Note: This project does NOT include [source access](#).#

Build It -- Downloading the Neutrino OS/Kernel Source[#](#)

This is the *easy* way to build the whole OS source (kernel, process manager, libraries, services, utilities ... the whole shebang). After you have done this, you will want to be much more discerning and want to understand [how the source tree works](#) and [how the build system works](#).

Pre-requisites[#](#)

To build the Neutrino OS source you need to have:

1. [Momentics Development Environment (6.4.1)](#) installed for the build tools.
2. Have [the OS source](#) available to your local machine.

Supported build environments include: Neutrino (x86 self-hosted), Windows, Linux and Solaris.
NOTE: At this time you can only build GNU configure components on Neutrino self-hosted.

These instructions assume you have pulled the source down to a single directory that will serve as the build root. This directory is referenced below as <BUILD_ROOT> and the contents should be as follows:

Unix	Windows

```% pwd```
```/home/somepath```
```% ls```
```trunk```
<BUILD_ROOT> is */home/somepath*
	```c:\> pwd```
	```c:/home/somepath```
	```c:\> dir```
	```...```
	```06/09/2007 01:07 PM <DIR> trunk```
	<BUILD_ROOT> is *c:/home/somepath*
	NOTE: Use forward slashes (/) for the environment variables used in the QNX build process.

In all cases below, <BUILD_ROOT> is just a placeholder, use the **full path** instead of actually putting in **<BUILD_ROOT>**.

Build It[#](#)

1. If you are **not** running Neutrino self-hosted, then you will need to tell the build process to ignore the content of the GNU configure style source modules. To do this run this [hide-gnu.sh](#) script.
   ```% ksh hide-gnu.sh <BUILD_ROOT>```
2. Create a staging directory for installed binaries and headers to go
   ```% cd <BUILD_ROOT>```
   ```% mkdir stage```
3. Configure the override makefile stub ```% cd <BUILD_ROOT>```
   ```% cat >qconf-override.mk```
   ```USE_INSTALL_ROOT=1```
   ```INSTALL_ROOT_nto=<BUILD_ROOT>/stage```
   ```^D```
4. Set the override environment variable to point at the makefile stub
   ```% export QCONF_OVERRIDE=<BUILD_ROOT>/qconf-override.mk```
5. Install all of the Header Files in the local stage
   ```% cd <BUILD_ROOT>/trunk```

```
% make OSLIST=nto hinstall
```
... lots of output, including copying to <BUILD_ROOT>/stage

6. Build the Neutrino OS binaries
```
% cd <BUILD_ROOT>/trunk
% make OSLIST=nto install
```
... lots of build output, binaries copied to <BUILD_ROOT>/stage...
One potential gotcha to be aware of here: if you installed the x86-only development environment, you also need to add "CPULIST=x86" to that make command, or the make will fail when it tries to build for a non-x86 platform
```
% make OSLIST=nto CPULIST=x86 install
```

The final binaries (libraries, executables) that you have built should now be found in the <BUILD_ROOT>/ stage tree. If the build failed, read the next section.

For a more detailed explanation of these build steps and what is going on, read on to the [understanding the Neutrino build process](#)

Hey! The Build Failed ... What's Up With That![#](#)

Some straightforward possibilities to consider if the build is not working for you:

1. Did you download the latest development environment as per the pre-requisites? If you are an existing QNX developer and you have a Momentics development environment, then some aspects of the build will work for you, but you will most likely be missing the makefile modifications done to ignore the *.svn* directories as part of the source build.
2. Is your staging directory properly populated? If you skipped the **make hinstall** step, then you will be missing headers. If you updated the source (via svn up) and pulled in new headers, then without a **make hinstall** you won't propagate the new headers into the staging area.
3. Are you trying to build a **GNU configure** style source module (identified by the presence of a GNUmakefile):
   a. You are trying to do this on a non-Neutrino host ... this is unsupported and you should re-run the hide-gnu.sh script above.
   b. You are missing some program **configure** is trying to launch (ie autoconf, m4, etc) ... you'll have to satisfy the missing pieces (**TODO:** Provide an ancillary package for these)

If those don't apply and you are still having problems, then let's face it, the source is likely busted.

We strive for a continuously building system, however in these exciting times of transparent development it is inevitable that there will be times where builds will fail because new source dependencies have been introduced that are not yet satisfied in the public published source tree. In these cases we will try to either:

- Accelerate the publishing of the required source module(s) from the [community roadmap](#)
- Request your patience that you work around the build by adding in a **Makefile.dnm** to the component that is not building (Makefile.dnm == Makefile Do No Make)

In all cases, please post to the [OS forums](#) if you run into trouble.

Other methods[#](#)

You can also use [Momentics IDE](#) to build the kernel.