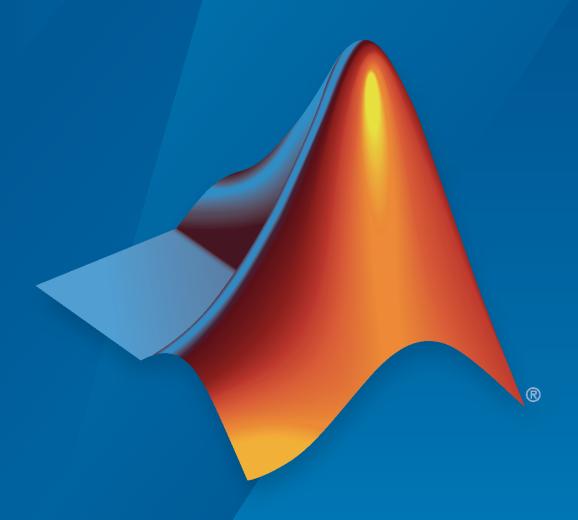
### MATLAB® Compiler SDK™ Release Notes



# 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

**T** 

Phone: 508-647-7000



The MathWorks, Inc. 1 Apple Hill Drive Natick, MA 01760-2098

MATLAB® Compiler SDK™ Release Notes

© COPYRIGHT 2004-2021 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.

## Contents

R20	21b
Packaging: Obfuscate file structures and file names	1-2
Support Packages: Additional options for specifying support packages	1-2
Excel Add-In for MATLAB Production Server: Create an Excel add-in for MATLAB Production Server using the excelClientForProductionServer function	1-2
C++ Shared Library Integration: Support for strongly typed MATLAB code	1-2
Python Version Support for Python Package Integration	1-2
Functionality Being Removed or Changed	1-3
-build and -package options in the productionServerCompiler function will	1-3
be removed	1-3 1-3
R20	21a
COM Component Integration: Create a COM component using the compiler.build.comComponent function	2-2
C++ Integration: Create a C++ shared library using the compiler.build.cppSharedLibrary function	2-2
C Integration: Create a C shared library using the compiler.build.cSharedLibrary function	2-2
.NET Integration: Create a .NET assembly using the compiler.build.dotNETAssembly function	2-2
Java: Create a Java package using the compiler.build.javaPackage	

2-2

Python integration: Create a Python package using the compiler.build.pythonPackage function	2-3
Functionality Being Removed or Changed	2-3
-build and -package options in the libraryCompiler function will be removed	2-3
-build and -package options in the productionServerCompiler function will	
be removed	2-3 2-3
R20	20b
Java Package Integration: Support for MATLAB string array	3-2
MATLAB Production Server: Create deployable archives using the compiler.build.productionServerArchive function	3-2
Python Version Support for Python Package Integration	3-2
Functionality Being Removed or Changed	3-2
-build and -package options in the productionServerCompiler function will	3-2
be removed	3-2 3-2
R20	20a
N20	<b>20</b> 0
.NET Assembly Integration: Support for MATLAB string array	4-2
C Shared Library Integration: Support for MATLAB string array	4-2
C++ Shared Library Integration: Support for MATLAB string array	4-2
Functionality Being Removed or Changed	4-2
-build and -package options in the productionServerCompiler function will be removed	<b>4-2</b> <b>4-2</b>

NET Assembly Integration: Run .NET assemblies integrated into .Net applications on Linux and macOS using .NET Core		5-2
ile Versioning: Generate system-level file versioning on Windows systems for files created using deployment apps		5-2
	<b>R20</b> 1	19a
ibrary Compiler: Generate programming language specific sample code		6-2
unctionality Being Removed		6-2
	<b>R20</b> 1	18b
ug Fixes		
ug Fixes	R201	l8a
ug Fixes ++ Shared Library Integration: Generate C++ shared libraries usi MATLAB Data API	ing	
++ Shared Library Integration: Generate C++ shared libraries usi	ing 	8-2 8-2
++ Shared Library Integration: Generate C++ shared libraries usi MATLAB Data API	i <b>ng</b> 	8-2
++ Shared Library Integration: Generate C++ shared libraries usi MATLAB Data API	ing	8-2 8-2 8-2
++ Shared Library Integration: Generate C++ shared libraries usi MATLAB Data API  ompiler Support for C/C++ Shared Library Integration  ython Version Support for Python Package Integration  ESTful API for Service Discovery: Discover MATLAB functions dep	ing bloyed B and	8-2

