

Migrating Code Composer v4.12 Projects to Code Composer Studio v2

Ki-Soo Lee

Code Composer Studio, Applications Engineering

ABSTRACT

This application report describes the process of migrating an existing Code Composer v4.12 project to a Code Composer Studio™ Integrated Development Environment (IDE) v2 project.

Contents

1	Introduction	1
2	Converting a v4.12 Project File to a v2 Project File	2
2.1	Converting a *.mak File to a *.pj1 File	2
2.2	Migration Issues	4
2.2.1	Project Build Options	4
2.2.1	C Runtime Libraries	5
2.2.3	Include Search Path	7
2.2.4	Linker Command File	7
2.2.5	Project Configuration	8

1 Introduction

Code Composer Studio IDE v2 contains new project management features that improve the usability of the Code Composer Studio IDE. Such features include the ability to open multiple projects in the workspace and support for multiple build configurations/targets. To support these new features, both the format and syntax of Code Composer Studio project files have been modified. Also, Code Composer Studio v2 project files use a new file extension (*.pj1) to differentiate them from v4.12 project files. Because of these differences, projects from v4.12 must be migrated to the v2 project format. This application report shows each step of the process.

NOTE: The TMS320C24x™ examples in this document show a migration from Code Composer v4.12 to Code Composer Studio IDE v2.

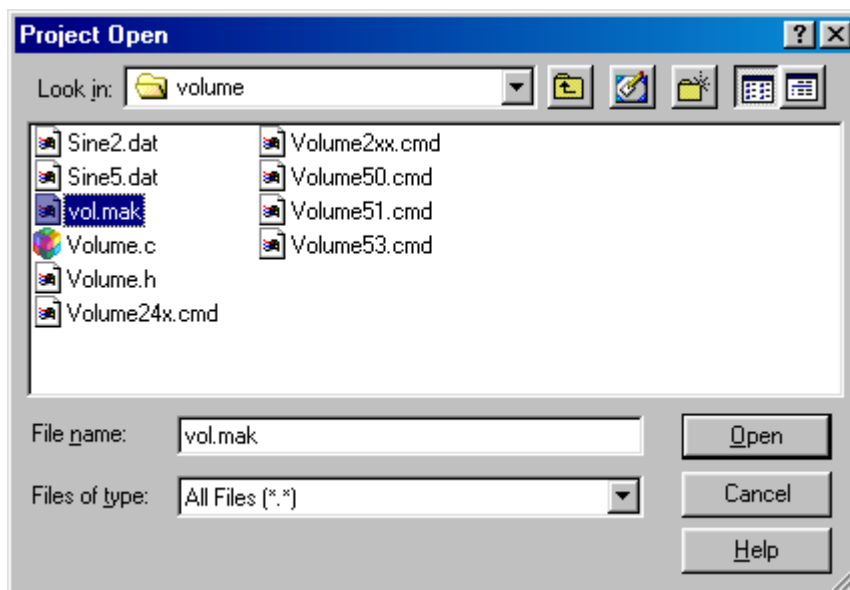
2 Converting a v4.12 Project File to a v2 Project File

2.1 Converting a *.mak File to a *.pjt File

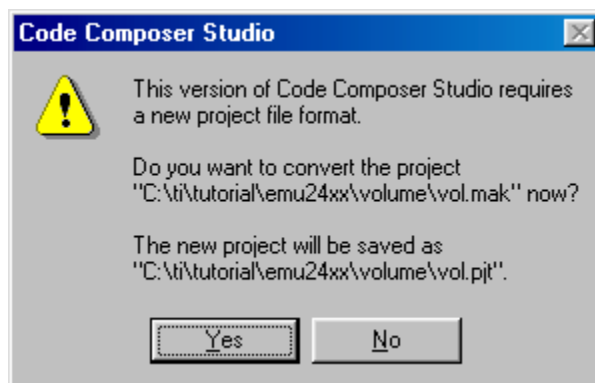
When you open a v4.12 project file in Code Composer Studio IDE v2, a new project file is automatically created. The new project file (*.pjt) is created in the same directory. The old project file (*.mak) remains unchanged. In this example, we will open a v4.12 project file called vol.mak and convert it into a v2 project file (vol.pjt). For this demonstration, the project folder is in `C:\ti\tutorial\emu24xx\volume\`.

