

MATLAB[®] Compiler Release Notes

Summary by Version	1
About Release Notes	1
Version 4.5 (R2006b) MATLAB Compiler	4
Support for Java Components	4
Deployment Tool: Graphical User Interface	4
Version 4.4 (R2006a) MATLAB Compiler	5
Support for .NET Components	5
Support for Microsoft Visual C/C++ Version 8.0	5
HP-UX No Longer Supported	6
Version 4.3 (R14SP3) MATLAB Compiler	7
Platform Support	7
Version 4.2 (R14SP2) MATLAB Compiler	8
Platform Support	8
Version 4.1.1 (R14SP1+) MATLAB Compiler	9
Fixed Bugs	9
Version 4.1 (R14SP1) MATLAB Compiler	12
Solaris Support	12
Installing a New Version of the MCR on Windows	14
Fixed Bugs	14
Version 4.0.1 (R14+) MATLAB Compiler	16
Fixed Bugs	16
Version 4.0 (R14) MATLAB Compiler	19
Targets	19
Language Support	20
Improved C++ Interface	20
MATLAB Component Runtime	20
Component Technology File	20
Compatibility Considerations	20

Compatibility Summary for MATLAB Compiler 25

Summary by Version

This table provides quick access to what's new in each version. For clarification, see "About 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 V4.5 (R2006b)	Yes Details	No	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation
V4.4 (R2006a)	Yes Details	No	Bug Reports Includes fixes	No
V4.3 (R14SP3)	Yes Details	No	Bug Reports Includes fixes	No
V4.2 (R14SP2)	Yes Details	No	No bug fixes	No
V4.1.1 (R14SP1+)	No	No	Fixed bugs	No
V4.1 (R14SP1)	Yes Details	No	Fixed bugs	No
V4.0.1 (R14+)	No	No	Fixed bugs	No
V4.0 (R14)	Yes Details	Yes Summary	No bug fixes	V3.0 product documentation

About Release Notes

Use release notes when upgrading to a newer version to learn about new features and changes, and the potential impact on your existing files and practices. Release notes are also beneficial if you use or support multiple versions.

If you are not upgrading from the most recent previous version, review 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 New Features and Changes, Version Compatibility Considerations, and Bug Reports for V1.1 and V1.2.

New Features and Changes

These include

- New functionality
- Changes to existing functionality
- Changes to system requirements (complete system requirements for the current version are at the MathWorks Web site)
- Any version compatibility considerations associated with each new feature or change

Version Compatibility Considerations

When a new feature or change introduces a known incompatibility between versions, its description includes a **Compatibility Considerations** subsection that details the impact. For a list of all new features and changes that have compatibility impact, see the “Compatibility Summary for MATLAB Compiler” on page 25.

Compatibility issues that become known after the product has been released are added to Bug Reports at the MathWorks Web site. Because bug fixes can sometimes result in incompatibilities, also review fixed bugs in Bug Reports for any compatibility impact.

Fixed Bugs and Known Problems

MathWorks Bug Reports is a user-searchable database of known problems, workarounds, and fixes. The MathWorks updates the Bug Reports database as new problems and resolutions become known, so check it as needed for the latest information.

Access Bug Reports at the MathWorks Web site using your MathWorks Account. If you are not logged in to your MathWorks Account when you link to Bug Reports, you are prompted to log in or create an account. You then can view bug fixes and known problems for R14SP2 and more recent releases.

The Bug Reports database was introduced for R14SP2 and does not include information for prior releases. You can access a list of bug fixes made in prior versions via the links in the summary table.

Related Documentation at Web Site

Printable Release Notes (PDF). You can print release notes from the PDF version, located at the MathWorks Web site. The PDF version does not support links to other documents or to the Web site, such as to Bug Reports. Use the browser-based version of release notes for access to all information.

Product Documentation. At the MathWorks Web site, you can access complete product documentation for the current version and some previous versions, as noted in the summary table.

Version 4.5 (R2006b) MATLAB Compiler

This table summarizes what's new in Version 4.5 (R2006b):

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

New features and changes introduced in this version are

- “Support for Java Components” on page 4
- “Deployment Tool: Graphical User Interface” on page 4

Support for Java Components

MATLAB Compiler, in conjunction with the optional MATLAB Builder for Java, lets you convert MATLAB M-files into Java components.

