

Placing Code Composer Studio Under Version Control

Bhavesh Mistry
Code Composer Studio, Applications Engineering

ABSTRACT

This application report describes the procedure for placing Code Composer Studio™ Integrated Development Environment (IDE) v2.0 under version control. With Code Composer Studio checked into a version control system, developers can maintain a high degree of control over tools available for builds. The developer can now produce bit-exact versions of production code. This application report does not cover the use of version control from within the Code Composer Studio environment.

Contents

1	Introduction	2
2	Setting Up Code Composer Studio v2.0 and Version Control	2
	2.1 Creating a View	2
	2.2 Setting Up Component Manager File	3
3	Exporting Registry Files	4
	3.1 Customizing Registry Files	4
4	Checking Code Composer Studio v2.0 Into ClearCase	5
5	Using Version-Controlled Code Composer Studio	7
	5.1 Running Code Composer Studio	8
6	Registration of Plug-In Tools	9
	6.1 Plug-In Tools	9
7	Further Issues	10
	7.1 Creating a Dynamic View	10
	7.2 Issues With Plug-In Tool Registration	10
	7.3 ClearCase Errors and Issues	11
	7.4 Visual Linker Projects	11
8	Using the Update Advisor	11
	8.1 Downloading Updates	12
	8.2 Update Advisor and Version Control	12
	8.3 Disabling Update Advisor	13
	8.4 Update Advisor Errors	13
Appendix A Component Manager Information		14
Appendix B Component Manager File		15

List of Tables

Table 1. .oxc Files to be Registered if Corresponding Plug-In is Required	9
Table 2. Update Advisor Issues and Procedures for Solving Them	13

Code Composer Studio is a trademark of Texas Instruments.
Trademarks are the property of their respective owners.

1 Introduction

Code Composer Studio version 2.0 provides an upgradeable environment which, through the Update Advisor (UA), downloads new patches or adds additional tools to Code Composer Studio. This may enhance the compiler and other build tools, which are referred to as Codegen (code generation) tools.

When performing a build of production code, these enhancements to the build services can alter the way in which that build will be performed, which will in turn alter the executable. Although the functionality of the executable will be the same, its composition will have changed. It will not be a bit-exact version compared to the same project built on a non-updated version of Code Composer Studio.

For those that require builds of their software that are bit-exact reproductions on every machine, Code Composer Studio can be placed under version control.

Version control is a large data repository for production code. One such system is Rational ClearCase, which will be used as the example of a version control system in this application report. Other systems such as SourceSafe may require different steps. By placing Code Composer Studio under a version control system, any developer has a specific version of Code Composer Studio available to them. They can use the version-controlled Code Composer Studio to perform builds. This will allow one to create bit-exact reproductions of code on any machine.

Code Composer Studio itself must be trimmed down in size before it is placed under version control, so that only the essential tools are checked in. A version control server may have limited resources, depending on the type and size of application in development. For efficiency reasons, the information checked in will be minimized.

This application report will alter the system registry of the computer as well as re-register a number of tools and components before a build can take place.

2 Setting Up Code Composer Studio v2.0 and Version Control

Prior to having access to all of the Codegen tools on your version control system, a number of steps must be taken in preparing Code Composer Studio for version control. This application report uses Rational ClearCase as the version control system.

2.1 Creating a View

NOTE: If you will be using a snapshot view, you will be following steps 4 – 10. If you will be creating a generals view, go to section 7.1 of this application report.

1. Create a VOB in ClearCase (this may have to be done by a ClearCase administrator).
2. Right-click on the drive that ClearCase is mounted upon. In the case of this application report, it is assumed to be the V drive. Select ClearCase and then Create View.

[v:\ → ClearCase → Create View0]

3. In the dialog box which appears, select the No radio button and then Next.

NOTE: If the view you will be working on is one of the projects described in the ClearCase project view, choose Yes in the dialog box, and then select Next.