NOTE: When a path name is given, <install path> will be used to refer to the path where Code Composer Studio IDE v2 is installed. The default install path name is `C:\ti`. Please be aware of this when full path names are shown in screen shots.

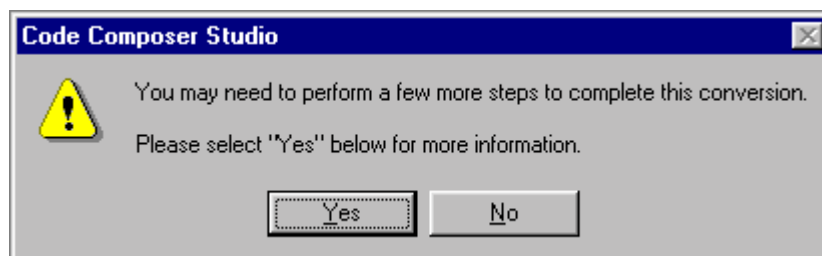
1. From the Project menu, choose Open.
2. In the Files of type drop-down list, select All Files or CCS 1.x Project Files (*.mak). Browse to and select vol.mak.



3. Click Open.
4. A prompt informs you that the project is stored in an old format and asks if you want to convert the project to the new format.



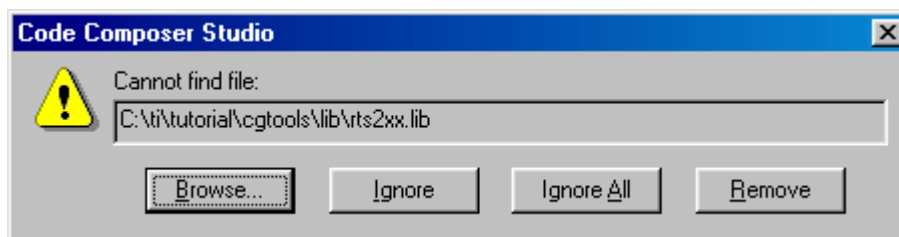
5. Click Yes.
6. A prompt informs you that more steps may be necessary to completely migrate the project successfully.



7. Click Yes. This brings up the on-line help documentation for migrating v4.12 projects to v2. Close the help documentation for now.
8. The project file has been successfully migrated to a v2 project file and the new project file is saved and renamed vol.pjt.

During project conversion, a Code Composer Studio dialog box may inform you that a particular file cannot be found at the location specified in the v4.12 project file.

Specify the correct location of the file by clicking on the Browse button, and specifying the correct location of the file in the Open dialog file, and clicking Open.



In this example, you browse to and select rts2xx.lib.

