

Embedded IDE Link™ CC Release Notes

How to Contact The MathWorks



www.mathworks.com Web
comp.soft-sys.matlab Newsgroup
www.mathworks.com/contact_TS.html Technical Support



suggest@mathworks.com Product enhancement suggestions
bugs@mathworks.com Bug reports
doc@mathworks.com Documentation error reports
service@mathworks.com Order status, license renewals, passcodes
info@mathworks.com Sales, pricing, and general information



508-647-7000 (Phone)



508-647-7001 (Fax)



The MathWorks, Inc.
3 Apple Hill Drive
Natick, MA 01760-2098

For contact information about worldwide offices, see the MathWorks Web site.

Embedded IDE Link™ CC Release Notes

© COPYRIGHT 2005–2009 by The MathWorks, Inc.

The software described in this document is furnished under a license agreement. The software may be used or copied only under the terms of the license agreement. No part of this manual may be photocopied or reproduced in any form without prior written consent from The MathWorks, Inc.

FEDERAL ACQUISITION: This provision applies to all acquisitions of the Program and Documentation by, for, or through the federal government of the United States. By accepting delivery of the Program or Documentation, the government hereby agrees that this software or documentation qualifies as commercial computer software or commercial computer software documentation as such terms are used or defined in FAR 12.212, DFARS Part 227.72, and DFARS 252.227-7014. Accordingly, the terms and conditions of this Agreement and only those rights specified in this Agreement, shall pertain to and govern the use, modification, reproduction, release, performance, display, and disclosure of the Program and Documentation by the federal government (or other entity acquiring for or through the federal government) and shall supersede any conflicting contractual terms or conditions. If this License fails to meet the government's needs or is inconsistent in any respect with federal procurement law, the government agrees to return the Program and Documentation, unused, to The MathWorks, Inc.

Trademarks

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See www.mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Patents

The MathWorks products are protected by one or more U.S. patents. Please see www.mathworks.com/patents for more information.

Summary by Version	1
Version 3.4 (R2009a) Embedded IDE Link CC	4
Version 3.3 (R2008b) Embedded IDE Link CC	8
Version 3.2 (R2008a) Embedded IDE Link CC	11
Version 3.1 (R2007b) Link for Code Composer Studio ..	13
Version 3.0 (R2007a) Link for Code Composer Studio ..	17
Version 2.1 (R2006b) Link for Code Composer Studio Development Tools	22
Version 2.0 (R2006a+) Link for Code Composer Studio Development Tools	25
Version 1.5 (R2006a) Link for Code Composer Studio Development Tools	31
Version 1.4.2 (R14SP3) Link for Code Composer Studio Development Tools	32
Compatibility Summary for Embedded IDE Link CC ..	33

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "Using Release Notes" on page 1.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V3.4 (R2009a)	Yes Details	Yes Summary	Bug Reports	Printable Release Notes: PDF Current product documentation
V3.3 (R2008b)	Yes Details	Yes Summary	Bug Reports	No
V3.2 (R2008a)	Yes Details	No	Bug Reports	No
V3.1 (R2007b)	Yes Details	Yes Summary	Bug Reports	No
V3.0 (R2007a)	Yes Details	Yes Summary	Bug Reports	No
V2.1 (R2006b)	Yes Details	Yes Summary	Bug Reports	No
V2.0 (R2006a+)	Yes Details	Yes Summary	Bug Reports	No
V1.5 (R2006a)	Yes Details	Yes Summary	Bug Reports	No
V1.4.2 (R14SP3)	Yes Details	Yes Summary	Bug Reports	No

Using Release Notes

Use release notes when upgrading to a newer version to learn about:

- New features
- Changes
- Potential impact on your existing files and practices

Review the release notes for other MathWorks™ products required for this product (for example, MATLAB® or Simulink®) for enhancements, bugs, and compatibility considerations that also might impact you.

If you are upgrading from a software version other than the most recent one, review the release notes for all interim versions, not just for the version you are installing. For example, when upgrading from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What's in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

When a new feature or change introduces a reported incompatibility between versions, the **Compatibility Considerations** subsection explains the impact.

Compatibility issues reported after the product is released appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so you should also review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

The MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. This includes provisions for any known workarounds or file replacements. Information is available

for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Version 3.4 (R2009a) Embedded IDE Link CC

This table summarizes what's new in V3.4 (R2009a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes Summary	Bug Reports	Printable Release Notes: PDF Current product documentation

New features and changes introduced in this version are described here.

