29 AM E
.bashrc	124 bytes	plain text document	Wed 16 Sep 2009 06:15:29 AM E
.dmrc	42 bytes	plain text document	Sat 13 Mar 2010 09:21:12 AM ES
.esd_auth	16 bytes	unknown	Thu 04 Mar 2010 06:02:15 PM ES
.gtk-bookmarks	132 bytes	plain text document	Sat 13 Mar 2010 09:21:14 AM ES

".bash_profile" selected (179 bytes)

Figure 4. Finding your .bash_profile File Using the File Browser

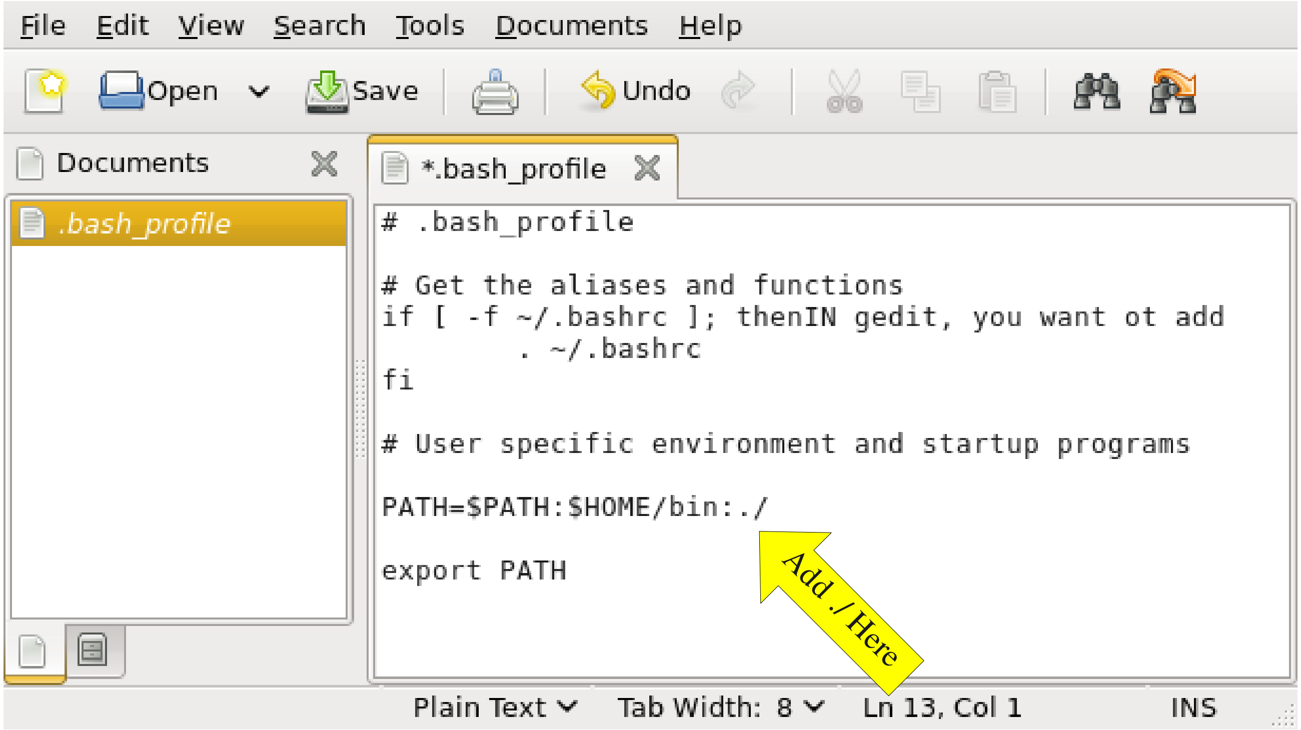
In gedit, you need to add a "./" (without the quotes) to your PATH environment variable, and save the file as indicated in Figure 5 below. Close gedit. Log out by selecting Quit after clicking on your user account name in the upper right corner of your screen. Then log back in. Now when you execute commands using the Terminal, the command will be executed from your current directory.

