

# **What's New in Code Composer Studio Development Tools v2.2**

Steve White

Code Composer Studio, Applications Engineering

## **ABSTRACT**

This application report highlights new features and functionality in Code Composer Studio™ v2.2. In addition to providing enhancements and bug fixes from Code Composer Studio v2.1, the latest version includes support for a number of new devices.

### **Requirements:**

- Windows® 98 SP1, Windows 98 SE, Windows NT® SP6, Windows 2000 SP1 and SP2, Windows XP Home and Pro versions.
- To upgrade to v2.2, you must have Code Composer Studio 2.1 already installed on your system.
- Users with a valid subscription service can download and install Code Composer Studio v2.2 at no cost. To purchase a new seat of Code Composer Studio v2.2 or to upgrade your subscription service, please contact Lee Marketing or the Texas Instruments Product Information Center at <http://www-k.ext.ti.com/sc/technical-support/product-information-centers.htm>
- An upgrade requires 60MB to 100 MB of free disk space. A full installation of v2.2 requires about 600MB of free disk space.

<b>1</b>	<b>Introduction</b> .....	<b>1</b>
<b>2</b>	<b>New Device Support</b> .....	<b>1</b>
	2.1 C2000 Family .....	1
	2.2 C5000 Family .....	2
	2.3 C6000 Family .....	2
	2.4 OMAP Devices .....	3
	2.5 ARM Devices .....	3
<b>3</b>	<b>Operating Systems Supported</b> .....	<b>3</b>
<b>4</b>	<b>Code Composer Studio IDE Enhancements</b> .....	<b>3</b>
<b>5</b>	<b>Subprojects</b> .....	<b>4</b>
<b>6</b>	<b>Simulator Analysis</b> .....	<b>5</b>
	6.1 Simulator Pipeline Analysis .....	5
	6.2 Enhanced Simulator Analysis Plugin for C6000 and C5500 Simulators .....	5
<b>7</b>	<b>Emulation</b> .....	<b>6</b>
<b>8</b>	<b>Code Generation Tools</b> .....	<b>6</b>
<b>9</b>	<b>Target Content</b> .....	<b>6</b>
<b>10</b>	<b>DSP/BIOS</b> .....	<b>6</b>
<b>11</b>	<b>Analysis Toolkit</b> .....	<b>7</b>

Trademarks are the property of their respective owners.

## 1 Introduction

Code Composer Studio v2.2 includes changes to the number of supported host operating systems. It also includes DSP/BIOS updates, IDE enhancements, new releases of Code Generation Tools for various ISA's, support for ARM processors and new simulators for C5000 and C6000 platforms. This release now has new chip support and chip support libraries. C2000 now supports functionality previously available only to the C5000, C6000, OMAP and ARM platforms.

## 2 New Device Support

Code Composer Studio v2.2 provides support for various new and enhanced DSP devices, both on the simulator and emulator. Code Composer Studio simulators are now much faster than previous versions. This particular release adds support for the C2000 family of DSPs.

### 2.1 C2000 Family

Because of popular demand, Code Composer Studio v2.2 supports these C2000 devices:

C242, F240, F241, F243, F2810, F2812, LC2402A, LC2404A, LC2406A, LF2401A, LF2402A, LF2403A, LF2406A, LF2407A and LC2401A.

Code Composer Studio v2.2 also supports these simulators:

*CPU Only Simulator:* F27xx Simulator, F28xx Simulator

*Device Simulators:* F2810, F2812

To reduce the learning curve for C2000 users, Code Composer Studio 2.2 offers build support, configuration support, debug support via emulator, and stop mode emulation as well as examples and tutorials.

### 2.2 C5000 Family

Code Composer Studio v2.2 supports these devices from the C5000 ISA family:

- Legacy: C541, C542, C543, C545, C546, C547, C548, C549, C5401, C5402, C5403, C5404, C5406, C5407, C5409, C5410, C5420, C5421, C5440, C5441, C5444, C5470, C5471, C5510 (rev. 1)
- New: C5416, C5502, C5509, C5510 (revs. up to 2.2), C5560, C5561
- C5416/C5510 DSK support (post v2.2) – Board Support Package available via Update Advisor to support v2.2 installations.