4. A dialog box may appear, indicating that certain machines on your network are not responding. Select OK.
5. Choose the type of view you will be accessing. In this case, it will be a snapshot.
6. Browse to where you wish your view to be located. In this case, the view is placed in:
`c:\snapshot_views\ccstudio_tools`
 An option will appear to create the directory if it does not already exist. Select OK to create the directory.
7. Select the Advanced Options button. Since the view is accessing a Unix-based VOB, make sure that the “use interop (insert_cr) text mode” button is selected. Select OK.
8. Select Finish.
9. A confirmation dialog box will appear. Select OK.
10. Select the VOB that you will be using in the left pane of the dialog box. Select ADD, and the name of the VOB will appear in the right pane of the same dialog box. Select OK.
11. Locate your view in Windows Explorer.
12. Expand the view until you reach the final branch in its structure.
13. In the final branch of the view, create three new folders named CCStudio, Registry, and Component File.

The view has now been created, and will allow you to check in and check out files from your VOB.

2.2 Setting Up Component Manager File

The Component Manager file contains information regarding the plug-ins that are available with each installation of Code Composer Studio on a particular machine. Code Composer Studio and Windows accesses this file for information on which plug-ins have been registered for Code Composer Studio.

The plug-in information will be cleared from the file so that Code Composer Studio can be customized to access only necessary plug-ins.

1. Go to `c:\program files\common files\texas instruments` in Windows Explorer.
2. Select the file `ccs_Compdb.ini`.
3. Copy the file and place it on the Windows desktop. Open up the file and erase all information after (and including) the header `[PlugIn_1]`.
4. Also in the section entitled `[CcsInstallation_1]`, and go to the subheading, `PlugInLst_GrpName =`.
5. Delete all of the plug-in information after the equals sign for that subheading.
6. If you wish to disable Update Advisor, under the subheading of “AutoLaunch =”, set to `FALSE`.

NOTE: A copy of how the Component Manager File should look has been included in Appendix A.

3 Exporting Registry Files

To properly use Code Composer Studio under version control, you are required to extract the registry information from the original machine that Code Composer Studio is installed upon. The registry contains all of the information regarding the computer's configuration. This information is continually referenced by the computer for information regarding what programs are installed on the computer, what hardware is available, and the property settings for objects in the Windows environment.

NOTE: A Windows 98 registry can be merged into the registry of a Windows NT system and vice-versa. However, Windows 2000 and Windows XP are not backwards-compatible; hence, the registries for these operating systems cannot be merged into that of NT or 98.

It is recommended that the registries being checked into ClearCase be from the same operating system as that running the destination computer.

1. Install Code Composer Studio v2.0 on the hard-drive of your computer. Take note of the directory structure under which it has been installed. In this case, consider the directory of `c:\ti`.
2. In the Windows environment, click Start, and then click Run. Type Regedit to open up the Registry Editor software provided with Windows.
3. The Regedit window displays the entire registry of the computer. The registry contains all information with regard to how and where applications are run through Windows. Select the `hkey_local_machine` branch of the registry tree. Expand the tree, and then select the "software" branch:

```
[hkey_local_machine → à software]
```

4. Select the GODSP key.
5. After selecting this, the branch exports the registry information into a .reg file named GODSP, which is the same name as the original registry. This is done by going to the registry menu item.

```
[Registry → Export Registry File]
```

6. Repeat the same process for the "Texas Instruments" registry, located under the same branch.

```
[hkey_local_machine → Software → Texas Instruments]
```

All of the prevalent registry information for running the entire Code Composer Studio software package is now contained within the two registry files, `GODSP.reg` and `Texas Instruments.reg`. They will be merged into the system registry of other computers that will run the version-controlled Code Composer Studio v2.0.

3.1 Customizing Registry Files

The registry files that are now in use can be customized. Many users may feel that they do not wish to install Code Composer Studio under the same directory structure as it was originally taken from. For this functionality, you will be required to make changes to the actual registry file itself.