Python Package Integration: Support for Python 3.6	9-2
R20	017a
Bug Fixes	
R20	016b
Neural Network Training Models: Deploy Neural Network Toolbox functions that train a model	11-2
R2	016a
HTTPS Support for Excel Add-ins: Securely integrate Excel Add-ins with MATLAB Production Server using HTTPS	12-2
RESTful API and JSON: Develop clients for MATLAB Production Server in any programming language that supports HTTP	1 12-2
R2015a	aSP1
Bug Fixes	
R20	015 <b>b</b>

Packaging of your MATLAB programs as C/C++ shared libraries, Microsoft .NET assemblies, and Java classes	15-2
Royalty-free distribution of software components to users who do not need MATLAB	15-2
Development and test framework for MATLAB Production Server for integration with web and enterprise systems	15-2
Encryption of MATLAB code to protect your intellectual property	15-2

### R2021b

Version: 6.11

**New Features** 

**Bug Fixes** 

#### Packaging: Obfuscate file structures and file names

You can use the mcc command with the -s option to customize your archive for all deployment targets. This option obfuscates folder structures and file names in the deployable archive (.ctf file) from the end user.

#### Support Packages: Additional options for specifying support packages

You can use two new options to manually include support packages for all deployment targets.

- Use the mcc command with the -Z option to specify the method of adding support packages to the deployable archive.
- Use a compiler.build command with the SupportPackages option to specify the method of adding support packages.

To list installed support packages or those used by a specific file, see compiler.codetools.deployableSupportPackages.

# Excel Add-In for MATLAB Production Server: Create an Excel add-in for MATLAB Production Server using the excelClientForProductionServer function

You can create Excel® add-ins for MATLAB Production Server™ from the MATLAB command prompt using the compiler.build.excelClientForProductionServer function. This function, along with the compiler.build.ExcelClientForProductionServerOptions function, provides an improved interface to specify various options associated with creating Excel add-ins. You can deploy the generated Excel add-in to MATLAB Production Server. These functions are in addition to the existing Production Server Compiler (MATLAB Production Server) app you use to create Excel add-ins.

### C++ Shared Library Integration: Support for strongly typed MATLAB code

When creating C++ shared libraries from MATLAB functions or classes, you can stipulate how MATLAB data types should be represented in C++ application code using standard and custom data type mappings between MATLAB and C++. The data type requirements are specified using an arguments block within a MATLAB function or using a properties block and arguments block within a MATLAB class. The strongly typed MATLAB function, class, or package is compiled using the mcc command to generate a C++ shared library header (.hpp file) and a deployable archive (.ctf file). The generated header file (.hpp file) is included in the C++ application code using a #include directive. You can then compile and run the application.

For details, see "C++ MATLAB Data API Shared Library Support for Strongly Typed MATLAB Code".

#### Python Version Support for Python Package Integration

Support	Python® Version	Platform
Added	Python 3.9	All platforms

Support	Python <sup>®</sup> Version	Platform
Discontinued	Python 3.6, as of R2021a	All platforms

#### **Functionality Being Removed or Changed**

#### -build and -package options in the library Compiler function will be removed Warns

The -build and -package options in the libraryCompiler function will be removed in a future release. To build applications, use one of the compiler.build family of functions or the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the libraryCompiler function.

Also, there are no changes to the **Library Compiler** app. You can continue to the use the app to generate libraries.

