Manually Packaging a 6.4.x BSP#

This page provides details on how to package a BSP for use with QNX Software Development Platform 6.4.x. The BSP packaging has been greatly simplified for 6.4.x. If you based your work on a previously released BSP, there should not be much to do, other than archiving and testing.

Layout#

The BSP layout should be as described in the "Building Embedded Systems" guide.

The `module.tmpl` file#

IDE 4.6#

The IDE no longer needs `module.tmpl` files, so you can omit them when you're packaging a BSP.

Prior to IDE 4.6#

Each software module (e.g. driver) needs a `module.tmpl` file in order to successfully import a BSP into the IDE. It is highly recommended that the user upgrades to IDE 4.6 to eliminate this requirement.

The `source.xml` file#

The root directory should have a `source.xml` file, which the IDE uses to import the BSP. The format of the `source.xml` is documented here.

New to IDE 4.6#

Some new XML tags have been added to the source.xml file format to facilitate the process of importing BSPs in IDE 4.6:

Tag	Description	Example	
<startup>	The startup module name used by the project builder	<startup>*startup-boardname*</startup>	
<ipl>	The IPL name used by the project builder	<ipl>ipl-boardname</ipl>	
<maturity>	The maturity level of the BSP. Must be one of { GA, Experimental }	<maturity>GA</maturity>	
<license>	Overall license information for the BSP	<license>QDL and Apache II</license>	

The `common.mk` files#

Ensure that each `common.mk` file has **`INSTALL_ROOT_nto`** and **`USE_INSTALL_ROOT`** variables defined. The **`INSTALL_ROOT_nto`** variable needs to be set to the relative path from the directory containing `common.mk` to the `install` sub-directory. The **`USE_INSTALL_ROOT`** variable needs to be set to **`1`**.

For example, in the `src/hardware/i2c/common.mk` makefile the variables will be defined as follows:

```
INSTALL_ROOT_nto = $(PROJECT_ROOT)/../../../install
```

USE_INSTALL_ROOT=1

Packaging the BSP#

Here's an overview of the steps required to create a BSP archive:

1. First run **`make clean`** from the root of the BSP development tree.
2. Make a copy of the BSP development tree:
   - On Linux and self-hosted Neutrino systems, use **`cp -r`**
   - On Windows, use explorer or **`cp`**.
3. Delete any files from the copy that should not ship with the BSP, such as binary build output under the `src` sub directory, temporary files created during development, any build files under images, and any revision-control files (such as `.svn` directories).
4. Also remove the source from the copy for any components for which the source is not being included.
5. If prebuilt binaries are required, build the master BSP, and copy the required prebuilt binaries from the development tree's `install` directory to the equivalent location under the copy's `prebuilt` directory.
6. Then archive the BSP using the following command from the root directory of the copy:

```
zip -r ../<bsp_packge_name>.zip .
```

Test the BSP#

After creating a BSP package, test the resulting zip file:

- by unzipping the archive in a temporary directory, building it, and loading it onto the target platform, and,
- by importing and building the BSP with the IDE.

If there are any problems, fix the issue, and create a new archive following the steps above.