Build standalone executables from MATLAB code

P. Legrand

ALEA INRIA Team
IMB, institut de mathématiques de Bordeaux, UMR CNRS 5251
UFR Sciences et Modélisation

19/06/2012

Introduction

In this presentation,

we will learn how to build a windows standalone executable code from a MATLAB *.m code.

This opportunity can become very interesting when:

1. you collaborate with someone who does not work with MATLAB.
2. you want to deploy a software or a full toolbox working without MATLAB.

MATLAB Compiler lets you share your MATLAB application as an executable. Executables created with MATLAB Compiler use a runtime engine called the MATLAB Compiler Runtime (MCR). The MCR is provided with MATLAB Compiler for distribution with your application and can be deployed royalty-free.

Build executable

```
mcc -mv yourcode
```

Build library

```
mcc -lv yourcode
```

Passing Arguments to and from a Standalone Application

Input

- To pass a file called *helpfile* to the compiled function called *filename*, use ***filename helpfile***
- To pass numbers or letters (e.g., 1, 2, and 3), do not separate the arguments with commas, use ***filename 1 2 3***
- To pass matrices as input, use ***filename "[1 2 3]" "[4 5 6]"***
- You have to use the double quotes around the input arguments if there is a space in it

The input arguments you pass to your application from a system prompt are considered as string input

You can determine at run time whether or not to do this by using the ***isdeployed*** function. If your MATLAB file expects numeric inputs in MATLAB, the code can check whether it is being run as a standalone application. For example:

function myfun (n1, n2) if (isdeployed) n1 = str2num(n1); n2 = str2num(n2); end

Output

You cannot return back values from your standalone application to the user. The only way to return values from compiled code is to either display it on the screen or store it in a file.

Prerequisites for Deployment and Files to Deploy

1. Verify the MATLAB Compiler Runtime (MCR) is installed and ensure you have installed matching version.
2. If the MCR is not installed, run MCRInstaller, located in: $\backslash toolbox \setminus compiler \setminus deploy \setminus win64 \setminus MCRInstaller.exe$
3. Files to Deploy and Package for Standalone
   - MCRInstaller.exe
   - Yourcode.exe

Very important

Use the MCRInstaller.exe file coming from the computer where you compiled the code.

Build a graphical interface

We will build a program with a graphical interface.

