

# MATLAB Compiler Release Notes

---

The “MATLAB Compiler 4.0.1 Release Notes” on page 1-1 describe the changes introduced in the latest version of the MATLAB Compiler.

The MATLAB Compiler Release Notes also include information regarding MATLAB Compiler 4.0 including:

- “Introduction to the MATLAB Compiler 4.0” on page 2-2
- “Upgrading from an Earlier Release” on page 2-4
- “Known Software and Documentation Problems” on page 2-9

## **Printing the Release Notes**

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



## MATLAB Compiler 4.0.1 Release Notes

1

<b>MATLAB Compiler 4.0.1</b> .....	<b>1-2</b>
<b>Major Bug Fixes</b> .....	<b>1-3</b>
buildmcr Utility Works on Linux .....	1-3
extractCTF Utility Works on Linux .....	1-3
Files Added Using addpath Are Found .....	1-3
genpath Function Works in Deployed Applications .....	1-3
input Function Works in Deployed Applications .....	1-3
loadlibrary Function Works in Deployed Applications .....	1-3
MCRInstaller Works on Windows NT .....	1-4
Private Directories Work Properly .....	1-4
-V2.0 Option Removed .....	1-4
-w (Warning) Requires Option .....	1-4
Warning About set_param Removed .....	1-5
<b>Upgrading from an Earlier Release</b> .....	<b>1-6</b>
Upgrading from Version 4.0 .....	1-6
Upgrading from Earlier Versions .....	1-6
<b>Known Software and Documentation Problems</b> .....	<b>1-7</b>
Accessibility Support .....	1-7
Creating C++ Shared Libraries .....	1-7
Deprecated Functions .....	1-7
mccsavepath Is Not Available .....	1-7
MCRInstaller Generates setup.exe Error .....	1-8
Using Figure Windows in Shared Libraries .....	1-8
Using the Lcc Compiler .....	1-8
Using startup.m Files to Modify Directories .....	1-8
Using clear all Between Compilations .....	1-8
Using Microsoft Visual Studio with MATLAB Compiler 4 ....	1-9

<b>Introduction to the MATLAB Compiler 4.0</b> .....	<b>2-2</b>
Features .....	<b>2-2</b>
MATLAB Component Runtime .....	<b>2-3</b>
Component Technology File .....	<b>2-3</b>
<b>Upgrading from an Earlier Release</b> .....	<b>2-4</b>
Compiling MATLAB and Toolboxes .....	<b>2-4</b>
Differences Between Release 14 and Previous Versions of the MATLAB Compiler .....	<b>2-4</b>
<b>Known Software and Documentation Problems</b> .....	<b>2-9</b>
Deploying Applications and C++ Shared Libraries on Linux ..	<b>2-9</b>
Avoiding Warnings on Linux When Using Graphics .....	<b>2-9</b>
Using the MATLAB Compiler with the Mapping Toolbox ....	<b>2-9</b>
Documentation Updates .....	<b>2-10</b>

# MATLAB Compiler 4.0.1 Release Notes

---

<b>MATLAB Compiler 4.0.1</b> . . . . .	1-2
<b>Major Bug Fixes</b> . . . . .	1-3
<b>Upgrading from an Earlier Release</b> . . . . .	1-4
<b>Known Software and Documentation Problems</b> . . . . .	1-5

## **MATLAB Compiler 4.0.1**

MATLAB® Compiler 4.0.1 is a Web release for MathWorks Release 14. MATLAB Compiler 4.0 was included in Release 14. MATLAB Compiler 4.0.1 contains important bug fixes.

## Major Bug Fixes

MATLAB Compiler 4.0.1 includes bug fixes incorporated since Version 4.0. These bug fixes include the following.

### **buildmcr Utility Works on Linux**

You can use the `buildmcr` utility to create an MCRInstaller on Linux. For more information on `buildmcr`, see the MATLAB Compiler User's Guide.

### **extractCTF Utility Works on Linux**

The `extractCTF` utility has been moved to the `<matlabroot>/toolbox/compiler/deploy/glnx86` directory on Linux. You can use this utility to extract the CTF archive into the current working directory.

### **Files Added Using `addpath` Are Found**

The MATLAB Compiler can locate the files that are in directories that have been added to the MATLAB path using the `addpath` command.

### **genpath Function Works in Deployed Applications**

You can now compile and deploy an M-file that calls the `genpath` function.

### **input Function Works in Deployed Applications**

The `input` function works properly in deployed applications. It no longer waits for the input first and then displays the input prompt.

### **loadlibrary Function Works in Deployed Applications**

The MATLAB Compiler supports the compilation of the `loadlibrary` function.