Deployment Tool: Graphical User Interface

As an alternative to the `mcc` command to invoke MATLAB Compiler, you can invoke the graphical user interface for MATLAB Compiler by issuing the following command at the MATLAB prompt:

```
deploytool
```

Use the Deployment Tool to perform the tasks shown in the following conceptual illustration:



Version 4.4 (R2006a) MATLAB Compiler

This table summarizes what's new in Version 4.4 (R2006a):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes	Printable Release Notes: PDF Version 4.4 product documentation

New features and changes introduced in this version are

- “Support for .NET Components” on page 5
- “Support for Microsoft Visual C/C++ Version 8.0” on page 5
- “HP-UX No Longer Supported” on page 6

This release provides support for a new target, .NET components. With the optional MATLAB Builder for .NET product, you can create both COM components that can be used in native code applications and .NET components that can be used in managed code applications.

Support for .NET Components

The MATLAB Compiler, in conjunction with the optional MATLAB Builder for .NET, lets you convert MATLAB M-files into .NET components that are accessible from any Common Language Specification (CLS)-compliant client code. In addition, you can build Common Object Model (COM) components that are accessible from Visual Basic, C/C++, Microsoft Excel, or any other COM client.

Support for Microsoft Visual C/C++ Version 8.0

The MATLAB Compiler supports Microsoft Visual C/C++ Version 8.0 as a third-party compiler. However, if you use Microsoft Visual C/C++ Version 8.0 to generate applications/components, you must have the Microsoft Visual

Studio 2005 run-time libraries available on the computer used for deployment. If you distribute your applications/components, you must make sure that any machine used for deployment has these run-time libraries installed. For more information, see solution 1-2223MW.

HP-UX No Longer Supported

MATLAB Compiler 4.4 (R2006a) does not support HP-UX; MATLAB Compiler 4.3 (R14SP3) was the last release to support HP-UX.

Version 4.3 (R14SP3) MATLAB Compiler

This table summarizes what's new in Version 4.3 (R14SP3):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Bug Reports Includes fixes	No

New features and changes introduced in this version are

- “Platform Support” on page 7

Platform Support

You can use MATLAB Compiler 4.3 on the following supported systems to create redistributable, stand-alone applications or software components. These applications or components can then be deployed to other systems with the same operating system.

Supported Platforms

MATLAB Compiler 4.3 is supported on these platforms:

- Windows
- Linux
- Solaris
- HP-UX
- Linux x86-64
- Mac OS X

The MATLAB Compiler documentation has been updated to include configuration information for all the supported platforms.

Version 4.2 (R14SP2) MATLAB Compiler

This table summarizes what's new in Version 4.2 (R14SP2):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	No bug fixes	No

New features and changes introduced in this version are

- “Platform Support” on page 8

Platform Support

You can use MATLAB Compiler 4.2 on the following supported systems to create redistributable, stand-alone applications or software components. These applications or components can then be deployed to other systems with the same operating system.

Supported Platforms

MATLAB Compiler 4.2 is supported on these platforms:

- Windows
- Linux
- Solaris
- HP-UX
- Linux x86-64

The MATLAB Compiler documentation has been updated to include configuration information for all the supported platforms.

Version 4.1.1 (R14SP1+) MATLAB Compiler

This table summarizes what's new in Version 4.1.1 (R14SP1+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
No	No	Fixed bugs Details below	No

Fixed Bugs

MATLAB Compiler 4.1.1 includes bug fixes incorporated since Version 4.1 (Release 14SP1). These bug fixes include the following.

buildmcr Function Supports Use of ~ in Filename on UNIX

With this release, you can use the ~ character as part of the destination filename or directory on UNIX systems. For example:

```
zipfile = buildmcr('~\mcr')
```

Before this release, using the ~ character as part of the destination filename or directory on UNIX systems caused the error

```
Error opening '~\mcr\MCRInstaller.zip'
```

C++ Shared Libraries Fully Supported with Borland Compilers

With this release, you can use supported versions of the Borland Compiler (see the Supported Compilers list) to build C++ shared libraries. In previous releases, Borland compilers could build C shared libraries, but building C++ shared libraries resulted in link-time errors. These errors have been resolved.