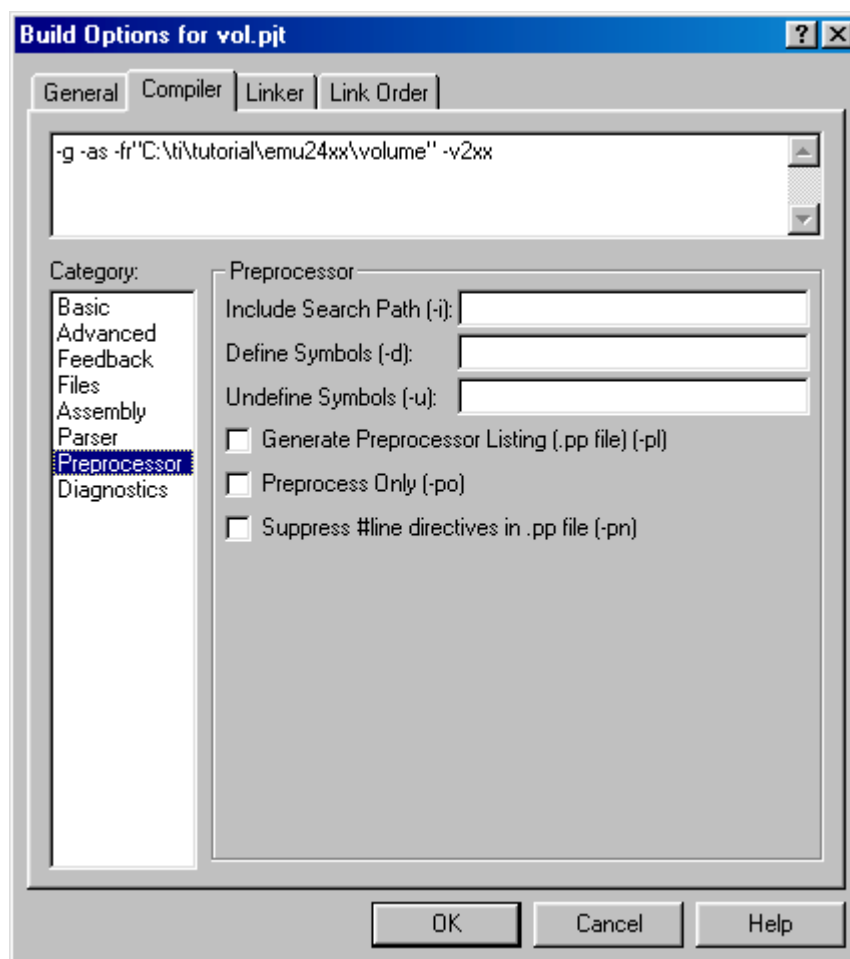
2.2 Migration Issues

Even after a clean v4.12 to v2 project conversion, more changes may still need to be made to the project file to ensure a smooth conversion. More complex project files that use additional libraries, have additional include paths, or have other build options enabled may still have some migration issues that need to be resolved. The following sections describe what these issues may be, how to resolve them and how to avoid other potential issues.

2.2.1 Project Build Options

The converted project uses the same build options that were set for your Code Composer v4.12 project. The installed directory structure for Code Composer Studio v2 differs from that of v4.12. If your v4.12 project defines search paths in the build options, it may be necessary to update these paths to point to the correct v2 installed directories. For example, if you used the Include Search Path option to specify the location of installed header files, open the Build Options dialog box and specify the path to the desired Include folder in v2. For example:

1. Select Project→Build Options.
2. In the Compiler dialog, select the Preprocessor category.



3. Edit the Include Search Path field as needed.

The Compiler options dialog contains additional options that define paths such as Asm Directory and Obj Directory. The Linker options dialog contains those that define Library Search Path and Include Libraries. Also note that the assembler tab is no longer a part of the Project Build Options. In v2, the Compiler tab is used to set compiler and assembler options.

To change the Library Search Path and library names in the Include Libraries field:

1. Select Project→Build Options. Then select the Linker tab.

2. Modify the Library search path as necessary.
3. Select OK to close the Build Options dialog.

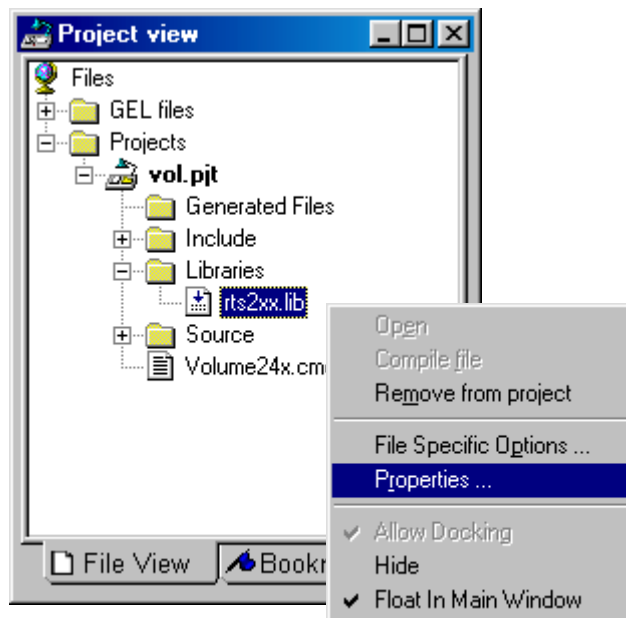
For more information on setting build options, see the Code Composer Studio v2 online help: Help>Contents>Using CCS IDE->Project Environment->Working with Projects->Building a Project->Setting Build Options.

