Chapter 2. Installing the IQmath Library

2.1. IQmath Package Contents

The TI IQmath library can be used in both C and C++ programs and it consists of 5 parts:

1) The IQmath header files
   The header files include the definitions needed to interface with the IQmath library.
   C programs use IQmathLib.h
   a) C++ programs use both IQmathLib.h and IQmathCPP.h
2) The IQmath object library. The library contains all of the IQmath functions and look-up tables. There are two builds of the library:
   a) IQmath.lib: This build of the library can be linked with code built for fixed-point.
   b) IQmath_f32.lib: This build of the library can be linked with code built with the --float_support=fpu32 switch. This can be useful for mixing IQmath with native floating-point code on devices with the C28x+FPU.
3) Example linker command files. The example linker command files allocate the sections used by the IQmath library. For some sections the location is device specific. For example, the tables used by the IQsin and IQcos functions are located within the boot ROM of the device.
4) Legacy IQmath GEL file. These gel functions are useful when using a version of Code Composer Studio that does not support IQ data types directly. The .gel file is not needed when using Code Composer Studio V3.3 or Code Composer Studio V4.
5) Example programs

2.2. How to Install the IQmath Library

The IQmath library is distributed in the form of a self-extracting ZIP file. By default, the install restores the IQmath library individual components in the directory structure shown below.

NOTE:

The directory structure of some earlier versions differed from what is shown below. It has been reorganized to reduce duplication of files. The <base> directory has been changed to correspond to ControlSUITE.

<base> install directory is C:\ti\controlSUITE\libs\math\IQmath\<version>

<base>\doc	Contains this file
<base>\include	The IQmath header files C code uses IQmathLib.h C++ code uses IQmathLib.h and IQmathCPP.h
<base>\lib	The IQmath library files. These are used by both C and C++ Refer to Chapter 4.
<base>\gel	Legacy GEL file for debug Refer to section 3.10.

The following example project run on CCS V4:

<base>\examples_ccsv4\C	**C example: Refer to ReadMe_SampleC.txt**
<base>\examples_ccsv4\C\<device>	Each device family has its own project folder
<base>\examples_ccsv4\C\source	Shared source code for the C example
<base>\examples_ccsv4\Cpp	**C++ code example: Refer to ReadMe_SampleCpp.txt**
<base>\examples_ccsv4\Cpp\source	Shared source code for the C++ example
<base>\examples_ccsv4\Cpp\<device>	Each device family has its own project folder
<base>\examples\cmd	Linker command files used by the examples

<base>\examples_ccsv4\graph_properties
> The files in this directory can be used to setup the watch window and graphs in CCS 4.

The following example projects run on CCS V3.3:

<base>\examples\C	C example: Refer to ReadMe_SampleC.txt
<base>\examples\C\projects	CCS projects for the C example
<base>\examples\C\source	Source code for the C example
<base>\examples\Cpp	C++ code example: Refer to ReadMe_SampleCpp.txt
<base>\examples\Cpp\projects	CCS projects for the C++ example
<base>\examples\Cpp\source	Source code for the C++ example
<base>\examples\cmd	Linker command files used by the examples