



wolfSSL Now Compatible with OpenEmbedded and the Yocto Project

wolfSSL has developed two products for use within OpenEmbedded and the Yocto Project. These products include a wolfSSL lightweight, embedded SSL/TLS library recipe and a cURL with wolfSSL *.bbappend* file. Both of these tools were tested on the MinnowBoard using the Yocto Project's Poky distribution, the base for developing custom Linux systems, and the Minnowboard board support package (BSP), [meta-minnow](#).



Why wolfSSL with OE and the Yocto Project

wolfSSL is a lightweight, open source SSL/TLS library specifically designed for embedded systems. Developers building custom Linux with OpenEmbedded and/or the YoctoProject will benefit from using wolfSSL due to its ease of use, lightweight design, and low runtime memory. Specifications of wolfSSL include:

- Up to TLS 1.2 and DTLS 1.2
- Small Size: 20-100kB
- Runtime Memory: 1-36kB
- 20X smaller than OpenSSL
- Progressive algorithm and cipher support

cURL with wolfSSL

cURL is a command line utility developed for sending and receiving files through URL syntax.

In order to include cURL with wolfSSL in a custom Linux build with the YoctoProject/OE, the wolfSSL recipe and the [meta-wolfssl](#) layer must be included. Once both of these files have been added to a bitbake image (the custom Linux build created using Poky), any cURL commands entered using **--ssl** will be transferred with wolfSSL encrypting the data.

See the next section, **wolfSSL Bitbake Recipe**, for instructions on implementing the wolfSSL library into a custom Yocto Project image.



wolfSSL Bitbake Recipe

The wolfSSL bitbake recipe is located in **meta-oe/meta-networking/recipes-connectivity/cyassl/**. To use this recipe, [meta-oe](#) must be cloned and included in the *bbayers.conf* file under the **BBLAYERS** = "" section. To build an image with wolfSSL, the image must be customized by adding the line **IMAGE_INSTALL** = "cyassl" to the image file. To test if wolfSSL is properly installed, a bitbake command may be run on the file by entering **bitbake cyassl** within the build directory. The output should look similar to this:

```
~/poky/build$ bitbake cyassl
Loading cache: 100%
#####| ETA: 00:00:00
Loaded 1270 entries from dependency cache.
Parsing recipes: 100%
#####| Time: 00:00:00
Parsing of 902 .bb files complete (901 cached, 1 parsed). 1270 targets, 28
skipped, 0 masked, 0 errors.
NOTE: Resolving any missing task queue dependencies
```

```
Build Configuration:
BB_VERSION      = "1.23.1"
BUILD_SYS       = "x86_64-linux"
NATIVELSBSTRING = "Ubuntu-14.04"
TARGET_SYS      = "i586-poky-linux"
MACHINE         = "minnow"
DISTRO          = "poky"
DISTRO_VERSION  = "1.6+snapshot-20140813"
TUNE_FEATURES   = "m32 core2"
TARGET_FPU      = ""
meta
meta-yocto
meta-yocto-bsp  =
"master:7c1a975a1c2fd884aa9f6f4736656d854a6c5edb"
meta-intel     = "master:6aa0c6a6a75f6a4c2fda8e59be1212464307c69a"
meta-minnow    =
"master:7bdcd1140b729598bae6246a4bbc21c3950aadd8"
meta-wolfssl   =
"master:84158b515206003eb217f2f99c07030890e8aaa4"
```

```
NOTE: Preparing runqueue
NOTE: Executing SetScene Tasks
NOTE: Executing RunQueue Tasks
NOTE: Tasks Summary: Attempted 395 tasks of which 383 didn't need to
be rerun and all succeeded.
```

Learn More

For more information about the wolfSSL OpenEmbedded and YoctoProject components, please contact us at info@wolfssl.com, or visit our website www.wolfssl.com.

www.wolfssl.com

Copyright © 2014 wolfSSL Inc. All Rights Reserved