This is the easiest way for studding/modify NAS4Free. 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 NAS4Free development environment.

1. Install FreeBSD 9.1 RELEASE.
On your dedicated PC (or under a VMware/Qemu) install and setup FreeBSD .
It need less than 14 gigs total hard drive space if setting up a dedicated disk slice for only building NAS4Free. (more space like 20gigs is recommended for development)

When installing FreeBSD create two partitions', a swap (3.5 gigs 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 NAS4Free build environment only you can ignore them.)

Setup the networking services. Those are required to download the source files & it's ports.
Hint: If you use the same machine for both a build environment and as your NAS4Free server use a different machine IP-address (assuming you are not using DNS) than you use for your NAS4Free box. 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 patch;

```
# 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.

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 NAS4Free source code from SVN.
Now we are ready to create the NAS4Free directory and grab the source files

3.2. Create the NAS4Free directory
This is the place where all source files will be stored and it's scripts can be used.

```
# mkdir /usr/local/nas4free
```

3.3. Fetching the latest NAS4Free source files
Enter following to get the sources on its right location:

```
# cd /usr/local/nas4free

# svn co http://svn.code.sf.net/p/nas4free/code/trunk svn
```

(Only registered NAS4free developers use: https://svn.code.sf.net/p/nas4free/code/trunk **if they wants to upload code to svn)**

4. Compile and build NAS4Free from scratch
Now login as root user and start the compile/build script.

```
# cd /usr/local/nas4free/svn/build

# ./make.sh
```

Note: The ./ in front of make.sh says; use only the current directory, do not use files found in the path

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

NAS4Free Build Environment

Menu Options:

1 - Update NAS4FREE Source Files to CURRENT.
2 - Compile NAS4FREE from Scratch.
10 - Create 'Embedded' (IMG) File (rawrite to CF/USB/DD).
11 - Create 'LiveUSB' (IMG) File.
12 - Create 'LiveCD' (ISO) File.
13 - Create 'LiveCD-Tiny' (ISO) File without 'Embedded' File.
14 - Create 'Full' (TGZ) Update File.
* - Exit.
Press #

Select Menu option 2 - Compile NAS4FREE from Scratch
The following BUILD menu should come up and the selections again self-explanatory:

Compile NAS4FREE 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 - Build Ports.
6 - Build Bootloader.
7 - Add Necessary Libraries.
8 - 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 so any errors can be corrected before proceeding)
Notes:
· READ the README files in the various svn/build directories
· Kernel patches should be applied only once. 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 local. Those files are fetched at the /usr/ports/distfiles location. Just remember to keep those up-to-date.

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) File (rawrite to CF/USB/DD).
(This will create the embedded upgrade file)

11 - Create 'LiveUSB' (IMG) File.
(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)

5. Making a updated translation template for launchpad.
This is needed for the nas4free developers only!
NAS4Free make use of translation.po files to display the WebGUI in a other language than English.
Only the nas4free.pot has to be uploaded as we only download the translations.po online.

Run below to update the template:

```
# cd /usr/local/nas4free/svn/build
# ./nas4free-create-pot.sh
```

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

Some important notes!

* Before fuppes can be compiled; it's important to copy all files in /usr/local/nas4free/svn/build/ports/distfiles to /usr/ports/distfiles manual as the current fuppes can't be fetched from its download page!

* Security patch p3 for FreeBSD 9.1 can give troubles when you perform freebsd-update fetch install!

Before you run do perform # mkdir –p /usr/src/crypto/openssl/ssl
Now again # freebsd-update fetch install

Good luck, now you can do it all yourself!