MATLAB Link for Code Composer Studio Development Tools Release Notes

The "MATLAB Link for Code Composer Studio® Development Tools 1.3 Release Notes" on page 1-1 describe the changes introduced in the latest version of the MATLAB Link for Code Composer Studio Development Tools, which is available from the MathWorks Web site. These Release Notes discuss the following topics:

- "New Features" on page 1-2
- "Major Bug Fixes" on page 1-8
- "Upgrading from an Earlier Release" on page 1-9
- "Known Software and Documentation Problems" on page 1-14

Note We have changed the Version 1.2 upgrade process. You do not need to install the patch `R13_Target_patch.exe` as specified in the previous release notes. This release notice no longer refers to the patch and the patch is not shipped with the product.

If you are upgrading from a version earlier than Version 1.2, you should also see

- "MATLAB Link for Code Composer Studio® Development Tools 1.2 Release Notes" on page 1-1
- "MATLAB Link for Code Composer Studio™ Development Tools 1.1 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

MATLAB Link for Code Composer Studio® Development Tools 1.3 Release Notes

MATLAB Link for Code Composer Studio® Development Tools 1.2 Release Notes

MATLAB Link for Code Composer Studio™ Development Tools 1.1 Release Notes

MATLAB Link for Code Composer Studio® Development Tools 1.3 Release Notes

New Features

This section introduces features and enhancements added in MATLAB 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 *MATLAB 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 MATLAB 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 *MATLAB Link for Code Composer Studio Developement Tools User's Guide*.

Support TI Code Composer Studio 2.2

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

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

MATLAB 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.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving to the MATLAB Link for Code Composer Studio® Development Tools 1.3 from Version 1.2.

If you are upgrading from a Release earlier than Release 13, you should see "Upgrading from an Earlier Release" on page 2-7 in the MATLAB Link for Code Composer Studio Development Tools 1.1 Release Notes.

Required Software Products and Versions

Version 1.3 of the product requires that you install the following software from The MathWorks and Texas Instruments. Note the versions required.

Product	Version
MATLAB	6.5.1
Signal Processing Toolbox	6.1
One or more of the following as appropriate:	
• Code Composer Studio for C2000	2.2
• Code Composer Studio for C5000	2.2
• Code Composer Studio for C6000	2.2
• OMAP™ Code Composer Studio Integrated Development Environment (IDE)	2.2

MATLAB must be installed first.

You can install the Signal Processing Toolbox, Code Composer Studio IDE, and MATLAB Link for CCS Development Tools, Version 1.3 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.

Known Software and Documentation Problems

This section includes a link to a description of known software and documentation problems in Version 1.3.

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.

If you are upgrading from a Release earlier than Release 13, then you should see "Known Software and Documentation Problems" on page 3-5 in the MATLAB Link for Code Composer Studio Development Tools 1.0 Release Notes.

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

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

MATLAB 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 MATLAB 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
```

MATLAB 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.

MATLAB Link for Code Composer Studio® Development Tools 1.2 Release Notes

New Features

This section introduces features and enhancements added in MATLAB Link for Code Composer Studio® Development Tools 1.2.

Note The new features included in this release are not documented in the online help for the MATLAB Link for Code Composer Studio Development Tools. For now, this release notice contains all of the documentation for both the new features and the extensions to the existing methods.

Support for OMAP1510 Platform

With this release you can use the debugging and data manipulation features of the product with your OMAP1510 platform. The following new capabilities have become part of the product:

- Debug support for
  - OMAP1510 platform
  - TMS470R1x, also called ARM, processor—simulator only
  - TMS470R2x, also called ARM, processor—ARM925
- Data manipulation support for
  - OMAP1510 platforms
  - TMS320C55x platform
  - TMS470R1x, also called ARM, processor—simulator only
  - TMS470R2x, also called ARM, processor—ARM925
- Methods for debugging the multiple processors on the OMAP1510 platform

Note Although the OMAP5910 platform uses the same DSP and ARM as the OMAP1510—the TMS320C55x and ARM925—we have not tested the configuration.

Support Boards with Two or More Processors

Working with multiprocessor boards required that we make a change to the ccsdsp function. We changed the function to behave differently in all syntaxes:

```
cc = ccsdsp
cc = ccsdsp('boardnum',value)
cc = ccsdsp('boardnum',value,'procnum',value)
```

In the case where you do not specify a board or processor:

```
cc = ccsdsp

Array of CCSDSP Objects:
  API version              : 1.2
  Board name               : OMAP 3.0 Platform Simulator [Texas
Instruments]
  Board number             : 0
  Processor 0  (element 1) : TMS470R2127 (MPU, Not Running)
  Processor 1  (element 2) : TMS320C5500 (DSP, Not Running)
```

Where you choose to identify your target as an input argument to ccsdsp, for example, when your target board contains two processors:

```
cc = ccsdsp('boardnum',2)

Array of CCSDSP Objects:
  API version              : 1.2
  Board name               : OMAP 3.0 Platform Simulator [Texas
Instruments]
  Board number             : 2
  Processor 0  (element 1) : TMS470R2127 (MPU, Not Running)
  Processor 1  (element 2) : TMS320C5500 (DSP, Not Running)
```

cc returns a two element object handle with cc(1) corresponding to the first processor and cc(2) corresponding to the second.

You can include both the board number and the processor number in the ccsdsp syntax, as shown here:

```
cc = ccsdsp('boardnum',2,'procnum',[0 1])

Array of CCSDSP Objects:
  API version              : 1.2
  Board name              : OMAP 3.0 Platform Simulator [Texas
Instruments]
  Board number            : 2
  Processor 0  (element 1) : TMS470R2127 (MPU, Not Running)
  Processor 1  (element 2) : TMS320C5500 (DSP, Not Running)
```

Enter procnum as either a single processor on the board (a single value in the input arguments to specify one processor) or a vector of processor numbers, as shown in the example, to select two or more processors.

Support Co-emulation

Co-emulation, defined by Texas Instruments to mean simultaneous debugging of two or more CPUs, allows you to coordinate your debugging efforts between two or more processors within one device. Efficient development with OMAP™ hardware requires co-emulation support. Version 1.2 of MATLAB Link for Code Composer Studio provides co-emulation by changing the way the ccsdsp function operates. Instead of creating one cc object when you issue the command

```
cc = ccsdsp
```

for your target that has multiple processors, the resulting cc object comprises a vector of cc objects cc(1), cc(2), and so on, each of which accesses one processor on your target device, say an OMAP1510. When your target has one processor, cc is a single object. With a multiprocessor target, the cc object returns the new vector of objects. For example, for board 2 with two processors,

```
cc = ccsdsp
```

returns the following information about the board and processors:

```
cc = ccsdsp('boardnum',2)

Array of CCSDSP Objects:
  API version             : 1.2
  Board name              : OMAP 3.0 Platform Simulator [Texas
Instruments]
  Board number            : 2
  Processor 0  (element 1) : TMS470R2127 (MPU, Not Running)
  Processor 1  (element 2) : TMS320C5500 (DSP, Not Running)
```

Checking the existing boards shows that board 2 does have two processors:

```
ccsboardinfo

Board Board                              Proc Processor                      Processor
 Num  Name                                  Num  Name                             Type
 ---  ---------------------------------  ---  ----------------------------------
 2   OMAP 3.0 Platform Simulator [T ...  0   MPU                    TMS470R2x
 2   OMAP 3.0 Platform Simulator [T ...  1   DSP                    TMS320C5500
 1    MGS3 Simulator [Texas Instruments]  0   CPU                         TMS320C5500
 0     ARM925 Simulator [Texas Instru ...  0    CPU                          TMS470R2x
```

Updated Methods for Multiprocessor Boards

To let you work with boards that contain more than one processor, we added to or changed the arguments available for some of the existing debugging methods, such as `halt` and `animate`. The following table lists the methods that now work for both single and multiprocessor ccsdsp objects.

Note Functions not listed in this table accept only scalar ccsdsp objects—objects that access only one processor on one board.

Function	Description
`animate`	Calls the `animate` function on each processor represented by the ccsdsp object.
`disp`	Returns a display in MATLAB that shows you information about the ccsdsp object, and all its members.
`halt`	Calls the `halt` function for each processor. represented in the ccsdsp object. It stops each processor accessed by the object.
`info`	Returns the information about each processor accessed by the ccsdsp object in the command.
`isrtdxcapable`	Calls the `isrtdxcapable` function on each processor represented by the ccsdsp object to report whether the processor supports RTDX. You get back a vector that contains information about each processor.

Function	Description
`isrunning`	Calls the `isrunning` function on each processor represented by the ccsdsp object to report whether the processor is presently running a program. You get back a vector that contains information about each processor.
`reload`	Calls the `reload` function for each processor represented by the ccsdsp object.
`reset`	Resets each processor represented by the ccsdsp object.
`restart`	Restarts the loaded program on each processor accessed by the ccsdsp object.
`run`	Calls `run` for each processor, running the loaded program on each one.

New Demonstration Model for Multiprocessor Boards

To show you how the new multiprocessor target support works with the new and updated methods, we added a demonstration model named `multiproctutorial`. In this model we introduce how you use the debugging features with co-emulation to debug your multiprocessor targets.

To start the new model:

1. Select **Help->Demos** from the MATLAB menu bar.
2. When the Help browser appears with **Demos** selected, expand the **Toolboxes** directory and look for **MATLAB Link for Code Composer Studio** in the directory tree.
3. Expand this directory to find **Link for Code Composer Studio Demos**.
4. Find the `Multi-Processor Tutorial` entry in this directory. Click the model name to run the program. Follow the instructions in the program to continue.

workspace Option Restored to open Function

With this release, the function `open` accepts the input option `workspace` for opening workspace files. In Version 1.1, we removed the `workspace` input keyword from the available input arguments. These commands, and others like them, work once again.

- `open(cc,filename)` where `filename` is the name of a Code Composer Studio workspace file
- `open(cc,'filename','workspace')` where `filename` is the name of a Code Composer Studio workspace file and the keyword `workspace` identifies the type of file to open
- `open(cc,'filename','workspace',timeout)` where the `timeout` input argument sets the time-out period explicitly

Major Bug Fixes

MATLAB Link for Code Composer Studio® Development Tools 1.2 includes several bug fixes made since Version 1.1. This section describes the particularly important Version 1.2 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.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving to the MATLAB Link for Code Composer Studio® Development Tools 1.2 from Version 1.1.

If you are upgrading from a Release earlier than Release 13, then you should see “Upgrading from an Earlier Release” on page 2-7 in the MATLAB Link for Code Composer Studio Development Tools 1.1 Release Notes.

Required Software Products and Versions

Version 1.2 of the product requires that you install the following software from The MathWorks and Texas Instruments. Note the versions required.

Product	Version
MATLAB	6.5
Signal Processing Toolbox	6.1
One or more of the following as appropriate:	
• Code Composer Studio for C2000	2.12.01 or 2.12.07*
• Code Composer Studio for C5000	2.12.01 or 2.12.07*
• Code Composer Studio for C6000	2.12.01
• OMAP™ Code Composer Studio Integrated Development Environment (IDE)	2.12.01 or 2.12.07*

* Code Composer Studio 2.12.07 consists of Version 2.12.01 with the C5510 DSK patch applied.

MATLAB must be installed first.

You can install the Signal Processing Toolbox, Code Composer Studio IDE, and MATLAB Link for CCS Development Tools, Version 1.2 in any order.

Using Version 1.2 with Code Composer Studio IDE 2.2

To make MATLAB Link for Code Composer Studio 1.2 compatible with Code Composer Studio 2.2, download a patch from the MathWorks technical support FTP site.

Follow these steps to download the patch `Link1p2PatchForCCS2p2.exe`.

1. Go to the following FTP site:

   ```
   ftp://ftp.mathworks.com/pub/tech-support/solutions/s35109/
   ```

2. Copy `Link1p2PatchForCCS2p2.exe` into your `$MATLAB\toolbox\ccslink` directory (where `$MATLAB` is your root MATLAB directory).
3. Double-click on the file `Link1p2PatchForCCS2p2.exe`. WinZip opens.
4. Click **Unzip**.
5. A window displaying `6 file(s) unzipped succesfully` appears. Click **OK** and close the WinZip window.
6. Restart MATLAB

Problems and Solutions for Working with CCS IDE 2.2

You may encounter the following issues when you use MATLAB Link for Code Composer Studio 1.2 (R13.0.1+) with Code Composer Studio 2.2. The next sections describe the problems and how to correct them.

Time-out Issue with Demos on OMAP Simulator

Set up OMAP Code Composer Studio IDE 2.2 for OMAP3.1 Platform Simulator and run `cstutorial/ccsfirdemo/multiproctutorial`. At some point the demo seems to stop and it generates a time-out error in MATLAB.

This is caused by a bug in the OMAP3.1 Platform Simulator on Code Composer Studio IDE 2.2. When you open and load any ARM or C55x program, it takes a long time to load and run to any point in the program.

Demo Issues on the C5416 DSK Using XDS-510

When running `rtdxtutorial`, the demo terminates and you get the error:

```
Enable: Enable failed on channel: ichan
```

Also, when you run `rtdxlmsdemo`, and see the following text displayed in MATLAB

```
Reading coefficient updates and filtered results from target -
frame 1...
```

the process continues to take a long time. It eventually fails with this or a similar message:

```
ReadMsg: Timeout waiting for CCS to confirm completion: waiting
for 1 message. 50 seconds might not be enough time for reading the
data messages. Click view script and increase the value of the
TimeOut variable at the beginning of the script.
```

To correct this problem, perform the following steps:

1. Open the CCS IDE.
2. In the left pane, click (+) on `GEL Files` and remove all loaded GEL file(s). You are allowed only one GEL file in a project.
3. Right-click `GEL Files` and choose **Load GEL...** from the context menu.
4. In the **Load Gel file** dialog, go to `<tiroot>\cc\gel` and choose `c5416_dsk.gel` as the GEL file to load. Click **Open** to load the file.
5. Rerun the demo. You do not need to rebuild the project.

ccsfirdemo Times Out on the C2812 DSK Using XDS-510

When you run `ccsfirdemo`, it times out during the run target stage.

Resolve this problem by downloading and applying the patch `Link1p2PatchForCCS2p2.exe` as described in “Using Version 1.2 with Code Composer Studio IDE 2.2” above.

RTDXTUTORIAL/RTDXLMSDEMO on the OMAP Simulator

You cannot run the programs `rtdxtutorial` and `rtdxlmsdemo` on the OMAP 3.1 Platform Simulator. Although your OMAP 3.1 hardware includes the C5500 processor, the OMAP 3.1 Simulator does not support RTDX.

OMAP 3.1 hardware runs the demos successfully; the simulator does not. To run the demos on a simulator, use the C55xx simulator as the target.

Here is what happens when you run either program on the OMAP simulator:

1 When the demo prompts you to select a target, choose the C5500 processor.

   During the program load stage, a dialog opens in the CCS IDE:

```
RTDX target application does not match emulation protocol!
Loaded program was created with an rtdx library which does not
match the target device.
```

2 Click **OK** and continue with the demo. Eventually the demo errors out and exits. You see this error in MATLAB.

```
Ccsdsp:The server threw an exception.
```

CCSTUTORIAL/CCSFIRDEMO on ARM 7 Big Endian Simulator

During the load phase of running the demo, a dialog opens in the Code Composer Studio IDE.