- “C6000 Target Function Library Support for Multiply Extended to New Data Types ” on page 4
- “Demo for Using Target Function Libraries to Optimize Code” on page 5
- “Ability to Replace Any Generated File in a Project with a Custom File” on page 5
- “Source File Replacement Diagnostic Option” on page 5
- “PIL Block Behavior with Goto and From Blocks” on page 6
- “Features, Functions, and Arguments Being Removed” on page 6

C6000 Target Function Library Support for Multiply Extended to New Data Types

This release extends the multiply intrinsics to cover new data types. Additions include the signed data type combinations shown in the following table

Input Data Type 1 (Bits)	Input Data Type 2 (Bits)	Output Data Type (Bits)
Signed 32	Signed 32	Signed 16

Input Data Type 1 (Bits)	Input Data Type 2 (Bits)	Output Data Type (Bits)
Signed 32	Signed 16	Signed 16
Signed 32	Signed 8	Signed 16
Signed 32	Signed 32	Signed 8
Signed 32	Signed 16	Signed 8
Signed 32	Signed 8	Signed 8
Signed 16	Signed 8	Signed 8
Signed 8	Signed 8	Signed 8

To see the complete list of intrinsics available for C6000 processors, enter `RTW.viewTf1` at the Command prompt.

Demo for Using Target Function Libraries to Optimize Code

A new demo program in the online Help system shows you how to use target function libraries to optimize generated code. Access the demo *Optimizing Embedded Code via Target Function Library* in the Embedded IDE Link™ CC Project Generator demos on the **Demos** tab in the online Help.

Ability to Replace Any Generated File in a Project with a Custom File

When you generate a project, you can replace any generated file with a file you provide. The **Board custom code** options in the Target Preferences block allow you to specify a custom file with the same name as a generated file. When you generate code, the build process replaces the generated file with your custom file. For more information about replacing files, refer to Target Preferences in the Embedded IDE Link CC documentation in the online Help system.

Source File Replacement Diagnostic Option

To support the custom file replacement capability, a new option in the Embedded IDE Link CC configuration parameters, **Source file replacement**,

lets you control the messages you see when you replace source code with custom code.

PIL Block Behavior with Goto and From Blocks

Currently, it is possible but not recommended to use Goto and From blocks for I/O data that crosses the boundary of the PIL block component. For nonatomic subsystems, the right-click PIL build transforms boundary-crossing Goto blocks into outports and From blocks into inports. The resulting PIL block has extra I/O ports and you must rework the model to connect the PIL block. Starting in the next release, you will see an error if your PIL component includes any Goto or From blocks that cross the boundary of the PIL component. For more information on PIL support, refer to “PIL Feature Support and Limitations” in the Real-Time Workshop® Embedded Coder™ documentation.

Features, Functions, and Arguments Being Removed

The following table lists features, functions, and arguments that will be removed from the product in a future version.

Feature, Function, or Argument Name	What Happens Now When You Use the Feature, Function, or Argument	Use This Instead
createobj— Data Manipulation	Errors	None
createobj — Function Call	Errors	None
Function Call block (demo feature)	Errors	None
cexpr	Errors	None
close(, 'text')	Errors	None
configure(rx,)	Warns	None
datatypemanager	Warns	None

Feature, Function, or Argument Name	What Happens Now When You Use the Feature, Function, or Argument	Use This Instead
display with RTDX objects on C5000™ and C6000™ processors	Warns	None
goto	Errors	None
list(,type)	Errors	None
new(,'text')	Errors	None
open(,'text')	Errors	None
open(,'program')	Errors	load
open(,'workspace')	Errors	None
RTDX™ Support on C5000 and C6000 processors	Warns	None
save(,'text')	Errors	None

Version 3.3 (R2008b) Embedded IDE Link CC

This table summarizes what's new in V3.3 (R2008b):