Code Composer Studio v2.2 now includes a C55x imaging library. Previously, this library was available only on the TI website.

Debug time for the simulator is significantly faster (anywhere from 2 to 21 times faster) than in previous versions. Performance of core simulators (C55x Functional and C55x Cycle Accurate) is now faster than previous simulator versions.

Code Composer Studio v2.2 has added support for the C5502 Functional Simulator and C5502 Device Simulator. The C5502 Functional Simulator supports Cache Modeling, Functional DMA, Four Functional Timers, Pin/Port Connect, Simulation Analysis, RTDX and DSP/BIOS. Code Composer Studio v2.2 also includes device simulator support for the C5403, C5407, C541, C548 and C5410 DSP's .

## 2.3 C6000 Family

All existing C6000 simulators (device *and* functional), show faster speed. Functional simulators, for example, show increases of 2 to 3 times over that of previous simulators. New simulator support includes the following:

- C6713 and C6416. These now have functional simulation support:
- C6411/C6413/C6414/C6415/C6516/C6713. These now have device simulator support.

The functional simulators listed above offer the following functionality: L1P, L1D and L2 cache along with L2 cache mode configurability; L2 control registers; MAR Registers; 256K L2 cache (inclusive of 196K S2 RAM in L2); and 512K R2 ROM starting at address 0x3000000. The functional simulators support: four pairs of ROM patch registers; Functional QDMA; Functional 16 Channel EDMA; two 32 bit timers; Pin Connect for external interrupt pins; Port Connect for I/O and data memory; Simulator Analysis for L1P, L1D, L2Cache and XDMA events; DSP/BIOS and RTDX. Global Breakpoints are also supported via cTools on C64xx targets.

In addition, the following C6000 devices will be supported for this release:

- Legacy: C6201, C6202, C6211, C6211B, C6414 rev.1, C6415 rev.1, C6416 rev.1, C6701, C6711C, C6712C, C6202B, C6414 rev.1.1, C6416 rev.1.1, C6411 rev.1.1
- New: C6202B, C6203B, C6204, C6205, C6712C, C6415 rev. 1.1

## 2.4 OMAP Devices

The following OMAP devices are supported with this version of Code Composer Studio:

- OMAP 1510 ES2, OMAP 710, OMAP 310, OMAP 5910, OMAP 1509
- New OMAP simulator support includes: OMAP 3.1 Platform

## 2.5 ARM Devices

With this release, one new ARM device and a number of legacy ARM products will continue to be supported:

- New – ARM925T rev.2.5
- Legacy – ARM7, ARM9, ARM9TDMI, ARM925T rev.0, ARM925T, rev.2

## 3 Operating Systems Supported

The following operating systems will be supported with the Code Composer Studio 2.2 release:

- Windows 98 Second Edition and Windows 98 Service Pack 1
- Windows NT Service Pack 6
- Windows 2000 Service Pack 1 and Windows 2000 Service Pack 2
- Windows XP Pro and Windows XP Home

## 4 Code Composer Studio IDE Enhancements

The amount of memory that Code Composer Studio IDE v.2.2 uses when loading a large symbol file has declined by 3x. The latest version also includes bug fixes for the IDE, Watch Window and Symbol Browser.

Small changes have been made to how the Output Window behaves. First, a new item (OUTPUT WINDOW) will be added to the VIEW menu. This allows the user to make the Output Window appear if it is hidden or minimized. Second, users can save the output from this window to a file by right-clicking anywhere on the Output Window. Third, another right-click option (CLEAR) allows the user to clear contents from the Output Window.

Another option, RECENT GEL FILES, has been added to the FILE menu.

The FILE menu has replaced the two options LOAD SYMBOL and ADD SYMBOL with four new options: LOAD SYMBOLS, RELOAD SYMBOLS, UNLOAD SYMBOLS and RECENT SYMBOLS. The LOAD SYMBOL WITH OFFSETS has been moved to a submenu under LOAD SYMBOLS. A user who chooses this option will be prompted for data and code offsets. (This option will no longer be found in the OPTION >> CUSTOMIZE >>PROGRAM LOAD OPTIONS tab). Offsets for recent symbol files and currently loaded symbol files will be saved and reused upon reload of the executable. Another new menu item allows loading of symbols without removing currently loaded symbols.