```
Data verification failed at address 0x20.
Please verify target memory and memory map.
```

Before you run `ccstutorial` or `ccsfirdemo`, run the Code Composer Studio IDE and do the following operations:

1 Go to the **Option** menu.

2 Select **Memory Map**.

   The **Memory Map** dialog opens.

3 Select the **Enable Memory Mapping** check box to enable memory mapping on your target.

4 In **Starting Address**, enter `0x20`.

5 In Length, enter `0x7FFE0`.

6 From the **Attributes** list, select `RAM-Read and Write`.

7 Click **Add**.

8 Click **Done** to close the the dialog

Also, running the demo `ccstutorial` may generate the message

```
Can't Run Target CPU: Error number -2164 Error address 0
```

in the **Messages** window in the Code Composer Studio IDE. This does not affect the demo.

RTDXLMSDEMO on C6211 Big Endian Simulator

When you load the `rtdxlmsdemo` on the C6211 simulator, in big endian mode, a message dialog opens in the CCS IDE.

```
File
C:\MATLAB6p5\toolbox\ccslink\ccsdemos\rtdxlms\c6x\rtdxdemo_sim6xe.out
Does not match the target type, not loaded.
```

You have to rebuild the project `rtddemo_sim6xe.pjt` in `$MATLAB\toolbox\ccslink\ccsdemos\rtdxlms\c6x` to run the demo successfully. Perform the following steps to correct the problem:

1 Open the project `rtddemo_sim6e.pjt` in the CCS IDE.

2 Go to **Project->Build Options**.

3 On the **Compiler** pane, select `C621x (-mv6210)` under **Target Version**. Click **OK**.

4 Build the project and rerun the demo.

Note Since other processors use the same demo project, remember to return the project build options to the original settings and rebuild the project.

Known Software and Documentation Problems

This section includes a link to a description of known software and documentation problems in Version 1.2.

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.

If you are upgrading from a Release earlier than Release 13, then you should see "Known Software and Documentation Problems" on page 3-5 in the MATLAB Link for Code Composer Studio Development Tools 1.0 Release Notes.

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

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

MATLAB 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.

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.

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 MATLAB 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
```

MATLAB 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.

MATLAB Link for Code Composer Studio™ Development Tools 1.1 Release Notes

New Features

This section introduces features and enhancements added in MATLAB Link for Code Composer Studio™ Development Tools 1.1 (which is available via Web download), since the release of MATLAB Link for Code Composer Studio 1.0 (the Release 13 CD).

New function Object

In addition to the existing collection of objects for working with embedded objects, Version 1.1 adds a new object for working with functions in your programs or project in CCS.

The new object, called a function object, enables you to run functions on your target from MATLAB, making your development process easier and faster. In addition, function objects let you access functions so you can work with them, such as seeing the input arguments and the output argument for function declarations. For more information about the new object, refer to function object in the MATLAB Link for Code Composer Studio Development Tools Help browser.

New type Object

Because custom type definitions, called typedefs, are a fact of life in many C programs, MATLAB Link for Code Composer Studio now offers a type object.

Use the type object to tell MATLAB about the typedefs in your project. The new type object is a part of each ccsdsp object that you create. By adding your type definitions to the type object, you can work with them in MATLAB and use them as you develop programs for your target processor. For details about type objects, refer to the online documentation.

New addressof Method for memoryobj and struct Objects

To enable you to determine the address of a structure object or a memoryobj object (or derived objects that inherit from memoryobj objects), use the `addressof` method with a handle to the object. For example, to get the address of a structure named `bfield`, here is what the code looks like:

```
cc=ccsdsp('boardnum',1)
```

```
CCSDSP Object:
  API version      : 1.2
  Processor type   : TMS320C6711
  Processor name   : CPU_1
  Running?         : No
  Board number     : 1
  Processor number : 0
  Default timeout  : 10.00 secs

  RTDX channels    : 0

mm=createobj(cc,'bfield')

STRUCTURE Object stored in memory:
  Symbol name             : bfield
  Address                 : [ 2147501052 0]
  Address units per value : 4 au
  Size                    : [ 1 ]
  Total Address Units     : 4 au
  Array ordering          : row-major
  Members                 : 'b_2', 'b_22', 'b_10'

>> addressof(mm)

ans =

  1.0e+009 *

    2.1475         0
```

There is no online documentation for this new method.