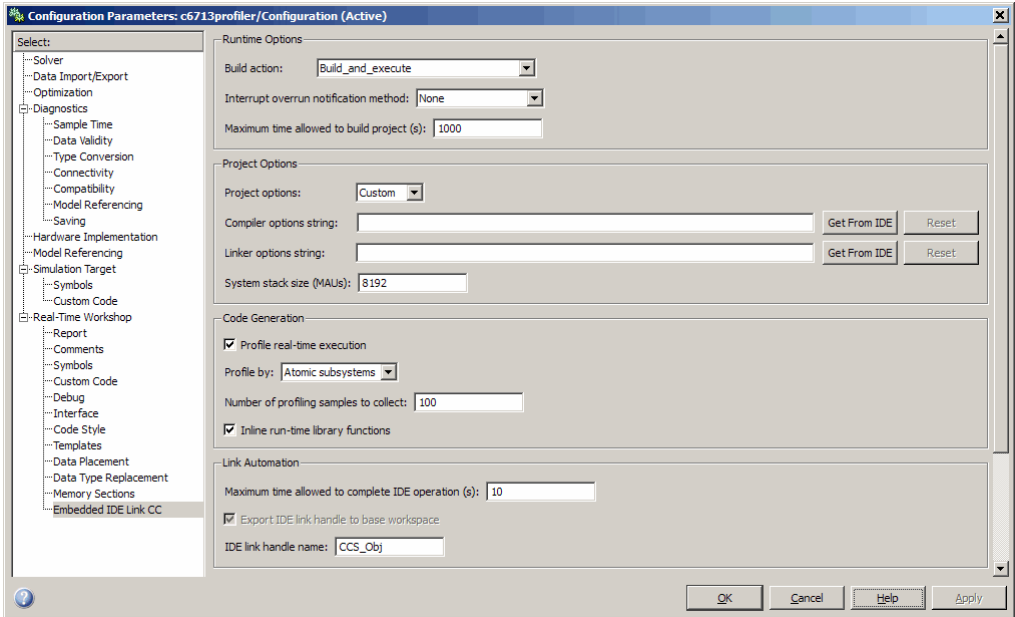
New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes Summary	Bug Reports	Printable Release Notes: PDF Current product documentation

New features and changes introduced in this version are described here.

- “Real-Time Execution Profiling Extended to Atomic Subsystems” on page 8
- “Support for C2823x Processors” on page 9
- “Features, Functions, and Arguments Being Removed” on page 9

Real-Time Execution Profiling Extended to Atomic Subsystems

If your models use atomic subsystems, you can use the real-time execution profiling feature to investigate how your subsystems perform when you execute your program on a processor. A new option on the Embedded IDE Link CC pane in Configuration Parameters, **Profile by**, lets you select whether to profile by task or by atomic subsystem. The following figure shows the new option:



For more information, refer to “Profiling Execution By Subsystems”.

Support for C2823x Processors

You can develop applications for C2823x processors from Texas Instruments™. Both the automation interface and project generation components support the new processor.

Features, Functions, and Arguments Being Removed

The following table lists features, functions, and arguments that will be removed from the product in a future version.

Feature, Function, or Argument Name	What Happens When You Use Feature, Function, or Argument	Use This Instead
createobj— Data Manipulation	createobj warns	None

Feature, Function, or Argument Name	What Happens When You Use Feature, Function, or Argument	Use This Instead
createobj — Function Call	createobj warns	None
Function Call block (demo feature)	Warns	None
cexpr	Warns	None
close(, 'text')	Warns	None
goto	Warns	None
list(, type)	Warn	None
new(, 'text')	Warns	None
open(, 'text')	Warns	None
open(, 'program')	Warns	load
open(, 'workspace')	Warns	None
save(, 'text')	Warns	None

Version 3.2 (R2008a) Embedded IDE Link CC

This table summarizes what's new in V3.2 (R2008a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports	Printable Release Notes: PDF Current product documentation

New features and changes introduced in this version are described here.

- ““What’s This?” Context-Sensitive Help Available for Simulink Configuration Parameters Dialog” on page 11
- “Profiling Stack Usage” on page 12
- “Support for C2833x Floating Point Processor” on page 12

“What’s This?” Context-Sensitive Help Available for Simulink Configuration Parameters Dialog

R2008a introduces “What’s This?” context-sensitive help for parameters that appear in the Simulink Configuration Parameters dialog. This feature provides quick access to a detailed description of the parameters, saving you the time it would take to find the information in the Help browser.

To use the “What’s This?” help, do the following:

- 1 Place your cursor over the label of a parameter.
- 2 Right-click. A **What’s This?** context menu appears.

For example, the following figure shows the **What’s This?** context menu appearing after a right-click on the **Start time** parameter in the **Solver** pane.



- 3 Click **What's This?** A context-sensitive help window appears showing a description of the parameter.

Profiling Stack Usage

The profile command includes new input arguments that enable you to profile how your application uses the stack. Input arguments setup and report configure and return stack use information. For more details, refer to `profile` and “System Stack Profiling” in the Embedded IDE Link CC User’s Guide.

Support for C2833x Floating Point Processor

You can develop applications for the C2833x floating-point processor from Texas Instruments. Both the automation interface and project components support the new processor.

Version 3.1 (R2007b) Link for Code Composer Studio

This table summarizes what's new in V3.1 (R2007b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports	Printable Release Notes: PDF V3.1 product documentation

New features and changes introduced in this version are described here.

- “Support for CCS 3.3” on page 13
- “Support for Real-Time Workshop Target Function Library Replacement” on page 14
- “set and get Methods with Updated Syntax” on page 14
- “Refactored PIL Project Options in Model Configuration Parameters” on page 15
- “Function ccdsp Returns ticcs Object” on page 15
- “Link and Target Products Regrouped in New Start, Help, and Demos Category” on page 15

Support for CCS 3.3

This release supports only Code Composer Studio Version 3.3.

