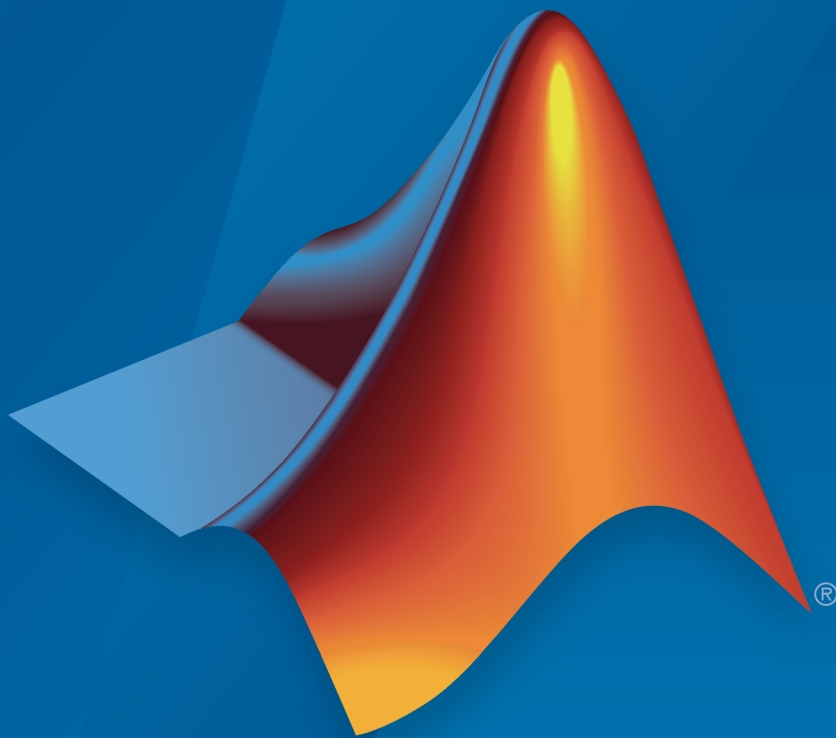


MATLAB[®] Compiler[™] Release Notes



MATLAB[®]

How to Contact MathWorks



Latest news: www.mathworks.com
Sales and services: www.mathworks.com/sales_and_services
User community: www.mathworks.com/matlabcentral
Technical support: www.mathworks.com/support/contact_us



Phone: 508-647-7000



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

MATLAB[®] Compiler[™] Release Notes

© COPYRIGHT 2004–2017 by The MathWorks, Inc.

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

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

Trademarks

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

Patents

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

R2017a

Cloudera CDH: Deploy MATLAB applications against Apache Spark in Cloudera CDH	1-2
Apache Spark 2.x Support	1-2
Functionality Being Removed or Changed	1-2

R2016b

Deployment of MATLAB Applications with Tall Arrays Against Apache Spark: Run MATLAB applications that use tall arrays as standalone executables against a Spark enabled cluster	2-2
Deployment of MATLAB Applications against Apache Spark: Run MATLAB applications as standalone executables against a Spark enabled cluster using the MATLAB API for Spark	2-2
Neural Network Training Models: Deploy Neural Network Toolbox functions that train a model	2-2
Packaging Standalone Applications: Automatically include dependent files when packaging standalone applications	2-3
Functionality Being Removed or Changed	2-3

R2016a

MATLAB Runtime Size: Smaller MATLAB Runtime footprint when deploying numerical applications without graphics or MATLAB user interfaces	3-2
---	------------

R2015aSP1

Bug Fixes

R2015b

32-bit Excel Add-In Compatibility: Create Excel add-ins in a 64-bit MATLAB that are capable of running in 32-bit Microsoft Office.	5-2
Hadoop Support for ImageDatastore: Deploy Hadoop packages with data imported from image collections that are too large to fit in memory	5-2

R2015a

MATLAB Compiler and builders repackaged as MATLAB Compiler and MATLAB Compiler SDK	6-2
Generation of an application-specific MATLAB Runtime based on requirements for GPU support	6-2

MATLAB Compiler support for MATLAB objects for Hadoop integration	6-2
--	------------

R2014b

Deployable applications and libraries for Hadoop using MATLAB MapReduce	7-2
--	------------

R2014a

Faster packaging of compiled applications and libraries ...	8-2
Packaging of hardware support packages using the toolstrip user interface	8-2

R2013b

Automatic download of version- and platform-specific MATLAB Compiler Runtime from MathWorks website ..	9-2
Toolstrip-style user interface for MATLAB Compiler apps ..	9-2
Customizable installers for deployed applications and libraries, including graphics, splash screen, and icons ..	9-2