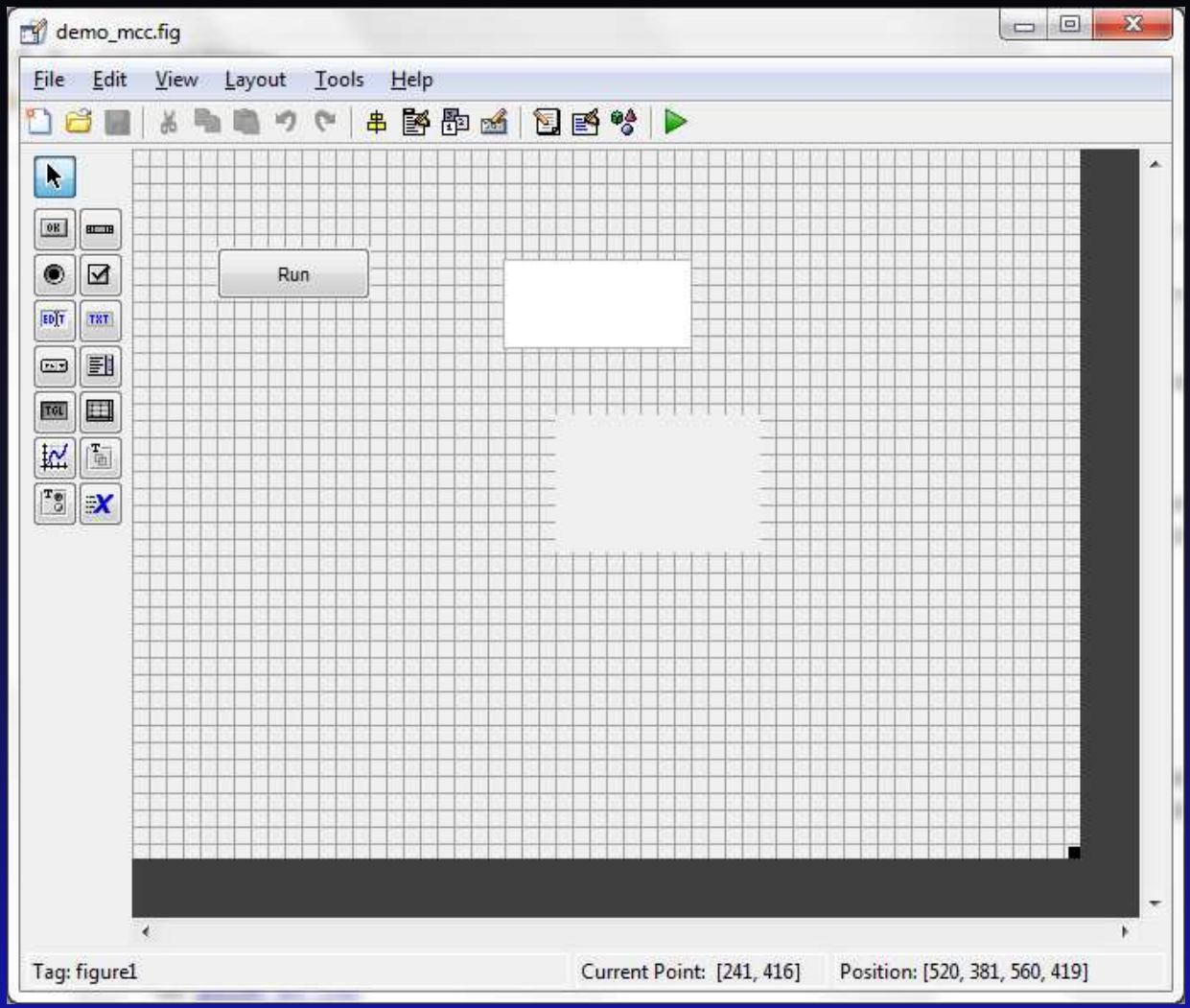


Figure: Build a graphical interface with Guide

With a call to an external function

We link the run button with an external function.