The next step is to get rid of the binary executable and library files that come with the Phase 5.2 distribution. Unless you happen to have the same computer architecture and environment as the program authors, you need to eliminate these files.

Open a Terminal window from the Applications/System Tools menu, and navigate (using Linux cd commands) to your CBWM/code/bin folder. This folder contains all the CBWM binary executable files (but not the library files). Be certain that you are in the CBWM/code/bin folder, and execute the following command to delete everything in this folder:

```
rm *
```

Now things get a little more tedious, and you have to know a little something about make files and clean rules, scripting, and compiling. So here we go.

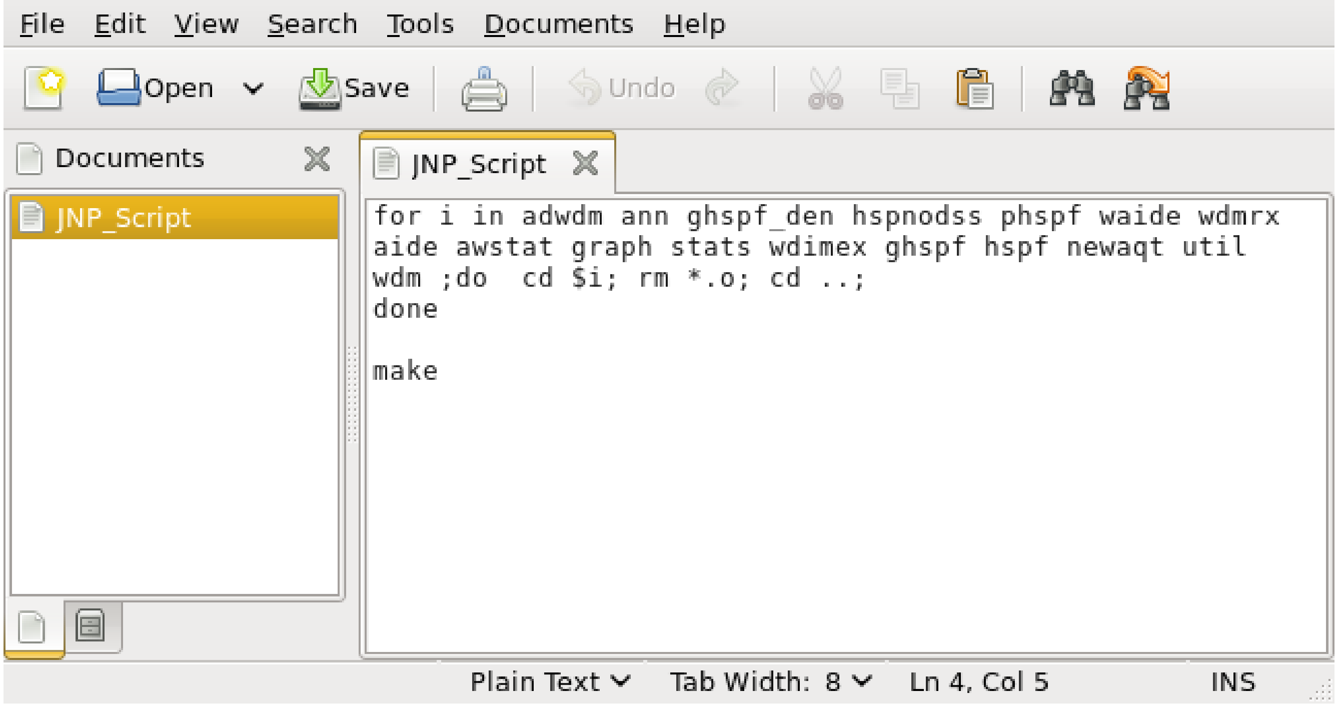


```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; thenIN gedit, you want ot add
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin:./

export PATH
```

Add ./ Here

Figure 5. Using gedit to Edit Your .bash_profile File

Use gedit to create a script file that reads as follows (I cleverly named mine JNP_Script)



```
for i in adwdm ann ghspf_den hspnodss phspf waide wdmrx
aide awstat graph stats wdimex ghspf hspf newaqt util
wdm ;do  cd $i; rm *.o; cd ..;
done

make
```

Figure 6. Script to Clean Up Old Binary Files

(Special thanks to Rob Burgholzer for the above script). Note that there should be no hard returns (i.e.

line breaks) in this file until just before the "done" command. Save the file in the CBWM/code/src/hspf/lib3.2/src folder.