Support for C2800 Digital Signal Processors

MATLAB Link for Code Composer Studio now supports debugging on C2800 processors. You can use the existing objects to debug your projects in CCS. You cannot use new function and type objects. With support for debugging on the C2800, you can run projects on your C2800 processor from MATLAB, as well as work with projects in CCS that target your C2800.

Note that support for the C2800 processor family does not include using the new function and type objects.

Limited Support for C5500 Digital Signal Processors

MATLAB Link for Code Composer Studio now supports debugging on C5500 processors. You can use the existing objects to debug your projects in CCS. You cannot use new function and type objects. With support for debugging on the C5500, you can run projects on your C5500 processor from MATLAB, as well as work with projects in CCS that target your C5500.

Note that support for the C5500 processor family does not include using the new function and type objects.

Compatibility testing with the C5500 was limited to the C5500 simulator in Code Composer Studio.

Additional Demonstration Program Support

The MATLAB Link for Code Composer Studio includes a number of demonstration programs that use the product. The following table identifies each demo program and which hardware and simulators it supports.

Demo Program	C6000 DSK	C6000 Simulator	C5500 DSK	C5500 Simulator	C5400 DSK	C5400 Simulator	C2800 DSK	C2800 Simulator
`ccsfirdemo`	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
`ccsinspect`	Yes	Yes	No	No	Yes	Yes	No	No
`ccstutorial`	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
`hiltutorial`	Yes	Yes	No	No	Yes	Yes	No	No
`rtdxlmsdemo`	Yes	Yes	No	Yes	Yes	Yes	No	No
`rtdxtutorial`	Yes	Yes	No	Yes	Yes	Yes	No	No

Support for the Texas Instruments XDS560™ JTAG Emulator

Version 1.1 supports RTDX communications using the XDS560™ JTAG Emulator from Texas Instruments. This supplements the support for the XDS

510 provided by the earlier version. You should be able to transition directly from your XDS510 to the XDS560 with little change.

Major Bug Fixes

MATLAB Link for Code Composer Studio™ Development Tools 1.1 includes several bug fixes made since Version 1.0 (Release 13). This section describes the particularly important Version 1.1 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.

Upgrading from an Earlier Release

This section describes the upgrade issues involved in moving to the MATLAB Link for Code Composer Studio™ Development Tools 1.1 from Version 1.0.

If you are upgrading from a Release earlier than Release 13, then you should see “Upgrading from an Earlier Release” on page 3-4 in the MATLAB Link for Code Composer Studio Development Tools 1.0 Release Notes.

Code Composer Studio 2.1x Support

Version 1.1 of the MATLAB Link for Code Composer Studio requires Code Composer Studio Development Tools Version 2.1 or higher.

goto Method Modified Behavior

The `goto` method now positions the cursor at the beginning of the specified function in your project, but does not run the code or do any initialization for variables. This changes the method behavior—previous code that uses `goto` may behave differently as a result of this change. Now use `run`, rather than `goto`, to position the program counter to a specified location or function and initialize variables in your project. For more information, refer to `ccshelp/run` and `ccshelp/goto` in the online documentation.

Known Software and Documentation Problems

This section includes a link to a description of known software and documentation problems in Version 1.1.

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.

If you are upgrading from a Release earlier than Release 13, then you should see "Known Software and Documentation Problems" on page 3-5 in the MATLAB Link for Code Composer Studio Development Tools 1.0 Release Notes.