



Protecting and Licensing MATLAB Compiler Standalone Applications with CodeMeter

What is MATLAB Compiler?

MATLAB/Simulink (MathWorks, Inc.) is numerical computing software that is used by engineers in a variety of industries for its rich functions and outstanding data visualization capabilities. MATLAB/Simulink is an excellent choice for AI developing algorithms and designing simulators, including physical models, although the finished algorithms and simulators can be difficult for non-engineers to use as such. MATLAB Compiler enables wider distribution to the public, as it creates an executable file that can be run as a standalone application. Recently, Simulink Compiler was added to do the same for Simulink models, turning them into standalone GUI applications.

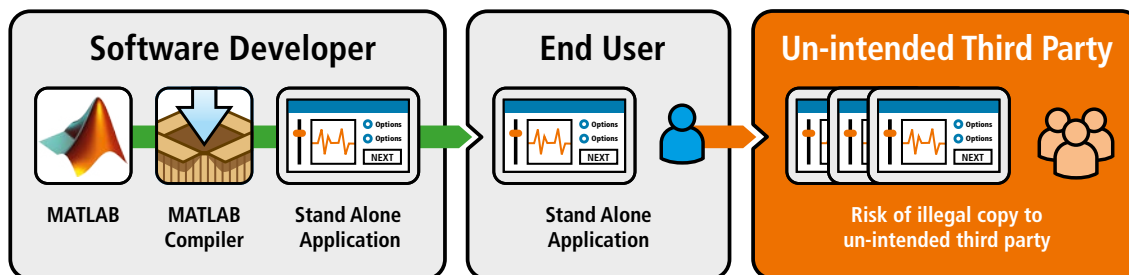
require a MATLAB license and can be readily duplicated, making it easy to distribute them to large audiences. However, there is no technical means to restrict the number of uses or period of use. This calls for a third-party licensing solution whose advanced algorithms can control the intended scope and duration of use, track the allowed number of copies, and secure the intellectual property invested in the application. CodeMeter, Wibu-Systems' proprietary technology, meets these needs by encrypting, protecting, and licensing the application for delivery to end users with the right license configuration and terms of use.

Benefits of being able to license

Standalone applications created using MATLAB Compiler do not

The CodeMeter licenses can be stored on a hardware dongle, in a software container bound to a known device, or in a cloud container.

When no licensing is applied to distributed stand alone application



When CodeMeter is applied to distributed stand alone application

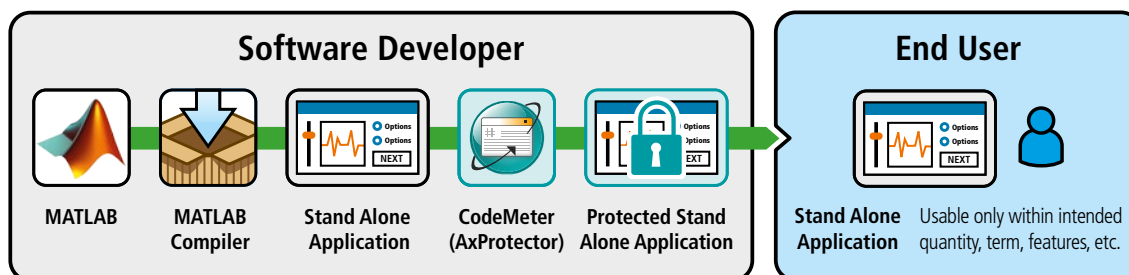


Figure 1: Comparing use cases with and without CodeMeter licensing.

Applying CodeMeter to a MATLAB Compiler sample applications

In a sample MATLAB Compiler application, the included MATLAB code encases the MATLAB function logic to create a magic square as a standalone application. CodeMeter technology will protect this sample application and store the license and the keys in a CmDongle. The protected application will run only when the required stick is connected. For other scenarios, CodeMeter licenses can also be stored directly in software containers on the target device or in cloud containers.

Try to run the standalone application as it is.

The standalone application has already been created on the MATLAB side, and the MATLAB Runtime installation and the "For_redistribution" folder have already been copied to the distribution side (target PC).

Launch the MyAppInstaller_web.exe installer in the For_redistribution folder.