Now you need to make the script executable. At the Terminal window in the same folder, execute:

```
chmod +x JNP_Script
```

and now you have an executable Linux script. Go ahead and let 'er rip:

```
JNP_Script
```

Continuing, there are a bunch of hspf library files that have to be compiled in the folders under CBWM/code/src/hspf (such as CBWM/code/src/hspf/boinc_hspg11.1, CBWM/code/src/hspf/hspf11.1, and CBWM/code/src/hspf/lib3.2). Under each of these three folders are other folders that contain Makefile files. Makefiles contain rules, flags, and compiler instructions. They are executed by navigating to a folder that contains a file called "Makefile" and issuing a simple

```
make
```

command at the Terminal window. The make command executes the Makefile in that particular folder. But..., because there are existing binary library files in the target folders, you will run into some architecture complaints from your compiler if you try to execute the Makefile without cleaning the old binary libraries out first. So how do you do that?

Most Makefile files contain "clean" rules that act as instructions to get rid of all the old binaries before compiling, so you don't end up trying to combine old stuff with your new compilation. Mostly you are trying to get rid of object files (ending with .o) or library archives (ending with .a). All you have to do is navigate to the folder containing the Makefile, and execute two commands at the Terminal window:

```
make clean
make
```

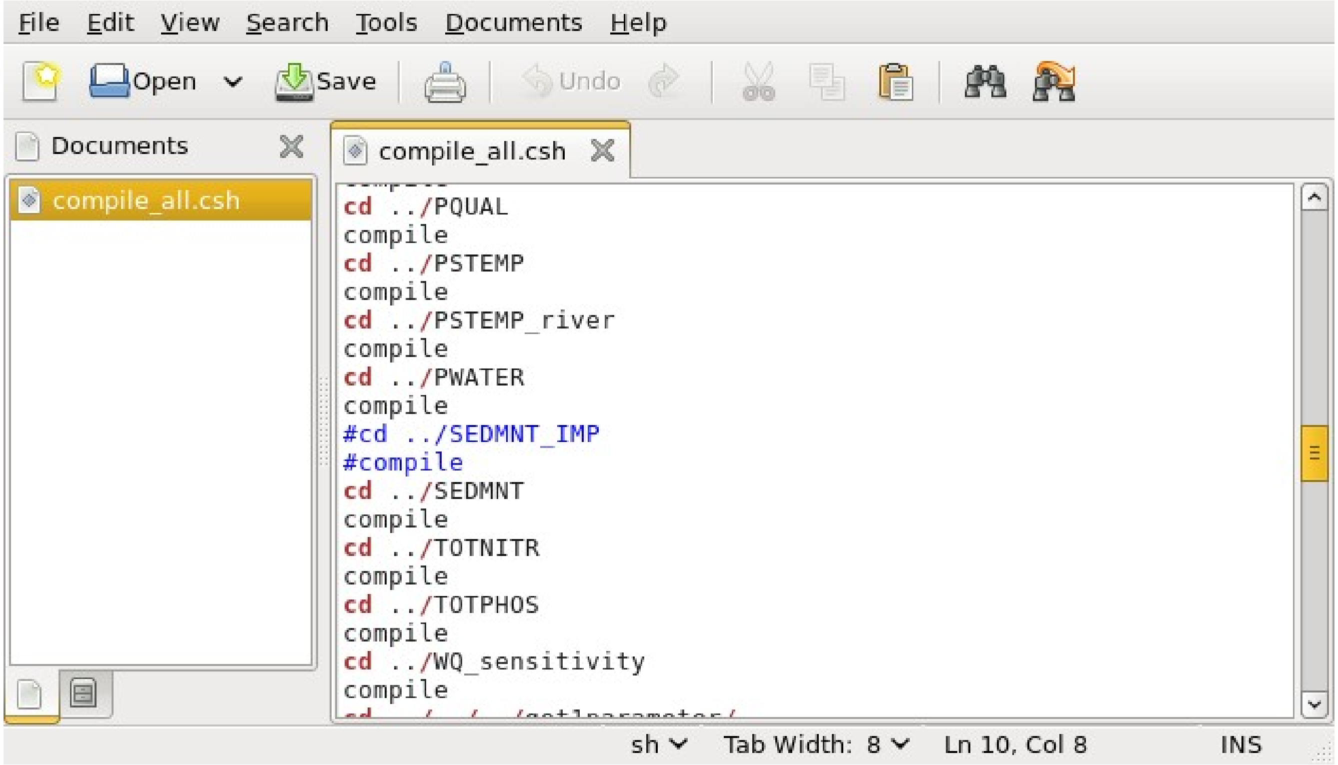
and the old binary libraries will be removed, and the new libraries will be created. Well and good, and not very complicated, so here comes one more wrinkle. There are numerous renamed Makefiles sitting idly by, that have different compilation options, and in many cases produce new and different libraries than the original Makefile you find in each folder. So you need to carefully put your phone on do-not-disturb, and proceed through all of the folders under CBWM/code/src/hspf running the following three-step process:

1. run `make clean`, then run `make` in each folder that has a Makefile.
2. Then look for other renamed Makefiles, sitting around in the same folder with names like quietmake, gmakefile, g_den_makefile, etc. and rename them (one at a time) as Makefile.
3. Then rerun `make clean`, and `make` on each of the renamed Makefiles.

If you keep Nautilus open to the side of the Terminal window, you should be able to see the library files being created with the current date in the CBWM/code/src/hspf/lib3.2/lib folder as you proceed. When you are done stepping through all of these folders you should NOT see any libraries that have prior dates associated with them. If you do, go back and find the appropriate folder and Makefile to repeat steps 1 through 3 above. You need to end up with **no old library files**.

You're on the home stretch now. The next step is to compile the CBWM binary files comprising the executable "model."

Use gedit to open the file compile_all.csh in the CBWM/code/src folder. Halfway down in the file, comment out the two lines as indicated in Figure 7.



```
cd ../PQUAL
compile
cd ../PSTEMP
compile
cd ../PSTEMP_river
compile
cd ../PWATER
compile
#cd ../SEDMNT_IMP
#compile
cd ../SEDMNT
compile
cd ../TOTNITR
compile
cd ../TOTPHOS
compile
cd ../WQ_sensitivity
compile
```

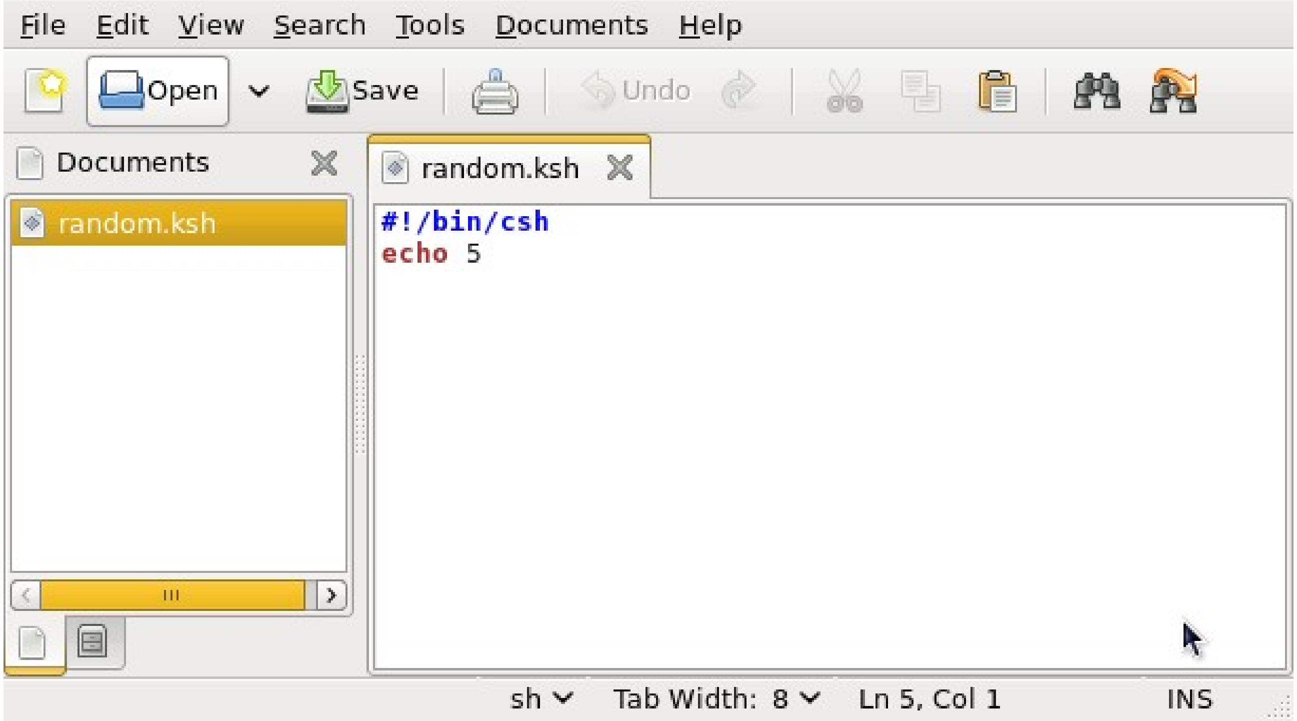
Figure 7. Comment Out Two Lines in compile_all.csh File

Save and exit gedit. At the Terminal window, in this folder, compile the CBWM model by executing:

```
compile_all.csh
```

and you should see a relatively calm progression of compiler object file listings with one or two warnings.

One last detail. In order to run the StMary example data to confirm that your CBWM model installation is successful, you will need to use gedit to edit the random.ksh file in the CBWM/run/useful folder. (You can check the CBWM listserver at http://sourceforge.net/mailarchive/forum.php?forum_name=npsource-cbpmodel if you really need to know about this file.) For now, all you want to do is make the random.ksh file return an integer when it is called during CBWM execution. Use gedit to make random.ksh read like Figure 8 (don't miss the change on the very first line from ksh to csh):



```
#!/bin/csh
echo 5
```