Compatibility Consideration

When you upgrade to this release of Link for Code Composer Studio, you must upgrade Code Composer Studio to version 3.3.

Support for Real-Time Workshop Target Function Library Replacement

If you are using `ccslink_ert.tlc` as your system target file, Link for CCS now supports generating code that is optimized for the processor by using compiler intrinsics and assembly code to replace certain mathematical operator functions. Link for CCS accomplishes this optimization through the Target Function Library (TFL) replacement mechanism that Real-Time Workshop provides.

TFL replacement requires Real-Time Workshop Embedded Coder and the embedded real-time target—`ccslink_ert.tlc`.

For more general information about TFL, look for TFL in the Interface options in “Configuring Real-Time Workshop® Code Generation Parameters”. For information about using TFL in Link for CCS, refer to “Optimizing Embedded Code with Target Function Libraries”.

set and get Methods with Updated Syntax

`set` and `get` now require both an object variable and property name input arguments to work. In earlier releases, you could use `set` and `get` with only the object variable name input to see a list of all of the properties for an object.

The following command syntaxes are no longer supported by `set` and `get`. In the examples, `cc` is a `ticcs` object created by `ccsdsp`:

- `get(cc)`
- `value = get(cc)`
- `set(cc)`
- `set(cc, 'propertyname1', propertyvalue1, 'propertyname2', propertyvalue2, ...)`
- `value = set(cc)`

The following syntax options work as before:

- `get(cc, 'propertyname')`
- `value = get(cc, 'propertyname')`

- `set(cc, 'propertyname', propertyvalue)`

Compatibility Consideration

The syntax changes to `set` and `get` affect scripts that use `set` and `get` to return lists of object properties. `Set` and `get` no longer return lists, only individual property values.

Refactored PIL Project Options in Model Configuration Parameters

The existing PIL block option—**Configure model to build PIL algorithm object code**—has been replaced by a new option `Create_Processor_in_the_Loop_project` on the **Build action** list and a new parameter called **PIL block action**.

To build the PIL block with the new options, select `Create_Processor_in_the_Loop_project` for the **Build action**, and then select an option from the **PIL block action** list. For more information about the new configuration parameters and using PIL blocks, refer to [Using Processor-in-the-Loop](#).

Function `ccsdsp` Returns `ticcs` Object

In previous releases, the basic object constructor `ccsdsp` returned a `ccsdsp` object. `ccsdsp` now returns a `ticcs` object, an instance of the `ticcs` class. This represents a change to the name of the object, not the way the object behaves or the properties.

Link and Target Products Regrouped in New Start, Help, and Demos Category

A new product category, `Links and Targets`, now contains all MathWorks software products that link, target, or cosimulate code.

Compatibility Considerations

This change impacts you in the following ways:

- Finding and viewing these products through the MATLAB Desktop **Start** button and in the Help browser **Contents** and **Demos** panes.
- Using the `demo` command to access the product demos.

For more about this new product category, see “Demos and Help Browser Contents Now Include New Category for Links and Targets”, in the *MATLAB Release Notes*.

Version 3.0 (R2007a) Link for Code Composer Studio

This table summarizes what's new in V3.0 (R2007a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports	Printable Release Notes: PDF

New features and changes introduced in this version are described here.

- “Project Generator Enables Project Generation for C2000, C5000, and C6000 Processors from Simulink Models” on page 17
- “Processor in the Loop Cosimulation with C2000, C5000, and C6000 Processors” on page 18
- “Blocks for Interrupt Handling and Accessing Memory in Projects Generated by Project Generator” on page 18
- “Real-Time Execution Profiling for Projects Generated with Project Generator and Running on Target Processors” on page 19
- “Demos That Introduce PIL Cosimulation and Real-Time Execution Profiling” on page 19
- “Adding a New Processor” on page 20
- “General Notes for the Release” on page 20

Project Generator Enables Project Generation for C2000, C5000, and C6000 Processors from Simulink Models

A new component, Project Generator, enables you to generate projects into Code Composer Studio from Simulink models. Embedded IDE Link CC now comprises the following components:

- Automation Interface component that provides all the earlier product features, such as project debugging, data manipulation, and function call. For more information, refer to [Getting Started with Automation Interface](#)
- Project Generator component that provides the new project generation features. Refer to [Introducing Project Generator](#) for more information.

Note Project Generator component use requires Simulink and Real-Time Workshop.