To make the changes to the registry files as easy as possible, you can edit the file in a text-based editor, such as notepad.

Flags will be entered at the point of the occurrence of any location-specific information, that can then be changed at a later time.

1. Right-click on the GODSP.reg registry file. Select Edit.
2. Select the Edit menu item.
3. Select Replace...
4. In the Find What: combo edit box, type:
`c:\ccs\rel_ver2`
5. In the Replace With: combo edit box, type:
`COMPANY`
 This will act as your flag to replace at a later time.
6. Repeat steps 3 through 5 for the following paths, and replace them with the corresponding flags.

Path	Flag
<code>c:\ccs\rel_ver2</code>	COMPANY2
<code>c:/ccs/rel_ver2</code>	COMPANY3

4 Checking Code Composer Studio v2.0 Into ClearCase

With the registry files and Component Manager file properly set up, Code Composer Studio can now be placed under version control. For the purpose of size efficiency, a number of steps must be taken prior to checking files into the ClearCase repository.

1. Open up ClearCase Explorer through the following procedure:
`[Start → Programs → Rational ClearCase → ClearCase Explorer]`
2. In the left pane of the Explorer window, click the Views tag. Right-click in the pane, and select Add View Shortcut. Under the Snapshot Location field, browse to the location of the view created in section 2.1.
3. A view shortcut will now appear in the left pane. Select that view, and browse to the CCStudio folder.
4. Copy the entire CCStudio directory from your original directory, and place it under the CCStudio folder in your VOB.
5. Certain folders and tools in Code Composer Studio can be omitted from version control for the purpose of size and efficiency. Selected parts of the application, such as the example files and tutorials, do not need to be included for an application to be built properly.

6. The following directories can now be deleted from the ClearCase view.

c:\ti\tutorial	c:\ti\my projects
c:\ti\examples	c:\ti\uninstall
c:\ti\docs	c:\ti\bin\src
The docs directory contains 75 MB of data. It is recommended that it be deleted from the folder for efficiency reasons. If the help files are necessary, this folder can be left in.	Gmake information allows builds of Code Composer Studio applications in an external environment through a command line. If all builds will be within the Code Composer Studio environment, this folder can be deleted.

Generally, any information gel files, driver information, and documents that have to do with hardware that you will not be using can also be left out of the build. For example, if you will be using c64xx, all gel files other than the following can be deleted.

```
ini64xx_csim.gel      init64xx_isim.gel      init64xxsim.gel
```

There are a number of situations such as this throughout Code Composer Studio. By deleting all files that will not be necessary for your specific build, Code Composer Studio can be trimmed down to less than 20% of its original size.

7. To add each of these files to version control, right-click on a folder and choose to “add to source control”.
8. A dialog box will appear, allowing you to place a note regarding the files. Enter any information you deem appropriate. Uncheck the “keep files checked-out” box in the dialog box, and select “apply to all”.

NOTE: If during this phase any additional dialog boxes or errors occur, see section 7.3.

9. ClearCase will not automatically place the contents of a folder under source control. For this reason, you will have to open up each folder, and then place the contents into version control. Repeat steps 6 through 8 for all of the files and folders under source control.
10. In the Registry folder, place the registry files Texas Instruments.reg and GODSP.reg, which were created earlier. Add them to source control.
11. In the “Component file” folder, place the “ccs_Compdb.ini” file in, which was edited earlier. Add it to source control.

NOTE: If any icons appear next to a file after it has been checked in, refer to section 7.3.

5 Using Version-Controlled Code Composer Studio

All necessary Codegen, debug, and build tools are now available to any user with a ClearCase connection. A number of steps must now be taken to use Code Composer Studio on another computer.