C:\Program Files\magicsquare folder will be created.

Begin by starting the magic square application without changing the folder.

There is no GUI for the magic square application, just a command prompt where you can enter the size of the magic square (matrix size) as a parameter.

```
>application>magicsquare 5
```

In the screenshot below, you can see the 5x5 and 3x3 magic squares. The application starts and shows these 5x5 and 3x3 magic squares, as no license restrictions are in place.

```
Microsoft Windows [Version 10.0.18363.1310]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\>cd C:\Program Files\magicsquare\application
C:\Program Files\magicsquare\application>magicsquare 5

m =

    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9

C:\Program Files\magicsquare\application>magicsquare 3

m =

     8     1     6
     3     5     7
     4     9     2

C:\Program Files\magicsquare\application>
```

Figure 2: Execution screen of the standalone application (Magic Square) created with MATLAB Compiler.

Apply licensing with CodeMeter

Use CodeMeter AxProtector to protect the standalone application (magicsquare.exe) and assign a "Product Code" to bind the license to. With AxProtector, you can protect and license the application by simply specifying the file, without changing the source code.

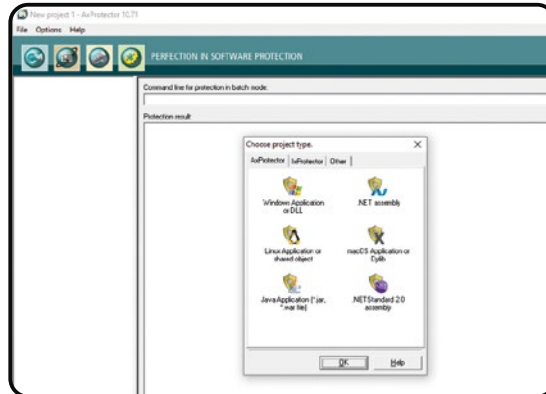


Figure 3: Choosing the project type

In this case, magicsquare.exe is a Windows application, so you select „Windows Application or DLL“ as the AxProtector project type.

As Source File (original/unprotected file), you select magicsquare.exe, while for the Destination File (protected/licensed file), the folder is structured as shown in Figure 4, and the file is given the new name magicsquareP.exe.

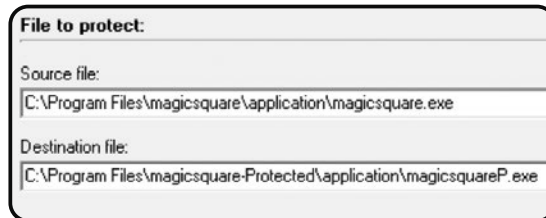


Figure 4: Specifying the name of the original file and the name of the licensed file.

In the Licensing Systems tab, set Firm Code and Product Code (see Figure 5).

Firm Code: 6000010 (Firm Code for evaluation)

Product Code: 14

The Firm Code is a number that Wibu-Systems assigns to an Independent Software Vendor to make sure that all licenses used worldwide are unique. The Product Code that the ISV selects and assigns to individual licenses, each characterized by a specific set of products or functions.

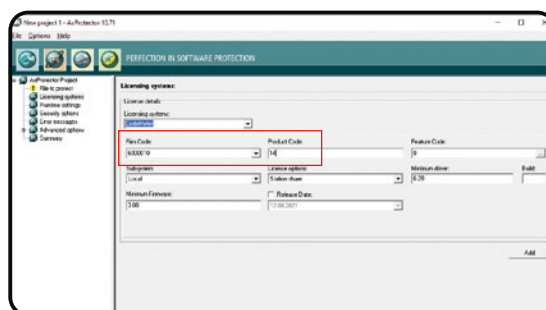


Figure 5: Setting Firm Code and Product Code to bind the application to the license.

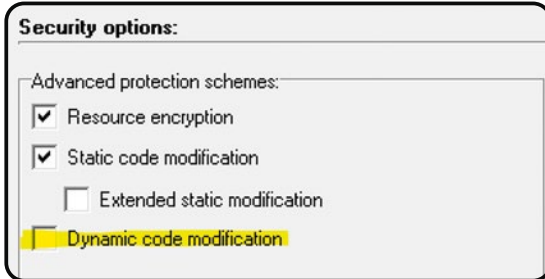


Figure 6: Security Options screen

In the Security Options tab, uncheck „Dynamic code modification“ (see Figure 6).

