Compile XigmasNAS Yourself

This is the easiest way for studying/modify XigmaNAS. It also assumes that the user has an bit of understanding of FreeBSD, including adding ports, packages and using the vi text editor.

Start prepare the XigmaNAS development environment.

1. Install FreeBSD 13.0 RELEASE.

On your dedicated PC (or under a Virtualbox (on XigmaNAS system), VMware/Qemu) install and setup FreeBSD.

It need less than 28GB total hard drive space if setting up a dedicated disk slice for only building XigmaNAS. (more space like 40GB is recommended for active development)

When installing FreeBSD create two partitions', a swap (3.5 GB is a good swap size) and a / partition. Avoid using the A (auto) command to partition the slice. You may get some warnings later about mail security. Since this is a dedicated XigmaNAS build environment only, you can ignore them.

Setup the networking services. Those are required to download the source files and ports.

Hint: If you use the same machine for both a build environment and as your XigmaNAS server, use a different IP-address for the build environment from the NAS environment (if you're not using DHCP). Could be done by bridged adapter in the vm settings. That way if you later SSH into either environment you won't get warnings about a fingerprint change.

Reboot to complete system install.
You now can login as root.

2. Update FreeBSD

Now, update your installed copy of FreeBSD with the latest patches and delete uncontrolled source files (for proper kernel build), remember this is a FreeBSD Build Only Environment!

2.1. Delete old source files

Delete of the /usr/src directory contents.

```
# rm -rf /usr/src
# mkdir /usr/src
```

2.2. Security update

Begin with installing the latest security patches;

```
# freebsd-update fetch install
```

2.3. Port update

Install the latest ports source. **The first time we use this command:**

```
# portsnap fetch extract
```

When done you can forward to step 2.4.

Only for subsequent port updates use:

```
# portsnap fetch update
```

2.4. Now reboot the system and login as root

Root login is not necessary but really recommend. It's only the development environment.

3. Install the required ports:

Go into the system ports and make install clean.

```
# cd /usr/ports/shells/bash
  make install clean
# cd /usr/ports/sysutils/cdrtools
 make install clean
# cd /usr/ports/ports-mgmt/portupgrade
  make install clean
# cd /usr/ports/devel/subversion
  make install clean
```

3.1. Download the XigmaNAS source code from SVN.

Now we are ready to create the XigmaNAS directory and grab the source files

3.2. Create the XigmaNAS directory

This is the place where all source files will be stored and it's scripts can be used.

```
# mkdir /usr/local/xigmanas
```

3.3. Fetching the latest XigmaNAS source files

Enter following to get the sources on its right location:

```
# cd /usr/local/xigmanas

# svn co https://svn.code.sf.net/p/xigmanas/code/trunk svn
```

(Only registered XigmaNAS developers use:
https://svn.code.sf.net/p/xigmanas/code/trunk **if they want to upload code to svn)**

4. Compile and build XigmaNAS from scratch

Now login as root user and start the compile/build script.

```
# cd /usr/local/xigmanas/svn/build
# ./make.sh
```

Note: The ./ in front of make.sh forces the shell to use the file in the current directory, and not search the path for the command.

The following BUILD ENVIROMENT menu should come up and the selections are self-explanatory:

XigmaNAS® Build Environment

---

1 - Update XigmaNAS® Source Files to CURRENT.

2 - Select Compile Menu.

10 - Create 'Embedded.img.xz' File. (Firmware Update)

11 - Create 'LiveUSB.img.gz MBR' File. (Rawrite to USB Key)

12 - Create 'LiveUSB.img.gz GPT' File. (Rawrite to USB Key)

13 - Create 'LiveCD' (ISO) File.

14 - Create 'LiveCD-Tin' (ISO) without 'Embedded' File.

15 - Create 'Full' (TGZ) Update File.

16 - Create All Release Files at once.

---

17 - Create 'xigmanas.pot' file from Source files.

* - Exit. Press #

Now select Menu option 2 - Select Compile Menu

The following BUILD menu should come up and the selections again self-explanatory:

Compile XigmaNAS from Scratch

Menu Options:

1 - Update FreeBSD Source Tree and Ports Collections.

2 - Create Filesystem Structure.

3 - Build/Install the Kernel.

4 - Build World.

5 - Copy Files/Ports to their locations.

6 - Build Ports.

7 - Build Bootloader.

8 - Add Necessary Libraries.

9 - Modify File Permissions.

* - Exit.

Press #

Select each menu item in order. (Hint: When it gives you a choice of multiple choices, do one at a time. This way many errors can be corrected/prevented before proceeding.)

When you compiled the ports in option 6-1 “build ports” you need to compile a second time the ports with only the ports “CA_ROOT_NSS, ICU and python2” selected before you perform 6-2 install the ports.

Notes:

· READ the README files in the various svn/build directories
· Kernel patches should be applied only once. Trying multiple times will fail. (see README)

· Building ports can be the most troubling as source locations change, revisions numbers change, etc. So once you get all the ports compiled, you may want to save those locally. Those files are fetched into "/usr/ports/distfiles". Just remember to keep those up-to-date.

· 3 - Build/Install the Kernel. (Second screen has by default all options set!), do not change those options if you don't know what you are doing! It can cause errors on building/install the kernel!)

When done press * to exit which takes you back to the main menu to select the option(s) for the final product:

10 - Create 'Embedded.img.xz' File. (Firmware Update)

(This will create the embedded upgrade file)

11 - Create 'LiveUSB.img.gz' File. (Rawrite to USB Key)

(This will create the LiveUSB file and embedded file)

12 - Create 'LiveCD' (ISO) File.

(This will create theLiveCD and embedded upgrade file with the checksum files)

14 - Create 'Full' (TGZ) Update File.

(This will create the full upgrade file to upgrade full installs)

5. Making a updated translation template for launchpad.

This is needed for the XigmaNAS Developers only!

XigmaNAS makes use of translation.po files to display the WebGUI in another language than English.

Only the xigmanas.pot has to be uploaded as we only download the translations.po online.

Run below to update the template:

cd /usr/local/xigmanas/svn/build

./xigmanas-create-pot.sh

Hint: Now you are able to locally update a translation.po with a program like Poedit. Google for it.

Good luck, now you can do it all yourself!