1. In Windows Explorer, select the drive upon which ClearCase is mounted. Right-click and follow the steps outlined in section 2.1 to set up a view. The location and name of the view will be:

```
c:\snapshot_views\ccstudio_tools
```

2. In step 9, choose the same VOB as in section 2.1. This will provide access to all of the same information that has been checked in during the previous stages.
3. Copy all checked-in information from Code Composer Studio, and place it on your computer under the same directory structure as the original computer. Use Windows Explorer to navigate to, or to create, the right directory e.g., the original computer had the directory structure of c:\ccs\rel_ver2.
4. Copy the registry files, Texas Instruments.reg and GODSP.reg to the desktop of the computer. Double-click on a file. The option to merge the file into the registry will appear. Select OK. Repeat for the other file. The registry information on how to run Code Composer Studio has now been entered into the host computer.
5. Once the information has been placed, open up the MS-DOS command window and go to the following directory:

```
c:\ccs\rel_ver2\c6000\cgtools\bin
```

6. Type the following to register additional tools such as the compiler, and for the ability to add source files.

```
regsvr32 bti6000.dll /s
```

7. In Windows Explorer, navigate to:

```
c:\ccs\rel_ver2\plugins\bios
```

8. Double-click on gconf.exe file. This will open up a DSP/BIOS window as well as register the tool in the system registry. Close the DSP/BIOS window.
9. Select all the files in c:\ccs\rel_ver2. Right-click on the files and select Properties. Uncheck the Read Only setting, so that all files are now write-able.

NOTE: In Windows 2000, by selecting the main root directory, you can change the properties of all the files in all the subdirectories. In Window NT, the property settings of a root folder do not affect all of the subdirectories. You will have to go through each folder, changing the properties of each of those contained files.

5.1 Running Code Composer Studio

The Component Manager file contains important information regarding the plug-in properties of your Code Composer Studio install. Code Composer Studio rewrites the Component Manager file every time that it is opened, to make sure that the registered plug-in tools are documented and up-to-date. Therefore, to ensure that the file remains in the format needed, the following procedure must be undertaken.

Once the steps are taken to keep the Component Manager file intact, Code Composer Studio is ready to be run and build code.

1. Run Code Composer Studio by double-clicking the `cc_app.exe` file in the following folder:

```
c:\ccs\rel_ver2\cc\bin
```

NOTE: A dialog box may appear that indicates a new component has been detected. If so, select OK.

A dialog box may also appear indicating that hardware will need to be selected for Code Composer Studio. Select OK and choose your hardware. Then, open up Code Composer Studio.

2. Close Code Composer Studio, and open up the MS-DOS command prompt. Go to the directory:

```
c:\ccs\rel_ver2\plugins\vislink
```

3. You will now be registering the linker configuration tool (for more information regarding why this tool must be registered, see section 6). Type:

```
regsvr32 linkerconfig.ocx /s
```

4. Close the MS DOS command window.
5. In your view located in `c:\snapshot_views` and named `ccstudio_tools`, locate the “component file” folder. Open up the `ccs_Compdb.ini` file in the folder. Copy the information between the headings `[CcsInstallation_1]` and `[PlugIn_1]`. This is the same information as displayed in Appendix A.
6. Code Composer Studio has automatically generated a Component Manager file also named `ccs_Compdb.ini` in the `c:\program files\common files\texas instruments` folder. Right-click the file, and select “properties”. Uncheck the Read Only property. Now, open up the file.
7. You will see that under the heading `[PlugIn_1]` there is information regarding the linker configuration tool that you have just registered. Paste the information copied from step 13 into this file so that it appears between the last semi-colon of the comments and the subheading `[PlugIn_1]`. It should now appear like the file in Appendix B. Close and save the file.
8. Right click on the file again and select “properties”. Set the file as Read Only.
9. Code Composer Studio can now be run. Go to `c:\ccs\rel_ver2\cc\bin` and select `cc_app.exe`.