---

**Note** MATLAB Compiler-generated libraries can not be loaded into the MATLAB workspace using the `loadlibrary` function.

---

## MCRInstaller Works on Windows NT

The MCRInstaller works properly on Windows NT.

## Private Directories Work Properly

Private directories deploy properly and work as expected in this release of the MATLAB Compiler.

## -V2.0 Option Removed

The -V2.0 option that was available in previous releases of the MATLAB Compiler has been obsolete and is no longer available. Using this option will give an error message.

## -w (Warning) Requires Option

You must specify an option (list, disable, enable, error) when using the -w option to display warnings. This table shows the valid forms.

Syntax	Description
-w list	Generates a table that maps <string> to warning message for use with enable, disable, and error.
-w enable	Enables complete warnings.
-w disable[:<string>]	Disables specific warning associated with <string>. Leave off the optional :<string> to apply the disable action to all warnings.
-w enable[:<string>]	Enables specific warning associated with <string>. Leave off the optional :<string> to apply the enable action to all warnings.
-w error[:<string>]	Treats specific warning associated with <string> as error. Leave off the optional :<string> to apply the error action to all warnings.



## **Warning About set\_param Removed**

You no longer get a warning/error message regarding SET\_PARAM not being found when using the MATLAB Compiler.

## **Upgrading from an Earlier Release**

### **Upgrading from Version 4.0**

MATLAB Compiler Version 4.0.1 is fully compatible with Version 4.0. Any files that were compilable with Version 4.0 will compile with Version 4.0.1.

### **Upgrading from Earlier Versions**

For information about upgrading from versions prior to Version 4.0, see “Upgrading from an Earlier Release” on page 2-4.

## Known Software and Documentation Problems

This section describes known software and documentation problems in Version 4.0.1.

### Accessibility Support

In the MATLAB Compiler Release Notes for Release 14, the section entitled “Accessibility Support for Deployed Applications” should not have been included. Note that this has been corrected in this version of the MATLAB Compiler release notes.

### Creating C++ Shared Libraries

You can use MATLAB Compiler Version 4 to create C++ shared libraries using the command

```
gcc -W cpplib:library_name -T link:lib mfile
```

This creates a C++ library wrapper and a header file. You can use the exported signature call in your external C++ code. Refer to the generated header file for the signature of the C++ interface to the M-functions that were compiled.

### Deprecated Functions

As of MATLAB Compiler Version 4, the set of imputed functions including `mbchar`, `mbcharscalar`, `mbcharvector`, `mbint`, `mbintscalar`, `mbintvector`, `mbreal`, `mbrealscalar`, `mbrealvector`, `mbscalar`, and `mbvector` are no longer supported by the Compiler. The new Compiler makes the need for these functions obsolete.

### mccsavepath Is Not Available

In previous versions of the MATLAB Compiler, you needed to use `mccsavepath` if the MATLAB Compiler was going to be invoked from a shell (DOS or UNIX) prompt. With this release of the Compiler, this step is no longer needed. Consequently, `mccsavepath` is no longer available with MATLAB Compiler Version 4.0.

## **MCRInstaller Generates setup.exe Error**

On Windows, if you run MCRInstaller in a directory that does not have write permission, you get a series of errors beginning with

```
error: cannot create Setup.exe
```

To work around this, use the `-d` option to specify a directory that does have write permission. For example,

```
MCRInstaller.exe -d D:\MCRInstall
```

## **Using Figure Windows in Shared Libraries**

If you create a shared library that displays a figure window and you call the shared library from a C stand-alone application, in some cases the figure window may flash and close with the warning message

```
Warning: Objects of graph2d.lineseries class exist - not clearing  
this class or any of its super-classes.
```

To avoid this problem, add `mclWaitForFiguresToDie(NULL)` to your code.

## **Using the Lcc Compiler**

If you compile a shared library using the Lcc compiler and use the `-d` option to direct the files to a specified directory, the `.lib` file does not get copied to the specified output directory.

## **Using startup.m Files to Modify Directories**

If you use a `startup.m` file to modify your directories, it can cause the Compiler not to find necessary files. To work around this

- 1 Use an `~isdeployed` check around any calls to `addpath`.
- 2 Use the Compiler `-a` option to add any additional files to the CTF archive.

## **Using clear all Between Compilations**

If you compile a MATLAB program into a stand-alone application, then issue a `clear all` command, and then recompile the same program, a pop-up window may display the message

```
Runtime error! Program:  
D:\Applications\MATLAB7\bin\win32\MATLAB.exe  
This application has requested the Runtime to terminate it in an  
unusual way. Please contact the application's support team for  
more information.
```

A workaround to this situation is to issue a different command, such as `inmem`, before calling `mcc` the second time.

