

Using wolfSSL with TI-RTOS

| |
|---|
| Contents |
| 1 Supported targets and products |
| 1.1 Targets |
| 1.2 Products |
| 2 Download |
| 2.1 TI-RTOS |
| 2.2 wolfSSL |
| 3 Installation |
| 3.1 TI-RTOS |
| 3.2 wolfSSL |
| 4 Build Instructions |
| 4.1 For wolfSSL v3.8.0 or later |
| 4.1.1 Setup |
| 4.1.2 Build |
| 4.2 For older wolfSSL products |
| 4.2.1 Setup |
| 4.2.2 Build |
| 5 Linking with the wolfSSL library in your TI-RTOS application |
| 5.1 TI Compiler builds within CCS |
| 5.2 GNU Compiler builds within CCS |
| 5.3 IAR Compiler builds within IAR Embedded Workbench |
| 5.4 For Command line builds |
| 5.5 For TI Command line builds |
| 5.6 For GNU Command line builds |
| 5.7 For IAR Command line builds |
| 6 wolfSSL's TI-RTOS examples |
| 7 Benchmarks |
| 8 CyaSSL support |
| 9 References |
| 9.1 User guides and manuals |
| 9.2 Support |

The **wolfSSL Embedded SSL Library** is a lightweight Secure Sockets Layer (SSL)/Transport Layer Security (TLS) library written in ANSI C and targeted for embedded, RTOS, and resource constrained environments primarily because of its small size, speed, and feature set.

TI-RTOS for TivaC product integrates wolfSSL with TI-RTOS NDK stack and provides secure TCP and HTTP examples that show how a TI-RTOS application can use the wolfSSL stack.

Supported targets and products

Targets

- ARM Cortex-M4F on the [TM4C1294XL Connected Launchpad](#)
- ARM Cortex-M4F on the [TM4C129EXL Crypto Connected Launchpad](#)

Products

- TI-RTOS v2.14.04.31 or later
- wolfSSL v3.6.8 - v3.11.0

Download

Download TI-RTOS and wolfSSL as follows:

TI-RTOS

Download the **TI-RTOS for TivaC** product from either of these locations:

- The App Center in Code Composer Studio
- [TI-RTOS webpage](#)

WolfSSL

Download the **wolfSSL** product from:

- [WolfSSL github downloads](#) -- a GPL licensed product
- For commercial licenses, please contact [wolfSSL](#)

Installation

Install TI-RTOS and wolfSSL as follows:

TI-RTOS

- Run the installer and follow the instructions. We recommend installing TI-RTOS in the **default location (i.e. C:\ti)**.

WolfSSL

- If you downloaded an installer, run the installer and follow the instructions.
- If you downloaded a *.zip file, extract the contents to your disk (for example, in C:\ti\wolfssl).

Build Instructions

For WolfSSL v3.8.0 or later

Setup

- Open a terminal or command prompt and type:

```
cd wolfssl_install_dir\tirtos
```
- Edit the `products.mak` file. Update the `XDC_INSTALL_DIR`, `BIOS_INSTALL_DIR`, `NDK_INSTALL_DIR` and `TIVAWARE_INSTALL_DIR` variables. XDCtools, BIOS, NDK, Tivaware products are part of the TI-RTOS installation. Update the code generation tools path for any of the tool chains - `ti.targets.arm.elf.M4F` (i.e. TI), `iar.targets.arm.M4F` (i.e. IAR) or `gnu.targets.arm.M4F` (i.e. GCC). After modification, these variable definitions should look similar to the following if you are working on Windows. (Windows users: note the use of "/" in the path).

```
XDC_INSTALL_DIR = C:/ti/xdctools_3_31_01_33
BIOS_INSTALL_DIR = C:/ti/tirtos_tivac_2_16_00_00/products/bios_6_45_01_23
NDK_INSTALL_DIR = C:/ti/tirtos_tivac_2_16_00_00/products/ndk_2_25_00_08
TIVAWARE_INSTALL_DIR = C:/ti/tirtos_tivac_2_16_00_00/products/TivaWare_C_Series-2.1.1.71b
ti.targets.arm.elf.M4F = C:/ti/ccsv6/tools/compiler/ti-cgt-arm_5.2.4
```

Note: Please set `XDCTOOLS_JAVA_HOME` variable if you are using XDCtools from the CCS installation. The variable has to be set to the JRE path (which is available in CCS).