#### -build and -package options in the production ServerCompiler function will be removed Warns

The -build and -package options in the productionServerCompiler function will be removed in a future release. To generate deployable archives, use the compiler.build.productionServerArchive function, or the mcc command, or the **Production Server Compiler** app. There are no changes to the other options of the productionServerCompiler function.

Also, there are no changes to the **Production Server Compiler** app. You can continue to the use the app to generate deployable archives.

#### -build and -package options in the deploytool function will be removed ${\it Warns}$

The -build and -package options in the deploytool function will be removed in a future release. To build applications, use the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the deploytool function.

### R2021a

Version: 6.10

**New Features** 

**Bug Fixes** 

### COM Component Integration: Create a COM component using the compiler.build.comComponent function

You can create COM components from the MATLAB command prompt using the compiler.build.comComponent function. This function, along with the compiler.build.COMComponentOptions function, provides an improved interface to specify various options associated with creating COM components. You can then integrate the generated COM component into any COM-compliant application. These functions are in addition to the existing Library Compiler app you use to create COM components.

### C++ Integration: Create a C++ shared library using the compiler.build.cppSharedLibrary function

You can create a C++ shared library from the MATLAB command prompt using the compiler.build.cppSharedLibrary function. This function, along with the compiler.build.CppSharedLibraryOptions function, provides an improved interface to specify various options associated with creating C++ shared libraries. You can then integrate the generated C++ shared library into any C++ application. These functions are in addition to the existing **Library Compiler** app you use to create shared libraries.

### C Integration: Create a C shared library using the compiler.build.cSharedLibrary function

You can create a C shared library from the MATLAB command prompt using the compiler.build.cSharedLibrary function. This function, along with the compiler.build.CSharedLibraryOptions function, provides an improved interface to specify various options associated with creating C shared libraries. You can then integrate the generated C shared library into any C application. These functions are in addition to the existing **Library Compiler** app you use to create shared libraries.

### .NET Integration: Create a .NET assembly using the compiler.build.dotNETAssembly function

You can create a .NET assembly from the MATLAB command prompt using the compiler.build.dotNETAssembly function. This function, along with the compiler.build.DotNETAssemblyOptions function, provides an improved interface to specify various options associated with creating a .NET assembly. These functions are in addition to the existing Library Compiler app you use to create .NET assemblies.

### Java: Create a Java package using the compiler.build.javaPackage function

You can create a Java® package from the MATLAB command prompt using the compiler.build.javaPackage function. This function, along with the compiler.build.JavaPackageOptions function, provides an improved interface to specify various options associated with creating Java packages. You can then integrate the generated Java package into any Java application. These functions are in addition to the existing Library Compiler app you use to create Java packages.

### Python integration: Create a Python package using the compiler.build.pythonPackage function

You can create a Python package from the MATLAB command prompt using the compiler.build.pythonPackage function. This function, along with the compiler.build.PythonPackageOptions function, provides an improved interface to specify various options associated with creating Python packages. You can then integrate the generated Python package into any Python application. These functions are in addition to the existing **Library Compiler** app you use to create Python packages.

#### Functionality Being Removed or Changed

#### -build and -package options in the library Compiler function will be removed ${\it Warns}$

The -build and -package options in the libraryCompiler function will be removed in a future release. To build applications, use one of the compiler.build family of functions or the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the libraryCompiler function.

Also, there are no changes to the **Library Compiler** app. You can continue to the use the app to generate libraries.

#### -build and -package options in the production ServerCompiler function will be removed Warns

The -build and -package options in the productionServerCompiler function will be removed in a future release. To generate deployable archives, use the compiler.build.productionServerArchive function, or the mcc command, or the **Production Server Compiler** app. There are no changes to the other options of the productionServerCompiler function.

Also, there are no changes to the **Production Server Compiler** app. You can continue to the use the app to generate deployable archives.

#### -build and -package options in the deploytool function will be removed ${\it Warns}$

The -build and -package options in the deploytool function will be removed in a future release. To build applications, use the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the deploytool function.