Figure 8. Changes to random.ksh File

Checking Your CBWM Installation

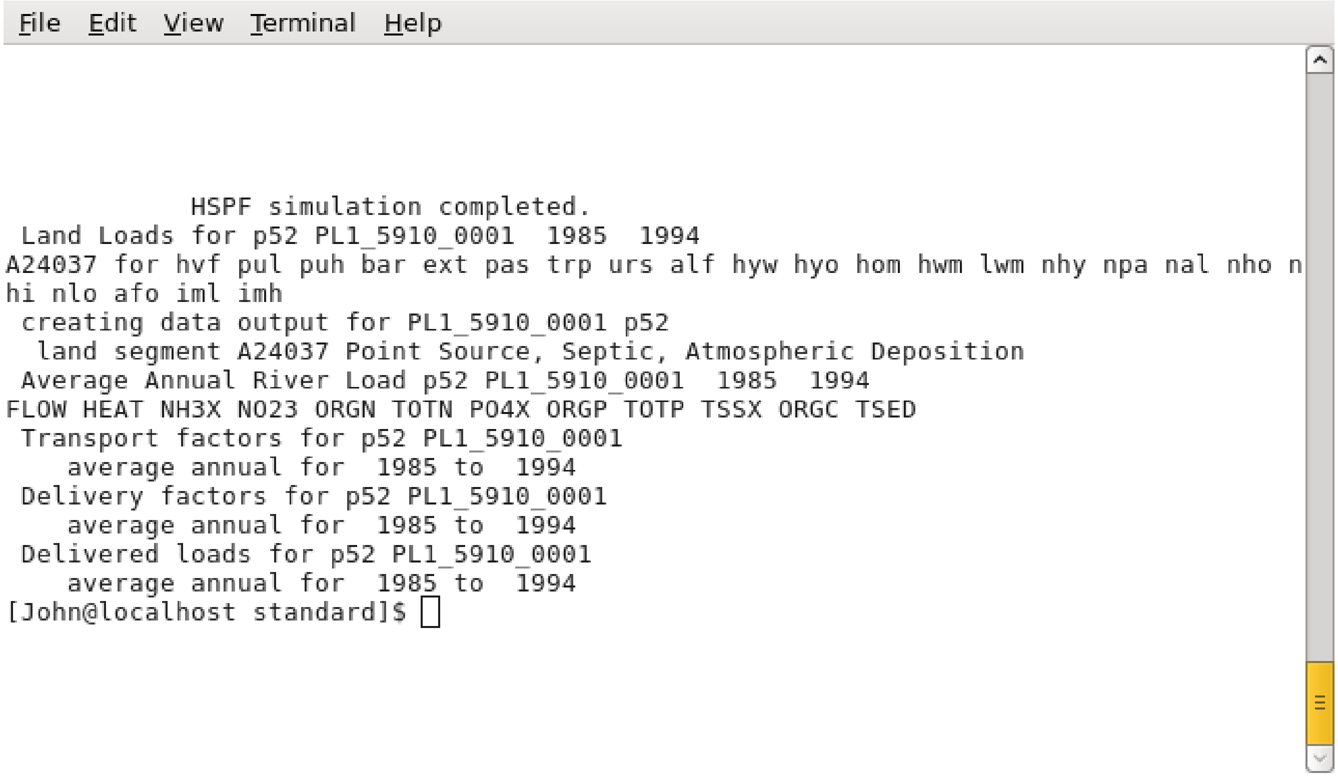
Some more tedium, but this stuff is easy if you've made it this far. It turns out the StMary files referred in the CBWM Model Operations Manual "Quick Start" section have been moved and renamed. You can find them at various locations in the Input2.tar file on the CBWM web site. The easiest way to confirm that your model is indeed working (as far as I know at this time) is to start executing the StMary script, read and heed the various "file not found messages," and copy, rename, and move the required files to the directories indicated by the error messages until the whole shebang runs without complaint. (Sorry, wish I could point you to a simple, clean example to run, but that's the way it is today).

To launch your new modeling tools, navigate in the Terminal window to the CBWM/run/standard folder and type:

```
run_all.csh p52 stmary
```

and fix the missing folders.

When you finally get it all sorted, after a successful run you will see an "HSPF simulation completed" message, followed by several lines of process echo, followed by your terminal prompt, as indicated below in Figure 9.



```
            HSPF simulation completed.
 Land Loads for p52 PL1_5910_0001  1985  1994
A24037 for hvf pul puh bar ext pas trp urs alf hyw hyo hom hwm lwm nhy npa nal nho n