R2013a

Bug Fixes

R2012b

Apple Xcode 4.2 compiler support for generating or using C/C++ shared libraries on Mac OS X	11-2
MATLAB Compiler Runtime (MCR) available for Web download	11-2
Support for lcc Compiler To Be Removed	11-2

R2012a

MATLAB Compiler Runtime (MCR) Available for Web Download	12-2
Support for Linux gcc 4.4 Compiler	12-2
Example of Deploying Graphical Function with Apple Xcode	12-2
Changes in MCR User Data Interface Arguments for Compiling Parallel Computing Toolbox Applications ...	12-2
Support Updates for Microsoft Visual Studio Versions ...	12-3
Support for lcc Compiler To Be Removed	12-3
Functions and Function Elements Being Removed	12-3

R2011b

Faster Installation of MATLAB Compiler Runtime (MCR) with Improved MCR Installer	13-2
Support for Graphics Processing Unit (GPU) in Parallel Computing Toolbox Applications	13-2
Functions and Function Elements Being Removed	13-2

R2011a

Local Worker Support for Parallel Computing Toolbox Applications	14-2
Error Reporting for Graphical Applications	14-2
Microsoft Visual Studio Express 2010 Compilers Supported	14-2
Functions and Function Elements Being Removed	14-2

R2010bSP1

Bug Fixes

R2010b

Support for Microsoft Visual Studio 2010 Added	16-2
---	------

Include the MCR Installer from a Link on Your Local Network	16-2
mbuild -setup No Longer Needed Before Compiling Standalone Targets	16-2
mcc Caching Automatically Managed for Linux Systems ..	16-2
Mixing MATLAB Files and C or C++ Files Workflow Deprecate d for Alternate Workflow	16-2
mcc -F Option Deprecate d with Warning	16-3
%#EXTERNAL Pragma Deprecate d For Standalones	16-3
New Deployment Tool Project Format	16-3
Intel Macintosh 32-Bit Systems (Maci) No Longer Supported	16-4

R2010a

Desktop Icon Available for Launching 64-Bit Mac Applications	17-2
New Video Demo Available	17-2
Support for MATLAB Sparse Matrices	17-2
Deployment Tool Now Available from Command Line	17-2
linkdata on Not Supported in Deployed Standalones	17-2

R2009bSP1

MCR Version Number Change	18-2
--	-------------

R2009b

Redesigned Deployment Tool GUI	19-2
Customizable MCR Start-Up Message	19-2
MATLAB Memory Shielding Available	19-2
Mac 64-Bit Support Available	19-2
New Format for Deployment Tool Projects	19-2
Change in Windows Library Path	19-2
INLINE Option to MEX Function Deprecated	19-3

R2009a

Reduced MCR Size Saves on Transfer Time	20-2
MATLAB Pcode Files Can Now Be Compiled	20-2
New Functions Return Information About MCR State	20-2
Microsoft Visual Studio Express 2008 Compilers Supported	20-2
Customized readme.txt Produced with Each Compilation .	20-2

Applications Created with Parallel Computing Toolbox Can Be Compiled	21-2
Data Sharing Between MATLAB Code, MCR Instance, and Wrapper Code Available	21-2
64-Bit Addressing of mxArray Arrays Is New Default, 32-Bit Addressing No Longer Supported	21-2
Microsoft Visual Studio 2008, 32-Bit and 64-Bit Editions, Now Supported	21-2
Warning Results When Running Figure-Generating Applications or Printing with -nojvm Flag	21-2
Same Name Class Objects Shared Between MCR Instances Will Not Work Correctly	21-3

C++ API Now Supports 64-Bit Default	22-2
CTF Archives Now Embedded in a Single Binary Executable for Convenient Deployment	22-2
Support Added for Microsoft Visual Studio 2008 Compiler	22-2
Replacement of the mclmcr Header File	22-2
Run-Time Libraries Required for Applications Built with Microsoft Visual Studio 2008 Compiler	22-3
Large Array Support Now Default for C and C++ Code ...	22-3
Support Dropped for Borland Compilers	22-3

MATLAB Component Runtime Renamed	22-3
MATLAB Application Deployment Web Example Guide Available	22-4
Enforcement of Proper Return Types For C/C++ Methods .	22-4
Changes in <code>mclmcr rt.h</code> and <code>mclmcr.h</code> Include Order	22-4