2.2.2 C Runtime Libraries

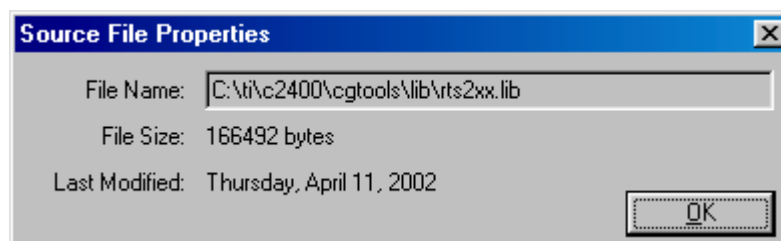
If the converted project contains a C runtime library, and you installed Code Composer v4.12 and Code Composer Studio IDE v2 under different paths, it is possible that a v4.12 runtime library path is specified for your converted v2 project.

The following steps describe how to ensure that the correct version of the runtime library is specified:

1. View the current runtime library specification.
 - a. In the Project View, expand the Libraries folder to view the runtime library file (*.lib).
 - b. Right-click on the runtime library filename and select Properties.



- c. In the Source File Properties dialog box, view the File Name field and verify that the correct path to the library file is specified. In most cases, the names of the C runtime libraries are the same in v4.12 and v2; therefore, it is necessary to use the path to verify the correct version.

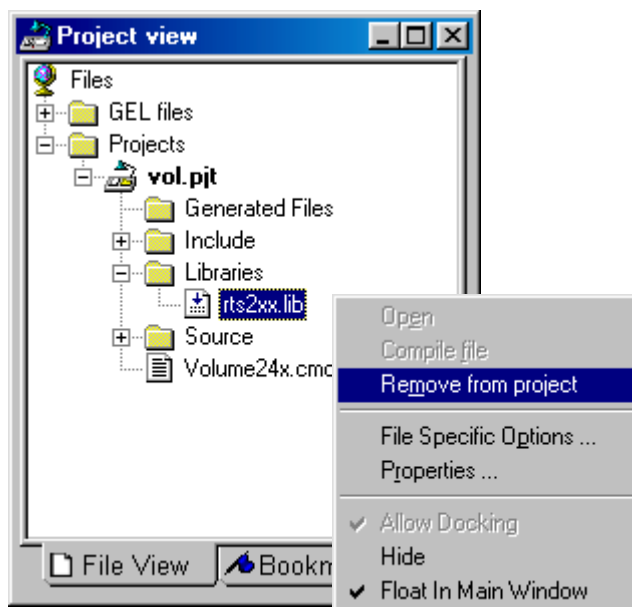


NOTE: If you installed Code Composer Studio v2 in C:\ti, the C runtime libraries are located in C:\ti\target\cgttools\lib*.lib, where target represents your target device: c2800, or c2400.

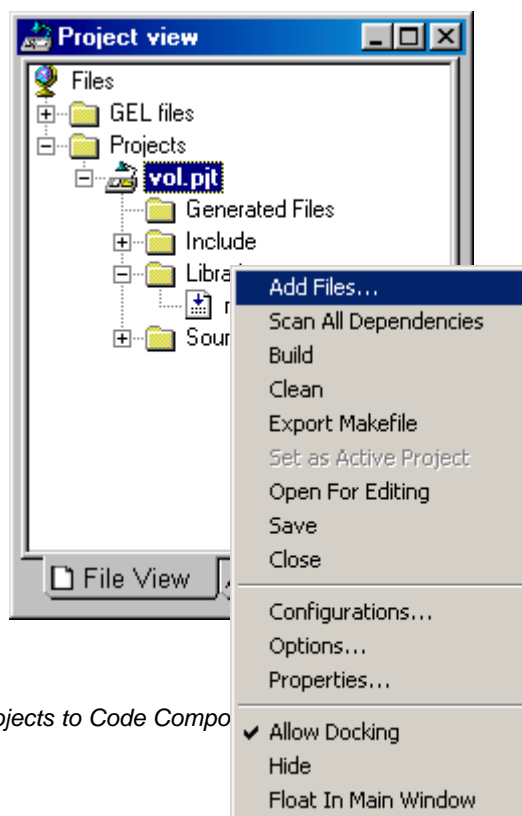
2. If necessary, change the runtime library specification.

To specify a new runtime library, you must remove the existing file from the project, and then add the desired file.

