

Welcome to Make World. Theme park rides in this area are know as Panic, Confusion,Dismay, and our all time favorite Segfault Error Operating System not Found. Please note these rides are not for the faint of heart or the easily frustrated as Make World rides will emphasis these handicaps. Please note all of these rides are designed solely for the amusement of the engineers who designed the system and are thousands of miles away. And for people around you who love to laugh at you while you suffer miserably on these knee jerk rides. And since Make World is free (as in beer) you will be getting your money's worth.

Okay lets boil this down, you can make a trip through Make World with this command:

```
cd /usr/src/ && mergemaster -p && make buildworld && make buildkernel KERNCONF=KERNNAME && make installkernel KERNCONF=KERNNAME && make installworld && mergemaster && shutdown -r +3
```

Now if you just went ahead and did that without reading any further. I wish you the best of luck on your journey into hell.

Make World:

is what FreeBSD user call it when you do an upgrade by using the make world group of commands.

The Obligatory Warnings:

BACKUP
BACKUP
BACKUP

If you don't to bad for you. You won't even get cheese with that whine.

I try to keep my articles as simple as possible and not overload you with useless information that you probably don't need. This article will go completely overboard in that area. Don't like it, to damn bad. What happens when you do this to your computer is pretty extensive. If messes with just about every critical file it can get it's hands on. And one wrong move will screw it up to the point where it won't even boot. Scared ? Good. That means you will have loads of fun.

And yes you probably don't need half the information I am throwing at you so I will try to keep critical stuff on the forefront and trivia in the back. Some people actually want to be a bit smarter others just want to upgrade.

This article deals with an upgrade on a FreeBSD system that originally had 4.7 Stable and upgraded to 4.8 Stable. Current users don't need this article. Because they like to live on the edge and help beta test new and risky stuff. I am not so brave.

Everything is done here is done as root.

Final Warning: The handbook really won't help you.

The Breakdown:

[uname -a](#)
[cvsup stable-supfile](#)
[make.conf](#)
[cd /usr/src](#)
[mergemaster -p](#)
[make buildworld](#)
[make buildkernel KERNCONF=KERNNAME](#)
[make install kernel KERNCONF=KERNNAME](#)
[make installworld](#)
[mergemaster](#)
[shutdown -r +3](#)
[uname -a](#)
[df -h](#)
[cd /usr/src; make clean](#)
[cd /usr/ports; make clean](#)
[df -h](#)
[The FAQ you never really asked](#)
[The Credits](#)

With the exception of the FAQ and Credits all of the above are the commands you will need to issue in order if you want to do this one step at a time. Please read the whole article. If you have never done this before some of the files need to be customized.

uname -a

```
FreeBSD TheBaron.bjh.homenet 4.7-STABLE FreeBSD 4.7-STABLE #0: Sun Oct 13 23:32:36  
EDT 2002  
root@TheBaron.bjh.homenet: /usr/src/sys/compile/THEBARON i386
```

Please make a note as to what version you are running so when you are done you can see if you actually upgraded. Definitions [here](#). At the [prompt](#) type:

```
uname -a
```

When it is done it will return to a command prompt

cvsup stable-supfile:

The first step to upgrade is to make sure we have the files we need to upgrade. In order to do this I use [cvsup](#). It is available in the [ports](#) tree. You can also use the [package version](#) if you wish. One way or another you need to [install](#) it.

Once you have installed it. You will need to have a file known as [stable-supfile](#). I recommend you copy one from the `/usr/share/examples/cvsup/stable-supfile` file or the `/usr/src/share/examples/cvsup/stable-supfile` file. To a simple location I keep mine in my `/` directory. You can make a copy with the `cp` command. type this:

```
cp /usr/share/examples/cvsup/stable-supfile /stable-supfile
```

You will need to edit the stable-supfile a little to suit you needs. You can look at an edited stable-supfile [here](#). I recommend [Easy Editor](#) (`ee`) when editing files.

```
ee /stable-supfile
```

You will need to change the following line:

```
*default host=CHANGE_THIS.FreeBSD.org
```

to read like this :

```
*default host=cvsup3.FreeBSD.org
```

That tells cvsup where to go to get the latest stable release files. The rest of the file is already setup to bring in all the files needed to make a successful trip through Make World. Go to you `/` directory. If that is where you saved your stable-supfile and type the following:

```
cvsup stable-supfile
```

