

MATLAB[®] Compiler[™] Release Notes

How to Contact The MathWorks



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



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



508-647-7000 (Phone)



508-647-7001 (Fax)



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

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

MATLAB® Compiler™ Release Notes

© COPYRIGHT 2004–2008 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

Summary by Version	1
Version 4.9 (R2008b) MATLAB® Compiler Software ...	4
Version 4.8 (R2008a) MATLAB® Compiler Software ...	7
Version 4.7 (R2007b) MATLAB® Compiler Software ...	11
Version 4.6 (R2007a) MATLAB® Compiler Software ...	14
Version 4.5 (R2006b) MATLAB® Compiler Software ...	17
Version 4.4 (R2006a) MATLAB® Compiler Software ...	18
Version 4.3 (R14SP3) MATLAB® Compiler Software ...	20
Version 4.2 (R14SP2) MATLAB® Compiler Software ...	21
Version 4.1.1 (R14SP1+) MATLAB® Compiler	22
Version 4.1 (R14SP1) MATLAB® Compiler Software ...	25
Version 4.0.1 (R14+) MATLAB® Compiler Software	29
Version 4.0 (R14) MATLAB® Compiler Software	32
Compatibility Summary for MATLAB® Compiler Software	38

Summary by Version

This table provides quick access to what's new in each version. For clarification, see “Using Release Notes” on page 2.

Version (Release)	New Features and Changes	Version Compatibility Considerations	Fixed Bugs and Known Problems	Related Documentation at Web Site
Latest Version V4.9 (R2008b)	Yes Details	Yes Summary	Bug Reports Includes fixes	Printable Release Notes: PDF Current product documentation
V4.8 (R2008a)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V4.7 (R2007b)	Yes Details	Yes Summary	Bug Reports Includes fixes	No
V4.6 (R2007a)	Yes Details	No	Bug Reports Includes fixes	No
V4.5 (R2006b)	Yes Details	No	Bug Reports Includes fixes	No
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	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	No

Using Release Notes

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

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

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

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

What's in the Release Notes

New Features and Changes

- New functionality
- Changes to existing functionality

Version Compatibility Considerations

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

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

Fixed Bugs and Known Problems

The MathWorks offers a user-searchable Bug Reports database so you can view Bug Reports. The development team updates this database at release time and as more information becomes available. This includes provisions for any known workarounds or file replacements. Information is available for bugs existing in or fixed in Release 14SP2 or later. Information is not available for all bugs in earlier releases.

Access Bug Reports using your MathWorks Account.

Version 4.9 (R2008b) MATLAB Compiler Software

This table summarizes what's new in Version 4.9 (R2008b):

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

New features and changes introduced in this version are:

- “Applications Created with Parallel Computing Toolbox Now Able to Be Compiled” on page 4
- “Data Sharing Between M-Code, MCR Instance, and Wrapper Code Available” on page 5
- “64-Bit Addressing of mxArray Is New Default, 32-Bit Addressing No Longer Supported” on page 5
- “Microsoft® Visual Studio 2008, 32-Bit and 64-Bit Editions, Now Supported” on page 5
- “For More Information About Compilers Supported by MATLAB” on page 5
- “Warning Results When Running Figure-Generating Applications or Printing With -nojvm Flag” on page 6
- “Same Name Class Objects Shared Between MCR Instances Will Not Work Correctly ” on page 6

Applications Created with Parallel Computing Toolbox Now Able to Be Compiled

MATLAB applications that make use of the Parallel Computing Toolbox™ are now able to be compiled. Resulting executables and components can scale to multicore and multiprocessing environments using the MATLAB Distributed Computing Server. For more information, see ““Improving Data Access Using

the MCR User Data Interface” in the MATLAB® Compiler™ User’s Guide for more information.

Data Sharing Between M-Code, MCR Instance, and Wrapper Code Available

It is now possible to share data between an MCR instance, the M-code running on that MCR, and the wrapper code that created the MCR, by implementing two M-functions and four external C-functions in a new API that may be called from within deployed application wrapper code. Using these functions may potentially improve performance and promote efficient use of computing resources. See “Improving Data Access Using the MCR User Data Interface” in the MATLAB Compiler User’s Guide for more information.