Excluded Functions List Stored in a Log File

If you use the -v option to generate the verbose output of the compilation steps, the MATLAB Compiler creates a log file called `mccExcludedFiles.log`. This file contains, if any, the list of functions that have been excluded from the compiled application. This list does not include the core MATLAB functions

that are unsupported. Refer to the Unsupported Functions table in the MATLAB Compiler User's Guide.

loadlibrary Function Works in Compiled Mode

The `loadlibrary` function did not work in certain cases in compiled mode. This problem has been resolved with this release of the MATLAB Compiler. Note that the libraries created using the MATLAB Compiler cannot be loaded into the MATLAB workspace using the `loadlibrary` function.

mclcppMlfEval No Longer Multiply Defined

In R14, before Service Pack 1, C++ applications that linked against two or more MATLAB Compiler generated C++ shared libraries would get a link-time error indicating that `mclcppMlfEval` was multiply defined. In R14 Service Pack 1, this function, which is found in `extern/include/mclcppclass.h`, has been declared `inline`; this resolves the problem.

MCR Started by MATLAB Compiler Returns true for isdeployed

You can use the `isdeployed` function to determine if an MCR was started by the MATLAB Compiler. The `mcc` command starts a new MCR. The `isdeployed` function will return `true` for this MCR, thus replicating the deployed environment.

Misleading -e Error Is Resolved

On Windows, when `mbuild` (which is called by `mcc`) fails, you will no longer get the misleading error about `-e` not being an internal or an external command.

Reduced Compilation Time and Size of the CTF File

In certain scenarios, the compilation time is reduced and the size of the CTF file is smaller.

Translation Files for Various Toolboxes Included in the CTF File

When various toolboxes are compiled, the translation files are now included in the CTF archive. This will let Japanese users see the translated output for certain toolbox messages.

web Function Compiles Successfully

In MATLAB Compiler 4.0 (R14), M-files that used the `web` function would compile (apparently without error), but fail to execute at run time. This occurred because a function that the `web` command depends on, `ibrowse`, was not being shipped. `ibrowse` is now shipped with the MCR.

Version 4.1 (R14SP1) MATLAB Compiler

This table summarizes what's new in Version 4.1 (R14SP1):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Yes Details below	No	Fixed bugs Details below	No

New features and changes introduced in this version are

- “Solaris Support” on page 12
- “Installing a New Version of the MCR on Windows” on page 14
- “Fixed Bugs” on page 14

Solaris Support

You can use MATLAB Compiler 4.1 on supported Solaris systems to create redistributable, stand-alone applications or software components. These applications or components can then be deployed to other Solaris systems.

We encourage you to use this beta version for Solaris and we appreciate your feedback. We expect to release a fully qualified version of the MATLAB Compiler for Solaris as soon as possible.

buildmcr on Solaris

The `buildmcr` command that ships with MATLAB Compiler 4.1 may fail on some Solaris systems. You must upgrade your `buildmcr` command if you see the error message

```
"Could not redirect CTFARCHIVER output to the MATLAB Desktop"
```

We recommend that all Solaris users upgrade their `buildmcr` command. For instructions on how to upgrade, see our Support site and search for “Solaris BUILDMCR.” This upgrade is not necessary for Windows or Linux users.

Modifying the Dynamic Library Path on Solaris

Development Machine. To run a MATLAB Compiler-generated application on your development machine, add the following directories to your dynamic library path.

Note For readability, the following command appears on separate lines, but you must enter it all on one line.

```
setenv LD_LIBRARY_PATH
/usr/lib/lwp:
<matlabroot>/bin/sol2:
<matlabroot>/sys/os/sol2:
<matlabroot>/sys/java/jre/sol2/jre1.4.2/lib/sparc/native_threads:
<matlabroot>/sys/java/jre/sol2/jre1.4.2/lib/sparc/client:
<matlabroot>/sys/java/jre/sol2/jre1.4.2/lib/sparc:
<matlabroot>/sys/opengl/lib/sol2:${LD_LIBRARY_PATH}
setenv XAPPLRESDIR <matlabroot>/X11/app-defaults
```

Target Machine. To run a MATLAB Compiler-generated application on a target machine (a machine without MATLAB that has the MCR installed), add the following directories to your dynamic library path.

Note For readability, the following command appears on separate lines, but you must enter it all on one line.

```
setenv LD_LIBRARY_PATH
/usr/lib/lwp:
<mcr_root>/runtime/sol2:
<mcr_root>/sys/os/sol2:
<mcr_root>/sys/java/jre/sol2/jre1.4.2/lib/sparc/native_threads:
<mcr_root>/sys/java/jre/sol2/jre1.4.2/lib/sparc/client:
<mcr_root>/sys/java/jre/sol2/jre1.4.2/lib/sparc:
<mcr_root>/sys/opengl/lib/sol2:${LD_LIBRARY_PATH}
setenv XAPPLRESDIR <mcr_root>/X11/app-defaults
```

<mcr_root> is the directory where the MCR is installed.