This setting adds dynamic security checks to the licenses application, which are not compatible with the MATLAB Compiler and needs to be turned off.

If you try to launch the licensed application (magicsquareP.exe), as expected, you will get an error message alerting you that the license is not found, as shown in Figure 7.

Next, create a license with access rights for this standalone application and save it on the USB dongle (see Figure 8).

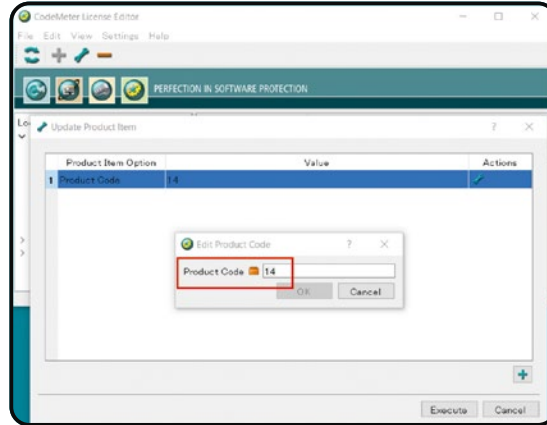


Figure 8: Writing license information to the dongle in CodeMeter License Editor.

The Product Code should be the same as the Product Code „14“ that you set when you encrypted the standalone application (magicsquare.exe) using AxProtector.

The application was launched successfully, and you are able to create a magic square with matrix sizes of 5 and 7. Although this example uses a dongle for the container, CodeMeter also allows you to create a software or a cloud container.

