BSP & Drivers Source (for QNX 6.4.0 and 6.3.0 BSPs)#

Introduction#

As part of the QNX hybrid software model, QNX is making the BSP and driver source code available for public access. The source is available under the QNX Development Licenses (QDL) and under Apache copyright license version 2.0. The initial availability will be limited: no libraries, services, or utility source will be released. It will be a progressive release process; we will release code components once we have verified the intellectual property rights for all the related code.

Getting Started#

The BSP and driver code is best built in a staging area. The staging area is a directory used by the QNX BSP Makefiles to install the driver and look for headers, utilities, and prebuilt binaries (utilities, libraries). A staging area can be a simple directory or a BSP installation. A full BSP staging area has advantages over a simple directory; it includes most -- if not all -- of the utilities, header files, and libraries required by the driver. The simplest way to create this structure is to download a currently released BSP package for the target hardware from Foundry27 and unzip it into the chosen directory.

Important things to do remember about staging areas:

- They allow the source to build.
- They prevent the build / installation from polluting/overlaying your "standard" installation.
- A "make install" will install driver binaries into the directory pointed to by the QCONF_OVERRIDE environment variable.

This is always good practice when building QNX source anyway, but it's particularly important when building the BSP and Drivers tree, since a 'make install' generally has to be done from the root to build components in the correct order to satisfy dependencies. **Note:** Since it's possible to check out the entire development tree and have multiple development branches at once, it's important to set up a staging area at the trunk level or within a desired development branch. We also recommend that you have one staging area per development branch to avoid mixing releases.

This link, OS Source Build, covers some **Very Important Details**, most notably a description of how to create a staging area and a qconf-override.mk file (see steps 2, 3, and 4). Just to re-emphasize the importance of this: if you don't properly set things up, you stand a good chance of overwriting and possibly corrupting your base installation, and that would require a de-installation / re-installation to correct.

Just a couple of notes on the QCONF_OVERRIDE environment variable. On Windows, in a DOS shell, use the SET command (there is no export). You can make the environment variable "stick" by right-clicking on "My Computer" (select Properties). Select the "Advanced" tab and click on the "Environment Variables" button (bottom left in XP). Click on the "New" button and add in QCONF_OVERRIDE and the value and "OK" your way out. The next time you start a DOS shell, QCONF_OVERRIDE will already be set up for you. If you're running self-hosted (or under Linux), you can add the export command to your .profile script and it will also automatically get set when you start a new terminal session.

Source Tree Organization#

The source is laid out with a top level similar to other projects. There is the *trunk* directory, which contains the most up-to-date development source code. We expect that all code in the *trunk* directory will work its way into the general product at some point in time, however we don't guarantee that the *trunk* source will build without error, or operate perfectly if it does build. While we will make every effort to ensure that the *trunk* source will

indeed build and operate, we all make mistakes from time to time, so code will slip in from time to time that won't have everyone's best interests in mind. Hey, that's what you get when you sign up for live development!

The *branches* directory contains stable, released code that does build and work correctly. This is where source code used for official releases is derived. It may also contain source for highly experimental development that would adversely affect the trunk branch, or "specialized" code that might not be included as part of the general product. For 6.4.0 and 6.3.2, maintenance release branches (branches/6.4.0/trunk and branches/6.3.0/trunk respectively) have been created. These branches contain the source used to create 6.4.0 and 6.3.2 plus critical maintenance fixes (which will be released in upcoming 6.x.x.x upgrade patches if needed).

The *tag* directory is used to shelter BSP packages in their SVN form. Checking out a BSP tag is equivalent to unzipping the latest archive for a specific BSP (with its associated QNX Momentics distribution). Starting in release 6.4.1 of QNX Momentics, it's possible to directly import the SVN tags and build a BSP (see IDE documentation for more details).

Some relevant portions of the tree are:

- **bsp**
  - **/trunk/**
    - **hardware** - Contains device drivers, startup & IPL code
    - **lib** - Libraries (Not populated)
    - **services** - BSP & Driver services (Grouped alphabetically). **(Not populated)**
    - **tools** - BSP & Driver tools. **(Not populated)**
    - **utils** - BSP & Driver utilities (Grouped alphabetically). **(Not populated)**
  - **branches/6.4.0**
    - **/trunk/**
      - **hardware** - Contains device drivers, startup & IPL code
      - **lib** - Libraries (Not populated)
      - **services** - BSP & Driver services (Grouped alphabetically). **(Not populated)**
      - **tools** - BSP & Driver tools. **(Not populated)**
      - **utils** - BSP & Driver utilities (Grouped alphabetically). **(Not populated)**
  - **branches/6.3.0** - **(Not populated)**
    - **/trunk/**
      - **hardware** - Contains device drivers, startup & IPL code
      - **lib** - Libraries (Not populated)
      - **services** - BSP & Driver services (Grouped alphabetically). **(Not populated)**
      - **tools** - BSP & Driver tools. **(Not populated)**
      - **utils** - BSP & Driver utilities (Grouped alphabetically). **(Not populated)**

How do I get the source?#

The source code is available in the BSP and Drivers repository located on http://community.qnx.com/sf/scm/do/listRepositories/projects.bsp/scm.

To download the entire source code into your source directory, use :

- `svn checkout --username <userid> http://community.qnx.com/svn/repos/bsp`

where **<userid>** is the email address used to create your account on the QNX site.

To check out only the trunk development branch source, use:

- **`svn checkout --username <userid> http://community.qnx.com/svn/repos/bsp/trunk`**

To check out only the 6.4.0 development branch source, use:

- **`svn checkout --username <userid> http://community.qnx.com/svn/repos/bsp/branches/6.4.0/trunk`**

To check out only the 6.3.0 development branch source, use:

- **`svn checkout --username <userid> http://community.qnx.com/svn/repos/bsp/branches/6.3.0/trunk`**

NOTE: There is currently no information available for the 6.3.0 branch.

How do I build the source?#

If you don't properly create a staging area and suitable qconf-override.mk file, you risk the chance of corrupting your standard installation.

If you haven't already done so, install the srcversion patch for specific QNX distribution you are using from the download section of the BSP project as root.

1. cd $QNX_TARGET/../..
2. tar -xpf srcversion-patch-6.X.X.tar

Not required for 6.4.1
srcversion-patch-6.4.0.tar
srcversion-patch-6.3.2.tar

Note: The following sections aren't supported yet, as we haven't yet released any sub-component code. You'll have dependency issues unless you point your staging area to the install directory of a previously downloaded BSP.

From the trunk directory of your selected development branch:

- **`make install`**

Or if you prefer to only build for a specific CPU and variant:

- **`make CPULIST=arm VARIANTLIST=le install`**

That's it! All the associated header files for the build and binaries created will get installed in your staging directory.

Note: Building from the top-level directory builds everything in the correct order so that all dependencies are satisfied from your staging area. Simply doing a (cd <subcomponent> && make) probably won't work until you've built the world at least once.

Note 2:: You'll need to install the srcversion patch for QNX 6.4.0 from the download section of the networking project as root.

1. cd $QNX_TARGET/../..

2. tar -xpf srcversion-patch-6.4.0.tar

Back to Main Wiki#