The old DEBUG toolbar will be replaced by several new toolbars on the VIEW menu (accessed by choosing VIEW >> DEBUG TOOLBARS). This DEBUG TOOLBARS submenu includes the following toolbar changes:

- The ASM/SOURCE STEPPING toolbar now is shown by default and contains a STEP OUT and ASM/SOURCE button.
- Stepping buttons from previous versions of Code Composer Studio now appear in a new CONTEXT SENSITIVE DEBUG toolbar (hidden by default).
- The TARGET CONTROL toolbar contains the RUN, HALT, ANIMATE and SET PC TO CURSOR buttons (visible by default).
- The DEBUG WINDOW toolbar contains buttons for the WATCH WINDOW, DISASSEMBLY WINDOW, VIEW MEMORY and VIEW STACK.

Another new feature is "SETTING BREAKPOINTS ON LABELS." In the DEBUG >> BREAKPOINTS dialogue box, the developer can create a "label" (such as "main") for a breakpoint by entering the label name in the LOCATION field. The breakpoint will be saved as a label rather than as a specific memory address. Even if the program is reloaded or if a different program is loaded, the program will "remember" the breakpoint set on that label. Breakpoints set on labels are saved in the workspace in which they were created. Labels are permitted for all types of breakpoints and also Probe Points.

Another option added to the PROGRAM LOAD options tab of the CUSTOMIZE dialog is the "CLEAR ALL BREAKPOINTS WHEN LOADING A NEW PROGRAM" feature. When enabled, all breakpoints will be removed when a different file is loaded. Note, however, that if no file is loaded, the next file loaded will show all previous breakpoints regardless of whether the "CLEAR ALL BREAKPOINTS" option was checked.

There have also been a number of performance enhancements with this version of Code Composer Studio. Benchmarks (compared to Code Composer Studio 2.1) on a 10,000 file project show decreases of:

- 10x in time to open a project
- 20x in time to close a project
- 20x improvement on doing a full dependency scan in the IDE

The Project Manager also supports multiple linker command files, which will be passed to the linker. The link order is user configurable and allows up to 1000 linker command files.

Code Composer Studio v2.2 now reports device failures in less than one second to eliminate the impression that the environment has hung while Code Composer Studio tries to recover. Error reporting and error descriptions in the Code Composer Studio drivers have improved.

## 5 Subprojects

Subprojects offers the user an easy way to manage and build a more complex Code Composer Studio project consisting of multiple small projects with dependencies between them. This allows the user to reuse project modules in a large scale project and specify which projects must be built before others.

To configure subprojects and project dependencies, choose **PROJECTS >> PROJECT DEPENDENCIES** or right-click an open project and choose **PROJECT DEPENDENCIES**.

Code Composer Studio v.2.2 allows the user to specify *project dependencies*, (i.e., which projects are subprojects of a containing project). These project dependencies are stored in the containing project. To work on the same projects in the Code Composer Studio IDE later on, the user can simply load the containing project. To build and debug a containing project and its subprojects, the user needs only to build and debug the containing project. Some more key points:

- The user can specify dependencies between projects that exist in a project collection.
- Dependencies are displayed graphically in the project view in the same way that multiple projects are displayed.
- The user can view dependencies in text form in the Dependencies tab of the Output Window.
- The user now can build a containing project without building its subprojects. Right click the project in the Project View Window and select the “**BUILD (SELECTION)**” menu item.
- Subprojects are recorded in the project file (.pjf).
- Building a containing project causes its subprojects to be built first.
- Exporting a containing project will cause its subprojects to be exported.
- Code Composer Studio v2.2 provides the ability to detect a cyclic dependency and handle it accordingly. It does this by scanning and displaying the valid candidates of subprojects in the Dependency Window so the user cannot create a cyclic dependency through the window.
- APIs and GEL functions are provided for adding and removing subprojects.

## 6 Simulator Analysis

### 6.1 Simulator Pipeline Analysis

When optimizing code for DSP's, the developer uses an instruction pipeline (a sequence of operations that occur when instructions are being executed). Pipelines allow for multiple instructions to access CPU resources simultaneously. Because CPU resources are limited, conflicts can occur when one resource is accessed by more than one pipeline stage. Some of these conflicts are resolved automatically by the CPU delaying access. Other conflicts are unprotected and must be resolved by the programmer.