### R2020b

Version: 6.9

**New Features** 

**Bug Fixes** 

#### Java Package Integration: Support for MATLAB string array

You can now create Java packages from MATLAB code that consists of string arrays.

### MATLAB Production Server: Create deployable archives using the compiler.build.productionServerArchive function

You can create deployable archives from the MATLAB command prompt using the compiler.build.productionServerArchive function. This function, along with the compiler.build.ProductionServerArchiveOptions function, provides an improved interface to specify various options associated with creating deployable archives. You can then copy or upload the deployable archives to the server. These functions are in addition to the existing mcc command you use to create deployable archives.

#### Python Version Support for Python Package Integration

Support	Python Version	Platform
Added	Python 3.8	All platforms

#### **Functionality Being Removed or Changed**

#### -build and -package options in the library Compiler function will be removed Warns

The -build and -package options in the libraryCompiler function will be removed in a future release. To build applications, use the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the libraryCompiler function.

Also, there are no changes to the **Library Compiler** app. You can continue to the use the app to generate libraries.

#### -build and -package options in the production ServerCompiler function will be removed Warns

The -build and -package options in the productionServerCompiler function will be removed in a future release. To generate deployable archives, use the mcc command or **Production Server Compiler** app. There are no changes to the other options of the productionServerCompiler function.

Also, there are no changes to the **Production Server Compiler** app. You can continue to the use the app to generate deployable archives.

#### -build and -package options in the deploytool function will be removed $\it Warns$

The -build and -package options in the deploytool function will be removed in a future release. To build applications, use the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the deploytool function.

### R2020a

Version: 6.8

**New Features** 

**Bug Fixes** 

#### .NET Assembly Integration: Support for MATLAB string array

You can now create .NET assemblies from MATLAB code that consists of string arrays.

#### C Shared Library Integration: Support for MATLAB string array

You can now create C shared libraries from MATLAB code that consists of string arrays.

#### C++ Shared Library Integration: Support for MATLAB string array

You can now create C++ shared libraries from MATLAB code that consists of string arrays.

#### **Functionality Being Removed or Changed**

#### -build and -package options in the library Compiler function will be removed $\mathit{Still\ runs}$

The -build and -package options in the libraryCompiler function will be removed in a future release. To build applications, use the mcc command. To package and create an installer, use the compiler.package.installer function. There are no changes to the other options of the libraryCompiler function.

Also, there are no changes to the **Library Compiler** app. You can continue to the use the app to generate libraries.

#### -build and -package options in the production ServerCompiler function will be removed $Still\ runs$

The -build and -package options in the productionServerCompiler function will be removed in a future release. To generate deployable archives, use the mcc command or **Production Server Compiler** app. There are no changes to the other options of the productionServerCompiler function.

Also, there are no changes to the **Production Server Compiler** app. You can continue to the use the app to generate deployable archives.

### R2019b

Version: 6.7

**New Features** 

**Bug Fixes** 

### .NET Assembly Integration: Run .NET assemblies integrated into .NET applications on Linux and macOS using .NET Core

Previously, MATLAB functions packaged into .NET assemblies and integrated with .NET applications could only run on a Windows® platform. Starting in R2019b, you can build these .NET applications on a Windows platform and run them on Linux® and macOS using .NET Core. To use this functionality, you must have .NET Core 2.0 or higher installed. For an example, see Create a .NET Core Application That Runs on Linux and macOS.

#### **Compatibility Considerations**

- To use this functionality, you must have Visual Studio<sup>®</sup> and .NET Core 2.0 or higher.
- If you have version 15.8.2 of Visual Studio 2017 installed, then you do not need to install .NET Core 2.0 or higher separately.
- Visual Studio 2017 is not required for building .NET Core applications. .NET Core comes with its own command line tools that let you create, build, and run .NET Core applications. For more information, see the .NET Core documentation. However, you will need Visual Studio to generate a .NET assembly using the Library Compiler app in MATLAB Compiler SDK.

