

MATLAB[®] Coder[™] Release Notes

How to Contact MathWorks



www.mathworks.com
comp.soft-sys.matlab
www.mathworks.com/contact_TS.html

Web
Newsgroup
Technical Support



suggest@mathworks.com
bugs@mathworks.com
doc@mathworks.com
service@mathworks.com
info@mathworks.com

Product enhancement suggestions
Bug reports
Documentation error reports
Order status, license renewals, passcodes
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.

MATLAB® Coder™ Release Notes

© COPYRIGHT 2011 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

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 2.0 (R2011a) MATLAB® Coder	3
Compatibility Summary for MATLAB® Coder	13

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
New Product V2.0 (R2011a)	Yes Details	Yes Summary	Bug Reports

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®). Determine if enhancements, bugs, or compatibility considerations in other products impact you.

If you are upgrading from a software version other than the most recent one, review the current release notes and all interim versions. For example, when you upgrade from V1.0 to V1.2, review the release notes for V1.1 and V1.2.

What Is 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 release appear under Bug Reports at the MathWorks Web site. Bug fixes can sometimes result in incompatibilities, so review the fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

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. Bug Reports include 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.

Documentation on the MathWorks Web Site

Related documentation is available on mathworks.com for the latest release and for previous releases:

- Latest product documentation
- Archived documentation

Version 2.0 (R2011a) MATLAB Coder

This table summarizes what's new in V 2.0 (R2011a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems
Yes Details below	Yes—Details labeled as Compatibility Considerations , below. See also Summary.	Bug Reports

New features and changes introduced in this version are

- “New User Interface for Managing Projects” on page 3
- “Migrating from Real-Time Workshop emlc Function” on page 5
- “New coder.Type Classes” on page 8
- “New coder Package Functions” on page 8
- “Script to Upgrade MATLAB Code to Use MATLAB® Coder Syntax” on page 9
- “Embedded MATLAB Now Called Code Generation from MATLAB” on page 9
- “MATLAB® Coder Uses rtwTargetInfo.m to Register Target Function Libraries” on page 9
- “New Getting Started Tutorial Video” on page 9
- “New Demos” on page 9
- “Functionality Being Removed in a Future Version” on page 11
- “Function Elements Being Removed in a Future Release” on page 11

New User Interface for Managing Projects

The new MATLAB® Coder™ user interface simplifies the MATLAB to C/C++ code generation process. Using this user interface, you can:

- Specify the MATLAB files from which you want to generate code

- Specify the data types for the inputs to these MATLAB files
- Select an output type:
 - MEX function
 - C/C++ Static Library
 - C/C++ Executable
- Configure build settings to customize your environment for code generation
- Open the code generation report to view build status, generated code, and compile-time information for the variables and expressions in your MATLAB code

To Get Started

You launch a MATLAB Coder project by doing one of the following:

- From the MATLAB main menu, select **File > New > Code Generation Project**
- Enter `coder` at the MATLAB command line

To learn more about working with MATLAB Coder, see “Tutorial: Generating C Code from MATLAB Code Using the MATLAB Coder Project Interface”.

Migrating from Real-Time Workshop emlc Function

In MATLAB Coder, the codegen function replaces emlc with the following differences:

New codegen Options

Old emlc Option	New codegen Option
-eg	-args
emlcoder.egc	coder.Constant
emlcoder.egs	coder.typeof Creates <code>coder.Type</code> objects for use with the <code>codegen -args</code> option. For more information, see “New <code>coder.Type</code> Classes” on page 8.
-F	No codegen option available. Instead, use the Global fimath. For more information, see “Working with the Global fimath” in the Fixed-Point Toolbox™ documentation.
-global	-globals Note -global continues to work with codegen
-N	This option is no longer supported. Instead, set up <code>numericType</code> in MATLAB.
-s	-config Use with the new configuration objects, see “New Code Generation Configuration Objects” on page 6.
-T rtw:exe	-config:exe Use this option to generate a C/C++ executable using default build options. Otherwise, use <code>-config</code> with a <code>coder.CodeConfig</code> or <code>coder.EmbeddedCodeConfig</code> configuration object.

