

Configuring the UEFI Development Environment

Tim Schooley [*tim@sbdev.net*]

Abstract

This document serves as a development environment configuration guide for anyone wishing to develop UEFI applications or drivers. The guide assumes basic familiarity with Linux.

1 Basic Requirements

The following requirements are given for the setup:

- Gentoo Linux, \geq 2007.1
- \geq GCC 4.1.1
- \geq sys-libs/glibc-2.5-r3
- \geq dev-java/sun-jdk-1.5.0.11

While the setup may work on earlier versions, this has not been tested by the author, and a basic Gentoo Linux setup will easily provide for the above.

2 Requirements of EDK2[1]

The following gentoo-installable packages are required by EDK2. The versions used for this guide are provided in brackets.

- Subversion[2] [dev-util/subversion-1.3.2-r4]
- E2FS Program Library [sys-fs/e2fsprogs-1.39-r2]
- Ant[3] [dev-java/ant-1.7.0]
- Ant-contrib[4] [dev-java/ant-contrib-1.0.beta2-r1]

These can be installed as follows. This assumes you are a user on a Linux system, logged into a console, with `sudo` access, and are in your home directory. Note that a symbolic link is required for the `ant-contrib.jar` file:

```
$ sudo emerge subversion dev-java/ant ant-contrib e2fsprogs
$ sudo ln -sf /usr/share/ant-contrib/lib/ant-contrib.jar \
/usr/share/ant/lib/ant-contrib.jar
```

Saxon[5] and XMLBeans[6] require manual installation due to their rather specific version requirements:

```
$ wget
http://www.mirrorservice.org/sites/ftp.apache.org/xmlbeans/binaries/xmlbeans-2.1.0.zip
$ sudo unzip xmlbeans-2.1.0.zip -d /opt/
$ rm xmlbeans-2.1.0.zip

$ wget http://prdownloads.sourceforge.net/saxon/saxonb8-1-1.zip
$ sudo mkdir /opt/saxonb8.1.1
$ sudo unzip saxonb8-1-1.zip -d /opt/saxonb8.1.1/
$ rm saxonb8-1-1.zip

$ sudo ln -sf /opt/saxonb8.1.1/saxon8.jar /opt/xmlbeans-2.1.0/lib/saxon8.jar
```

Some symbolic links are required for compilation:

```

$ ln -sf /usr/share/xerces-2/lib/xercesImpl.jar
/usr/share/xalan/lib/xercesImpl.jar
$ ln -sf /usr/share/xml-commons-external-1.3/lib/xml-apis.jar
/usr/share/xalan/lib/xml-apis.jar
$ ln -sf /usr/share/xml-commons-external-1.3/lib/xml-apis.jar
/usr/share/ant-core/lib/xml-apis.jar
$ ln -sf /usr/share/xerces-2/lib/xercesImpl.jar
/usr/share/ant-core/lib/xercesImpl.jar
$ ln -sf /usr/share/xalan/lib/xalan.jar /usr/share/ant-core/lib/xalan.jar

```

3 Checking out the EDK2 Code

The EDK2 Project Code is accessible using Subversion. In order to checkout, you need to have a registered account with them at <https://edk2.tianocore.org>.

```

$ cd ~
$ svn co https://edk2.tianocore.org/svn/edk2/trunk edk2

```

Now is a good time to read the build notes found at `edk2/edk2/BuildNotes.txt`.

4 Configuring the Environment Variables

There are a number of environment variables that require setting for compiling EDK2. For this, we write a shell script, and place it in `edk2/edk2/setupenv.sh`:

```

#!/bin/sh
export JAVA_HOME=/opt/sun-jdk-1.5.0.11/bin/
export XMLBEANS_HOME=/opt/xmlbeans-2.1.0
export ANT_HOME=/usr/share/ant
export TOOL_CHAIN="gcc"
export PATH="$PATH:$XMLBEANS_HOME/bin:$ANT_HOME/bin"
. edksetup.sh

```

This shell script should then be made executable, and then run:

```

$ chmod 755 edk2/edk2/setupenv.sh
$ ./edk2/edk2/setupenv.sh

```

Next, we have to configure the tool definitions for building. Edit the file `edk2/edk2/Tools/Conf/target.txt` and make the following changes:

- Replace `ACTIVE_PLATFORM = EdkNt32Pkg/Nt32.fpd` with `ACTIVE_PLATFORM = EdkUnixPkg/Unix.fpd`
- Replace `TOOL_CHAIN_TAG = MYTOOLS` with `TOOL_CHAIN_TAG = ELFGCC`

5 Building the TianoCross GCC Toolchain

If you would like your own toolchain for compiling EDK2, the following instructions will suffice. It seems this step isn't needed when using a target of `ELFGCC`, as *GenFwImage* can convert `ELF` images to `PE32+` images (Thanks to Tristan Gingold for pointing this out!).

A script is provided for building this, although it requires some slight changes. The script is found in `edk2/edk2/Tools/gcc/tianoCross-gcc-4.1`. The `CYGWIN_SNAP` variable needs to be changed to a recent/existing snapshot version. This can be found at <http://cygwin.com/snapshots/> [as of writing, this is 20070616]. Once this is modified, you can run the script as root, and it will install the toolchain in `/opt/tiano`:

```

$ cd Tools/gcc/
$ chmod 755 tianoCross-gcc-4.1
$ sudo ./tianoCross-gcc-4.1

```

Having built the new toolchain, we should then recompile the EDK2 tools:

```

$ cd ~/edk2/edk2/
$ . edksetup.sh ForceRebuild

```

Note that if the build fails with a “header not found” error for `Pcibus.h`, then this can be rectified by changing `Pcibus.h` to `pcibus.h` in the file `edk2/edk2/EdkModulePkg/Bus/Pci/PciBus/Dxe/PciHotPlugSupport.c`.

6 Building EdkUnixPkg

The UEFI Emulator for Unix is called EdkUnixPkg. To build this, run the following commands in sequence:

```
$ cd ~/edk2/edk2
$ build
```

If everything builds correctly, you should then be able to run the emulator with the following commands:

```
$ cd ~/edk2/edk2/Build/Unix/
$ . run.cmd
```

When the emulator loads up, it gives the following message:

```
”map: Cannot find required map name.”
```

This is normal. To get access to the firmware volumes, you need to execute the following at the shell, in order:

```
Shell> reconnect -r
Shell> map -r
```

This basically connects all the virtual devices, then provides a mapping to the block devices.

References

- [1] [Open Source]. EFI Development Kit II.
<https://edk2.tianocore.org/>.
- [2] [Open Source]. Subversion - Version Control System.
<http://subversion.tigris.org/>.
- [3] [Open Source]. The Apache Ant Project.
<http://ant.apache.org/>.
- [4] [Open Source]. Ant Contrib Tasks.
<http://ant-contrib.sourceforge.net/>.
- [5] Michael Kay. SAXON: The XSLT and XQuery Processor.
<http://saxon.sourceforge.net/>.
- [6] [Open Source]. The Apache XMLBeans Project.
<http://xmlbeans.apache.org/>.