Processor in the Loop Cosimulation with C2000, C5000, and C6000 Processors

Use processor in the loop (PIL) cosimulation techniques to verify the performance of your generated code running on an instruction set simulator or hardware target. For further information, refer to [Using Processor-in-the-Loop](#).

Note Real-Time Workshop Embedded Coder Required to use the PIL feature

Blocks for Interrupt Handling and Accessing Memory in Projects Generated by Project Generator

A new library, `ccslinklib`, contains blocks that enable you to use interrupt handling and access the memory in the projects you generate with Project Generator. The blocks cover the C280x, C281x, C5000, and C6000 processor families. To open the new library, enter:

```
ccslinklib
```

at the MATLAB command prompt.

`ccslinklib` includes the following block libraries:

Library Name	Contents
C280x DSP Chip Support	Blocks for memory operations and asynchronous scheduling in generated code
C281x DSP Chip Support	Blocks for memory operations and asynchronous scheduling in generated code
C5000 DSP Chip Support	Blocks for memory operations and asynchronous scheduling in generated code
C6000 DSP Chip Support	Blocks for memory operations and asynchronous scheduling in generated code
Target Preferences	Custom target preferences block for configuring the processor and mapping the processor internal and external memory

Real-Time Execution Profiling for Projects Generated with Project Generator and Running on Target Processors

The profiler includes a set of utilities for recording, uploading, and analyzing execution profile data for synchronous and asynchronous tasks. It generates a display that shows when tasks are activated, preempted, resumed, and completed. In addition, the profiler generates an HTML report with statistical information on each synchronous task. Refer to the demo Real-Time Task Execution Profiling in the online Help for an introduction. For more information, refer to Real-Time Execution Profiling.

Demos That Introduce PIL Cosimulation and Real-Time Execution Profiling

Two new demo programs in the online Help system show you how to use the new PIL and execution profiling capabilities. Access these demos from the **Demos** tab in the online Help.

- Comparing Simulation and Target Implementation with Processor-in-the-Loop (PIL)
- Real-Time Task Execution Profiling

Adding a New Processor

In the custom Target Preferences block in the Target Preferences library, a new feature lets you add a new processor to support. On the **Board Info** pane of the block, **Add new** opens a dialog box where you configure your new processor for support and code generation. You set the processor name and class, the internal memory mapping, the clock speed, the cache, and more.

General Notes for the Release

The compatibility considerations from the Version 2.0 release (R2006a+) remain applicable.

Compatibility Considerations

The following issues reflect changes to the support provided by Embedded IDE Link CC:

- The product no longer supports the RTDX-based demos on the C54x simulator and on C6701 processor-based hardware. These include:
 - Real Time Data Exchange Tutorial
 - LMS Adaptive Filtering
 - Transferring Data Between Simulink and a Target Application
- When you use the C54x simulator, the `disp` method does not display the correct processor type for your target. The property `revfamily` is set to an improper processor revision.
- When you use the C6713 Device Cycle simulator, the `disp` method does not display the correct processor subfamily identifier. The property `subfamily` is set to an improper processor subfamily value of 100 instead of 103.
- If you use the C6713 processor with a USB connection, RTDX communications does not work when you launch Code Composer Studio from Embedded IDE Link CC. To work around this problem, launch CCS manually from Microsoft Windows®.

- With the C2808 processor as your target, you cannot enable more than one RTDX read channel from the host-side application. In some cases, enabling one channel from the host disables a previously enabled channel. To work around this problem, enable the RTDX channels programmatically in the C2808 application code.
- The product no longer support the Debugging a Target Application demo (ccsdebugdemo) on TMS470R2x hardware.
- When you launch CCS by issuing the command `cc=ccsdsp` at the MATLAB prompt, and then you use any DSP/BIOS plug-in or the `profile` function, you encounter an error that says `vbd.dll` could not be loaded. Work around this by opening CCS manually before you use `ccsdsp`.
- When you use `open` to open a project in a CCS IDE window and the project is already open in another CCS IDE window, Embedded IDE Link CC returns a time-out error in MATLAB. You can avoid this error by:
 - Opening the project in a new CCS IDE window manually.
 - Closing the existing project in the other IDE before you open the new instance of your project.

Version 2.1 (R2006b) Link for Code Composer Studio Development Tools

This table summarizes what's new in V2.1 (R2006b):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports	Printable Release Notes: PDF

New features and changes introduced in this version are described here.

- “Extended Support to Two Code Composer Studio Versions” on page 22
- “New Function Call Block in New Demo – Comparing Simulation and Target Implementation with Function Call” on page 23
- “New Demo Featuring Debug Component – Comparing Simulation and Target Implementation with Debug Methods” on page 23
- “Demo Support for New Processors” on page 23
- “General Notes for the Release” on page 23

Extended Support to Two Code Composer Studio Versions

- Added CCS 3.2 for the C64x+ and C67x+ processor families
- Continuing support for CCS 3.1 (C6000, C5000, C2000, Rxx, OMAP processor families)

Additional or continuing compatibility considerations are listed in “Compatibility Considerations” on page 26.