64-Bit Addressing of mxArray Is New Default, 32-Bit Addressing No Longer Supported

The `MX_COMPAT_32_OFF` variable, which allowed 32-bit addressing of `mxArrays`, is no longer supported. As in R2008a, the default is 64-bit (large array support). See “Addressing mxArray Above the 2 GB Limit” in the MATLAB Compiler User’s Guide for more information.

Microsoft Visual Studio 2008, 32-Bit and 64-Bit Editions, Now Supported

Support is now available for Microsoft® Visual Studio® 2008, 32-bit and 64-bit editions.

For More Information About Compilers Supported by MATLAB

For more information about changes in compiler-related software and interfaces supported by MATLAB, see “Changes to Compiler Support” in the MATLAB Compiler User’s Guide for more information.

Warning Results When Running Figure-Generating Applications or Printing With -nojvm Flag

As of R2008b, running a figure-generating application, or printing, with the `-nojvm` option results in a warning message. In some cases, figure rendering may succeed, and in other cases it may not. In a future release, the `-nojvm` option will no longer support figure-generating or printing and will be removed.

Same Name Class Objects Shared Between MCR Instances Will Not Work Correctly

If the same class name is used in two or more separate MCR instances within the same process, the object will not work correctly. This bug impacts all component based targets: C/C++ shared libraries, COM/.Net targets, and Java targets. This bug does not affect standalone executables. This is due to a bug in the MATLAB Object System and is being addressed.

Version 4.8 (R2008a) MATLAB Compiler Software

This table summarizes what's new in Version 4.8 (R2008a):

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

New features and changes introduced in this version are:

- “C++ API Now Supports 64-Bit Default” on page 7
- “CTF Archives Now Embedded in a Single Binary Executable for Convenient Deployment” on page 8
- “Support Added for Microsoft® Visual Studio 2008 Compiler” on page 8
- “Replacement of the mclmcr Header File” on page 8
- “Run-Time Libraries Required for Applications Built with Microsoft® Visual Studio 2008 Compiler” on page 8
- “Large Array Support Now Default for C and C++ Code” on page 9
- “Support Dropped for Borland Compilers” on page 9
- “MATLAB Component Runtime Renamed” on page 9
- “MATLAB Application Deployment Web Example Guide Available” on page 9
- “For More Information About Compilers Supported by MATLAB” on page 10

C++ API Now Supports 64-Bit Default

In R2007b, the `MX_COMPAT_32_OFF` variable allowed for 64-bit support. In this release, 64-bit is the default setting. If you still require explicit 32-bit execution, see “Addressing mxArray Above the 2 GB Limit” in the MATLAB Compiler User’s Guide for more information.

CTF Archives Now Embedded in a Single Binary Executable for Convenient Deployment

As of R2008a, CTF data is now automatically embedded directly in the C/C++, main and Winmain, shared libraries and standalones by default for convenient deployment of applications. In order to override this default functionality, you must compile with the `-C` option. See ““Overriding Default CTF Archive Embedding Using the MCR Component Cache”” in the MATLAB Compiler User’s Guide for more information.

Support Added for Microsoft Visual Studio 2008 Compiler

Support has been added for the compiler included with Microsoft Visual Studio 2008. See “System Requirements ” in the MATLAB Compiler User’s Guide for more details.

Replacement of the `mclmcr` Header File

Starting with R2008a, replace all occurrences of `#include mclmcr.h` with `#include mclmcrct.h`.

If the library header created by MATLAB Compiler software is included in the external code, neither of these files need to be included, since the library header includes the correct file.

Run-Time Libraries Required for Applications Built with Microsoft Visual Studio 2008 Compiler

If you distribute a MEX-file, an engine application, or a MAT-file application built with the Visual Studio® 2008 compiler, you must provide the Visual C++® run-time libraries. These files are required to run applications developed with Visual C++ on a computer that does not have Visual C++ 2008 installed. For information on locating the Microsoft® Visual C++® 2008 Redistributable Package (x86), containing `vc redistrib_x86.exe` and `vc redistrib_x64.exe`, consult your Microsoft® documentation.

Large Array Support Now Default for C and C++ Code

In R2008a, the default definition of `MX_COMPAT_32` has been removed, and large array support is now the default for both C and C++ code. This new default may, in some cases, cause compiler warnings and errors. You can define `MX_COMPAT_32` in your `mbuild` step to return to the previous default behavior.