R2007b

Support Added for the Microsoft Platform SDK	23-2
Project Files Use Relative Paths and Can Be Shared	23-2
Replacement of <code>MCRInstaller.zip</code> and <code>BUILDMCR</code> Functionality	23-2
Addressing <code>MWArrays</code> Above the 2 GB Limit	23-2
New Compiler Option <code>-e</code>	23-3
Support for Borland Compilers Removed	23-3
Functions Being Removed	23-3

R2007a

Support Added for Intel Mac and Solaris 64	24-2
Support Dropped for Solaris 2	24-2
Readme File Added	24-2

Warning About Future Borland Compiler Support	24-2
New Compiler Option -F	24-2
Issues with the Microsoft Windows Vista Operating System	24-2

R2006b

Support for Java Components	25-2
Deployment Tool Graphical User Interface	25-2

R2006a

Support for .NET Components	26-2
Support for Microsoft Visual C++ Version 8.0	26-2
HP-UX No Longer Supported	26-2

R14SP3

Platform Support	27-2
Supported Platforms	27-2

Platform Support	28-2
Supported Platforms	28-2

R2017a

Version: 6.4

New Features

Bug Fixes

Compatibility Considerations

Cloudera CDH: Deploy MATLAB applications against Apache Spark in Cloudera CDH

You can deploy MATLAB® applications against Apache Spark™ in Cloudera® CDH (version 5.7 and later). To deploy applications that use the MATLAB API for Spark™ to Cloudera CDH, see “Deploying Applications to Cloudera Spark Using the MATLAB API for Spark”. To deploy tall array applications to Cloudera CDH, see “Deploying Tall Arrays to a Cloudera Spark Enabled Hadoop Cluster”. Tall array applications use the `spark-submit` script in Spark’s `bin` directory to start applications on a cluster. Both workflows can be adopted for deployment against Spark "Hadoop Free" builds.

Apache Spark 2.x Support

Deploy MATLAB applications to Apache Spark 2.x. enabled clusters.

Functionality Being Removed or Changed

Functionality	Result	Use Instead	Compatibility Considerations
Support for running MATLAB MapReduce on Hadoop® 1.x clusters will be removed in a future release.	Warns	Use clusters that have Hadoop 2.x or higher installed to run MATLAB MapReduce.	Migrate MATLAB MapReduce code that runs on Hadoop 1.x to Hadoop 2.x.

R2016b

Version: 6.3

New Features

Bug Fixes

Compatibility Considerations

Deployment of MATLAB Applications with Tall Arrays Against Apache Spark: Run MATLAB applications that use tall arrays as standalone executables against a Spark enabled cluster

MATLAB applications that contain tall arrays can be packaged as standalone applications and deployed against a Spark enabled cluster.

For more information, see [Deploy Tall Arrays to a Spark Enabled Hadoop Cluster](#).

Deployment of MATLAB Applications against Apache Spark: Run MATLAB applications as standalone executables against a Spark enabled cluster using the MATLAB API for Spark

MATLAB applications can be executed against a Spark enabled cluster using the MATLAB API for Spark. The MATLAB API for Spark allows you to use Spark functions such as `flatMap`, `mapPartitions`, `aggregate` and others in your MATLAB code. It exposes the Spark programming model to MATLAB, allowing for MATLAB implementations of numerous Spark functions.

For more information, see [Deploy Applications Using the MATLAB API for Spark](#).

Neural Network Training Models: Deploy Neural Network Toolbox functions that train a model

Use MATLAB Runtime to deploy functions that can train a model. To deploy MATLAB code that trains neural networks, see [Create Standalone Application from Command Line](#). The following features are not supported in deployed mode:

- Training progress dialog, `nntraintool`
- `genFunction` and `gensim` to generate MATLAB code or Simulink® blocks
- `view` method
- `nctool`, `nftool`, `nnstart`, `nprtool`, `ntstool`
- Plot functions

Packaging Standalone Applications: Automatically include dependent files when packaging standalone applications

You no longer need to include files accessed using standard file format functions or low-level I/O functions. The Application Compiler App automatically includes:

- Files that are accessed using standard file format functions. These dependent files include text, spreadsheets, images, audio, and video files. Scientific data formats are the only standard file formats that are not supported.
- Files that are accessed using low-level I/O functions.

For a list of supported functions, see [Standard File Formats](#). For a list of supported I/O functions, see [Low-Level File I/O](#).

Functionality Being Removed or Changed

Functionality	Result	Use Instead	Compatibility Considerations
Support for running MATLAB MapReduce on Hadoop 1.x clusters will be removed in a future release.	Still runs	Use clusters that have Hadoop 2.x or higher installed to run MATLAB MapReduce.	Migrate MATLAB MapReduce code that runs on Hadoop 1.x to Hadoop 2.x.