6 Registration of Plug-In Tools

Code Composer Studio v2.0 is an open-architecture software system. This allows developers to easily integrate tools built and designed outside of Code Composer Studio into the development environment. These tools are named plug-ins.

Many of the tools that are used in Code Composer Studio, such as the symbol browser and watch window, are actually plug-ins, which have been integrated into the environment. These tools are not automatically registered in the Windows registry when Code Composer Studio is copied from ClearCase to the host machine.

To register all of the plug-in tools, open up an MS-DOS command window and go to:

```
c:\ti\plugins
```

From here, you will have to go through each individual directory and register the .ocx file corresponding to the plug-in needed.

To register the Symbol Browser, open up the generic folder:

```
c:\ti\plugins\generic
```

Then type:

```
regsvr32 Symbolbrowser.ocx /s
```

The next time Code Composer Studio is opened, a dialog box will appear, indicating that new components have been added to the Code Composer Studio environment.

6.1 Plug-In Tools

Table 1 shows the location of all the .ocx files that will need to be registered if the corresponding plug-in is required.

Table 1. .ocx Files to be Registered if Corresponding Plug-In is Required

Tool	Name and Location of File
Profiler	C:\ti\plugins\general\profiler.ocx
Symbol Browser	C:\ti\plugins\general\symbolbrowser.ocx
Watch Window	C:\ti\plugins\general\watch.ocx
Command Window	C:\ti\plugins\general\ticmdwindow.ocx
RTDX Channel Viewer	C:\ti\plugins\rtdx\rtdxchannelviewer.ocx
RTDX Config Tool	C:\ti\plugins\rtdx\rtdxconfig.ocx
RTDX Diagnostics Tool	C:\ti\plugins\rtdx\rtdxdiagnostics.ocx
C6000 Project Settings	C:\ti\plugins\c6x_project_options\c6xprojectsetting.ocx
Emulator Analysis	C:\ti\plugins\c6xemu_analysis\c6xemu_analysis.ocx
Simulator Analysis	C:\ti\plugins\c6x_sim_analysis\c6xsim_analysis.ocx
Profile-Based Compilation	C:\ti\plugins\pbc\pbc.ocx

7 Further Issues

The following sections highlight possible issues or further information that may be needed in relation to topics in this application report.

7.1 Creating a Dynamic View

This section is for those that are creating a dynamic view and were directed to this section from section 2.1. To create a dynamic view in ClearCase, begin by creating a VOB in ClearCase and mount it. You will then have to create a view to that VOB. To create a VOB, you will need to talk to your network/ClearCase administrator.

1. Mounting a VOB gives a user access to that section of the ClearCase drive. To mount the VOB in Windows Explorer, right-click on the V drive (if V is the mounted ClearCase drive), and choose the options of:

[v:\ → ClearCase → Mount VOB...]

2. You will now need to select the VOB that has been created for the version control of Code Composer Studio, and choose to mount it.
3. Right-click on the V drive (if V is the mounted ClearCase drive) again, and choose the following options:

[v:\ → ClearCase → Create View...]

4. Choose “no” in this dialog box.

NOTE: If the view you will be working on is one of the projects described in the ClearCase project view, choose “yes” in the next dialog box.

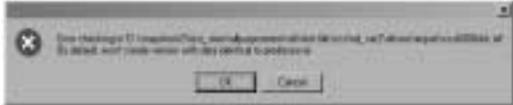
5. Choose the type of view you will be accessing. In this case, it will be dynamic.
6. Browse to where you would like the actual files to be copied to.
7. Since the view is accessing a Unix-based VOB, make sure that “use interop (insert_cr) text mode” is checked in the Advanced Options dialog box.

7.2 Issues With Plug-In Tool Registration

If, when Code Composer Studio is started, toolbars for plug-in tools do appear, yet the tool cannot be accessed, re-register the tool according to the procedure in section 6.

7.3 ClearCase Errors and Issues

If, in ClearCase, any of the following dialog boxes or errors occur, follow the listed procedure.

