Link for Code Composer Studio Development Tools Release Notes

The "Link for Code Composer Studio® Development Tools 1.3.1 Release Notes" on page 1-5 describe Version 1.3.1 of the Link for Code Composer Studio™ Development Tools.

These Release Notes discuss the following topics:

If you are upgrading from a version earlier than Version 1.3, see "Link for Code Composer Studio® Development Tools 1.3 Release Notes" on page 2-1.

Printing the Release Notes

If you would like to print the Release Notes, you can link to a PDF version.

Contents

1

Link for Code Composer Studio® Development Tools 1.3.1 Release Notes

Upgrading from an Earlier Release

There are no upgrade issues involved in moving to the Link for Code Composer Studio® Development Tools 1.3.1 from Version 1.3.

This section lists the required software products and versions and the supported software operating systems.

Required Software Products and Versions

Version 1.3.1 of the product requires that you install the following software from The MathWorks and Texas Instruments. Note the versions required.

Product	Version
MATLAB	7
Signal Processing Toolbox	6.2
One or more of the following as appropriate:	
• Code Composer Studio for C2000	2.12, 2.2, 2.21
• Code Composer Studio for C5000	same
• Code Composer Studio for C6000	same
• Code Composer Studio for OMAP	same

MATLAB must be installed first.

You can install the Signal Processing Toolbox, Code Composer Studio IDE, and Link for Code Composer Studio Development Tools, Version 1.3.1 in any order.

Supported Software Operating Systems

In this release, we drop support for Microsoft Windows® 98 and we add support for Microsoft Windows® XP. Note that this differs slightly from the MATLAB supported platforms. In addition, some hardware targets that work with the Embedded Target for TI C6000 do not work on Microsoft Windows NT platforms although Link for Code Composer Studio does. Refer to your

Embedded Target for TI C6000 DSP documentation for details about supported hardware targets and operating systems.

Known Software and Documentation Problems

Link to Open Bugs Report

This section includes a link to a description of known software and documentation problems in Version 1.3.1.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.

Additional Known Problems

The following errors are not in the list of problems accessed by the preceding link.

Error Messages with open(cc) and goto(cc) Functions

The functions `open(cc, )` and `goto(cc, )` sometimes return an error message

```
'Unspecified error'
```

This is an issue involving the API we use to communicate with Code Composer Studio. We are investigating the problem. Rerunning the function usually works.

Running the Demos on Targets

Link for CCS demos assume a specific memory mapping in the linker command file. Thus, the demo project memory map should match that of your target. If the mappings do not match, change the linker command file of your demo project to match your target's memory mapping and rebuild the program file.

Working with Multiple Installed Versions of Code Composer Studio

When you have more than one version of Code Composer Studio installed on your machine, you cannot select which CCS version Link for Code Composer Studio attaches to when you create a ccsdsp object. If, for example, you have both CCS for C5000 and CCS for C6000 versions installed, you cannot choose to connect to the C6000 version rather than the C5000 version.

When you issue the command

```
cc = ccsdsp
```

Link for Code Composer Studio starts the CCS version you last used. If you last used your C5000 version, the `cc` object access the C5000 version.

Workaround. To make your `ccsdsp` object access the correct target:

1. Start and close the appropriate CCS version before you create the `ccsdsp` object in MATLAB.
2. Create the `ccsdsp` object using the `boardnum` and `procnum` properties to select your target, if needed.

   Recall that `ccsboardinfo` returns the `boardnum` and `procnum` values for the targets that CCS recognizes.

Running Demos on the TMS320C54x Simulator

Demos that you run on your TMS320C54x simulator do not work if you enable memory mapping. Before you use the TMS320C54x simulator to run demos, do the following to disable memory mapping on the target:

1. Select **Option->Memory Map...** from the **Code Composer Studio** menu bar.
2. Clear the **Enable Memory Mapping** check box on the **Memory Map** dialog.
3. Click **Done** to close the dialog and return to CCS.

Now you can run the demos.

2

Link for Code Composer Studio® Development Tools 1.3 Release Notes

New Features

This section introduces features and enhancements added in Link for Code Composer Studio® Development Tools 1.3.

Support for New Platforms

This release include support for a number of new platforms or processors..

Supported Board Designation	Board Description
C2401 DSK	Starter kit for the C2401 processor
C2407 DSK	Starter kit for the C2407 processor
C27x Simulators	Simulators for the C27x DSP family
C2812 DSK and simulators	DSP starter kit and simulators for the C28x DSP family
C6416 DSK	DSP starter kit for the C6416 processor
C6713 DSK and simulators	DSP starter kit for the C6713 processor

For a little more information, the following features apply to the C2000 processor family:

- C24x processors work in Debug mode operations
- C27x processors work in Debug mode operations
- C28x processors work in
  - Data manipulation
  - RTDX—requires the C28x RTDX patch for CCS 2.20.18, available from Texas Instruments
  - Hardware-in-the-loop operations

To learn more about the details of the support for the new platforms, refer to Appendix A in the *Link for Code Composer Studio Development Tools User's Guide*.

New Tool for Managing Data Types in HIL

If you use the HIL features in Link for Code Composer Studio, you may encounter problems using custom type definitions from your projects. A new tool, the Data Type Manager, helps you define your custom data types so MATLAB can understand and interpret them through a graphical user interface. From within the DTM, you use the various options to add your custom data type definitions to a ccsdsp object. Then your functions that use the custom types work with MATLAB. Note that the same features were available using the `add` and `remove` methods for `cc.type`.

For more information about the Data Type Manager, use the Help browser to look in the *Link for Code Composer Studio Developement Tools User's Guide*.

Support TI Code Composer Studio 2.2 and 2.21

Version 1.3 of this product requires TI Code Composer Studio IDE 2.2 or 2.21.

Syntax Change to createobj

We changed createobj to allow you create a function object and allocate memory buffers for the input arguments to the function.

You use the following syntax to allocate memory buffers for two, or more, input arguments to a function:

```
ff = createobj(cc,functionname,'function','allocate',
{'input1',15,'input2',10});
```

The reference page for `createobj` incorporates the new syntax.

load Function Loads GEL Files

`load` includes the ability to load GEL files into Code Composer Studio from MATLAB. The new load syntax, as now added to the load reference page, is

```
load(cc,gelfilename,timeout);
```

Input argument `timeout` is optional. If you omit the timeout value, the load process uses the default time-out value in `cc`.

In addition, `remove` lets you remove GEL files

```
remove(cc,gelfilename);
```

Major Bug Fixes

Link for Code Composer Studio® Development Tools 1.3 includes several bug fixes made since Version 1.2. This section describes the particularly important Version 1.3 bug fixes.

If you are viewing these Release Notes in PDF form, please refer to the HTML form of the Release Notes, using either the Help browser or the MathWorks Web site and use the link provided.