R2016a

Version: 6.2

New Features

Bug Fixes

MATLAB Runtime Size: Smaller MATLAB Runtime footprint when deploying numerical applications without graphics or MATLAB user interfaces

Depending on application requirements, the memory footprint of MATLAB Runtime is now decreased by up to 60% when compared to previous releases.

R2015aSP1

Version: 6.0.1

Bug Fixes

R2015b

Version: 6.1

New Features

Bug Fixes

32-bit Excel Add-In Compatibility: Create Excel add-ins in a 64-bit MATLAB that are capable of running in 32-bit Microsoft Office.

Excel® add-ins created using a 64-bit installation of MATLAB Compiler™ can be installed and used by 32-bit installations of Microsoft® Office.

Hadoop Support for ImageDatastore: Deploy Hadoop packages with data imported from image collections that are too large to fit in memory

Use the `datastore` function to create an `ImageDatastore`, which can import data from image collections that are too large to fit in memory. `ImageDatastore` is a compatible `datastore` for use with `mapreduce`.

R2015a

Version: 6

New Features

Bug Fixes

MATLAB Compiler and builders repackaged as MATLAB Compiler and MATLAB Compiler SDK

- Functionality to create Excel add-in components for Microsoft Excel desktop applications has moved to MATLAB Compiler.
- Functionality to create C/C++ shared libraries has moved to MATLAB Compiler SDK™.
- Functionality to create CTF archive files for use with MATLAB Production Server™ has moved to MATLAB Compiler SDK.

Generation of an application-specific MATLAB Runtime based on requirements for GPU support

MATLAB Compiler analyzes your MATLAB code for use of graphics or a GPU and defines and calculates optimized MATLAB Runtime. A MATLAB Runtime specific for each application is available for download from the web or included in the package.

MATLAB Compiler support for MATLAB objects for Hadoop integration

MATLAB Compiler now supports MATLAB objects as output from the map and reduce functions for Hadoop Integration.

R2014b

Version: 5.2

New Features

Bug Fixes

Deployable applications and libraries for Hadoop using MATLAB MapReduce

MATLAB Compiler includes a MapReduce compiler app that packages MapReduce applications for deployment against Hadoop. The applications can be compiled into either standalone applications or libraries for use in developing other applications.

R2014a

Version: 5.1

New Features

Bug Fixes

Compatibility Considerations

Faster packaging of compiled applications and libraries

MATLAB Compiler uses a new dependency resolution function that is faster.

Compatibility Considerations

The new dependency resolution function requires that p-code being used in a compiled function has an associated `.m` file containing the original source for the p-code.

Packaging of hardware support packages using the toolstrip user interface

The compiler apps now scan your code and the installed support packages to determine what needs to be packaged. The compiler will package all required support packages for distribution.

R2013b

Version: 5.0

New Features

Bug Fixes

Automatic download of version- and platform-specific MATLAB Compiler Runtime from MathWorks website

The MATLAB Compiler apps generate a platform specific installer that will automatically download and install the proper MATLAB Compiler Runtime if it is needed.

Toolstrip-style user interface for MATLAB Compiler apps

The MATLAB Compiler apps use a toolstrip for commonly used interface components and required input fields.

Customizable installers for deployed applications and libraries, including graphics, splash screen, and icons

The MATLAB Compiler apps have fields for supplying custom graphics, icons, developer profile details, and version numbers to the generated installer.

R2013a

Version: 4.18.1

Bug Fixes

R2012b

Version: 4.18

New Features

Bug Fixes

Compatibility Considerations

Apple Xcode 4.2 compiler support for generating or using C/C++ shared libraries on Mac OS X

MATLAB Compiler now supports C and C++ shared libraries generation for Mac OS X, using Apple Xcode compiler, version 4.2.

For more information on using Mac with MATLAB Compiler, see Writing Applications for Mac or Linux.

MATLAB Compiler Runtime (MCR) available for Web download

The MATLAB Compiler Runtime (MCR) is available for downloading from the Web to simplify the distribution of your applications or components created with the MATLAB Compiler.

Direct your end users to the MATLAB Compiler Runtime page at <http://www.mathworks.com/products/compiler/mcr/> to download the MCR, as opposed to redistributing or packaging it with your applications or components.

Support for lcc Compiler To Be Removed

The lcc compiler that ships with Windows[®] will no longer be supported as of R2013a.

For release R2012b, selecting the lcc compiler using `mbuild -setup` results in a warning, though the compiler is available. In R2013a, lcc will no longer appear in the list of available compilers.