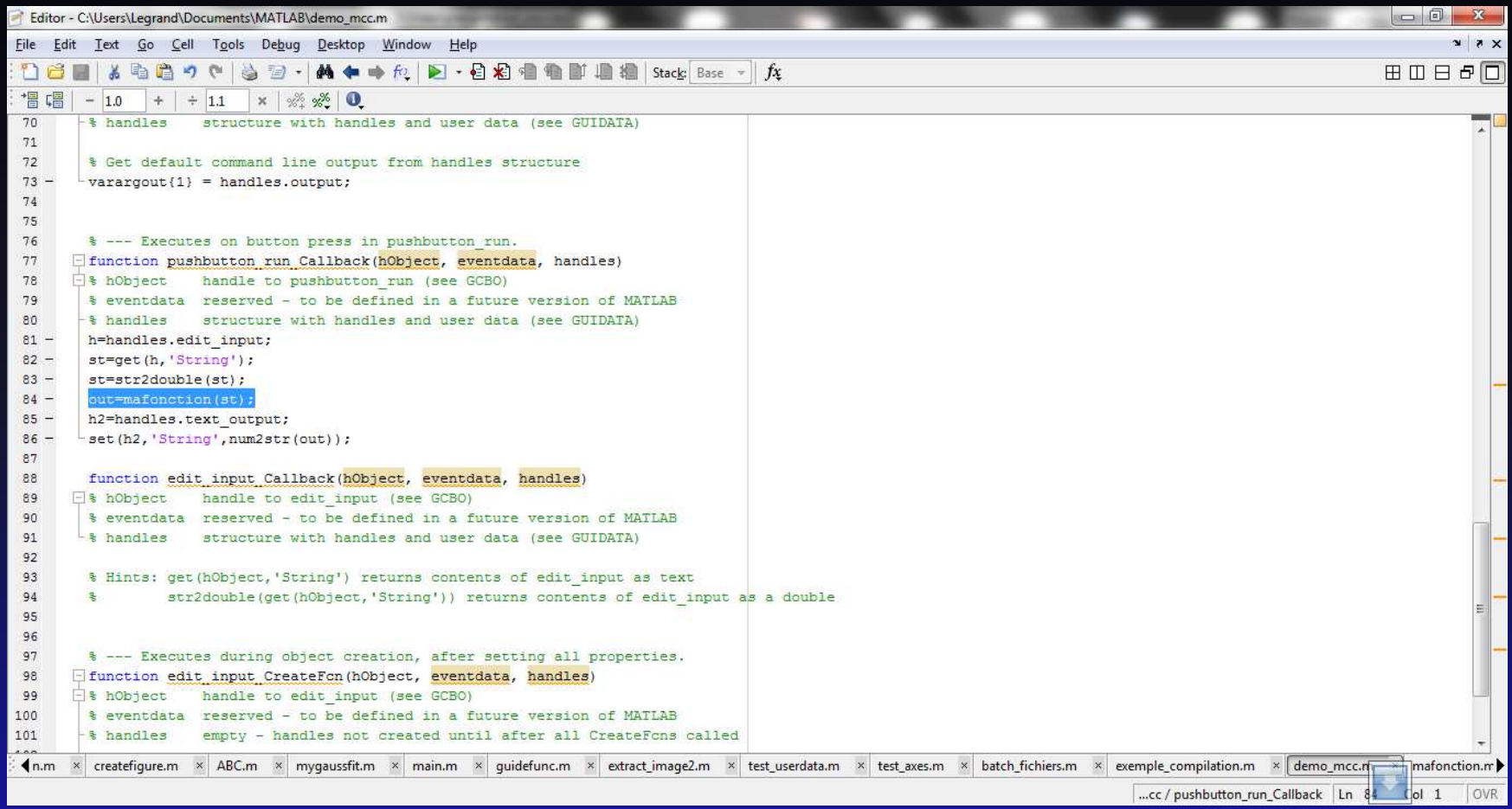


Figure: Matlab Desktop

With a call to an external function

We write the function.

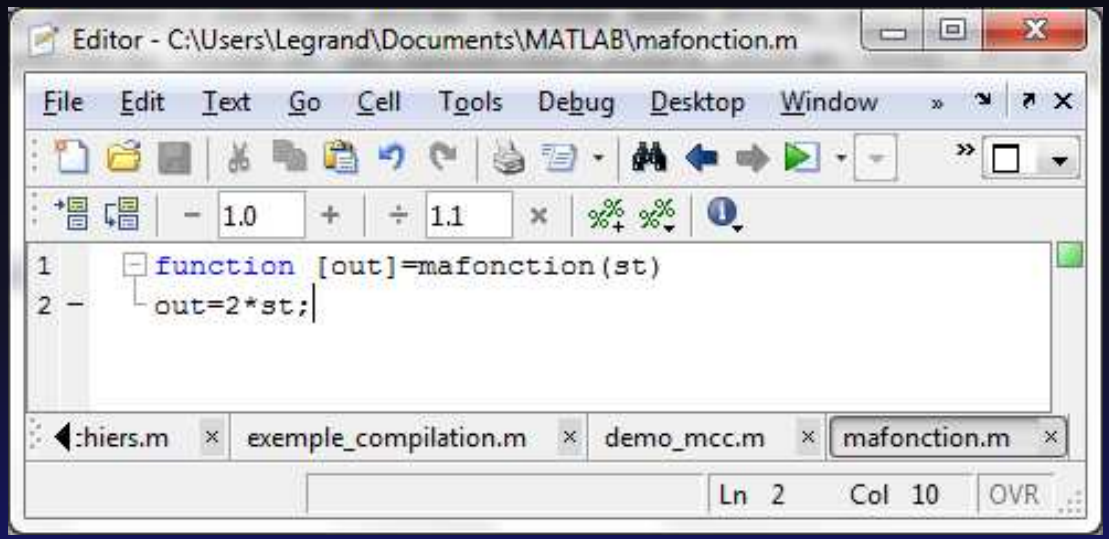


Figure: Matlab Editor

Use the compiler

From the Matlab command window, we start the compilation with:

$mcc\ -mv\ demo\_mcc$

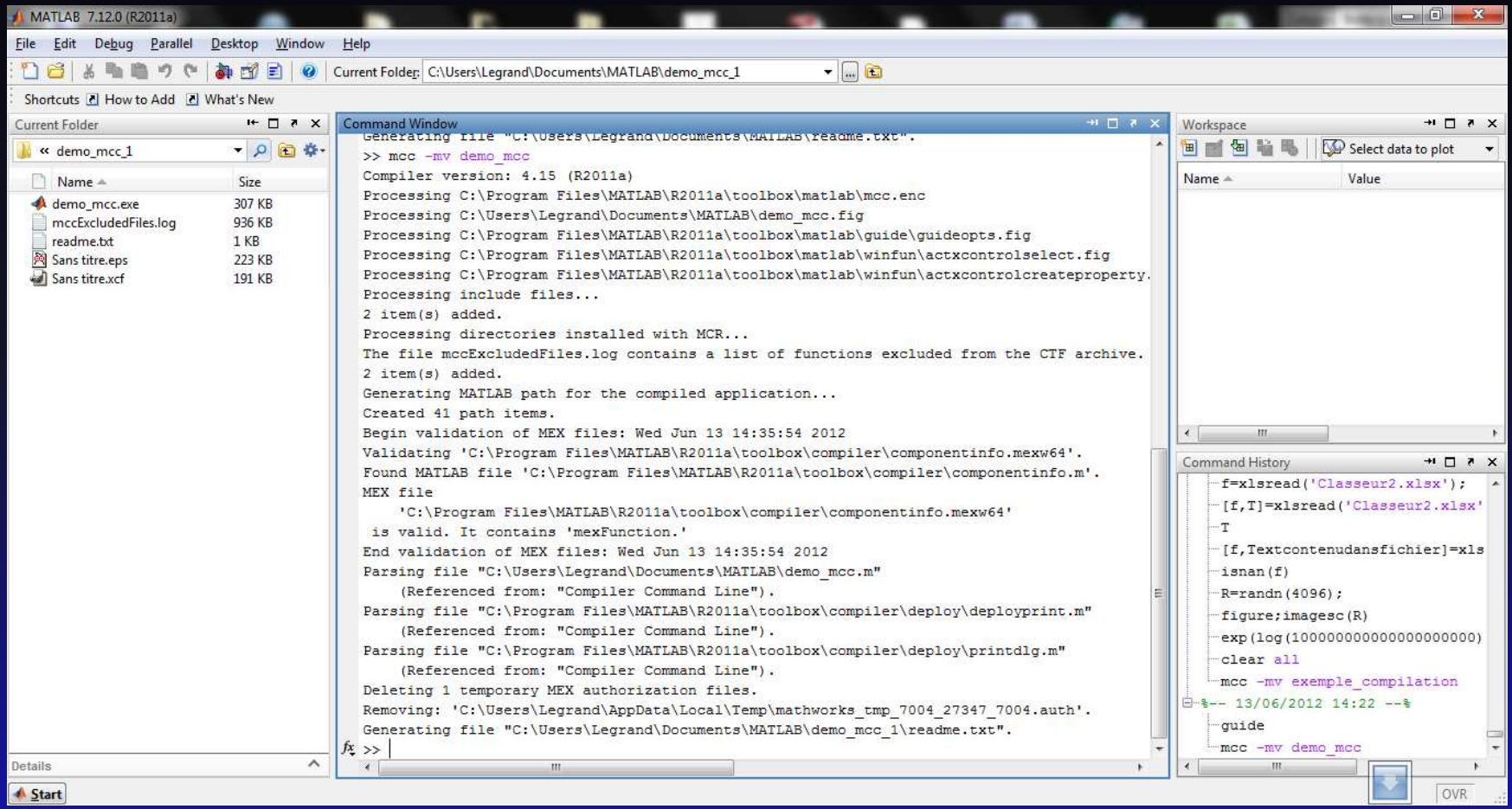


Figure: Matlab command window

Files

Some files are created inside the current directory.