## **Using Microsoft Visual Studio with MATLAB Compiler 4**

For more information on setting up Microsoft Visual Studio to work with the MATLAB Compiler, see Technical Solution 1-OQHIV at the Support area of the MathWorks Web site.



# MATLAB Compiler 4.0 Release Notes

---

<b>Introduction to the MATLAB Compiler 4.0 . . . . .</b>	<b>2-2</b>
<b>Upgrading from an Earlier Release . . . . .</b>	<b>2-4</b>
<b>Known Software and Documentation Problems . . . . .</b>	<b>2-9</b>

## Introduction to the MATLAB Compiler 4.0

MATLAB® Compiler Version 4 takes M-files as input and generates redistributable, stand-alone applications or software components. These resulting applications and components are platform specific.

### Features

#### Targets

The MATLAB Compiler can generate the following kinds of applications or components. None of these requires MATLAB on the end-user's system:

- Stand-alone applications
- C and C++ shared libraries (dynamically linked libraries, or DLLs, on Microsoft Windows)
- Excel add-ins; requires MATLAB Builder for Excel
- COM objects; requires MATLAB Builder for COM

#### Language Support

The MATLAB Compiler supports all the functionality of MATLAB, including objects. In addition, no special considerations are necessary for private and method functions; they are handled by the Compiler.

#### Improved C++ Interface

The C++ interface for Compiler-generated wrapper functions has been improved.

---

**Note** MATLAB Compiler 4 will not compile every toolbox, consequently, some MathWorks toolboxes will not be deployable. Portions of toolboxes may be nondeployable due to licensing restrictions (in general, compilation of toolbox graphical user interfaces will be restricted). The MATLAB Compiler will not compile Simulink®, Stateflow®, or products that require them. For more information regarding the compilability of toolboxes, see the MATLAB Compiler product page on the Web.

---



## **MATLAB Component Runtime**

MATLAB Compiler 4 uses the new MATLAB Component Runtime (MCR), which is a stand-alone set of shared libraries that enable the execution of compiled M-files, instead of the MATLAB C/C++ Math and Graphics Libraries. The MCR provides complete support for all features of the MATLAB language.

## **Component Technology File**

MATLAB Compiler 4 also uses a Component Technology File (CTF) archive to house the deployable package. All M-files are encrypted in the CTF archive using the Advanced Encryption Standard (AES) cryptosystem, where symmetric keys are protected by 1024-bit RSA keys.

## Upgrading from an Earlier Release

MATLAB Compiler 4 is compatible with previous releases of the Compiler. M-files that you compiled with a previous version of the MATLAB Compiler should compile with this version if your M-files contain only compilable Release 14 functions. There are no restrictions on the contents of your M-files other than compatibility with Release 14.

### Compiling MATLAB and Toolboxes

The MATLAB Compiler supports the full MATLAB language and almost all MATLAB based toolboxes. However, some limited MATLAB and toolbox functionality is not licensed for compilation:

- Functionality that cannot be called directly from the command line will not compile.
- Most of the prebuilt graphical user interfaces included in MATLAB and its companion toolboxes will not compile, for example, `sptool` from the Signal Processing Toolbox.
- Some toolboxes, such as the Symbolic Math Toolbox, will not compile.

The code generated by the MATLAB Compiler is not suitable for embedded applications.

To see an up-to-date list of noncompilable toolboxes and functionality, visit the MATLAB Compiler product page on the Web.

### Differences Between Release 14 and Previous Versions of the MATLAB Compiler

This section highlights significant differences between Compiler 4 and previous versions of the MATLAB Compiler.

MATLAB Compiler 4 is a deployment tool for creating software components and complete applications that can be distributed to other users. This version of the MATLAB Compiler fully supports all features of the MATLAB language including objects:

- Compiler 4 uses the new MATLAB Component Runtime (MCR), which is a stand-alone set of shared libraries that enable the execution of compiled M-files, instead of the MATLAB C/C++ Math and Graphics Libraries.

- Compiler 4 does not support the creation of MEX-files and Simulink S-functions from M-functions because features in MATLAB 7 make this functionality redundant. The MATLAB JIT makes compilation for speed obsolete, and the MATLAB pcode (preparsed code) function enables you to hide your proprietary algorithms.
- Compiler 4 is supported on Microsoft Windows and Linux only. We expect to add additional platforms in a future release.
- Compiler 4 does not include the MATLAB Add-in for Visual Studio.
- Compiler 4 does not speed up applications. There is no speed difference between a compiled application and running it in MATLAB. The compiled application will run as fast as MATLAB with the JIT Accelerator.
- MATLAB does not support the loading of MATLAB Compiler-generated libraries via the `loadlibrary` function.