For example: `export XDCTOOLS_JAVA_ROME = C:/ti/ccsv6/eclipse/jre`

Build

WolfSSL library for TI-RTOS has been configured to include the ciphers that are enabled by default. Users can update the configuration to include/exclude ciphers from the library. The TI-RTOS configuration in wolfSSL can be found `wolfssl_install_dir\wolfssl\wolfcrypt\settings.h` under `#ifdef WOLFSSL_TIRTOS`. Read [building wolfSSL manual](#) for the various build defines.

After configuration update (if any), build the wolfSSL libraries as follows:

- To build:

```
make -f wolfssl.mak all
```
- To clean:

```
make -f wolfssl.mak clean
```

If the "make" tool is not installed on your machine, you can use the "gmake" tool available in XDCtools which is installed along with TI-RTOS.

For older wolfSSL products

Setup

In order to build wolfSSL, you first need to update the wolfSSL installation directory path in TI-RTOS. Follow these steps:

- Open the `tirtos.mak` file in the TI-RTOS installation directory.
- In `tirtos.mak`, update the definition of `WOLFSSL_INSTALL_DIR` (or `WOLFSSL_INSTALLATION_DIR`) to point to the installed wolfSSL product path.
- (Optional) Update other build options, such as the code generation tools path and build targets. Information about these options can be found in the *TI-RTOS User's Guide* (SPRUHD4).

Build

WolfSSL library for TI-RTOS has been configured to include the ciphers that are enabled by default. Users can update the configuration to include/exclude ciphers from the library. The TI-RTOS configuration in wolfSSL can be found `wolfssl_install_dir\wolfssl\wolfcrypt\settings.h` under `#ifdef WOLFSSL_TIRTOS`. Read [building wolfSSL manual](#) for the various build defines.

Note: For WolfSSL versions 3.6.6 and later, define `HAVE_BCC` in `wolfssl_install_dir\wolfssl\wolfcrypt\settings.h` under `#ifdef WOLFSSL_TIRTOS`. This workaround is required to run the TI-RTOS HTTPS example.

After configuration update (if any), build the wolfSSL libraries as follows:

- Open your choice of terminal or command prompt.
- Type the following command:

```
cd tirtos_install_dir
```
- To build:

```
make -f tirtos.mak wolfssl
```
- To clean:

```
make -f tirtos.mak clean-wolfssl
```

If the "make" tool is not installed on your machine, you can use the "gmake" tool available in XDCtools which is installed along with TI-RTOS.

Linking with the wolfSSL library in your TI-RTOS application

Examples are provided as a part of the TI-RTOS product. See *TI-RTOS for TivaC Getting Started Guide* (SPRUHJ5) for detailed information on importing projects into CCS or IAR and a list of examples that use WolfSSL.

The wolfSSL library path must be added to both the compiler and linker build options.

This process differs based on the compiler toolchain that you are building for (TI, IAR, and GNU). The library file path `wolfssl.lib` will either be:

For software based cipher:

```
wolfssl_install_dir\tirtos\packages\ti\net\wolfssl\lib\wolfssl.atarget
```

For hardware accelerated cipher (works only on Crypto Connected Launchpad):

```
wolfssl_install_dir\tirtos\packages\ti\net\wolfssl\lib\wolfssl_tm4c_hw.atarget
```

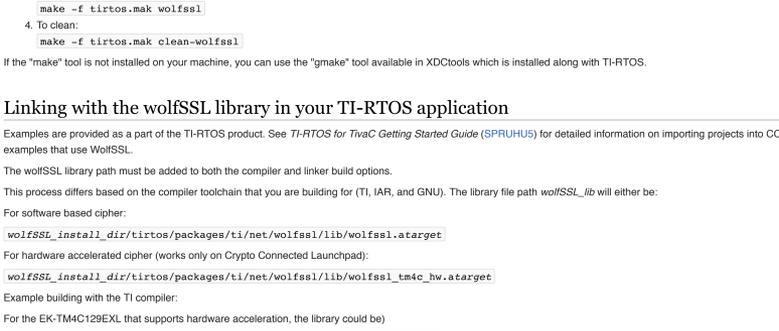
Example building with the TI compiler:

For the EK-TM4C129EXL that supports hardware acceleration, the library could be)