hi nlo afo iml imh
 creating data output for PL1_5910_0001 p52
  land segment A24037 Point Source, Septic, Atmospheric Deposition
 Average Annual River Load p52 PL1_5910_0001  1985  1994
FLOW HEAT NH3X NO23 ORGN TOTN PO4X ORGP TOTP TSSX ORGC TSED
 Transport factors for p52 PL1_5910_0001
    average annual for  1985 to  1994
 Delivery factors for p52 PL1_5910_0001
    average annual for  1985 to  1994
 Delivered loads for p52 PL1_5910_0001
    average annual for  1985 to  1994
[John@localhost standard]$ 
```

Figure 9. Final Terminal Screen After Successful StMary Example Run

The Omega and the Alpha

Congratulations, this process ends at the beginning.

I had a lot of help from the kind folks listed at the beginning of this document, particularly Rob Burgholzer and Gary Shenk.

This document is pasted together from scribbled notes and taken from my faulty memory (you can't record screen shots when you're installing the screen). I also did a lot of trial-and-error hacking around along the way, most of it not documented. If I have misstated anything or put something in the wrong order, please put something on the CBWM listserver so others won't have to toil as hard to get to where they can run the CBWM model.

I hope that this document was helpful to you, and wish you great success in your modeling endeavors.

John