It will now begin importing all of the file changes it needs for the upgrade. Once it has finished It will return to a prompt

make.conf

Your [make.conf](#) is a file that is read during the trip through Make World. When you issue your `make buildworld`, `make installworld` and other make commands on your trip through Make World. The file is actually located in the `/usr/src/etc/defaults/` directory. It needs to be copied over to the `/etc/` directory. You can do that with this command:

```
cp /usr/src/etc/defaults/make.conf /etc/make.conf
```

Once you have copied it you will need to do some editing.

```
ee /etc/make.conf
```

Uncomment the following line:

```
#CFLAGS= -O -pipe
```

So it looks like this:

```
CFLAGS= -O -pipe
```

Press Esc and save the changes.

You can see my make.conf [here](#).

cd /usr/src

This is a change directory ([cd](#)) command. You are changing to the `/usr/src/` directory. This is where you need to be in order to complete a successful trip through Make World.

Mergemaster -p

This is a prep phase for mergemaster. This a program that goes through your `/dev/MAKEDEV` file and your `/etc/` directory. Checking for differences between files on your computer and the upgrade files. The program starts by trying to set up a temp directory. Please take notes on the files that are dealt with. You don't have to write down everything but some notes on the name of the files would be helpful. Mergemaster `-p` saves sendmail users and openssh info. Not having that info can cause errors during `make installworld`. For the big definition on Mergemaster go [here](#). At the command prompt type:

```
mergemaster -p
```

```
*** The directory specified for the temporary root environment,  
/var/tmp/temproot, exists. This can be a security risk if untrusted users have  
access to the system.
```

```
Use 'd' to delete the old /var/tmp/temproot and continue  
Use 't' to select a new temporary root directory  
Use 'e' to exit mergemaster
```

```
Default is to use /var/tmp/temproot as is
```

```
How should I deal with this? [Use the existing /var/tmp/temproot]
```

Press Enter (which means use /var/tmp/temproot) and you should be good to go. This will keep the temporary root environment in /var/tmp/temproot. It will now start to compare a large number of files. And show you the differences between them.

Now it is going to start going through your /etc/ directory and /dev/MAKEDEV file .This is the beginning of the hellish part of Make World. Some notes on what you will see here. Mergemaster brings up the current file on the hard drive and the new version.If they are the same it smiles and moves on. If they are different it will point out the differences. It brings up what is different between the files and shows you just the things that are different in the file. It may not need to bring up everything in the file. The symbols used in Mergemaster:

--- 3 minus symbols in a row usually pertaining to the date of the file installation. I have only seen them at the top of a file that is to be changed. It means the line next to them really has to be removed.

+++ 3 plus symbols in a row usually pertaining to the date of the file installation. I have only seen them at the top of a file that is to be changed. It means the line next to them really has to be added (replace the existing line).

@@ means the line numbers that will be affected by the change. @@ -1,5 +1,24 @@ means line 1 and the next 4 lines will become line 1 and the next 24 lines. Look at the next symbols to understand.

+ means the line that will be added.

- means the line that will be replaced/ removed

. The mergemaster then brings up this dialogue.

```
Use 'd' to delete the temporary .name of file  
Use 'i' to install the temporary .name of file  
Use 'm' to merge the temporary and installed versions or parts of them.  
Use 'v' to view the diff results again
```

```
Default is to leave the temporary file to deal with by hand
```

```
How should I deal with this? [Leave it for later]
```

'd' will remove any new lines for the file leaving it the way it was. No changes.

'i' will install all of the new lines and remove all of the old lines.

'm' will place both old and /or new lines in the file.

'v' lets you look at it again.

Use 'i' if you want to upgrade. Use 'd' if you don't. Use 'm' if you want to screw with things. And do it without the quotes.

Okay important notes here. If you have custom files in the /etc/ directory And you just click through this without checking. I can assure you they will be gone when your trip to Make World is done. and you will not have had a good time on the rides.You did remember to backup didn't you?

So how do I know what to upgrade and what to not upgrade? And what to merge?