New Function Call Block in New Demo – Comparing Simulation and Target Implementation with Function Call

To complement the function objects and createobj in the product, this release includes a new demo that introduces a demonstration function call block — “Comparing Simulation And Target Implementation Through Function Call.” You use the block in a Simulink model to identify and run a function on your target. The ability to communicate between Simulink and your target lets you run models that exercise functions and algorithms that reside on your target, and move data back and forth between MATLAB, your Simulink model, and your target-based process. Refer to the demos for Link for Code Composer Studio in MATLAB to find and run the new demo.

New Demo Featuring Debug Component – Comparing Simulation and Target Implementation with Debug Methods

A Simulink-based demo featuring the Debug component that shows how you use the debug methods to work with targets.

Demo Support for New Processors

C64x+ and C67x+ processor families supported by all demos.

General Notes for the Release

The compatibility considerations from the Version 2.0 release (R2006a+) remain applicable as shown here.

Compatibility Considerations

The following issues reflect changes to the support provided by Embedded IDE Link CC:

- The product no longer supports the RTDX-based demos on the C54x simulator and on C6701 processor-based hardware. These include
 - Real Time Data Exchange Tutorial
 - LMS Adaptive Filtering

- Transferring Data Between Simulink and a Target Application
- When you use the C54x simulator, the `disp` method does not display the correct processor type for your target. The property `revfamily` is set to an improper processor revision.
- When you use the C6713 Device Cycle simulator, the `disp` method does not display the correct processor subfamily identifier. The property `subfamily` is set to an improper processor subfamily value of 100 instead of 103.
- If you use the C6713 processor with a USB connection, RTDX communications does not work when you launch Code Composer Studio from Embedded IDE Link CC. To work around this problem, launch CCS manually from Microsoft Windows.
- With the C2808 processor as your target, you cannot enable more than one RTDX read channel from the host-side application. In some cases, enabling one channel from the host disables a previously enabled channel. To work around this problem, enable the RTDX channels programmatically in the C2808 application code.
- The product no longer support the Debugging a Target Application demo (`ccsdebugdemo`) on TMS470R2x hardware.
- When you launch CCS by issuing the command `cc=ccsdsp` at the MATLAB prompt, and then you use any DSP/BIOS plug-in or the `profile` function, you encounter an error that says `vbd.dll` could not be loaded. Work around this by opening CCS manually before you use `ccsdsp`.
- When you use `open` to open a project in a CCS IDE window and the project is already open in another CCS IDE window, Embedded IDE Link CC returns a time-out error in MATLAB. You can avoid this error by
 - Opening the project in a new CCS IDE window manually.
 - Closing the existing project in the other IDE before you open the new instance of your project.

Version 2.0 (R2006a+) Link for Code Composer Studio Development Tools

This table summarizes what's new in V2.0 (R2006a+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports	Printable Release Notes: PDF

New features and changes introduced in this version are described here.

- “New architecture for Link for Code Composer Studio Development Tools” on page 25
- “Timeout Argument Added to insert and delete Methods” on page 27
- “insert and delete Now Blocking Operations” on page 27
- “RTDX Simulation Blocks in New Demo RTDX Simulation” on page 28
- “build Supports Output Arguments” on page 28
- “createobj Improved to Work With All Functions” on page 28
- “CCS 3.1 Compatibility Updates” on page 29

New architecture for Link for Code Composer Studio Development Tools

This release of the product uses a new internal structure to interact with Code Composer Studio™ (CCS). Link for CCS now installs components (also called plug-ins) in CCS IDE that enable the links and MATLAB commands to run in the IDE process. The internal changes provide improved performance when you read or write data to your target. The interface you see in MATLAB has not changed, only the implementation of the software. All of your scripts and existing code work as before in earlier versions.

Installation Notes

Link for CCS registers a plug-in tool for Code Composer Studio (CCS). After you install Link for CCS, start CCS. CCS will detect the plug-in and prompt you that "New components were detected." Click **Yes** to enable tools for all compatible CCS releases.

The New Components dialog appears only once. If you do not accept the new components, you cannot use Link for CCS without reregistering the new components and then using the Component Manager in CCS to enable them.

Compatibility Considerations

The following changes result from the new architecture and the improved performance.