R2012a

Version: 4.17

New Features

Bug Fixes

Compatibility Considerations

MATLAB Compiler Runtime (MCR) Available for Web Download

The MATLAB Compiler Runtime (MCR) is available for downloading from the Web to simplify the distribution of your applications or components created with the MATLAB Compiler.

Direct your end users to the MATLAB Compiler product page to download the MCR, as opposed to redistributing or packaging it with your applications or components.

Support for Linux gcc 4.4 Compiler

MATLAB Compiler now supports Linux[®] gcc 4.4.x compilers for both 32-bit and 64-bit architectures.

See http://www.mathworks.com/support/compilers/current_release/ for a complete up-to-date list of supported compilers and software.

Example of Deploying Graphical Function with Apple Xcode

An example demonstrating how to deploy a graphical MATLAB function with the Mac OS X Cocoa development layer, is available.

The function is compiled to a shared library target and then integrated into a deployed application using Apple Xcode.

See Writing Applications for Mac or Linux for more details.

Changes in MCR User Data Interface Arguments for Compiling Parallel Computing Toolbox Applications

Parallel Computing Toolbox™ configurations are now referred to as *parallel profiles*, as of R2012a.

See Profiles in Compiled Applications in the Parallel Computing Toolbox Release Notes for complete information.

Compatibility Considerations

The older MCR User Data Interface argument `ParallelConfigurationFile` and MAT files will still be supported until Parallel Computing Toolbox documentation states

otherwise, but note that if you use the `ParallelConfigurationFile` argument, you must supply a configuration `.mat` file, rather than a profile `.settings` file. If the `ParallelProfile` argument is used, a configuration `.mat` file or a profile `.settings` file can be used.

Support Updates for Microsoft Visual Studio Versions

The following changes to Microsoft Visual Studio® support apply for R2012a and R2012b.

Microsoft Visual Studio Version	What Happens When You Attempt to Use This Version in R2012a	What Happens When You Attempt to Use This Version in R2012b
Microsoft Visual Studio 2010 Express, for 32-bit or 64-bit architectures	Not Available	Not Available
Microsoft Visual C++® 2008 for 32-bit or 64-bit architectures	Warns	Not Available

Support for lcc Compiler To Be Removed

The `lcc` compiler that ships with Windows will no longer be supported as of R2013a.

For releases R2012a and R2012b, selecting the `lcc` compiler using `mbuild -setup` results in a warning, though the compiler is available. For R2013a, `lcc` will no longer appear in the list of available compilers.

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When You Use the Function or Element	Do This Instead	Compatibility Considerations
<i>LibraryGetMcrID</i>	Errors	Use one of the following functions without specifying an MCR ID: <ul style="list-style-type: none"> <code>getmcruserdata</code> <code>setmcruserdata</code> 	For more information, see Setting and Retrieving MCR Data for Shared Libraries .

Function or Function Element Name	What Happens When You Use the Function or Element	Do This Instead	Compatibility Considerations
mclGetMcrUserData	Errors	Use getmcusercontent.	For more information, see Setting and Retrieving MCR Data for Shared Libraries.
mclSetMcrUserData	Errors	Use setmcusercontent.	For more information, see Setting and Retrieving MCR Data for Shared Libraries.
mclSetCmdLineUserData	Errors	No replacement	Not applicable

R2011b

Version: 4.16

New Features

Bug Fixes

Compatibility Considerations

Faster Installation of MATLAB Compiler Runtime (MCR) with Improved MCR Installer

The Application Deployment products now feature faster installation of the MATLAB Compiler Runtime (MCR) through improved MCR Installer technology. In addition to improved performance, the new installer provides a common interface for all platforms.

Support for Graphics Processing Unit (GPU) in Parallel Computing Toolbox Applications

MATLAB Compiler generated executables and components now benefit from better performance when used with Parallel Computing Toolbox applications. The improvement is due to support of the Graphics Processing Unit (GPU) feature.

For more information, see GPU Computing in the Parallel Computing Toolbox User's Guide, as well as the Parallel Computing Toolbox examples in the deployment user's guides for additional information about compiling your deployable application with GPU.

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When You Use the Function or Element	Do This Instead	Compatibility Considerations
<code>%#external</code>	Errors	Rewrite the C or C++ function as a MEX-file and call it from your application.	For additional information, see <i>Interfacing MATLAB Code to C/C++ Code</i> in the User's Guide and <i>Creating C/C++ Language MEX-Files</i> in the <i>MATLAB External Interfaces User's Guide</i> .
<code>libraryGetMcrID</code>	Warnings	Use one of the following functions without specifying an MCR ID:	For more information, see <i>Setting and Retrieving MCR Data for Shared Libraries</i> .