### File Versioning: Generate system-level file versioning on Windows systems for files created using deployment apps

On Windows systems, you can now generate target files with system-level version numbers using deployment apps. System-level versioning of files is supported for the following targets:

- · C shared libraries
- C++ shared libraries
- .NET assemblies
- COM components

You can specify the version number in the version section of any of the deployment apps by using the mcc command. For more information, see mcc.

#### **Compatibility Considerations**

System-level file versioning for files created using deployment apps is supported on Windows systems only.

### R2019a

**Version: 6.6.1** 

**New Features** 

**Bug Fixes** 

### Library Compiler: Generate programming language specific sample driver code

When you package MATLAB code into a programming language specific component using the **Library Compiler** app, you can simultaneously generate sample driver code that can be used to build and run that component.

To generate sample driver code:

- 1 Open the **Library Compiler** app from the **Apps** tab in the toolstrip.
- 2 Select the type of component you want to generate and add the MATLAB files you want to package.
- **3** Expand the Samples section in the app and click **Create sample**.
- 4 In the resulting context menu, select the MATLAB file for which you want to generate sample code. This will automatically create a sample MATLAB script that calls the MATLAB function in the selected file. You can edit this file or leave it as it is.

Alternatively, if you have an existing MATLAB script that can serve as a sample, add that file instead of creating a sample.

If you do not create or add a sample MATLAB script, sample driver code will not be generated.

5 Click the **Package** button after creating or adding a sample MATLAB script. The **Library Compiler** app will create the corresponding programming language specific driver code.

The generated sample driver code can be found in the .../
for\_redistribution\_files\_only/samples folder. If you install the application, you can find
the generated sample code in the ...<installLocation>/<applicationName>/
application/samples folder.

Sample driver code generation is supported for the following targets: C++, Java,  $Microsoft^{\$}$  .NET, and Python.

Sample driver code generation is *not* supported for the following targets: C, COM, and Microsoft Excel.

#### **Functionality Being Removed**

Functionality	Result
WebFigures: Support for exporting figures to a deployed Java or .NET application has been removed. The webfigure function will no longer work.	Error

#### **Compatibility Considerations**

To deploy MATLAB apps to the web, consider using Web Apps (MATLAB Compiler).

### R2018b

Version: 6.6

### R2018a

Version: 6.5

**New Features** 

**Bug Fixes** 

### C++ Shared Library Integration: Generate C++ shared libraries using MATLAB Data API

You can generate C++ shared libraries using the new MATLAB Data API that leverages modern C++ semantics and design patterns. The **Library Compiler** app automatically generates shared libraries using both the old mwArray API and the new MATLAB Data API. The new shared library files can be found in a folder named v2 in the project folder.

For more information, see Integrate a C++ MATLAB Data API Shared Library into an Application.

#### **Compatibility Considerations**

Shared libraries created with the mwArray API cannot be interchangably used with files generated using the MATLAB Data API.

#### Compiler Support for C/C++ Shared Library Integration

Support	Compiler	Platform
Added	GNU® gcc 6.x	Linux
Added	Xcode 9.x	macOS

For an up-to-date list of supported compilers, see the Supported and Compatible Compilers website.

#### **Python Version Support for Python Package Integration**

Support	Python Version	Platform
Added	Python 3.6, as of R2017b	All platforms
Discontinued	Python 3.4	All platforms

### RESTful API for Service Discovery: Discover MATLAB functions deployed on MATLAB Production Server

You can get information about the archives deployed to MATLAB Production Server and the MATLAB functions contained within those deployed archives using the discovery service API. You can use the API to find out the names of the deployed functions and the number and size and of the inputs and outputs.

For more information, see RESTful API.