The pipeline stall analyzer tool was developed to give C55x and C6xx developers better data about the instruction pipeline when optimizing assembly code. It provides a visual representation of PPU stalls, instructions that are stalling and the instructions that caused the stall. The tool displays the IBQ (Instruction Buffer Queue), which shows the FETCH PC and DECODE PC instructions.

To access this function on C55x and C6x, go to TOOLS and select PIPELINE ANALYSIS.

### 6.2 Enhanced Simulator Analysis Plugin for C6000 and C5500 Simulators

The C6000 and C5500 simulators support detection of certain events in the target during execution of the program. Each of these events relates to a specific happening in the target. The list of events will vary from simulator configuration to configuration. The Simulator Analysis tool reports the occurrence of these events so you can accurately monitor and measure the performance of your program. The tool can be used to either halt the program execution when the event is triggered or to count the occurrence of this event over the execution period.

This version of Code Composer Studio makes usability improvements over the previous release. Previously, the number of shown events quickly became large and unmanageable. Because of this, the view window in the latest version organizes events in a hierarchical fashion, grouping them by type of event and assigning event names to reflect specific instances of a particular event.

To access this feature, go to TOOLS and select SIMULATOR ANALYSIS.

## 7 Emulation

Drivers for both XDS-510 and XDS-560 will be supplied in Code Composer Studio 2.2 for C2000, C5000, C6000, OMAP and ARM. Code changes have improved the XDS-560 emulator's download speed and robustness. Enhancements to the device drivers now offer more detailed error messages if the error condition is met with the target.

## 8 Code Generation Tools

This release of Code Composer Studio offers the following versions of Code Generation Tools (CGT) and these DSP Libraries and Image Libraries:

- C24xx – v.7.01 CGT
- C27xx/C28xx – v.3.03 CGT

- C54x – v.3.80 CGT
- C55x – v.2.40 CGT
- C6000 – v.4.30 CGT
- TMS470 – v.2.204 CGT

## 9 Target Content

In this release of Code Composer Studio, the help file contents have been updated to add references to DSPLIB, IMAGELIB and MATHLIB documentation.

LIB examples are now in the “examples” directory. Also, the root directory now has separate directories for the C6200, C6700 and C6400 ISA’s.

HELP >> CCS ON THE WEB contains a shortcut to the Reference Frameworks web page. From here, users can download Reference Frameworks and Reference Framework app notes.

## 10 DSP/BIOS

Texas Instruments has enhanced DSP/BIOS to provide improved semaphore and mailbox performance on C55x and C6000 BIOS. The streaming I/O interface (SIO) has also been modified to support SWI (software interrupts). In addition it is now possible to select just SIO\_issue and SIO\_reclaim if desired, further reducing the required footprint. This is supported in the following ISAs: C54xx, C55xx, C6000 and C28xx.

This version of Code Composer Studio includes cache support for C55xx. This feature provides more accurate data to the CPU Load graph plugin.

## 11 Analysis Toolkit

Registered users with a valid Code Composer Studio subscription can now obtain analysis tools for selected DSP platforms via the Update Advisor. The Analysis Toolkit was designed specifically for the new functional simulators included with Code Composer Studio v.2.2. It contains standalone utilities to help the DSP developer visualize and quickly identify problem areas in their applications.

Cache Analysis provides a window into the developer’s application for visualizing conflict, compulsory and capacity misses. The interface offers a two–dimensional view of cache activity so the programmer can “see” not only where cache conflicts occur but when they occur. This makes it easier to spot patterns in cache accesses and helps the developer to make more informed decisions about pre–fetching or code restructuring to achieve better cache performance.

The Toolkit also includes a new Code Coverage and Multi–Event Profiler to help increase integrity, reliability and robustness of application code. Code Coverage identifies source code that was not exercised during an application’s execution. This helps to uncover unnecessary code or highlight holes in a test program. Multi–event profiling provides profile data for functions over multiple events of interest during a single run of an application. This includes common CPU events (such as cycles, instructions or stalls) and memory events (such as cache hits and misses). Information from multi–event profiling assists in revealing hotspots in the code.

## 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