Function or Function Element Name	What Happens When You Use the Function or Element	Do This Instead	Compatibility Considerations
		<ul style="list-style-type: none"> • getmcruserdata • setmcruserdata 	
mclGetMcrUserData	Warnings	Use getmcruserdata.	For more information, see Setting and Retrieving MCR Data for Shared Libraries.
mclSetMcrUserData	Warnings	Use setmcruserdata.	For more information, see Setting and Retrieving MCR Data for Shared Libraries.
mclSetCmdLineUserData	Warnings	No replacement	No support as of version R2012a.
Control System Toolbox™ prescale GUI	Errors	No replacement	The Control System Toolbox prescale GUI is no longer compatible with MATLAB Compiler and cannot be deployed.

R2011a

Version: 4.15

New Features

Bug Fixes

Compatibility Considerations

Local Worker Support for Parallel Computing Toolbox Applications

MATLAB Compiler and the builders now support local workers in Parallel Computing Toolbox applications.

For information about this feature, see *Use a Local Cluster* for general information on Local Workers. Also, see the *Parallel Computing Toolbox User's Guide* for information about working with compilable PCT applications.

Error Reporting for Graphical Applications

MATLAB Compiler now provides error messages in pop-up windows for Windows and Macintosh graphical applications. You do not need to run a console in order to view these messages.

Microsoft Visual Studio Express 2010 Compilers Supported

Effective with this release, Microsoft Visual Studio Express 2010 compilers (32- and 64-bit versions) are now supported. Microsoft Visual Studio Express 2008 compilers (32- and 64-bit versions), which have been dropped from support. See *Installation and Configuration* in the *MATLAB Compiler User's Guide* for information about installing and configuring them. See http://www.mathworks.com/support/compilers/current_release/ for an up-to-date list of supported compilers.

Functions and Function Elements Being Removed

Function or Function Element Name	What Happens When you use the Function or Element	Use This Instead	Compatibility Considerations
<code>mcc -F</code>	Errors	<code>deploytool -n</code>	For information about using the command-line version of the Deployment Tool, see <i>Using the Deployment Tool from the Command Line</i> .

R2010bSP1

Version: 4.14.1

Bug Fixes

R2010b

Version: 4.14

New Features

Bug Fixes

Compatibility Considerations

Support for Microsoft Visual Studio 2010 Added

This release adds support for Microsoft Visual Studio software on both 32-bit and 64-bit systems. See the Installation and Configuration chapter in the *MATLAB Compiler User's Guide* for more information.

Include the MCR Installer from a Link on Your Local Network

This feature lets you add a link to an MCR Installer residing on a local area network. Adding this link allows you to invoke the installer over the network, as opposed to copying the installer physically into each deployable package.

See Packaging (Optional) in the MATLAB Compiler User's Guide, or in your respective Builder product User's Guide, for more details.

`mbuild -setup` No Longer Needed Before Compiling Standalone Targets

You no longer need to specify a third-party compiler with `mbuild -setup` when building a standalone target. This is the case for both standalones and Windows standalones (standalone applications that suppress the DOS command window when running). For more information about when `mbuild -setup` is not needed, see When Not to Use `mbuild -setup`.

`mcc` Caching Automatically Managed for Linux Systems

In previous releases, you could manually manage cache for `mcc` on Linux systems.

As of R2010b, caching for `mcc` is automatically managed and tuned. No user intervention is required.

Mixing MATLAB Files and C or C++ Files Workflow Deprecated for Alternate Workflow

In previous releases, the workflow described in the procedure “Mixing MATLAB Files and C or C++ Files” detailed how to mix MATLAB and C or C++ source files. This workflow has been replaced with a more straight-forward approach for release R2010b.

Compatibility Considerations

Running the obsolete workflow (documented prior to R2010b) now results in a warning.

See [Combining Your MATLAB and C/C++ Code](#) for details on the recommended replacement workflow.

mcc -F Option Deprecated with Warning

As of R2011a, the `mcc -F` option will no longer process Deploytool Tool project files passed in or generate project files when invoked with command line arguments.

Compatibility Considerations

In R2010b, a warning will appear when the `mcc -F` command is issued. In R2011a, the command will generate an error.

Instead of using `mcc -F`, see [Using the Deployment Tool from the Command Line in the User's Guide](#) for replacement functionality.