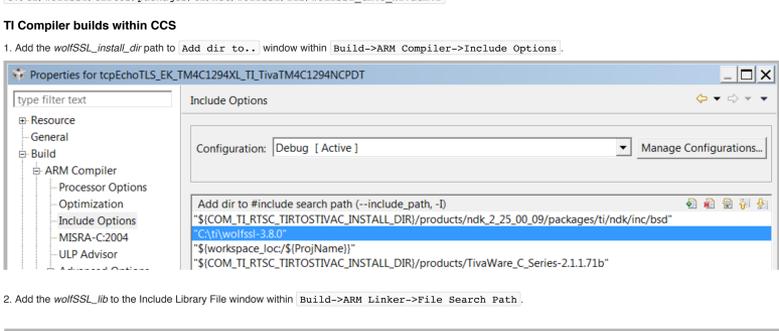
```
C:/ti/wolfssl/tirtos/packages/ti/net/wolfssl/lib/wolfssl_tm4c_hw.aem4f
```

TI Compiler builds within CCS

- Add the `wolfssl_install_dir` path to the `Add dir to..` window within `build->ARM Compiler->Include Options`.

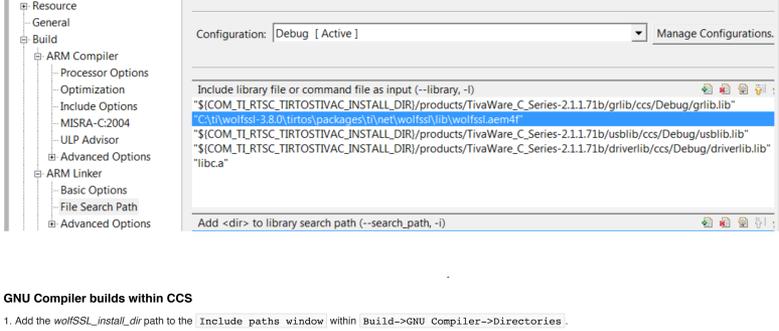


- Add the `wolfssl.lib` to the `Include Library File` window within `build->ARM Linker->File Search Path`.



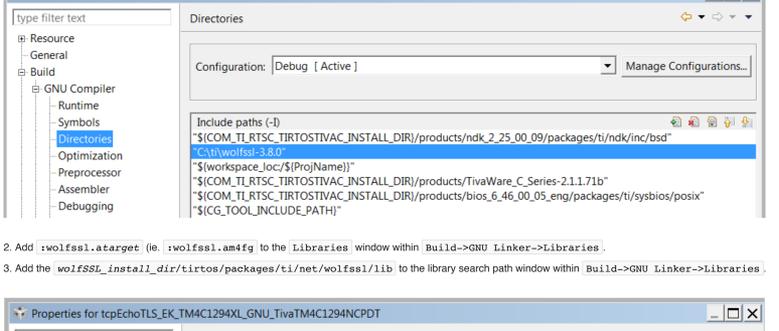
GNU Compiler builds within CCS

- Add the `wolfssl_install_dir` path to the `Include paths` window within `Build->GNU Compiler->Directories`.



- Add `wolfssl.atarget` (i.e. `wolfssl.am4fg`) to the `Libraries` window within `Build->GNU Linker->Libraries`.

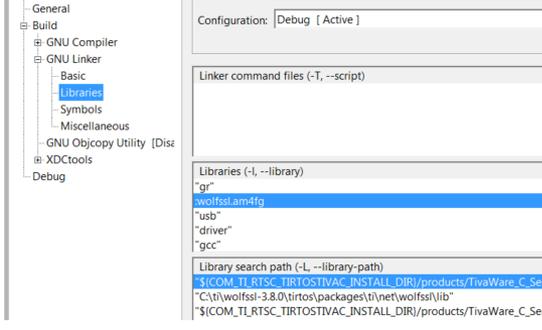
- Add the `wolfssl_install_dir\tirtos\packages\ti\net\wolfssl\lib` to the `library search path` window within `Build->GNU Linker->Libraries`.



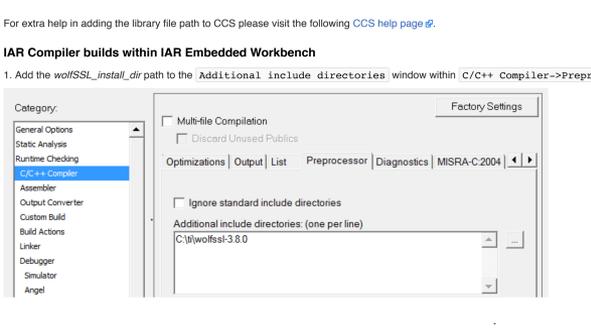
For extra help in adding the library file path to CCS please visit the following [CCS help page](#).