Error	Reason	Procedure
	ClearCase indicates that there is an identical version of the file currently checked out.	Click OK in the dialog box. In the ClearCase Explorer window, browse to the file. Right-click on the file, and select to "check in". In the dialog box, select the option "Check In Even If Identical to Previous Version".
	This will appear next to the file. The file is hijacked.	Right-click on the file, and select "Undo Hijack".
	This will appear next to the filename. The file is checked out.	Right-click on the file, and select "Check In".

7.4 Visual Linker Projects

The Visual Linker creates a number of different files with implementation. Each of these files is necessary for a Code Composer Studio project to compile and run without incident. These files will need to be checked into version control, along with any other project that is being developed.

All of the following types of files generated by the Visual Linker must be checked into version control.

.mem .rcp

.str .rcq

.mem .arg

.str .lkv

.mem .prj

8 Using the Update Advisor

The Update Advisor (UA) is a tool that is utilized by Code Composer Studio to keep the development environment completely up-to-date. It will download additional patches and plug-ins, to allow this constant level of functionality. The following sections outlines the use of Update Advisor, and the necessary steps required for UA utilization, when Code Composer Studio is checked into a version control.

8.1 Downloading Updates

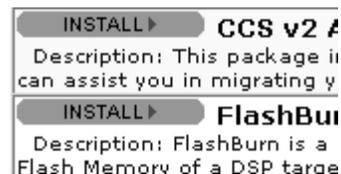
To gain access to the Code Composer Studio version 2.0 UA service, you will be required to take the following steps:

1. Access the Update Advisor by going to:
Help → CCS on the Web → Update Advisor
2. You will now be required to acquire a Username and Password at the TI&ME website. Follow the steps outlined on the web site.
3. You may need to now register yourself with the website. A red box may appear next to the patches, as seen in the image below.



If this is the case, select the register and follow the steps. **You must complete the survey directed to by the website for Update Advisor to work.** Fill in all required information. If there is an Install button, go to step 4.

4. Once registered, you can now download the appropriate patch to your computer by selecting the Install button, as shown below.



5. Once the patch has been downloaded, run the patch, and the appropriate updates will be made to your Code Composer Studio.

8.2 Update Advisor and Version Control

Update Advisor can make a number of changes to Code Composer Studio. If an update patch is applied to your version of Code Composer Studio, a few major issues should be considered. A number of patches will affect the compiler tools of Code Composer Studio. Although this does increase the functionality of the software package, it also changes the way in which the executable is built. For this reason, if you do choose to make an update standard for all of your build tools, this update must also be checked into version control.

It is recommended that the update patch be downloaded to one computer and be tested for its functionality. If the additional tools are advantageous to your development stream, it is recommended that it be checked into version control.

To place the patches under version control, you must delete the old version of Code Composer Studio currently checked in during the steps of section 4. Now, repeat all of the steps required to check in the new upgraded Code Composer Studio.

To delete the old version, you must contact your version control administrator so that all information is properly discarded.

Also, as of Code Composer Studio 2.1, the Update Advisor will provide a list of files that have been added or updated, as of the latest patch. In this situation, it is possible to find all of the new files and check them into ClearCase individually, rather than placing all of Code Composer Studio under versions control.

8.3 Disabling Update Advisor

For those developers that choose to manually disable the Update Advisor so that no patches at all are received, the following procedure should be followed.

1. To disable the Update Advisor application, go to:

Help Menu → CCS on Web → Update Setting

Un-check the “Enabled timed check for update on startup” box. This will stop automatic updates.

8.4 Update Advisor Errors

A few issues have arisen from the use of Update Advisor. They are listed in Table 2. Follow the listed procedure to solve them.