%#EXTERNAL Pragma Deprecated For Standalones

Use of the `%#external` pragma to combine C/C++ code with a MATLAB application has been deprecated for standalone targets in release R2010b.

Compatibility Considerations

Using the `%#external` pragma in release R2010b results in a warning. In a future release, usage will result in an error.

See [Interfacing MATLAB Code to C/C++ Code](#) for details of the replacement workflow.

New Deployment Tool Project Format

Beginning in release R2010b, the Deployment Tool will store projects in a new format.

Compatibility Considerations

If you open a supported legacy project file with `deploytool`, you will automatically be prompted to save it in the new format.

If your legacy project files are in a nonuser-customized format, there is no action you need to take beyond converting your projects to the new format, when prompted.

Intel Macintosh 32-Bit Systems (Maci) No Longer Supported

Intel Macintosh 32-Bit systems, also known as Maci, are no longer supported as of release R2010b.

The software still supports Intel Macintosh 64-Bit systems, also known as Maci 64.

Compatibility Considerations

All MacIntosh bundles produced by deployment products are designed to run on 64-bit architectures and specific releases of the Apple operating system. Incompatibilities will result in an error from the Apple operating system.

R2010a

Version: 4.13

New Features

Bug Fixes

Desktop Icon Available for Launching 64-Bit Mac Applications

Macintosh 64-bit applications can now be launched conveniently from the desktop. See [Using MATLAB Compiler on Mac or Linux](#) for details about the Mac Application Launcher and other guidelines for UNIX, Linux, and Mac users.

New Video Demo Available

Watch a video about deploying applications using MATLAB Compiler.

Support for MATLAB Sparse Matrices

This release provides support for MATLAB sparse matrices via a C++ API. See [Static Factory Methods for Sparse Arrays](#) for a listing of capable classes and methods.

Deployment Tool Now Available from Command Line

You can now invoke the Deployment Tool GUI from the command line. See [Using the Deployment Tool from the Command Line](#) for more information.

linkdata on Not Supported in Deployed Standalones

In MATLAB, setting `linkdata on` can be used to link data between multiple plots in the same figure. This feature is not available for standalone applications run in deployed mode.

R2009bSP1

Version: 4.12

Bug Fixes

MCR Version Number Change

As described in the MATLAB Release Notes, the version number of the MATLAB Compiler Runtime (MCR) in release R2009bSP1 is different from the MCR version number in release R2009b.

R2009b

Version: 4.11

New Features

Bug Fixes

Compatibility Considerations

Redesigned Deployment Tool GUI

The new Deployment Tool (`deploytool`) interface features intuitive task-based navigation, a cancellable progress dialog, fast loading of previously-created projects, and ability to add supporting files as folders. The GUI also features new context sensitive help.

Customizable MCR Start-Up Message

Console applications can now display a user-customizable message indicating that the MATLAB Compiler Runtime (MCR) is in the process of starting up. See [Displaying MCR Initialization Start-Up and Completion Messages For Users](#) for more information.

MATLAB Memory Shielding Available

The same memory shielding process available in MATLAB is now available for deployed applications. The process reserves memory to ensure resource-intensive applications are allocated proper run-time resources. See [Reserving Memory for Deployed Applications with MATLAB Memory Shielding](#) for more information.

Mac 64-Bit Support Available

Support for Macintosh[®] 64-bit processors is now available.

New Format for Deployment Tool Projects

As of this release, deployment projects are being created in a new file format.

Compatibility Considerations

If you have projects that were created with the Deployment Tool prior to R2009b, those projects will continue to work. However, projects created or changed in R2009b cannot be opened in previous versions of Deployment Tool (`deploytool`). Recreate old projects with Deployment Tool to ensure they function properly.

Change in Windows Library Path

The path `matlab\bin\arch` has been removed for Windows and replaced by `matlabroot\runtime\win32|win64`. You may be required to reconfigure your

deployment target computers. See [Set MCR Paths for Development and Testing](#) and [Set MCR Paths for Run-Time Deployment](#) for more information.

INLINE Option to MEX Function Deprecated

For more information on this change in support that will occur in R2010a, see [Functionality Being Removed in the *MATLAB Release Notes*](#).

R2009a

Version: 4.10

New Features

Bug Fixes

Reduced MCR Size Saves on Transfer Time

MCR copying and transfer time has been improved by 50% following a reduction in the file's contents. The change will not reduce processor time or memory consumption since the deleted files were not loaded into RAM.

MATLAB Pcode Files Can Now Be Compiled

You can now compile protected MATLAB files (Pcode files or "P-files"). For more information about Pcode and P-files, see the MATLAB Function Reference.