Redistribution of the Sun sunperf Library

On Solaris, the only compiler that the MATLAB Compiler supports is the Sun compiler. If you are a Solaris user who is licensed to use the Sun compiler, you can redistribute the Sun sunperf and dependent libraries, including BLAS, to your end users. For more information, consult your Sun license agreement.

Installing a New Version of the MCR on Windows

The MCRInstaller now supports the installation of multiple versions of the MCR on a target machine. This allows applications compiled with different versions of the MCR to execute side by side on the same machine.

If multiple versions of the MCR are not desired on the target machine, you can run **Add or Remove Programs** from the Control Panel to remove any of the previous versions. This can be done either before or after installation of a more recent version of the MCR, as versions can be installed or removed in any order.

MCR Version 7.0 Restriction

The only caveat to installing a new version of the MCR is that you cannot automatically install version 7.0 in the same directory as a previously installed version of the MCR. This restriction applies only to version 7.0 of the MCRInstaller. If you must install version 7.0 in the same directory as a previous version, you should first run the version 7.0 MCRInstaller and select that directory. After the installation completes, you should manually add the MCR run-time directory <TargetDir>\v70\runtime\win32 to the system path.

Fixed Bugs

MATLAB Compiler 4.1 includes bug fixes incorporated since Version 4.0.1, which was released via the Web in July 2004. These bug fixes include the following.

Calls to Java Code Works in Deployed Applications

In MATLAB Compiler 4.0 (Release 14), users were not able to deploy MATLAB applications that used certain Java files. For example, you could not deploy the Database Toolbox functions. This bug has been fixed in this release.

Deploying Applications to Non-U.S. Locale Systems

The SET_PARAM error that was generated when an application was deployed to a non-U.S. locale system has been fixed.

Printing Figure Windows

On Windows, the problem of printing figure windows has been resolved and should work as expected.

Using pause in Applications Generated by MATLAB Compiler

In Release 14 on Windows, MATLAB Compiler generated applications that used the pause command would occasionally hang. This was more likely to happen if you called pause frequently or you used very small timeout values. This problem has been resolved.

Version 4.0.1 (R14+) MATLAB Compiler

This table summarizes what's new in Version 4.0.1 (R14+):

New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
No	No	Fixed bugs Details below	No

Fixed Bugs

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 cannot 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 obsoleted 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`, or `error`) when using the `-w` option to display warnings. This table shows the valid forms.

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

Syntax	Description
-w enable[:string]	Enables specific warning associated with <i>string</i> . Leave off the optional <i>:string</i> to apply the enable action to all warnings.
-w error[:string]	Treats specific warning associated with <i>string</i> as error. Leave off the optional <i>:string</i> 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.

Version 4.0 (R14) MATLAB Compiler

This table summarizes what's new in Version 4.0 (R14):

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.	No bug fixes	V3.0 product documentation

New features and changes introduced in this version are

- “Targets” on page 19
- “Language Support” on page 20
- “Improved C++ Interface” on page 20
- “MATLAB Component Runtime” on page 20
- “Component Technology File” on page 20
- “Compatibility Considerations” on page 20

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 MATLAB Compiler.

Improved C++ Interface

The C++ interface for MATLAB 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.

Compatibility Considerations

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.
- Compiler 4 does not support the set of imputed functions including `mbchar`, `mbcharscalar`, `mbcharvector`, `mbint`, `mbintscalar`, `mbintvector`, `mbreal`, `mbrealscalar`, `mbrealvector`, `mbscalar`, and `mbvector`. Compiler 4 makes the need for these functions obsolete.
- Compiler 4 eliminates the need to use `mccsavepath` to invoke the MATLAB Compiler from a shell (DOS or UNIX) prompt. Consequently, `mccsavepath` is no longer available with Compiler 4.

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.

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

Compatibility Summary for MATLAB Compiler

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 V4.5 (R2006b)	None
V4.4 (R2006a)	None
V4.3 (R14SP3)	None
V4.2 (R14SP2)	None
V4.1.1 (R14SP1+)	None
V4.1 (R14SP1)	None
V4.0.1 (R14+)	None
V4.0 (R14)	<p>See the Compatibility Considerations subheading for each of these new features and changes:</p> <ul style="list-style-type: none"> • “Compiling MATLAB and Toolboxes” on page 21 • “Differences Between Release 14 and Previous Versions of the MATLAB Compiler” on page 21