### Wrapper Differences

- Compiler 4 only generates code for interface functions (wrappers), whereas previous versions generated code for the entire M-file. There are several differences to be aware of when calling Release 14 Compiler functions from C or C++:
  - Since Compiler 4 does not use the MATLAB C/C++ Math and Graphics Libraries, the various `m1f` functions previously available with the libraries are no longer available. Some of the Release 13 `m1f` functions have Release 14 equivalents in the MATLAB External Interface functions. For example, you can replace calls to `m1fScalar` with calls to `mxCreateDoubleScalar`.
  - The interface to the `m1f` functions generated by the Compiler from your M-file routines has changed. Unlike previous versions of the MATLAB Compiler, all the return values are passed as input to the function. The return value of these functions is `void`. See the MATLAB Compiler documentation for additional details.
  - The `initialize` routine now returns a status flag that can be used to test if the library was initialized properly.

---

**Note** These wrapper file differences only affect users who build libraries; they do not affect users who build executables.

---

## Deprecated Compiler Options

- Compiler 4 has deprecated options that involve code generation and formatting. The options below are no longer supported and will produce errors.

<b>Option</b>	<b>Description</b>
A	Code annotation
B pcode	Generate P-code
F	Format parameters
h	Helper functions
i	Include specified M-files
l	Line/file numbers (This option has changed and now means “library”.)
L	Target language
O	Optimized code
p	Generate C++ code (This option has changed and now means “add directory to compilation path in an order-sensitive context”.)
S	Macro to generate Simulink S-function
t	Translate M-code to C/C++ code
u	Specifies number of inputs for Simulink S-function
x	Macro to generate MEX-function
y	Specifies number of outputs for Simulink S-function

- Compiler 4 has deprecated some wrapper options and their associated bundle files. The following wrapper options and their associated bundle files are deprecated and are replaced by the new ones.

Wrapper Option/Bundle File	Replaced By
B csglcom	B ccom
B csglexcel	B cexcel
B csglsharedlib	B csharedlib
B cppsglcom	B cppcom
B cppsglexcel	B cppexcel
W comhg	W com
W excelhg	W excel
W libhg	W lib
W mainhg	W main

- You no longer need to use `-B sgl` and `-B sglcpp` to access Handle Graphics® functions. All compiled applications have access to graphics by default.

## New Compiler Options

Compiler 4 includes several new options.

Option	Description
a filename	Add filename to archive; specifies files to be directly added to the CTF archive.
l	Macro that generates a function library. (The meaning of this option has changed since Release 13.)
N	Clears the path of all but a minimal, required set of directories.

<b>Option</b>	<b>Description</b>
p <directory>	Add directory to compilation path in an order-sensitive context; requires -N option.
R -nojvm R -nojit	Run-time; provides MCR options; same as MATLAB startup options of the same name; only used with executable target.

## Known Software and Documentation Problems

### Deploying Applications and C++ Shared Libraries on Linux

Neither deploying applications nor building C++ shared libraries will work on Linux until you install a patch. See Solution 1-NZAD7 on our Support page for more information.

### Avoiding Warnings on Linux When Using Graphics

If your deployed application uses graphics, you may see this warning on Linux

```
Warning: could not access OpenGL library
```

To avoid this warning, add `${MCRROOT}/sys/opengl/lib/$ARCH` to the `$LD_LIBRARY_PATH`.

### Using the MATLAB Compiler with the Mapping Toolbox

When using the Mapping Toolbox, the MATLAB Compiler includes, by default, only the following low resolution MAT-files from the `mapdisp` directory: `coast.mat`, `worldlo.mat`, `oceanlo.mat`, and `usalo.mat`.

Functions that use other MAT-files from `mapdisp`, in particular the high resolution files, must include the MAT-file using the `-a Compiler` option.

For example, to compile `worldmap.m`, use

```
mcc -a mapdisp/worldhi.mat -m worldmap.m
```

To run `worldmap` as a stand-alone application and display a map of the world, first make sure `worldmap` is on the path or in the local directory. Then, from the command prompt (shell prompt on Linux, DOS prompt on Windows), type the application name (preceded by `./` on Linux if the local directory containing `worldmap` is not on the path).

```
worldmap world
```

To compile `usamap.m` use

```
mcc -a mapdisp/usahi.mat -m usamap.m
```

To display the state of Missouri, use the command

```
usamap missouri
```

### **Documentation Updates**

From time to time, the MATLAB Compiler documentation at the MathWorks Web site may be updated and additional examples added. Also, code examples may be added to the File Exchange section of MATLAB Central. You can check for these examples at the File Exchange.