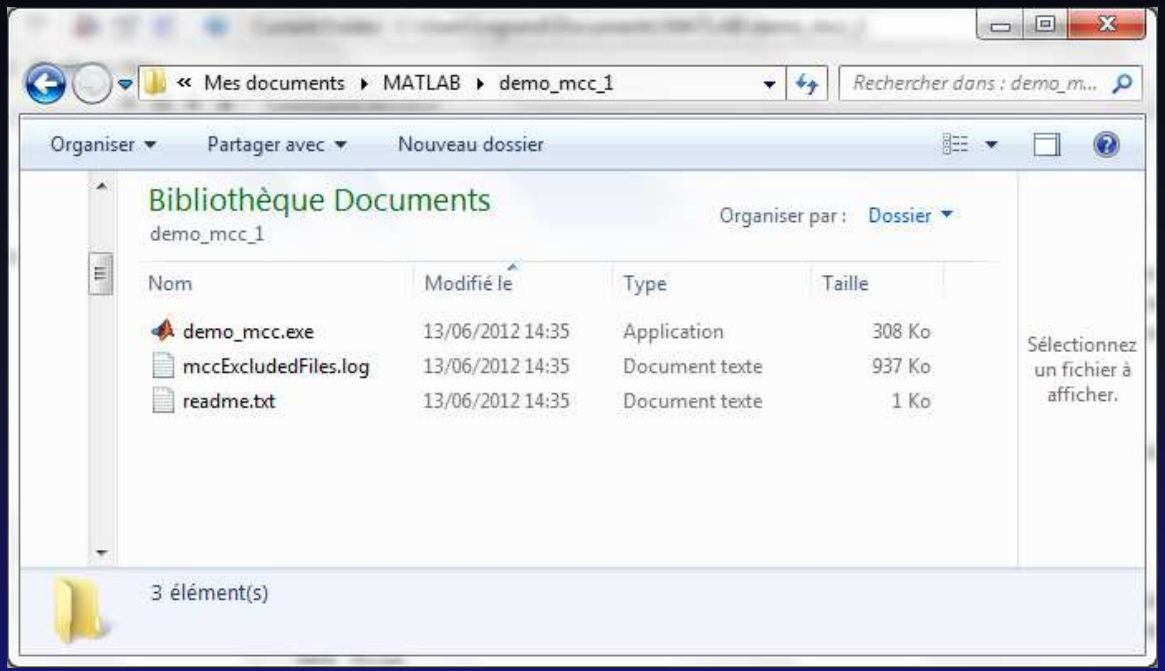


Figure: Current directory

Readme file

The content of the readme file

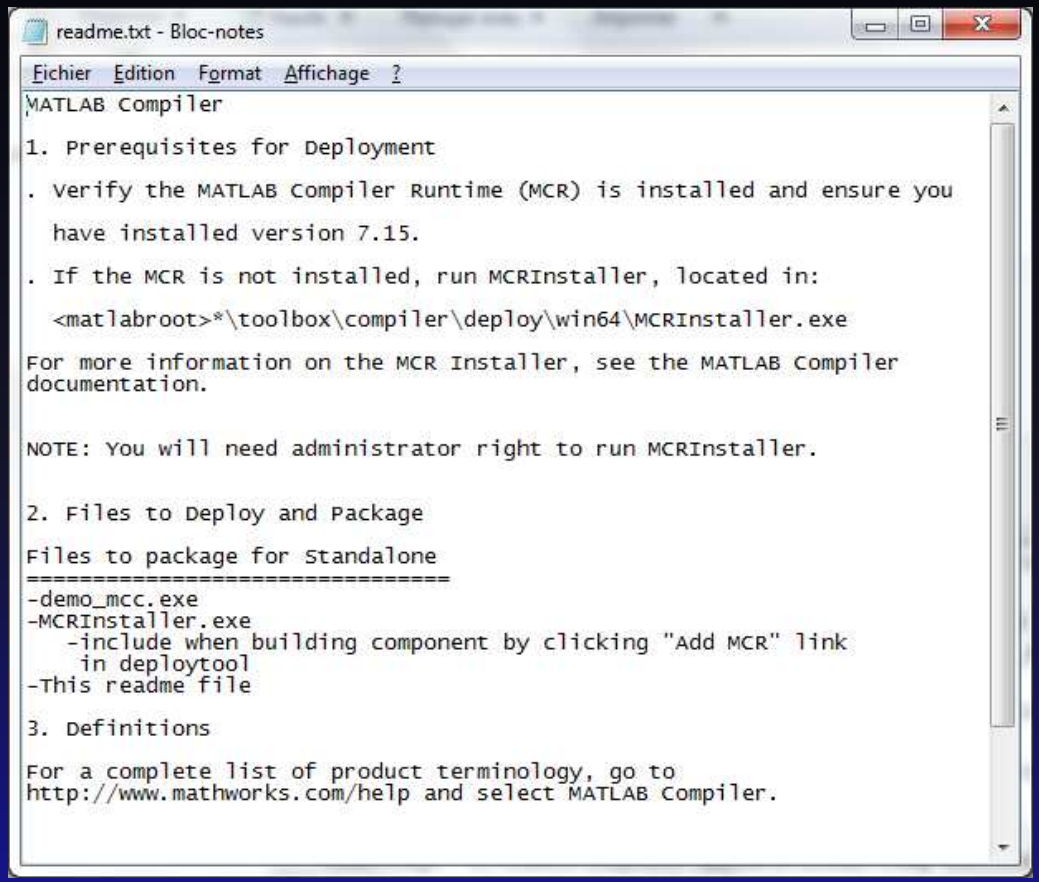
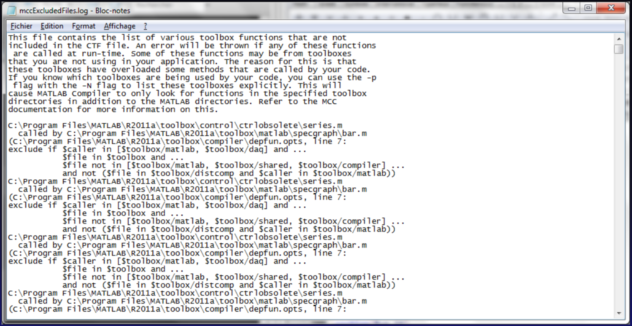


