

MATLAB[®] Compiler Release Notes

The “MATLAB Compiler 4.2 Release Notes” on page 1-1 describe the changes introduced in the latest version of the MATLAB Compiler. The following topics are discussed in these Release Notes:

- “MATLAB Compiler 4.2” on page 1-2
- “Major Bug Fixes” on page 1-3
- “Upgrading from an Earlier Release” on page 1-4
- “Known Software and Documentation Problems” on page 1-5

The MATLAB Compiler Release Notes also provide information about the earlier versions of the product.

- “MATLAB Compiler 4.1.1” on page 2-2
- “MATLAB Compiler 4.1 Release Notes” on page 3-1
- “MATLAB Compiler 4.0.1 Release Notes” on page 4-1
- “MATLAB Compiler 4.0 Release Notes” on page 5-1

Printing the Release Notes

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

MATLAB Compiler 4.2 Release Notes

1

MATLAB Compiler 4.2	1-2
Platform Support	1-2
Major Bug Fixes	1-3
Mixed Case Function Names on Windows	1-3
Upgrading from an Earlier Release	1-4
Upgrading from Version 4.0, 4.0.1, 4.1, and 4.1.1	1-4
Upgrading from Earlier Versions	1-4
Known Software and Documentation Problems	1-5
Borland C++ Compiler Limits Indexing to Arrays of 20 Dimensions or Less	1-5
Cabinet File Error During MCRInstaller Installation	1-5
Calling Compiler-Generated Components from Java Native Interface	1-5
Using the Lcc Compiler	1-5
Using Microsoft Visual Studio with MATLAB Compiler 4	1-6
Using startup.m Files to Modify Directories	1-6
Eliminating Incorrect Message Catalog Warning on HP-UX ..	1-6

MATLAB Compiler 4.1.1 Release Notes

2

MATLAB Compiler 4.1.1	2-2
Major Bug Fixes	2-3
buildmcr Function Supports Use of ~ in Filename on UNIX ..	2-3
C++ Shared Libraries Fully Supported with Borland Compilers	2-3
Excluded Functions List Stored in Log File	2-3

loadlibrary Function Works in Compiled Mode	2-3
mclcppMlfEval No Longer Multiply-Defined	2-4
MCR Started by MATLAB Compiler Returns true for isdeployed	2-4
Misleading -e Error Is Resolved	2-4
Reduced Compilation Time and Size of the CTF File	2-4
Translation Files for Various Toolboxes Included in the CTF File	2-4
web Function Compiles Successfully	2-4
Upgrading from an Earlier Release	2-5
Upgrading from Version 4.0, 4.0.1, and 4.1	2-5
Upgrading from Earlier Versions	2-5

MATLAB Compiler 4.1 Release Notes

3

MATLAB Compiler 4.1	3-2
Solaris Support	3-2
Installing a New Version of the MCR on Windows	3-3
Major Bug Fixes	3-5
Calls to Java Code Works in Deployed Applications	3-5
Deploying Applications to Non-US Locale Systems	3-5
Printing Figure Windows	3-5
Using pause in MATLAB Compiler-Generated Applications	3-5
Upgrading from an Earlier Release	3-6
Upgrading from Version 4.0 and 4.0.1	3-6
Upgrading from Earlier Versions	3-6

MATLAB Compiler 4.0.1 Release Notes

4

MATLAB Compiler 4.0.1	4-2
Major Bug Fixes	4-3
buildmcr Utility Works on Linux	4-3
extractCTF Utility Works on Linux	4-3
Files Added Using addpath Are Found	4-3
genpath Function Works in Deployed Applications	4-3
input Function Works in Deployed Applications	4-3
loadlibrary Function Works in Deployed Applications	4-3
MCRInstaller Works on Windows NT	4-4
Private Directories Work Properly	4-4
-V2.0 Option Removed	4-4
-w (Warning) Requires Option	4-4
Warning About set_param Removed	4-5
Upgrading from an Earlier Release	4-6
Upgrading from Version 4.0	4-6
Upgrading from Earlier Versions	4-6

MATLAB Compiler 4.0 Release Notes

5

Introduction to the MATLAB Compiler 4.0	5-2
Features	5-2
MATLAB Component Runtime	5-3
Component Technology File	5-3
Upgrading from an Earlier Release	5-4
Compiling MATLAB and Toolboxes	5-4
Differences Between Release 14 and Previous Versions of the MATLAB Compiler	5-4

MATLAB Compiler 4.2 Release Notes

MATLAB Compiler 4.2	1-2
Major Bug Fixes	1-3
Upgrading from an Earlier Release	1-4
Known Software and Documentation Problems	1-5

MATLAB Compiler 4.2

MATLAB® Compiler 4.2 is part of Release 14SP2, and requires MATLAB 7.0.4. MATLAB Compiler 4.1.1 was a Web release for Release 14SP1.

MATLAB Compiler 4.2 includes support for several new platforms as well as bug fixes incorporated since Version 4.1.1.

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:

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

The MATLAB Compiler User's Guide has been updated to include configuration information for all the supported platforms.

Major Bug Fixes

MATLAB Compiler 4.2 includes bug fixes incorporated since Version 4.1.1. MATLAB Compiler 4.2 requires MATLAB 7.0.4.