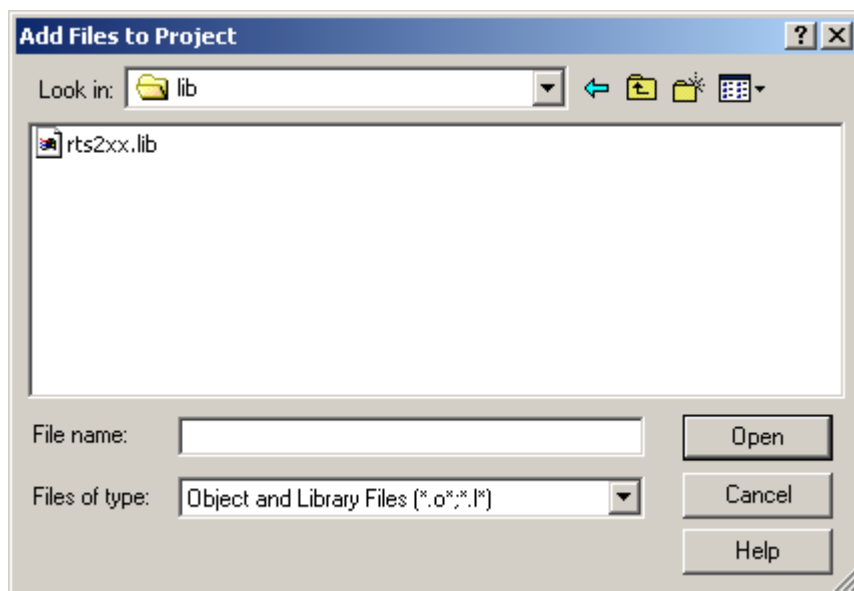
- a. In the Project View, expand the Libraries folder to view the runtime library file (*.lib).
- b. To remove the v4.12 C runtime library from the project, right-click on the filename and select Remove from project.



- c. To specify the new runtime library, right-click on the project name (*.pj) and select Add Files.



- d. In the Add Files to Project dialog box, browse to the directory that contains the C runtime libraries. See the above note for additional information on the location of runtime libraries.



- e. Select the desired runtime library.
- f. Click Open.

2.2.3 Include Search Path

The implementation-defined search path that is used to locate header files is changed in v2. If your source code uses the syntax

```
#include <file.h>
```

to include a header file that is not located in an installed directory, add the location of the header file to the search path by using the Include Search Path option.

1. Select Project→Build Options.
2. In the Compiler dialog, select the Preprocessor category.
3. Edit the Include Search Path field.
4. Click OK.

2.2.4 Linker Command File

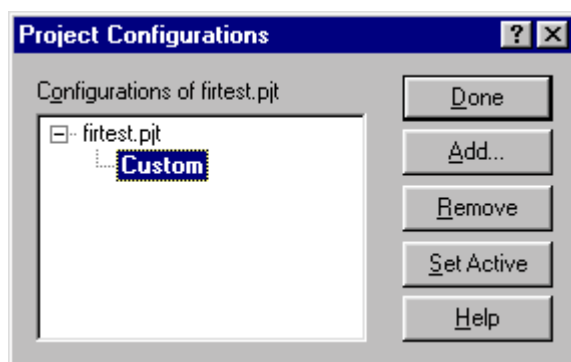
The linker command file (*.cmd) may contain references to object files and/or object library files. If those references exist, they must specify the correct paths.

NOTE: The recommended method is to add files to the project (in the Project View, right-click on the project file (*.pjf) and choose Add Files), rather than place references in the linker command file.

2.2.5 Project Configuration

Code Composer Studio IDE v2 supports multiple configurations for individual projects. Each project configuration defines a set of build options. Creating a new project in v2 creates two default configurations: *Debug* (includes all debug symbols) and *Release* (optimized with no debug symbols). Separate Debug and Release directories are created as subdirectories of the directory containing the project file. By default, object files and executables generated during the build process are located in these directories.

Converting a Code Composer v4.12 project to a Code Composer Studio IDE v2 project creates a single default project configuration named *Custom*. The location of the *Custom* configuration is defined to be the same directory that contains the converted project file. By default, object files and executables are generated in this directory. A single project configuration helps v2 support your existing directory structure and build options. To view all the Project Configurations, select Project->Configurations.



Additional project configurations can be added after conversion. Any added configurations will operate the same as those created with a new project. For more information on project configurations, see the Code Composer Studio online help: Help->Contents->Using CCS IDE->Project Environment->Working with Projects->Selecting a Project Configuration.