Matlab to C++ Code

Contents

[Compiling a Shared Library 2](#_Toc324494418)

[Testing Components on Development Machine 2](#_Toc324494419)

[Windows. 2](#_Toc324494420)

[Linux. 2](#_Toc324494421)

[Suppressing Warnings on Linux 2](#_Toc324494422)

*For version R14 and Compiler 4* (<http://www.mathworks.com/help/releases/R14/toolbox/compiler/>)

## Compiling a Shared Library

To create a shared library from the M-file mymfunction, use the command

* mcc -l mymfunction.m

## Testing Components on Development Machine

To test either component on your development machine, make sure you have your path set properly.

### Windows.

1. Add the following directory to your system PATH environment variable.

<matlabroot>\bin\win32

### Linux.

1. Add the following directories to your dynamic library path.

|  |
| --- |
| **Note**   For readability, the following command appears on separate lines, but you must enter it all on one line. |

* setenv LD\_LIBRARY\_PATH
* <matlabroot>/bin/glnx86:
* <matlabroot>/sys/os/glnx86:
* <matlabroot>/sys/java/jre/glnx86/jre1.4.2/lib/i386/client:
* <matlabroot>/sys/java/jre/glnx86/jre1.4.2/lib/i386:
* <matlabroot>/sys/opengl/lib/glnx86:${LD\_LIBRARY\_PATH}
* setenv XAPPLRESDIR <matlabroot>/X11/app-defaults

You can then run the compiled applications on your development machine to test them.

## Suppressing Warnings on Linux

Several warnings may appear when you run a stand-alone application on Linux. This section describes how to suppress these warnings.

To suppress the app-defaults warnings, set XAPPLRESDIR to point to <mcr>/X11/app-defaults.

To suppress the libjvm.so warning, place the following on the LD\_LIBRARY\_PATH.

* <mcr>/sys/java/jre/glnx86/jre1.4.1/lib/i386/i386
* <mcr>/sys/java/jre/glnx86/jre1.4.1/lib/i386/client

Alternately, you can use the MATLAB Compiler option -R -nojvm to set your application's nojvm run-time option, if the application is capable of running without Java.

## Compilation process

* Matlab Compiler 4 uses the Matlab Component Runtime (MCR)
  + A stand-alone set of shared libraries that enables the execution of m-files.
  + Provides complete support for all features of the MATLAB language.
* Matlab Compiler 4 also uses a Component Technology File (CTF) archive that houses the deployable package.
  + All m-files are encrypted in the CTF archive using AES cryptosystem.
  + Each shared library produced by the library compiler has an associated CTF archieve.
* Dependencies
* The compiler examines all the dependencies of the included m-file(s) and includes all existing dependencies.
* Wrapper Code Generation
* The compilation process produces:
  + C/C++ interface code for each m-file supplied to the command line.
  + Data file that has all information needed to run encrypted m-files.
* Compilation and Linking
* This step compiles the c code into object code and then links with object code with necessary matlab libraries to create the finished component.

### Porting Generated Code to a Different Platform

You cannot run the Compiler in Windows and then use it in linux. You have to run compiler within linux to use it in that machine. See the following description:

“Since binary formats are different on each platform, the various components generated by the MATLAB Compiler cannot be moved from platform to platform as is. You can distribute a MATLAB Compiler-generated application to any target machine that has the same operating system as the machine on which the application was compiled. For example, if you want to deploy an application to a Windows machine, you must use the Windows version of the MATLAB Compiler to build the application on a Windows machine.

To deploy an application to a machine, whose operating system is different than the machine used to develop the application, requires recompiling. You must recompile the application on the desired targeted platform. For example, If you want to deploy the previous application that was developed on a Windows machine to a Linux machine, you must use the MATLAB Compiler on a Linux machine and completely rebuild the application. Consequently, you must have a valid MATLAB Compiler license on both platforms in order to be able to do this.”

## Installing the MCR on a Deployment Machine

Before end users can run MATLAB Compiler-generated components on their machines, they need to install the MCR, if it is not already present. You only need to install the MCR one time on a deployment machine