Code compiled with `MX_COMPAT_32` is *not* 64-bit aware. In addition, `MX_COMPAT_32` controls the behavior of some type definitions. For instance, when `MX_COMPAT_32` is defined, `mwSize` and `mwIndex` are defined to `ints`. When `MX_COMPAT_32` is not defined, `mwSize` and `mwIndex` are defined to `size_t`. This can lead to compiler warnings and errors with respect to signed and unsigned mismatches.

For information about expected behavior of this feature in R2007b, see ““Addressing MWArrays Above the 2 GB Limit” on page 12” in the MATLAB Compiler User’s Guide for more information..

This feature changes how the MWArray C++ Library functions are implemented. For more details, see “Addressing mwArrays Above the 2 GB Limit” in the MATLAB Compiler User’s Guide.

Support Dropped for Borland Compilers

Borland compilers are no longer supported as of this release.

MATLAB Component Runtime Renamed

The MATLAB Component Runtime has been renamed to the MATLAB Compiler Runtime.

MATLAB Application Deployment Web Example Guide Available

A new publication, the *MATLAB Application Deployment Web Example Guide*, is now available from the MATLAB Compiler, MATLAB® Builder™ NE, and MATLAB Builder JA roadmap pages.

The guide provides full examples of common tasks performed by the MATLAB programmer, IT specialist, and others who play significant roles in deploying MATLAB applications to the Web.

For More Information About Compilers Supported by MATLAB

For more information about changes in compiler-related software and interfaces supported by MATLAB, see “Changes to Compiler Support” in the MATLAB Compiler User’s Guide for more information.

Version 4.7 (R2007b) MATLAB Compiler Software

This table summarizes what's new in Version 4.7 (R2007b):

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

New features and changes introduced in this version are:

- “Support Added for the Microsoft® Platform SDK” on page 11
- “Project Files Use Relative Paths and Can Be Shared” on page 11
- “Replacement of MCRInstaller.zip and BUILDMCR Functionality” on page 12
- “Addressing MWArrays Above the 2 GB Limit” on page 12
- “New Compiler Option `-e`” on page 12
- “Support for Borland Compilers Being Phased Out” on page 13
- “Functions Being Removed” on page 13

Support Added for the Microsoft® Platform SDK

Support has been added for the Microsoft Platform SDK Compiler. See System Requirements in the MATLAB® Compiler User’s Guide documentation for more details.

Project Files Use Relative Paths and Can Be Shared

Project files can now be enabled for use with other computers since project files now use relative paths. See Using Relative Paths with Project Files in the MATLAB Compiler User’s Guide documentation for more details.

Replacement of MCRInstaller.zip and BUILDMCR Functionality

In past releases, you needed to include `MCRInstaller.zip` in your packaged application (created by running the `buildmcr` command). Now you must run the following files, which trigger self-extracting archives, that replace the functionality previously provided by `MCRInstaller.zip`. These files ship with MATLAB Compiler:

Platform	File Replacing MCRInstaller.zip
Windows	<code>MCRInstaller.exe</code>
UNIX (Except for Mac)	<code>MCRInstaller.bin</code>
Mac	<code>MCRInstaller.dmg</code>

Since there is no longer a need to create `MCRInstaller.zip`, `buildmcr` is no longer supported.

Type `mcr` or `mcrinstaller` at the MATLAB command prompt for a list of all available MCR installers, compatible platforms, version numbers, and other information. In addition, typing `help mcr` or `help mcrinstaller` at the command prompt will provide further details and assistance.

See the MATLAB Compiler User's Guide documentation for more details.

Addressing MWArrays Above the 2 GB Limit

As of R2007b, you can address MWArrays above the 2 GB limit. This is enabled by defining `MX_COMPAT_32_OFF` in your `mbuild` step. This feature will be the default as of R2008a.

This feature changes how the MWArray C++ Library functions are implemented. For more details, see *Addressing mwArrays Above the 2 GB Limit* in the MATLAB Compiler User's Guide documentation.

New Compiler Option `-e`

This new `mcc` option suppresses the appearance of the MS-DOS command window when generating a standalone application. Use `-e` in place of the

-m option. See “-e Suppress MS-DOS Command Window” in the MATLAB Compiler User’s Guide documentation.

Support for Borland Compilers Being Phased Out

Borland compilers are supported in Version 7.5 (R2007b), but will not be supported in a future version of MATLAB. Please prepare and plan accordingly.