Figure: Content of the readme file

mccExcludedFiles.log file

The content of the mccExcludedFiles.log file




```
This file contains the list of various toolbox functions that are not
included in the CTF file. An error will be thrown if any of these functions
 are called at run-time. Some of these functions may be from toolboxes
that you are not using in your application. The reason for this is that
these toolboxes have overloaded some methods that are called by your code.
If you know which toolboxes are being used by your code, you can use the -p
 flag with the -N flag to list these toolboxes explicitly. This will
cause MATLAB Compiler to only look for functions in the specified toolbox
directories in addition to the MATLAB directories. Refer to the MCC
documentation for more information on this.

C:\Program Files\MATLAB\R2011a\toolbox\control\ctrlobsolete\series.m
  called by C:\Program Files\MATLAB\R2011a\toolbox\matlab\specgraph\bar.m
(C:\Program Files\MATLAB\R2011a\toolbox\compiler\depfun.opts, line 7:
exclude if $caller in [$toolbox/matlab, $toolbox/daq] and ...
           $file in $toolbox and ...
           $file not in [$toolbox/matlab, $toolbox/shared, $toolbox/compiler] ...
           and not ($file in $toolbox/distcomp and $caller in $toolbox/matlab))
C:\Program Files\MATLAB\R2011a\toolbox\control\ctrlobsolete\series.m
  called by C:\Program Files\MATLAB\R2011a\toolbox\matlab\specgraph\bar.m
(C:\Program Files\MATLAB\R2011a\toolbox\compiler\depfun.opts, line 7:
exclude if $caller in [$toolbox/matlab, $toolbox/daq] and ...
           $file in $toolbox and ...
           $file not in [$toolbox/matlab, $toolbox/shared, $toolbox/compiler] ...
           and not ($file in $toolbox/distcomp and $caller in $toolbox/matlab))
C:\Program Files\MATLAB\R2011a\toolbox\control\ctrlobsolete\series.m
  called by C:\Program Files\MATLAB\R2011a\toolbox\matlab\specgraph\bar.m
(C:\Program Files\MATLAB\R2011a\toolbox\compiler\depfun.opts, line 7:
exclude if $caller in [$toolbox/matlab, $toolbox/daq] and ...
           $file in $toolbox and ...
           $file not in [$toolbox/matlab, $toolbox/shared, $toolbox/compiler] ...
           and not ($file in $toolbox/distcomp and $caller in $toolbox/matlab))
C:\Program Files\MATLAB\R2011a\toolbox\control\ctrlobsolete\series.m
  called by C:\Program Files\MATLAB\R2011a\toolbox\matlab\specgraph\bar.m
(C:\Program Files\MATLAB\R2011a\toolbox\compiler\depfun.opts, line 7:
```

Figure: List of excluded files

Some functions are not included in the library.

blabla