Mixed Case Function Names on Windows

This version of the MATLAB Compiler preserves the case of user function names on Windows. In previous versions, if you compiled a function called `MixedCaseFcn`, the generated code included the line (some arguments omitted)

```
mclFeval(..., "mixedcasefcn", ...);
```

This caused a verbose, run-time warning about an inexact match for the function on Windows. MATLAB Compiler 4.2 now generates the code as

```
mclFeval(..., "MixedCaseFcn", ...);
```

In addition to handling user function names, this version of the MATLAB Compiler makes the case of generated `m1x` functions more closely match the case of the M-functions they invoke. In previous versions, the associated mixed case function for this example would have only capitalized the initial letter of the function name

```
m1xMixedcasefcn
```

MATLAB Compiler 4.2 now matches the case exactly and generates

```
m1xMixedCaseFcn
```

The rule for determining how to capitalize function names is:

- The 4th letter (first letter after the `m1x` prefix) will always be capitalized.
- The rest of the casing will match exactly the name of the M-file.

Note The code generation on UNIX already matches the case properly.

Upgrading from an Earlier Release

Upgrading from Version 4.0, 4.0.1, 4.1, and 4.1.1

MATLAB Compiler Version 4.2 is fully compatible with Versions 4.0, 4.0.1, 4.1, and 4.1.1. Any files that were compilable with these versions will compile with Version 4.2.

Upgrading from Earlier Versions

For information about upgrading from versions prior to Version 4.0, see “Upgrading from an Earlier Release” on page 5-4.

Known Software and Documentation Problems

This section describes known software and documentation problems in Version 4.2.

Note From time to time, solutions for known problems are posted at our Support site.

Borland C++ Compiler Limits Indexing to Arrays of 20 Dimensions or Less

Due to internal limitations of the Borland C++ compiler, the `mwArray` class supports indexing operations on arrays of 20 or fewer dimensions (with other manufacturers C++ compilers, this limit increases to 32 dimensions).

Cabinet File Error During MCRInstaller Installation

This Windows error during the installation of the `MCRInstaller` may occur if you manually run the `setup.exe` file before the `MCRInstaller` calls it. To avoid this problem, reinstall `MCRInstaller` by running `MCRInstaller.exe`. The `MCRInstaller` unzips several files including `setup.exe`. The `MCRInstaller` will automatically execute `setup.exe`. Do *not* manually execute this file by double-clicking on it.

Calling Compiler-Generated Components from Java Native Interface

You cannot invoke Compiler-generated DLLs or shared libraries from the Java Native Interface (JNI). However, your Compiler-generated components can invoke Java.

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 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.

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.

Eliminating Incorrect Message Catalog Warning on HP-UX

When you compile an M-file on HP-UX, you may see a series of warnings including

```
cc: 487: warning : Possibly incorrect message catalog.
```

To eliminate the warnings, set `NLSPATH` to

```
/opt/ansic/lib/nls/msg/%L/%N.cat:/opt/ansic/lib/nls/msg/C/%N.cat
```

MATLAB Compiler 4.1.1 Release Notes

MATLAB Compiler 4.1.1	2-2
Major Bug Fixes	2-3
Upgrading from an Earlier Release	2-5

MATLAB Compiler 4.1.1

MATLAB® Compiler 4.1.1 is a Web release for MathWorks Release 14SP1. MATLAB Compiler 4.1 was included in Release 14SP1. MATLAB Compiler 4.1.1 contains important bug fixes.

Major Bug Fixes

MATLAB Compiler 4.1.1 includes bug fixes incorporated since Version 4.1, which was released in 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')
```

Previous to 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 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 MATLAB Compiler User's Guide for the unsupported functions.

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.

mclcppMlfFeval 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 `mclcppMlfFeval` 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 enable Japanese users to 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.

Upgrading from an Earlier Release

Upgrading from Version 4.0, 4.0.1, and 4.1

MATLAB Compiler Version 4.1.1 is fully compatible with Versions 4.0, 4.0.1, and 4.1. Any files that were compilable with these versions will compile with Version 4.1.1.

Upgrading from Earlier Versions

For information about upgrading from versions prior to Version 4.0, see “Upgrading from an Earlier Release” on page 5-4.

MATLAB Compiler 4.1 Release Notes

MATLAB Compiler 4.1	3-2
Major Bug Fixes	3-5
Upgrading from an Earlier Release	3-6

MATLAB Compiler 4.1

MATLAB® Compiler 4.1 adds initial beta-level support for Solaris systems and contains important bug fixes. MATLAB Compiler 4.1 is part of Release 14SP1, and requires MATLAB 7.0.1.

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 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 Sun's 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 Sun's 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.

Major Bug Fixes

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-US Locale Systems

The SET_PARAM error that was generated when an application was deployed to a non-US 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 MATLAB Compiler-Generated Applications

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.

Upgrading from an Earlier Release

Upgrading from Version 4.0 and 4.0.1

MATLAB Compiler Version 4.1 is fully compatible with Versions 4.0 and 4.0.1. Any files that were compilable with Versions 4.0 and 4.0.1 will compile with Version 4.1.

Upgrading from Earlier Versions

For information about upgrading from versions prior to Version 4.0, see “Upgrading from an Earlier Release” on page 5-4.

MATLAB Compiler 4.0.1 Release Notes

MATLAB Compiler 4.0.1	4-2
Major Bug Fixes	4-3
Upgrading from an Earlier Release	4-6

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 5-4.

MATLAB Compiler 4.0 Release Notes

Introduction to the MATLAB Compiler 4.0	5-2
Upgrading from an Earlier Release	5-4

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.
- 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

Option	Description
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.)

Option	Description
N	Clears the path of all but a minimal, required set of directories.
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.