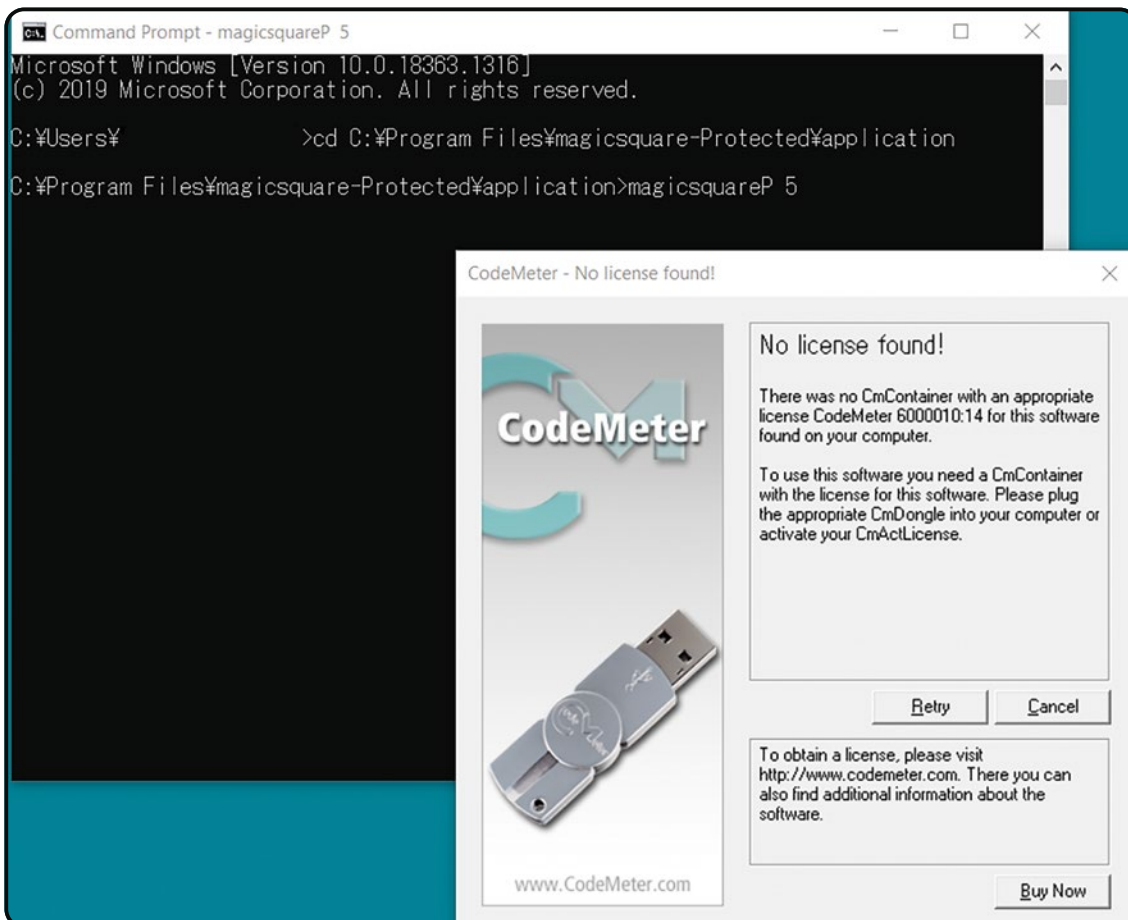


Figure 7: License error display.

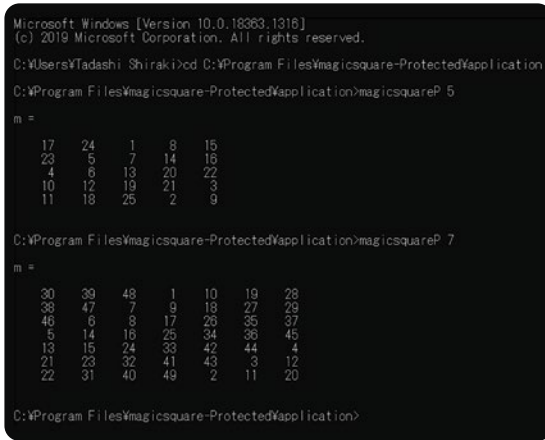


Figure 9: Magic square application execution screen after applying licensing.

Applying CodeMeter to Simulink Compiler sample applications

The spring, mass, and damper simulation applications are provided as examples in Simulink Compiler.

CodeMeter licensing can be applied in the same way as for MATLAB Compiler.

Summary

Standalone applications created with MATLAB Compiler or Simulink Compiler can be easily distributed, but are hard to control in terms of their intended scope of use. With CodeMeter, you can apply licensing in a simple step, not only to prevent leaks or unauthorized copies, but also to build a software business with the optimal license model to fit the business model, such as the number of products that can be used at the destination, license period, features, or versions as needed. Software is increasingly equipped with advanced algorithms and intellectual property that deserve to be controlled i.e. image recognition, machine learning, sensing, AI, physical simulation, etc. CodeMeter helps you protect your applications and monetize your software business.

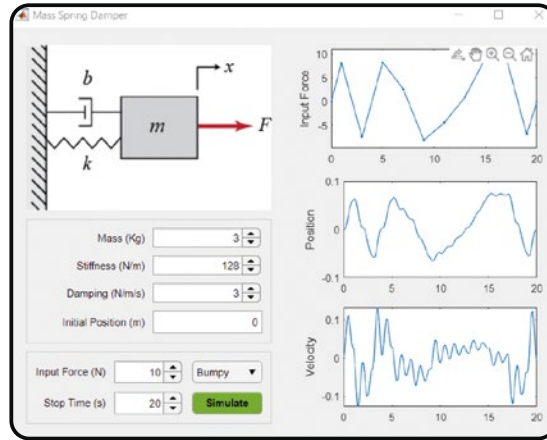


Figure 10: Simulink Compiler sample spring, mass, and damper simulation application.

Integration with CodeMeter API

In addition to encrypting the application's code, CodeMeter API can be integrated into the MATLAB code.

CodeMeter API is a licensing and cryptographic interface used for the following operations:

- Checking licenses
- Reading license properties
- Logging into other systems
- Encrypting communication
- Symmetrical and asymmetric cryptography
- Hash functions and challenge-response checks
- Encrypting own data in the MATLAB application
- Encrypting configuration or other data files

To integrate CodeMeter API, the CodeMeter .dll must be linked in MATLAB, for example using the following command line:

```
lcc-implib -u %WINDIR%\System32\WibuCm32.dll.
```