My rules of thumb on this is:

The 'd' is for files that I have customized. Files I don't want changes in. Files like [ppp.conf](#).

The 'i' is for files that I haven't customized that won't affect things I want my FreeBSD box to do. Most of my files in mergemaster will get this option. Some examples for me are rc.diskless, pam.conf, and rc.syscons. I didn't need to customize these files and their changes won't affect my box.

The 'm' is pretty cool, scary but cool. I have a file I want some changes in but not all. My example is [motd](#). I wanted motd to show the new version of FreeBSD, but not the huge greeting that follows it. It offers you choices:

```
FreeBSD 4.7-STABLE (THEBARON) #0: S | FreeBSD ??? (UNKNOWN)

l:      use the left version
r:      use the right version
e l:    edit then use the left version
e r:    edit then use the right version
e b:    edit then use the left and right versions concatenated
e:      edit a new version
s:      silently include common lines
v:      verbosely include common lines
q:      quit

%r
```

See that %r what happened was I got a prompt % and I wanted the right side so I typed r. and

```
Welcome to The Baron! | Before seeking technical support, p
>
> o Security advisories and updated
> at http://www.FreeBSD.org/releas
> for your release first as it's u
>
> o The Handbook and FAQ documents a
> along with the mailing lists, ca
> http://www.FreeBSD.org/search/.
> been installed, they're also ava
>
> If you still have a question or pro
> `uname -a', along with any releva
> as a question to the questions@Free
> unfamiliar with FreeBSD's directory
> man page. If you are not familiar
>
> You may also use /stand/sysinstall
> configuration utility. Edit /etc/m
>

l:      use the left version
r:      use the right version
e l:    edit then use the left version
e r:    edit then use the right version
e b:    edit then use the left and right versions concatenated
e:      edit a new version
s:      silently include common lines
v:      verbosely include common lines
q:      quit

%l

Use 'i' to install merged file
Use 'r' to re-do the merge
Use 'v' to view the merged file
Default is to leave the temporary file to deal with by hand

*** How should I deal with the merged file? [Leave it for later]

%i
```

See that %l that was for left side and %i was to install the merged file

In this case I told it in the upper boxes to use the right version (%r) to show what issue of FreeBSD (?.?.? goes to the new version) I am using and in the lower setting I used the left version (%l). Then I told it to install the merged version (%i).

Some other things you will also see:

```
*** There is no installed version of ./etc/login.conf.db

Use 'd' to delete the temporary ./etc/login.conf.db
Use 'i' to install the temporary ./etc/login.conf.db

Default is to leave the temporary file to deal with by hand

How should I deal with this? [Leave it for later]
```

This means that a new file would like to be added and that there is no current copy of this file. I have always told it to install ('i' without the quotes people).

When it is done You will see this:

```
*** Comparison complete

Do you wish to delete what is left of /var/tmp/temproot? [no]
```

Press Enter (which means no) and you have set up the temporary files. You want to keep those files there for when you go back into mergemaster at the end.

You get to go through this twice once at the beginning and once at the end and I have seen it choose different files from the first time to when it has gone through it's make world session. And that folks is the basics of Mergemaster. When it is done it will go back to a prompt.

make buildworld

Okay it is time for the rides to a more dangerous turn. Make buildworld is the command that tell your computer to grab those files in /usr/src/ and start putting them together. This will cause a lot of stuff to flash across the screen as it compiles, uncompresses and sorts through the whole nine yards. These are the files that are connected to the /dev/ and /etc/ and other directories. These are not the kernel files. This like when a cabinet is built all the parts are formed at a shop away from your house. At the prompt type:

```
make buildworld
```

When it is finished it will return to a prompt.

make buildkernel KERNCONF=KERNNAME

Now we are entering the big drop on the roller coaster ride of make world. Make buildkernel KERNCONF=KERNNAME is the command that tells the computer to grab the [kernel](#) files in /usr/src/ and start putting them together. Now if you typed it in just like it is shown you will get an error message. KERNNAME is suppose to be the name of your kernel. If you haven't given your kernel a name you use [GENERIC](#). So the default is to use GENERIC unless you gave your kernel a name. If you did give your kernel a name such as BOB you would need to replace KERNNAME with BOB. And yes the fact that they are in capitol letters is important. Make sure yours are also. This would be where the shelves that will go in the cabinet are built. Nothing is set in place yet. At the prompt type:

```
make buildkernel KERNCONF=KERNNAME
```