Functions Being Removed

The following functions were removed in R2007b:

Function Being Removed	What Happens When You Run the Function?	Use This Instead	Compatibility Considerations
buildmcr	Errors	MCRInstaller.exe (Windows), MCRInstaller.bin (UNIX), MCRInstaller.dmg (Mac)	See the MATLAB Compiler User’s Guide documentation.
comtool	Undefined Function Error	deploytool	Migrate to deploytool.
dotnettool	Undefined Function Error	deploytool	Migrate to deploytool.
mxltool	Undefined Function Error	deploytool	Migrate to deploytool.
opennb1	Undefined Function Error	deploytool	Migrate to deploytool.
openmx1	Undefined Function Error	deploytool	Migrate to deploytool.
opencb1	Undefined Function Error	deploytool	Migrate to deploytool.

Version 4.6 (R2007a) MATLAB Compiler Software

This table summarizes what's new in Version 4.6 (R2007a):

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:

- “Support Added for Intel Mac and Solaris 64” on page 14
- “Support Dropped for Solaris 2” on page 14
- “Readme File Added” on page 14
- “Warning About Future Borland Compiler Support” on page 15
- “New Compiler Option –F” on page 15
- “Issues with the Microsoft Windows Vista Operating System” on page 15

Support Added for Intel Mac and Solaris 64

Support has been added for the following operating systems:

- Intel Mac
- Solaris 64-bit

Support Dropped for Solaris 2

Support has been dropped for the Solaris 2 operating system. It has been replaced by support for Solaris 64.

Readme File Added

A readme file containing a customized checklist of deployment prerequisites is now generated in the output directory with each compiled application.

Warning About Future Borland Compiler Support

While still supported in this release, customers should prepare for discontinuance of support for the following Borland compilers:

- Borland C++Builder version 6.0
- Borland C++Builder version 5.0
- Borland C/C++ (free command-line tools) version 5.5

New Compiler Option -F

MATLAB Compiler 4.6 includes the -F option for both `mcc` and `deploytool`. Use this option to feed back a specific project file's settings to MATLAB Compiler. For more information, see the MATLAB Compiler User's Guide documentation.

Issues with the Microsoft Windows Vista Operating System

The following are known issues with Windows Vista™ as of this release:

- With User Account Control (UAC) enabled, a standard user is not able to write to a directory in the `c:\Program Files` directory. See the *MATLAB Release Notes* for more details regarding this issue.
- With User Account Control (UAC) enabled, a standard user is not able to register DLLs. When `mbuild -setup` attempts to register `mwcomutil.dll`, the following errors are displayed:

```
Trying to update options file:
  C:\Users\qe\AppData\Roaming\MathWorks\MATLAB\R2007a\compopts.bat
From template:
  C:\PROGRA-1\MATLAB\R2007a\bin\win64\mbuildopts\msvc80compp.bat

Done . . .

--> "C:\PROGRA-1\MATLAB\R2007a\bin\win64\mwregsvr C:\PROGRA-1\MATLAB\R2007a\bin\
win64\mwcomutil.dll"

Error: DllRegisterServer in C:\PROGRA-1\MATLAB\R2007a\bin\win64\mwcomutil.dll
failed
```

```
Undefined subroutine &mexsetup::expire called at C:\PROGRA-1\MATLAB\R2007a\bin\
mexsetup.pm line 839.
```

This is also the case when MATLAB Builder for Excel invokes `mbuild` in an attempt to register DLLs after a compilation completes.

- The following message is displayed when MATLAB Builder for .NET, attempts to install a DLL into the Global Assembly Cache from a network location:

```
Unhandled Exception: System.IO.FileLoadException: Could not load file or assembly
'GACInstaller, Version=1.0.2568.30711, Culture=neutral, PublicKeyToken=null' or
one of its dependencies. Failed to grant permission to execute. (Exception from
HRESULT: 0x80131418)
File name: 'GACInstaller, Version=1.0.2568.30711, Culture=neutral,
PublicKeyToken=null' ---> System.Security.Policy.PolicyException: Execution
permission cannot be acquired.
    at System.Security.SecurityManager.ResolvePolicy(Evidence evidence,
PermissionSet reqdPset, PermissionSet optPset, PermissionSet denyPset,
PermissionSet& denied, Boolean checkExecutionPermission)
    at System.Security.SecurityManager.ResolvePolicy(Evidence evidence,
PermissionSet reqdPset, PermissionSet optPset, PermissionSet denyPset,
PermissionSet& denied, Int32& securitySpecialFlags, Boolean
checkExecutionPermission)
```

- Printing from a compiled application is not currently available with Microsoft Windows Vista. The following message is displayed if you attempt to print on systems with the Microsoft Windows Vista 32-bit operating system installed:

```
PrintImage Error
StartPage failed with error 6: The handle is invalid.
```

Printing with Vista 64-bit installed results in an error message that lists a manifest as missing. Please check <http://www.mathworks.com/support/> for updates on these issues.

