NetBurner
Networking in 1 Day!

Product Change Notice

PCN 008

NetBurner Part Number: Eclipse IDE in 2.x Releases for macOS Big Sur
Implementation Date: February 22, 2021
Hardware Revision Number: n/a
Development Tools Revision Number: Eclipse in all 2.x NNDK Releases

1 Description

With Apple's release of Big Sur for macOS, 32-bit applications are no longer supported. All the NetBurner development tools will run on Big Sur with the exception of NBEclipse, which required 32-bit Java.

The latest NBEclipse in NetBurner tools 3.x is 64-bit and is not affected, but ColdFire based platforms prior to the MOD54415, MOD54417, NANO54415 and SB800EX must use the 2.x tool set.

However, you can still develop on macOS Big Sur with the methods described in this document. All references below are for running the NNDK 2.9.5 tool sets on Big Sur.

2 Method 1: Install a Virtual Environment

32-bit application can be run in a virtual environment running OSX Mojave or Windows. If using windows, Parallels is a common method.

3 Method 2: Command Line

The gcc compiler and build system work fine on Big Sur, and projects can be built with GNU Make and the editor of your choice. (even the general stand-alone 64-bit Eclipse). The process is to edit the source code, then in the terminal window, type "make" to build the project, or "make load" to build the project and flash it to your NetBurner device. The NetBurner examples are verified with the make system, and every example program has a makefile that can be used as an example. For most projects all that needs to be done is to edit a makefile and add your project's .c and .cpp files.

3.1 Build an Example Project

The makefiles already exist for NetBurner examples:

- Open a terminal window
- Navigate to any example directory
- Type "make" and verify it builds correctly
- Type "make load" to build and load it onto your NetBurner device

3.2 Create Your Own Project

The easiest way to start is to copy a makefile from an example project. If you are using the web server and an html folder, copy from an example that uses html as well so the proper commands are in the makefile.

- Create a folder for your new project
- Copy your source files and html folder if the web server feature is used
- Edit the makefile to add your source files (see description below)
- Build using the same method as in the example project description above

3.3 Editing Makefiles

The makefile for the SimpleHtml example is shown below:

```
NAME = SimpleHtml

CXXSRCS = main.cpp \


CXXSRCS += htmldata.cpp

#Uncomment and modify these lines if you have C or S files.
#CSRCS := foo.c
#ASRCS := foo.s

CREATEDTARGS = htmldata.cpp

include $(NBROOT)/make/main.mak

htmldata.cpp : $(wildcard html/*.*)
      comphtml html -ohtmldata.cpp
```

NAME Specifies the name of the resultant application .s19 file

CXXSRCS A list of source .cpp files. The '\' character is a line continuation. While you can list all source files on one line separated by spaces, it can be more readable if each .cpp file is on it's own line, such as:

```
CXXSRCS = main.cpp \
          foo1.cpp \
          foo2.cpp \
          foo3.cpp \
```

main.mak The system makefile that processes the build once all the source files are known

htmldata.cpp The HTML file that is automatically created from the file in your project's html folder. HTML is handled automatically, and you do not need to add makefile entries.