When it is finished it will return to a prompt.

make install kernel KERNCONF=KERNNAME

Make install kernel KERNCONF=KERNNAME is where the kernel is set into place. This is the core of the FreeBSD operating system. It is what is read on boot and helps the functions work on the computer. The KERNNAME deal pops up here again remember if you don't have a custom named kernel you replace KERNNAME with GENERIC. Remember to if you do have a specific KERNNAME it must be the same as in make buildkernel KERNCONF=KERNNAME. This is the boss file on your computer every other file works for it . This would be like the shelves going in the cabinet. They are just shelves you say. Shelves are easy to put in. Yes they are. And they keep everything in the cabinet organized. You will find out if you have no shelves that it is not nearly as easy to stock the cabinets. At the prompt type:

```
make install kernel KERNCONF=KERNNAME
```

When it is finished it will return to a prompt.

make installworld

Okay in make buildworld you built everything. Now you are going to install it. With this command everything starts to fit itself into place. Lots of gobbledygook flashes across the screen. The cabinet guys come to your house and start installing the cabinets and the shelves. At the prompt type:

```
make installworld
```

When it is finished it will return to a prompt.

mergemaster

Okay we are back here again. Please note that there is no -p switch. The deletes, installs, and merges here are permanent. This is where what you reviewed earlier in your first mergemaster really happens. It may go through a second time every file it went through with you before and it may even have a few new ones. The idea is still the same if you didn't customize it. Just install it. If you did you will have either delete or merge it as described above. Please have your notes from the previous session ready you did make notes didn't you? Please don't be like the waitress who thinks she can memorize my order and then come back and say "Could you repeat your order again." Write it down. When it is done you get this again:

```
*** Comparison complete
Do you wish to delete what is left of /var/tmp/temproot? [no]
```

This time you tell it yes. Because you are done and don't need the temp files anymore the permanent ones are installed. This is the beginning of cleaning up the mess. Just like with the cabinets there is going to be some cleaning that should be done. More on that in a bit.

shutdown -r +3

This is the [shutdown](#) reboot command. I tell it that I want the computer to shutdown in three minutes and then reboot. This prevents a hard shutdown and gives everything a chance to close. You don't have to make it that long and most everything will happen in the last 10 seconds but I have seen "shutdown -r now" after a trip through make world screw a few things up. You must reboot for the changes in make world to take place. At the prompt type:

```
shutdown -r +3
```

When it has finished you will need to log back on as root..

uname -a

Okay you did the reboot you did log in. And you didn't see any screwed up error messages. Marvelous. Now at the prompt type:

```
uname -a
```

This will bring up something like this:

```
FreeBSD TheBaron.bjh.homenet 4.8-RC FreeBSD 4.8-RC #2: Fri Apr  4 06:43:52 EST 2003
root@TheBaron.bjh.homenet: /usr/obj/usr/src/sys/THEBARON  i386
```

Please note that it should be a different version than the one you started with. That was the whole point. Uname displays the system information the -a switch means give me all of the information. Now that you have done a great job, you have to clean up. What you don't think you do, look at this:

df-h

At the command prompt you will type [df -h](#) and you will see how much space is used up on your hard disk. So at the command prompt type.

```
df-h
```

```
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/ad0s1a    194M   50M  129M    28%     /
/dev/ad0s1f    992M  513K  913M     0%     /tmp
/dev/ad0s1g    8.8G   8.1G  921M    93%     /usr
/dev/ad0s1e    1.9G   14M  1.8G     1%     /var
procfs         4.0K   4.0K   0B    100%    /proc
```

That /usr directory is a bit full. There are three things stuff to the hilt in there right now. They are /usr/ports, /usr/src, and /usr/obj.

cd /usr/src; make clean

The [make clean](#) command is the nice way to clean up the source ([src](#)) files used in your trip through Make World. This does a nice neat clean up. It takes time but when it is done you are worry free. At the prompt type:

```
cd /usr/src; make clean
```

When it is finished it will return to a prompt.

cd /usr/ports; make clean

The [make clean](#) command is the nice way to clean up the [ports](#) files used in your trip through Make World. This does a nice neat clean up. It takes time but when it is done you are worry free. At the prompt type:

```
cd /usr/ports; make clean
```

When it is finished it will return to a prompt.