- Changing the architecture affects the way MATLAB connects to the CCS IDE. As a result, after you create `ccsdsp` objects that link to the CCS IDE, you should not close the IDE associated with the `ccsdsp` object without clearing the handle (your `ccsdsp` object) from your MATLAB workspace. Closing the IDE without clearing the handle leaves invalid handles. Use this command to clear the specified handles:

```
clear(handlesname)
```

If you do not clear the handles, you get the following error when you try to use an invalid handle to the missing IDE instance:

```
??? CCS was closed independent of MATLAB. All existing handles are invalid.
```

```
Please clear all CCS handles and recreate them.
```

To proceed from here, you must clear the handles and then recreate your `ccsdsp` object.

- If you use the `load` and `build` functions from Embedded IDE Link CC in a MATLAB script, you might find that your script no longer builds and loads projects properly. Now MATLAB returns a message about an unspecified error. The error can occur because the `load` and `build` operations issued by your script are conflicting with the **Load Program After Build** option in CCS. If you select **Load Program After Build** in CCS, CCS automatically loads

programs after building them. When your script issues the build and load requests, they conflict with the CCS build and load and the process fails.

To prevent your scripts from failing in this case, go to **Option > Customize > Program/Project Load** in CCS IDE and clear the **Load Program After Build** option before you run your script.

Timeout Argument Added to insert and delete Methods

Both `insert` and `delete` now offer an input argument, `timeout`, that lets you specify explicitly the time the insert or delete operations wait while CCS processes the commands. You specify the time-out period in seconds. Please note the next new feature below that relates to this one. Note also that the `timeout` input option applies to all debug point options—break points, probe points, and profile points (called *debug* points).

insert and delete Now Blocking Operations

When you use `insert` or `delete` to add or remove a debug point (a break point, probe point, or profile point) in CCS, the process now operates in *blocking* mode. Blocking mode means that after you issue a `insert` or `delete` command in MATLAB, the MATLAB method retains control until your requested process reaches completion (it adds or deletes the debug point) or the process exceeds the time-out period as specified by either the optional `timeout` input argument or the default time-out value in `cc.timeout`. This represents a change to the earlier `insert` and `delete` operations.

Compatibility Considerations

The following changes result from the new `insert` and `delete` operation:

- Making `insert` and `delete` blocking means unsuccessful `insert` or `delete` operations return errors when the operation fails. In earlier releases, `insert` and `delete` did not return errors related to the success or failure of the operation. When you use the `line` input argument to insert a debug point on a specified line, `line` must represent a valid line. If `line` does not specify a valid line, `insert` returns an error and does not insert the debug point.

- `delete(..., 'all', ...)` removes all valid break points in the project code. It does not remove probe points.

RTDX Simulation Blocks in New Demo RTDX Simulation

A new demo, Transferring Data Between Simulink and a Target Application (`rtdxsumdiffdemo`), introduces two new blocks for working with RTDX in Simulink models. The new blocks, RTDX Read and RTDX Write, enable you to simulate RTDX communications with an application running on a hardware platform.

These new blocks are for demonstration purposes only. And they only apply in Simulink models on the host. They do not generate code for target applications.

build Supports Output Arguments

`build` now supports a pair of output arguments that return information about the build process. This syntax

```
[result,numwarns]=build(...)
```

returns two output values that report the results of the build operation. For a successful build, the output arguments are the following:

- `result` equals 1 for the build
- `numwarns` reports the number of build warnings that occurred during the build.

When the build is not successful, `build` displays an error and a message that contains the build error string in the MATLAB Command Window.

createobj Improved to Work With All Functions

`createobj` now returns complete function objects for library functions as well as C functions. In earlier releases, `createobj` could not return function objects from library functions that contained all necessary information to run the function.

Compatibility Considerations

The following changes result from the upgraded `createobj`.

- When you create a function object for a library function, you do not need to perform the `declare` operation to specify the function arguments. If you have scripts that use `createobj` followed by `declare`, remove the `declare` command from the scripts.
- If you are using any C28xx target, you cannot use the `declare` function after you create your function object with `createobj`. Using the `declare` causes an error. If you have scripts that use `createobj` followed by `declare`, remove the `declare` command from the scripts.
- Before you use a project built with any release of CCS before V3.0, rebuild the project in CCS 3.1.
- Always use the Full Symbolic Debug setting for the **Generate Debug Info** build option for your project in CCS.

CCS 3.1 Compatibility Updates

This release of Embedded IDE Link CC continues to support Code Composer Studio 3.1 with additional compatibility considerations as listed here.

Compatibility Considerations

The following issues reflect changes to the support provided by Embedded IDE Link CC:

- The product no longer supports the RTDX-based demos on the C54x simulator and on C6701 processor-based hardware. These include
 - Real Time Data Exchange Tutorial
 - LMS Adaptive Filtering
 - Transferring Data Between Simulink and a Target Application
- When you use the C54x simulator, the `disp` method does not display the correct processor type for your target. The property `revfamily` is set to an improper processor revision.