New Functions Return Information About MCR State

New functions have been introduced that return data about MCR state, such as whether the MCR has been properly initialized or whether the MCR was launched with a JVM™ instance enabled. For more information, see Retrieving MCR Attributes in the user guide.

Microsoft Visual Studio Express 2008 Compilers Supported

Microsoft Visual Studio Express 2008 compilers (32- and 64-bit versions) are now supported instead of Microsoft Visual Studio Express 2005 compilers (32- and 64-bit versions), which have been dropped from support. See Installation and Configuration in the MATLAB Compiler User's Guide for a complete list of supported compilers and information about installing and configuring them.

Customized readme.txt Produced with Each Compilation

An enhanced version of the `readme.txt` file, which is generated with each successful compilation, is now customized to include specific instructions about MATLAB Compiler deployment requirements.

R2008b

Version: 4.9

New Features

Bug Fixes

Compatibility Considerations

Applications Created with Parallel Computing Toolbox Can Be Compiled

You can now compile MATLAB applications that use the Parallel Computing Toolbox. 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 MATLAB Code, MCR Instance, and Wrapper Code Available

It is now possible to share data between an MCR instance, the MATLAB code running on that MCR, and the wrapper code that created the MCR, by implementing two MATLAB 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 `mxArrays` 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.

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.

Compatibility Considerations

In some cases, figure rendering may succeed, and in other cases it may not. Similarly, various MATLAB graphics functions dependent on Java[®], such as graphics passed with `-R -nojvm`, will not result in displayable graphics. 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.

R2008a

Version: 4.8

New Features

Bug Fixes

Compatibility Considerations

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.

Compatibility Considerations

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 “MCR Component Cache and CTF Archive Embedding” 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 mclmcr.h`.

Compatibility Considerations

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” 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.

Enforcement of Proper Return Types For C/C++ Methods

As of R2008a, the LCC compiler is more strict in enforcing `bool` return types from C and `void` returns from C++. For more information, see *Code Proper Return Types From C and C++ Methods* in the MATLAB Compiler User's Guide.

Changes in `mclmcr.h` and `mclmcr.h` Include Order

Changes introduced in this release now require that the header file `mclmcr.h` always be included after `mclmcr.h` in the preprocessing stage.

Compatibility Considerations

This change introduces a potential compatibility problem in subsequent releases of MATLAB Compiler. See the MathWorks support Web site for further information.

R2007b

Version: 4.7

New Features

Bug Fixes

Compatibility Considerations

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 Distribute MATLAB Code Using the MATLAB Compiler Runtime (MCR) for more information.

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 Removed

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
<code>buildmcr</code>	Errors	<code>MCRInstaller.exe</code> (Windows), <code>MCRInstaller.bin</code> (UNIX), <code>MCRInstaller.dmg</code> (Mac)	See Distribute MATLAB Code Using the MATLAB Compiler Runtime (MCR) in the MATLAB Compiler User's Guide documentation.
<code>comtool</code>	Undefined Function Error	<code>deploytool</code>	Migrate to <code>deploytool</code> .
<code>dotnettool</code>	Undefined Function Error	<code>deploytool</code>	Migrate to <code>deploytool</code> .
<code>mxltool</code>	Undefined Function Error	<code>deploytool</code>	Migrate to <code>deploytool</code> .

Function Being Removed	What Happens When You Run the Function?	Use This Instead	Compatibility Considerations
opennb1	Undefined Function Error	deploytool	Migrate to deploytool.
openmx1	Undefined Function Error	deploytool	Migrate to deploytool.
opencb1	Undefined Function Error	deploytool	Migrate to deploytool.

R2007a

Version: 4.6

New Features

Bug Fixes

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 folder 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. To resolve these issues, ensure you are logged in as Administrator.

- With User Account Control (UAC) enabled, a standard user is not able to write to a folder in the `c:\Program Files` folder. 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\msvc80comp.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.

R2006b

Version: 4.5

New Features

Bug Fixes

Support for Java Components

MATLAB Compiler, in conjunction with the optional MATLAB Builder JA, lets you convert MATLAB 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:



R2006a

Version: 4.4

New Features

Bug Fixes

Support for .NET Components

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.

MATLAB Compiler, in conjunction with the optional MATLAB Builder for .NET, lets you convert MATLAB 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++ 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.

R14SP3

Version: 4.3

New Features

Bug Fixes

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.

R14SP2

Version: 4.2

New Features

Platform Support

You can use MATLAB Compiler 4.2 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 4.2 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.