Another command you can run is this one, at the prompt type:

```
find /usr/ports -type d -name 'work' | xargs rm -vrf
```

That will do the same thing just a little quicker. It takes anything in a 'work' directory in /usr/ports/ and cleans it out.

df -h

Okay let's see how the clean up went at the prompt:.

```
df -h
```

Filesystem	Size	Used	Avail	Capacity	Mounted on
/dev/ad0s1a	194M	50M	129M	28%	/
/dev/ad0s1f	992M	513K	913M	0%	/tmp
/dev/ad0s1g	8.8G	2.9G	5.2G	36%	/usr
/dev/ad0s1e	1.9G	14M	1.8G	1%	/var
procfs	4.0K	4.0K	0B	100%	/proc

Look at the difference here. And that is with a nice clean up. Not to shabby.

We hope you enjoyed your trip through Make World please come visit again when the next release is issued.

The FAQ you never really asked.

1. What is the && mean?

The && means when you are finished with one thing go and automatically do the next. Provided you don't have any errors. Which causes the continue functioning to stop.

2. The tables that look like screenshots don't render in my Lynx browser. Are you going to fix that?

No. Blame Lynx not me.

3. How come you included KERNCONF=KERNNAME when you don't need it for a GENERIC kernel?

While it is true you don't need KERNCONF=KERNNAME for a GENERIC kernel. If you have a custom one and forget to put in KERNCONF=KERNNAME you will be screwed. Using KERNCONF=KERNNAME is a good habit to get into. Unlike smoking.

4. Something went wrong and I got an error. Can you help me fix it?

Probably not. I am still new to this stuff and I don't get to work or play on FreeBSD boxes all day everyday. You can post it in [Chucktips](#) and/or [comp.unix.bsd.freebsd.misc](#) and see if you get the answer you need. You can also post your question in the reply section to this article here. I can't guarantee you will get a quick response let alone a right one. but I will at least try.

5. Can I post some helpful advice or a suggestion/ comment?

Absolutely. Please do [here](#).

6. I want to upgrade from 4.4 to 4.8 (or some other multi release jump), will what you wrote work?

Yes provided you follow the instructions and backup. Mergemaster is your friend.

7. How come everything says I can delete /usr/obj and you don't?

I don't like deleting it. I use the cd /usr/src; make clean and it does a nice neat job and takes care of /usr/obj without creating dependency issues.

8. Why don't you drop into single user mode and use the tweaks?

Because you don't have to. Will doing it make your trip through Make World faster, yes. But that means more explaining. And this is for newbies. Once you have done it a few times, then go into single user mode and use the tweaks.

The Credits

[Jason Neuman](#), who probably wonders if I am taking all my meds but still lets me post my articles.

[acb](#), who has great ideas and lets me put them in my articles.

[Net_Fish](#), who understands all of this easily and with no effort. As I have to struggle with it. It just isn't fair.

The [Huntsville Macintosh Users Group](#), who supply most of the [man pages](#).

The [GSP](#), who have some nice man pages. Too bad they won't validate under W3C standards.

The big [article](#) on Make World by Nik Clayton

The [FreeBSD handbook section](#) on make world.

[comp.unix.bsd.freebsd.misc](#) is a very helpful newsgroup.

[University Computing Systems](#) has a nice [glossary](#) of Unix terms.

Kathy Russell with [Focus on Unix Guide](#) has a very good [glossary](#). Too bad they have crappy evil pop ups.

The FreeBSD man pages can also be accessed from [here](#). I can't use them in the glossary because they won't validate under W3C standards.


[The School of Mathematacis](#) had the easy editor man page for my glossary. I didn't see how many more man pages they had.

[The Kiev Polytechnic Institute](#) also has a nice set of [man pages](#)

[Visi.com](#) supplies the man page for make.conf.

Valid CSS!

[Home](#)

 Here is the article as a .pdf

Valid HTML
4.0!