- When you use the C6713 Device Cycle simulator, the `disp` method does not display the correct processor subfamily identifier. The property `subfamily` is set to an improper processor subfamily value of 100 instead of 103.
- If you use the C6713 processor with a USB connection, RTDX communications does not work when you launch Code Composer Studio from Embedded IDE Link CC. To work around this problem, launch CCS manually from Microsoft Windows.
- With the C2808 processor as your target, you cannot enable more than one RTDX read channel from the host-side application. In some cases, enabling one channel from the host disables a previously enabled channel. To work around this problem, enable the RTDX channels programmatically in the C2808 application code.
- The product no longer support the Debugging a Target Application demo (`ccsdebugdemo`) on TMS470R2x hardware.
- When you launch CCS by issuing the command `cc=ccsdsp` at the MATLAB prompt, and then you use any DSP/BIOS plug-in or the `profile` function, you encounter an error that says `vbd.dll` could not be loaded. Work around this by opening CCS manually before you use `ccsdsp`.
- When you use `open` to open a project in a CCS IDE window and the project is already open in another CCS IDE window, Embedded IDE Link CC returns a time-out error in MATLAB. You can avoid this error by
 - Opening the project in a new CCS IDE window manually.
 - Closing the existing project in the other IDE before you open the new instance of your project.

Version 1.5 (R2006a) Link for Code Composer Studio Development Tools

This table summarizes what's new in V1.5 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports	Printable Release Notes: PDF V1.5 product documentation

New features and changes introduced in this version are described here:

CCS 3.1 Compatibility

Link for CCS supports CCS 3.1 exclusively. To use V1.5 of Link for CCS, you must upgrade your CCS installation to Version 3.1.

Compatibility Considerations

Link for CCS V1.5 does not support operation with any version of Code Composer Studio before 3.1.

Version 1.4.2 (R14SP3) Link for Code Composer Studio Development Tools

This table summarizes what's new in V1.4.2 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports	Printable Release Notes: PDF

New features and changes introduced in this version are described here:

Improved Performance When Creating Function Objects

For this release, we changed the `createobj` method to reduce the time it takes to create new objects. When you create an object, `createobj` no longer saves any registers while creating the object. Earlier versions stored default registers from a list named `saveregs`. That list is now empty by default. As a result, the process for constructing new objects is faster. You can still add new registers to the saved registers list with `addregister`, and you can delete saved registers from the list with `deleteregister`. You must add the registers to the saved registers list immediately after you create the object.

Compatibility Considerations

Because we no longer save any default registers, the `cleanup` method does not perform any function unless you added registers to `saveregs`.

Compatibility Summary for Embedded IDE Link CC

This table summarizes new features and changes that might cause incompatibilities when you upgrade from an earlier version, or when you use files on multiple versions. Details are provided in the description of the new feature or change.

Version (Release)	New Features and Changes with Version Compatibility Impact
Latest Version V3.4 (R2009a)	See this change: <ul style="list-style-type: none"> • “Features, Functions, and Arguments Being Removed” on page 6
V3.3 (R2008b)	See this change: <ul style="list-style-type: none"> • “Features, Functions, and Arguments Being Removed” on page 9
V3.2 (R2008a)	No
V3.1 (R2007b)	See the Compatibility Considerations for these changes: <ul style="list-style-type: none"> • “Support for CCS 3.3” on page 13 • “set and get Methods with Updated Syntax” on page 14 • “Link and Target Products Regrouped in New Start, Help, and Demos Category” on page 15
V3.0 (R2007a)	See the Compatibility Considerations for this change “General Notes for the Release” on page 20
V2.1 (R2006b)	See the Compatibility Considerations for this change “General Notes for the Release” on page 23

Version (Release)	New Features and Changes with Version Compatibility Impact
V2.0 (R2006a+)	<p>See the Compatibility Considerations for these new features or changes:</p> <ul style="list-style-type: none"> • “New architecture for Link for Code Composer Studio Development Tools” on page 25 • “insert and delete Now Blocking Operations” on page 27 • “createobj Improved to Work With All Functions” on page 28 • “CCS 3.1 Compatibility Updates” on page 29
V1.5 (R2006a)	<p>See the Compatibility Considerations subheading for this new feature or change:</p> <ul style="list-style-type: none"> • “CCS 3.1 Compatibility” on page 31
V1.4.2 (R14SP3)	<p>See the Compatibility Considerations subheading for this new feature or change:</p> <ul style="list-style-type: none"> • “Improved Performance When Creating Function Objects” on page 32