Old emlc Option	New codegen Option
-T mex	-config:mex Use this option to generate a MEX function using default build options. Otherwise, use -config with a coder.MexCodeConfig configuration object.
-T rtw -T rtw:lib	-config:lib Use either of these options to generate a C/C++ library using default build options. Otherwise, use -config with a coder.CodeConfig or coder.EmbeddedCodeConfig configuration object.

New Code Generation Configuration Objects

The codegen function uses new configuration objects that replace the old emlc objects with the following differences:

Old emlc Configuration Object	New codegen Configuration Object
emlcoder.MEXConfig	coder.MexCodeConfig
emlcoder.RTWConfig emlcoder.RTWConfig('grt')	coder.CodeConfig The SupportNonFinite property is now available without an Embedded Coder™ license. The following property names have changed: <ul style="list-style-type: none"> • RTWCompilerOptimization is now CCompilerOptimization • RTWCustomCompilerOptimization is now CCustomCompilerOptimization • RTWVerbose is now Verbose

Old emlc Configuration Object	New codegen Configuration Object
emlcoder.RTWConfig('ert')	coder.EmbeddedCodeConfig The following property names have changed: <ul style="list-style-type: none"> • MultiInstanceERTCode is now MultiInstanceCode • RTWCompilerOptimization is now CCompilerOptimization • RTWCustomCompilerOptimization is now CCustomCompilerOptimization • RTWVerbose is now Verbose
emlcoder.HardwareImplementation	coder.HardwareImplementation

The codegen Function Has No Default Primary Function Input Type

In previous releases, if you used the emlc function to generate code for a MATLAB function with input parameters, and you did not specify the types of these inputs, by default, emlc assumed that these inputs were real, scalar, doubles. In R2011a, the codegen function does not assume a default type. You must specify at least the class of each primary function input. For more information, see “Specifying Properties of Primary Function Inputs in a MATLAB Coder Project”.

Compatibility Consideration. If your existing script calls emlc to generate code for a MATLAB function that has inputs and does not specify the input types, and you migrate this script to use codegen, you must modify the script to specify inputs.

The codegen Function Processes Compilation Options in a Different Order

In previous releases, the emlc function resolved compilation options from left to right so that the right-most option prevailed. In R2011a, the codegen function gives precedence to individual command-line options over options specified using a configuration object. If command-line options conflict, the right-most option prevails.

Compatibility Consideration. If your existing script calls `emlc` specifying a configuration object as well as other command-line options, and you migrate this script to use `codegen`, `codegen` might not use the same configuration parameter values as `emlc`.

New `coder.Type` Classes

MATLAB Coder includes the following new classes to specify input parameter definitions:

- `coder.ArrayType`
- `coder.Constant`
- `coder.EnumType`
- `coder.FiType`
- `coder.PrimitiveType`
- `coder.StructType`
- `coder.Type`

New `coder` Package Functions

The following new package functions let you work with objects and types for C/C++ code generation:

Function	Purpose
<code>coder.config</code>	Create MATLAB Coder code generation configuration objects
<code>coder.newtype</code>	Create a new <code>coder.Type</code> object
<code>coder.resize</code>	Resize a <code>coder.Type</code> object
<code>coder.typeof</code>	Convert a MATLAB value into its canonical type

Script to Upgrade MATLAB Code to Use MATLAB Coder Syntax

The `coder.upgrade` script helps you upgrade to MATLAB Coder by searching your MATLAB code for old commands and options and replacing them with their new equivalents. For more information, at the MATLAB command prompt, enter `help coder.upgrade`.

Embedded MATLAB Now Called Code Generation from MATLAB

MathWorks is no longer using the term *Embedded MATLAB* to refer to the language subset that supports code generation from MATLAB algorithms. This nomenclature incorrectly implies that the generated code is used in embedded systems only. The new term is *code generation from MATLAB*. This terminology better reflects the full extent of the capability for translating MATLAB algorithms into readable, efficient, and compact MEX and C/C++ code for deployment to both desktop and embedded systems.