Version 4.5 (R2006b) MATLAB Compiler Software

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	No

New features and changes introduced in this version are:

- “Support for Java Components” on page 17
- “Deployment Tool Graphical User Interface” on page 17

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 Software

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	No

New features and changes introduced in this version are:

- “Support for .NET Components” on page 18
- “Support for Microsoft® Visual C++ Version 8.0” on page 18
- “HP-UX No Longer Supported” on page 19

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

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++ Version 8.0

MATLAB Compiler supports Microsoft Visual 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 Software

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

Platform Support

You can use MATLAB Compiler 4.3 on the following supported systems to create redistributable, standalone 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 Software

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

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	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 the File Name on UNIX

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

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

Before this release, using the ~ character as part of the destination file name 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, 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. See “Unsupported Functions” in the MATLAB Compiler User’s Guide documentation.

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 MATLAB Compiler. Note that the libraries created using MATLAB Compiler cannot be loaded into 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 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 Software

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	No

New features and changes introduced in this version are:

- “Solaris Support” on page 25
- “Installing a New Version of the MCR on Windows” on page 27
- “Fixed Bugs” on page 27

Solaris Support

You can use MATLAB Compiler 4.1 on supported Solaris systems to create redistributable, standalone 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 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"
```

All Solaris users are advised to 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 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 Work 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 Software

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 Software

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	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 documentation.

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

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

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 MATLAB Compiler.

-V2.0 Option Removed

The `-V2.0` option that was available in previous releases of 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 MATLAB Compiler software.

Version 4.0 (R14) MATLAB Compiler Software

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	No

New features and changes introduced in this version are:

- “Targets” on page 32
- “Language Support” on page 33
- “Improved C++ Interface” on page 33
- “MATLAB Compiler Runtime” on page 33
- “Component Technology File” on page 33
- “Compatibility Considerations” on page 33

Targets

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

- Standalone 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

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 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). 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 Compiler Runtime

MATLAB Compiler 4 uses the new MATLAB Compiler Runtime (MCR), which is a standalone 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 MATLAB Compiler. M-files that you compiled with a previous version of 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

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 Signal Processing Toolbox™.
- Some toolboxes, such as Symbolic Math Toolbox™, will not compile.

The code generated by 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 MATLAB Compiler

This section highlights significant differences between Compiler 4 and previous versions of 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 MATLAB Compiler fully supports all features of the MATLAB language including objects:

- Compiler 4 uses the new MATLAB Compiler 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. Support is planned for 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 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 `mxCreatDoubleScalar`.
 - The interface to the `m1f` functions generated by MATLAB Compiler from your M-file routines has changed. Unlike previous versions of 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 following options are no longer supported and will produce errors if used.

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
<code>a filename</code>	Add <i>filename</i> to archive; specifies files to be directly added to the CTF archive.
<code>l</code>	Macro that generates a function library. (The meaning of this option has changed since Release 13.)
<code>N</code>	Clears the path of all but a minimal, required set of directories.
<code>p directory</code>	Add <i>directory</i> to compilation path in an order-sensitive context; requires <code>-N</code> option.
<code>R -nojvm</code> <code>R -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 Software

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 V4.9 (R2008b)	See “C++ API Now Supports 64-Bit Default” on page 7. See “Warning Results When Running Figure-Generating Applications or Printing With -nojvm Flag” on page 6.
V4.8 (R2008a)	See “C++ API Now Supports 64-Bit Default” on page 7. See “Replacement of the mclmcr Header File” on page 8.
V4.7 (R2007b)	See “Functions Being Removed” on page 13.
V4.6 (R2007a)	None
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)	See the Compatibility Considerations subheading for each of these new features and changes: <ul style="list-style-type: none"> • “Compiling MATLAB and Toolboxes” on page 34 • “Differences Between Release 14 and Previous Versions of MATLAB® Compiler” on page 34