Understanding the Neutrino Source Tree#

Directory Layout#

The Neutrino source tree is laid out in a hierarchal manner that quickly itemizes what a source's role in life is. The root of the tree contains broad source categories as it's top level directories:

services
: These are all of the main operating system services that run on a Neutrino system. An system service is a background process, often a Neutrino resource manager, that provides functionality to other applications via a message passing interface. In addition to services like pipe and mqueue, this is where you will find the Neutrino Kernel/Process Manager in services/system.

lib
: This directory contains the supporting libraries for the operating system. These libraries, both shared and dynamic, provide functionality that is common and shared among many applications. This is where you will find the C and C++ libraries (lib/c and lib/cpp), math and compression libraries (lib/m, lib/z, lib/ucl) and utility libraries (lib/qnx4, lib/compat).

utils
: This directory contains all of the operating system utilties. Utilities differ from services in that they are clients of system services and generally don't provide an API for other programs to access their data directly. The utilities tree is further split with directories a-z to facilitate navigation. For example the ls utility would be found in utils/l/ls while cat would be in utils/c/cat.

hardware
: This is home for most of the low level hardware, bit banging, drivers. The frameworks that these drivers fit into are generally located in the lib or services tree, but the hardware drivers for networking, block, serial, usb, graphics and others all live in this section of the source tree. This is also where you will find the home of the different BSP content (hardware/boards)

apps
: Sample applications, many of these based on the Photon windowing system, live here. You will find programs like the Photon File Manager (apps/pfm) as well as

ports
: Ported applications that are not necessarily maintained as part of the standard operating system distribution are included in this directory. These are often open-source projects that have been ported to Neutrino and are of general use. When an open source project is ported then the preference is always to work with the project to integrate the changes required to cleanly build and run the application on Neutrino.

Inside A Source Module#

Once you get inside one of the source modules, you will find that there is a lot of commonality in terms of the content(1). Much of this common structure is used by the Neutrino build process. The content will generally include:

- A Makefile file that contains a stub for the QNX recursive makefiles

```
% cat Makefile
LIST=CPU
include recurse.mk
```

- A common.mk file contains the build directives for the source module

```
% cat common.mk
ifndef QCONFIG
QCONFIG=qconfig.mk
endif
include $(QCONFIG)

INSTALLDIR=usr/sbin

define PINFO
PINFO DESCRIPTION=This is a nifty module
endef

include $(MKFILES_ROOT)/qtargets.mk
```

- A module.tmpl file that contains descriptive information about the source module and information used for automated build and packaging
- The CPU or OS variant directories, for example x86, mips, sh, arm and/or ppc
- A public directory. This directory is only present if the module provides any public header files. The content of this directory will be copied over entirely into the stage area when a make install or make hinstall command is performed on the module.

<hr> 1: Really, no two source modules are identical, they start out the same and then diverge as specific behaviour is required.

TODO: Continue with details about the layout, special directory names, module.tmpl