Leveraging MATHLIB RTS functions in DSP application

The MATHLIB provides optimized arithmetic, trigonometric and logarithmic functions. These routines are typically used in computationally intensive real-time applications where optimal execution speed is critical. By using these routines instead of the routines found in the existing run-time-support libraries, you can achieve execution speeds considerably faster without rewriting existing code. The MATHLIB library includes all floating-point math routines currently provided in existing run-time-support libraries. These new functions can be called with the current run-time-support library names or the new names included in the MATHLIB library. The library provides both single and double precision floating point functions.

BUILD MATHLIB with OVERRIDE_RTS Option: (Refer only to rebuild MATHLIB else use prebuilt library)

Default operation of the MATHLIB functions may require users to modify their code to use C, inlined or vector version of the optimized functions. However, there is an option OVERRIDE_RTS that allows users to take benefit of this library with no modification of the code by replacing links to existing RTS library functions.

This required users to rebuild the MATHLIB with the OVERRIDE_RTS flag in the build files:

mathlib_c66x_3_1_1_0\packages\ti\mathlib\lib

Locate OVERRIDE_RTS=0 in the .mk files and replace it with OVERRIDE_RTS=1 and rebuild the library.

Prebuilt MATHLIB required to override RTS library functions is provided:

- mathlib_rts.ae66

How do I know if MATHLIB benefits my application?

Check the map file for your application. If the map file indicates that you are using RTS function for any of the trigonometric, logarithmic or arithmetic functions provided by MATHLIB then overriding those functions will give improved performance. Check for following symbols:

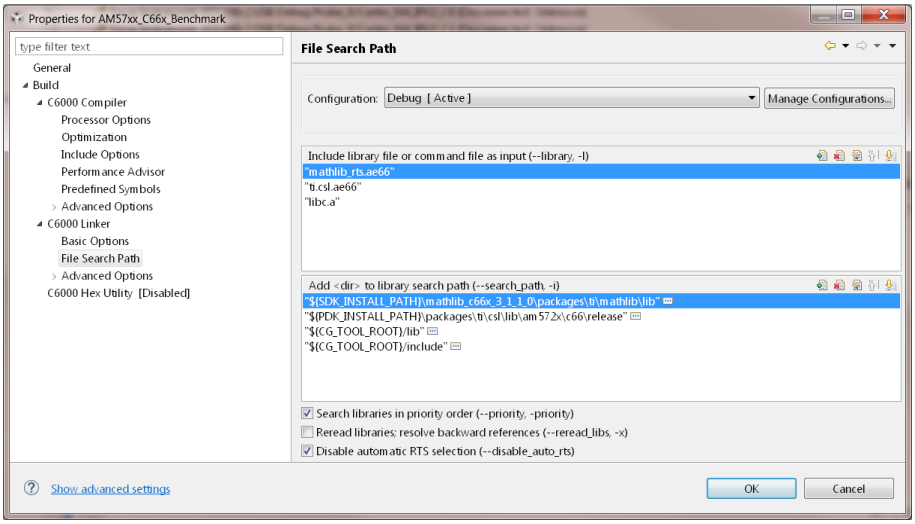
`__c6xabi_divd, sin, cos, log, etc in the .map file for the application. Example:`

```
C:\ti\AM57xx_PRSDK42\ti-cgt-c6000_8.2.2\lib\rts6600_elf.lib
   divd.obj                       1536    0         0
   sin.obj                        672     0         64
```

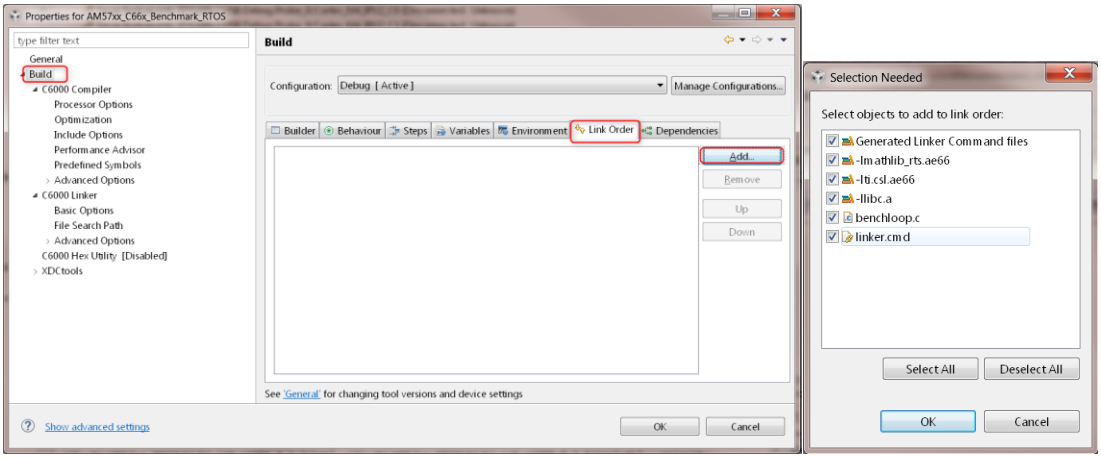
```
80001720  __c6xabi_divd
800020a0  sin
800020a0  sinl
```

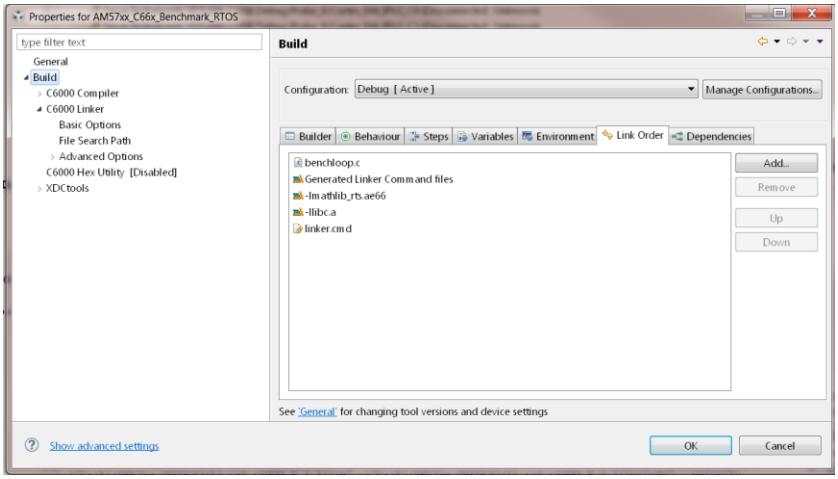
Steps to leverage MATHLIB in your project:

1. Copy mathlib_rts.ae66 to location_mathlib_c66x_3_1_1_0\packages\ti\mathlib\lib
2. In the project, add the library path and the library as shows in screenshot below:



3. Go to build options, and link Order Tab as shown below and link math_rts.ae66 before libc.a which links the RTS libraries:





4. You are now ready to rebuild the application. After you rebuild the application recheck the .map file to confirm mathlib is being used in the application by confirming symbols from mathlib are included. Example, map file will show:

```
80006400    00000080     mathlib_rts.ae66 : divdp.oe66 (.text:optci)

80007a80    00000020     mathlib_rts.ae66 : divdp.oe66 (.text)
80007aa0    00000020                      : sindp.oe66 (.text)



C:/ti/mathlib_c66x_3_1_1_0/packages/ti/mathlib/lib/mathlib_rts.ae66
 sindp.oe66                  384     0          0
divdp.oe66                   160     0          0
```

Example RTOS Application with MATHLIB optimization

Refer to the Example project: AM57xx_C66x_Benchmark_RTOS_MATHLIB project