IAR Compiler builds within IAR Embedded Workbench

- Add the `wolfssl_install_dir` path to the `Additional include directories` window within `C/C++ Compiler->Preprocessor`.



- Add the `wolfssl.lib` path to the `Additional libraries` window within `Linker->Library`.



For Command line builds

The wolfssl library path must be added to the `makedefs` file to ensure the library ordering is correct to avoid linker errors.

Within the generated TIRTOS examples directory, located within your TIRTOS install directory, edit the

```
"TIRTOS_examples_dir"/"COMPILER"/"BOARD_dir"/makedefs file and add the correct WolfSSL library(SIRTOS) to the LIBFLAGS variable.
```

For reference, example link lines are shown for each toolchain below

For TI Command line builds

```
Ex 1) LIBFLAGS = -l"<WOLFSSL_lib"> <LINKERFLAGS> -llibc.a
```

```
Ex 2)
```

```
LIBFLAGS = -l"<WOLFSSL_lib"> -I$(TIVAWARE_INSTALL_DIR)/grib/ccs/debug/grib.lib -I$(TIVAWARE_INSTALL_DIR)/driverlib/ccs/Debug/driverlib.lib
```

```
EK_TM4C1294XL.cmd -m$(NAME).map --warn_sections --rom_model -I$(CODEGEN_INSTALL_DIR)/lib -llibc.a
```

For GNU Command line builds

The "WOLFSSL_lib" must appear after the specified linker flags but before all math libraries.

```
Ex 1) LIBFLAGS = <LINKERFLAGS> --gc-sections wolfssl_lib "MATH_lib"
```

```
Ex 2)
```

```
"C:/ti/ccsv6/tools/compiler/gcc-arm-none-eabi-4.8-2014q3/bin/arm-none-eabi-gcc" EK_TM4C1294XL.obj httpsgnet.obj -Wl,-T,EK_TM4C1294XL.lds -Wl,-Map,httpsgnet.map -Wl,-T,httpsgnet/linker.cmd -ldriver -larch-arm7e-m -mthumb -mfloat-abi-hard -mcpu=fpv4-sp-d16 -nostartfiles -static -Wl,--gc-sections wolfssl_lib -lgcc -lc -lm -lrdimon -o .out
```

For IAR Command line builds

```
Ex 1) LIBFLAGS = wolfssl_lib "LINKERFLAGS"
```

```
Ex 2)
```

```
LIBFLAGS = wolfssl_lib $(TIVAWARE_INSTALL_DIR)/driverlib/ewarm/Exe/driverlib.a --config EK_TM4C1294XL.icf --map $(NAME).map --silent --cpu=Cortex-M4F --entry=_iar_program_start
```

Note: The crypto hardware accelerator library may need extra heap. Increase `BIOS.heapSize` in the `cfg` file.

wolfSSL's TI-RTOS examples

Examples from the wolfSSL team can be downloaded from [wolfSSL's examples GitHub](#).

Benchmarks

For more wolfSSL benchmark information, see [wolfSSL's benchmarks](#).

CyaSSL support

Read [Using CyaSSL with TI-RTOS](#) wiki for using older versions of wolfSSL (formerly CyaSSL) with TI-RTOS.

References

User guides and manuals

- TI-RTOS for TivaC Getting Started Guide* (SPRUHJ5)
- TI-RTOS User's Guide* (SPRUHD4)
- [WolfSSL Manual](#)

Support

- TI-RTOS support forum
- wolfSSL support forum

TI E2E™ Community **Engage in the TI E2E™ Community** Ask questions, share knowledge, explore ideas **For technical support please post your questions at <http://e2e.ti.com>. Please post only comments about the article *Using wolfSSL with TI-RTOS* here.**

Links

- [Amplifiers & Linear](#)
- [Audio](#)
- [Broadband RF/IF & Digital Radio](#)
- [Clocks & Timers](#)
- [Data Converters](#)
- [DLP & MEMS](#)
- [High-Reliability](#)
- [Interface](#)
- [Power Management](#)
- [Processors](#)
- [ARM Processors](#)
- [Digital Signal Processors \(DSP\)](#)
- [Microcontrollers \(MCU\)](#)
- [OMAP Applications Processors](#)
- [Switches & Multiplexers](#)
- [Temperature Sensors & Control ICs](#)
- [Wireless Connectivity](#)

Categories: [Operating Systems \(OS/RTOS\)](#) | [Pages with broken file links](#) | [Under Construction](#)