MATLAB Coder Uses `rtwTargetInfo.m` to Register Target Function Libraries

In previous releases, the `emlc` function also recognized the customization file, `sl_customization.m`. In R2011a, the MATLAB Coder software does not recognize this customization file, you must use `rtwTargetInfo.m` to register a Target Function Library (TFL). To register a TFL, you must have Embedded Coder software. For more information, see “Using the `rtwTargetInfo` API to Register a TFL with MATLAB Coder Software” in the Embedded Coder documentation.

New Getting Started Tutorial Video

To learn how to generate C code from MATLAB code, see the “Generating C Code from MATLAB Code” video in the MATLAB Coder Getting Started demos.

New Demos

The following demos have been added:

Demo...	Shows How You Can...
Hello World	Generate and run a MEX function from a simple MATLAB program
Working with Persistent Variables	Compute the average for a set of values by using persistent variables
Working with Structure Arrays	Shows how to build a scalar template before growing it into a structure array, a requirement for code generation from MATLAB.
Balls Simulation	Simulates bouncing balls and shows that you should specify only the entry function when you compile the application into a MEX function.
General Relativity with MATLAB Coder	Uses Einstein's theory of general relativity to calculate geodesics in curved space-time.
Averaging Filter	Generate a standalone C library from MATLAB code using <code>codegen</code>
Edge Detection on Images	Generate a standalone C library from MATLAB code that implements a Sobel filter
Read Text File	Generate a standalone C library from MATLAB code that uses the <code>coder.ceval</code> , <code>coder.extrinsic</code> and <code>coder.opaque</code> functions.
"Atoms" Simulation	Generate a standalone C library and executable from MATLAB code using a code generation configuration object to enable dynamic memory allocation

Demo...	Shows How You Can...
Replacing Math Functions and Operators	Use target function libraries (TFLs) to replace operators and functions in the generated code
	<hr/> <p>Note To run this demo, you need Embedded Coder software.</p> <hr/>
Kalman Filter	<ul style="list-style-type: none"> • Generate a standalone C library from a MATLAB version of a Kalman filter • Accelerate the Kalman filter algorithm by generating a MEX function

Functionality Being Removed in a Future Version

This function will be removed in a future version of MATLAB Coder software.

Function Name	What Happens When You Use This Function?	Compatibility Considerations
emlc	Still runs in R2011a	None

Function Elements Being Removed in a Future Release

Function or Element Name	What Happens When You Use the Function or Element?	Use This Element Instead
%#eml	Still runs	%#codegen
eml.allowpcode	Still runs	coder.allowpcode
eml.ceval	Still runs	coder.ceval
eml.cstructname	Still runs	coder.cstructname
eml.extrinsic	Still runs	coder.extrinsic

Function or Element Name	What Happens When You Use the Function or Element?	Use This Element Instead
<code>eml.inline</code>	Still runs	<code>coder.inline</code>
<code>eml.nullcopy</code>	Still runs	<code>coder.nullcopy</code>
<code>eml.opaque</code>	Still runs	<code>coder.opaque</code>
<code>eml.ref</code>	Still runs	<code>coder.ref</code>
<code>eml.rref</code>	Still runs	<code>coder.rref</code>
<code>eml.target</code>	Still runs	<code>coder.target</code>
<code>eml.unroll</code>	Still runs	<code>coder.unroll</code>
<code>eml.varsizes</code>	Still runs	<code>coder.varsizes</code>
<code>eml.wref</code>	Still runs	<code>coder.wref</code>

Compatibility Summary for MATLAB Coder

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 V2.0 (R2011a)	See the Compatibility Considerations subheading for this new feature or change: <ul style="list-style-type: none">• “The codegen Function Has No Default Primary Function Input Type” on page 7• “The codegen Function Processes Compilation Options in a Different Order” on page 7