### JSON Encode and Decode Functions: Convert data between MATLAB and JSON for MATLAB Production Server

You can convert data in both directions between MATLAB and JSON, including server requests and responses using MATLAB Production Server JSON schema. You can use these functions when making RESTful API calls to the server. For more information, see the following reference pages:

· mps.json.encode

- mps.json.decode
- mps.json.encoderequest
- mps.json.decoderesponse

### R2017b

Version: 6.4

**New Features** 

**Bug Fixes** 

#### **Functionality Being Removed or Changed**

Functionality	Result
WebFigures: Support for exporting figures to a deployed Java or .NET application will be removed in a future release.	Warns

# Compiler Support: Create C/C++ shared libraries and generic COM components using the freely available MinGW-w64 compiler on 64-bit Windows

You can use the MinGW-w64 version 5.3.0 compiler with MATLAB Compiler SDK to generate C/C++ shared libraries and generic COM components. The compiler can be installed as an add-on using the Add-On Explorer in the MATLAB desktop. Use the search term "MinGW-w64" in the Add-On Explorer to find and install the compiler.

You will also need to install the Windows 10 SDK to generate generic COM components using the MinGW-w64 compiler. You can download the Windows 10 SDK from Microsoft.

For an up-to-date list of supported compilers, see the Supported and Compatible Compilers website.

#### Python Package Integration: Support for Python 3.6

You can use Python 3.6 with MATLAB Compiler SDK to generate Python packages that can be integrated with Python applications.

### R2017a

Version: 6.3.1

### R2016b

Version: 6.3

**New Features** 

### 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

### R2016a

Version: 6.2

**New Features** 

### HTTPS Support for Excel Add-ins: Securely integrate Excel Add-ins with MATLAB Production Server using HTTPS

Excel add-ins can securely connect to MATLAB Production Server using HTTPS and evaluate deployed MATLAB functions. Using HTTPS provides transmission-layer encryption. A server certificate must be must be installed when using HTTPS, and can be either self-signed or signed by a trusted Certificate Authority (CA). Self-signed certificates need to be installed in the certificate store of the local machine running the Excel add-in.

For more information, see Change the Server Configuration.

### RESTful API and JSON: Develop clients for MATLAB Production Server in any programming language that supports HTTP

The MATLAB Production Server RESTful API enables you to evaluate MATLAB functions on remote servers using JSON representation of MATLAB data types. You can create client programs in any programming language with an HTTP library.

Client code that uses the MATLAB Production Server RESTful API and JSON representation of MATLAB data types can be written in web-based languages such as JavaScript® and embedded in HTML pages. These web pages can then be used to send and retrieve requests from a MATLAB Production Server instance.

For more information, see RESTful API and JSON.

### R2015aSP1

Version: 6.0.1

### R2015b

Version: 6.1

**New Features** 

### Python Integration: Deploy MATLAB components as native Python packages, for integration with applications written in Python

MATLAB Compiler SDK packages MATLAB functions into software components that you integrate into native Python applications. MATLAB Compiler SDK provides APIs to execute compiled MATLAB functions on the MATLAB Runtime.

For more information, see Python Package Integration.

### R2015a

Version: 6.0

**New Features** 

### Packaging of your MATLAB programs as C/C++ shared libraries, Microsoft .NET assemblies, and Java classes

MATLAB Compiler SDK packages MATLAB functions into software components that you integrate into applications. MATLAB Compiler SDK includes APIs to integrate the packaged components using C, C++, Microsoft .NET, or Java.

### Royalty-free distribution of software components to users who do not need MATLAB

MATLAB functions compiled with MATLAB Compiler SDK execute on the MATLAB Runtime. The MATLAB Runtime is a freely distributable execution engine made up of shared libraries that MATLAB uses to run MATLAB files on systems without an installed version of MATLAB.

### Development and test framework for MATLAB Production Server for integration with web and enterprise systems

MATLAB Compiler SDK includes everything you need to develop client applications for MATLAB Production Server:

- Client APIs execute MATLAB functions on the server.
- Excel add-in to execute MATLAB functions on the server.
- Production Server Compiler app tests the integration between clients and compiles MATLAB functions for deployment to the server.
- Test server tests client code and server configuration together.

#### **Encryption of MATLAB code to protect your intellectual property**

MATLAB Compiler SDK encrypts your MATLAB code so that it cannot be read or copied.