Table 2. Update Advisor Issues and Procedures for Solving Them

Error Number	Error	Reason	Procedure
1	The Update Advisor still has a red register box besides the plug-in after already being registered.	Although you followed the register procedure, you did not complete the TI survey that follows.	Register again, this time ensuring that the survey has been completed.
2	After having error #1, the procedure to correct is followed, but now an error appears, stating that the service code has been used.	The service code was already entered the first time that the registration process occurred, indicating that the code is already in use.	For technical support, go the the TI web site at http://www.ti.com and see the Software and Tools web page for Code Composer Studio, or contact the TI product information center or your nearest TI field sales office for assistance.
3	The UA web site still shows the Register button, despite using proper procedure, etc.	Your service code or subscription to Code Composer Studio has expired.	Go to the TI web site at, http://www.ti.com , and see the Code Composer Studio web page, or contact the TI upgrade center.
4	Next to the download instructions, there is no “install shield” icon, as it states there should be in the accompanying web text.	Install shield is not installed on your computer.	On the web page, select the “click here” option. A self-extracting executable will be downloaded. Now continue.

Appendix A Component Manager Information

The text being copied from one Component Manager file to the other should appear like the following.

NOTE: The subheading of [CcsInstallation_1] and other headings may be slightly different, depending upon the number of Code Composer Studio installations on your original machine.

```
[CcsInstallation_1]
DisplayName      =Code Composer Studio 2 ('C6000)
ProgLocation     =c:\ti\cc\bin\cc_app.exe
LastAccessDate  =(I) Monday, November 12, 2001 10:53:18
Status           =17
SdkVersion       =1.2
CpuSupport       =TMS320C62XX,TMS320C67XX,TMS320C64XX
PlugInLst_GrpName=
GroupDescription =
HelpFile         =
Sku
=C6000-2.00.00-FULL,ALL-2.00-SA,C6000-2.00-SA,C5000C6000-2.00-SA
UrlBase          =www-a.ti.com
Registration     =D416057-0521D
AutoLaunch       =FALSE
LastCheckDate   =04 01,2000 00:00:00
Section          =apps/advisor
Frequency        =30
```

Appendix B Component Manager File

NOTE: The subheading of [CcsInstallation_1] and other subheading information may be slightly different, depending upon the number and location of Code Composer Studio installations on your machine.

```

;-----
;Copyright (c) 2000 Texas Instruments
;-----
;NOTE:
;IMPORTANT: DO NOT ALTER OR DELETE THIS FILE.
;This file is maintained by Code Composer Studio Component Management tool
;
;-----;
[CcsInstallation_1]
DisplayName      =Code Composer Studio 2 ('C6000)
ProgLocation     =c:\ti\cc\bin\cc_app.exe
LastAccessDate  =(I) Monday, November 12, 2001 10:53:18
Status           =17
SdkVersion       =1.2
CpuSupport       =TMS320C62XX,TMS320C67XX,TMS320C64XX
PlugInLst_GrpName=PlugIn_1,PlugIn_2
GroupDescription =
HelpFile         =
Sku              =C6000-2.00.00-FULL,ALL-2.00-SA,C6000-2.00-SA,C5000C6000-2.00-SA
UrlBase          =www-a.ti.com
AutoLaunch       =FALSE
Registration     =D416057-0521D
LastCheckDate   =04 01,2000 00:00:00
Section          =apps/advisor
Frequency        =30
[PlugIn_1]
DisplayName      =Linker Configuration {8F3AC11A-D3DA-11d3-97FB-006097F0023C}
ProgLocation     =c:\ccs\rel_ver2\plugins\vislink\LINKER~1.OCX
LastAccessDate  =(I) Monday, November 26, 2001 16:18:32
Status           =19
SdkVersion       =1.2
CpuSupport       =*
PlugInLst_GrpName=Linker Configuration
GroupDescription =Linker Configuration
HelpFile         =LinkerConfig.hlp
Sku              =TI-LINKCFG-2.00
Family_InstDate =Linker Configuration {8F3AC11A-D3DA-11d3-97FB-006097F0023C}
    
```

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265