

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

Matlab Compiler

- Compiler Commands

- Passing Arguments to and from a Standalone Application

Deployment process

- Prerequisites for Deployment and Files to Deploy

- Very important

Exemple

- Build a graphical interface

- With a call to an external function

- Use the compiler

- Files

- Readme file

- mccExcludedFiles.log file

Run it

Introduction

Matlab Compiler

- Compiler Commands

- Passing Arguments to and from a Standalone Application

Deployment process

- Prerequisites for Deployment and Files to Deploy

- Very important

Exemple

- Build a graphical interface

- With a call to an external function

- Use the compiler

- Files

- Readme file

- mccExcludedFiles.log file

Run it

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.

Introduction

Matlab Compiler

Compiler Commands

Passing Arguments to and from a Standalone Application

Deployment process

Prerequisites for Deployment and Files to Deploy

Very important

Exemple

Build a graphical interface

With a call to an external function

Use the compiler

Files

Readme file

mccExcludedFiles.log file

Run it

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.

Introduction

Matlab Compiler

Compiler Commands

Passing Arguments to and from a Standalone Application

Deployment process

Prerequisites for Deployment and Files to Deploy

Very important

Exemple

Build a graphical interface

With a call to an external function

Use the compiler

Files

Readme file

mccExcludedFiles.log file

Run it

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:
`\toolbox \ compiler \ deploy \ win64 \ 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.

Introduction

Matlab Compiler

Compiler Commands

Passing Arguments to and from a Standalone Application

Deployment process

Prerequisites for Deployment and Files to Deploy

Very important

Exemple

Build a graphical interface

With a call to an external function

Use the compiler

Files

Readme file

mccExcludedFiles.log file

Run it

Build a graphical interface

We will build a program with a graphical interface.

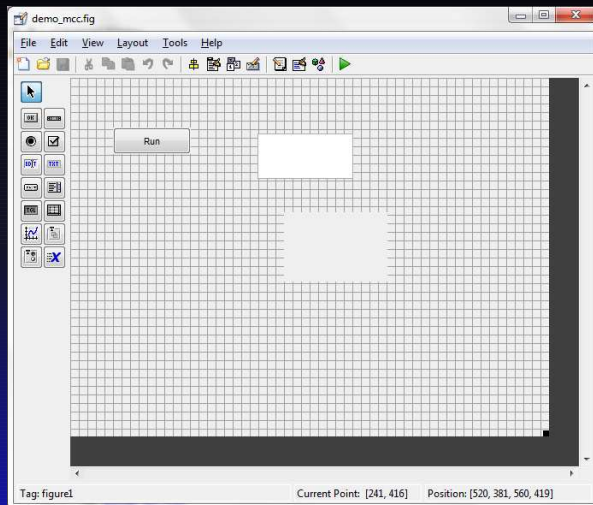
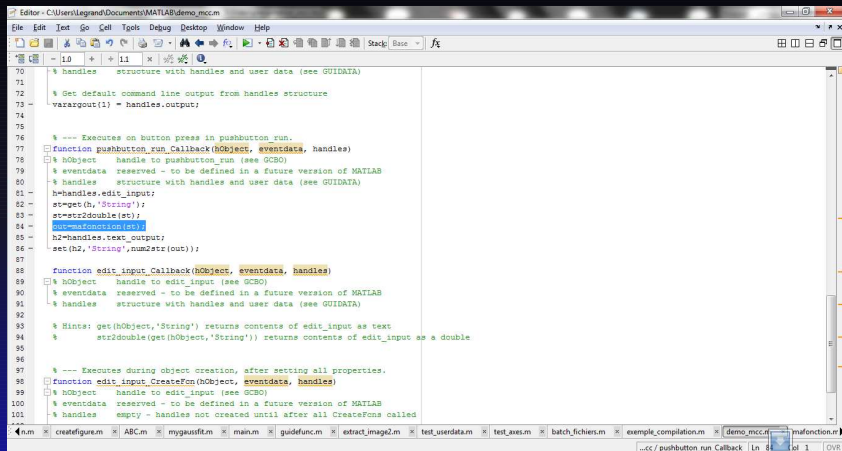


Figure: Build a graphical interface with Guide

With a call to an external function

We link the run button with an external function.



```
70 % handles structure with handles and user data (see GUIDATA)
71
72 % Get default command line output from handles structure
73 varargin(1) = handles.output;
74
75
76 % --- Executes on button press in pushbutton_run.
77 function pushbutton_run_Callback(hObject, eventdata, handles)
78 % hObject handle to pushbutton_run (see GCBO)
79 % eventdata reserved - to be defined in a future version of MATLAB
80 % handles structure with handles and user data (see GUIDATA)
81 h=handles.edit_input;
82 st=get(h,'String');
83 st=str2double(st);
84 out=mafuction(st);
85 h2=handles.text_output;
86 set(h2,'String',num2str(out));
87
88 function edit_input_Callback(hObject, eventdata, handles)
89 % hObject handle to edit_input (see GCBO)
90 % eventdata reserved - to be defined in a future version of MATLAB
91 % handles structure with handles and user data (see GUIDATA)
92
93 % Hints: get(hObject,'String') returns contents of edit_input as text
94 %       str2double(get(hObject,'String')) returns contents of edit_input as a double
95
96
97 % --- Executes during object creation, after setting all properties.
98 function edit_input_CreateFcn(hObject, eventdata, handles)
99 % hObject handle to edit_input (see GCBO)
100 % eventdata reserved - to be defined in a future version of MATLAB
101 % handles empty - handles not created until after all CreateFcns called
***
```

Figure: Matlab Desktop

With a call to an external function

We write the function.

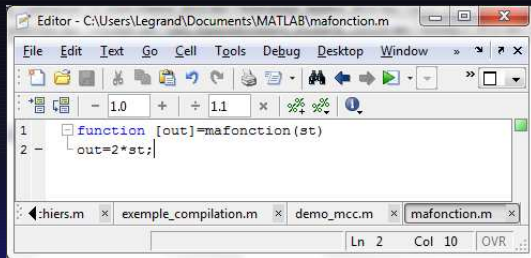


Figure: Matlab Editor

Use the compiler

From the Matlab command window, we start the compilation with:
mcc -mv demo_mcc

```

MATLAB 7.12.0 (R2011a)
File Edit Debug Parallel Desktop Window Help
Current Folder: C:\Users\Legrand\Documents\MATLAB\demo_mcc_1
Current Folder
demo_mcc_1
demo_mcc.exe 307 KB
mccExcludedFiles.log 936 KB
readme.txt 1 KB
Sans titre.eps 223 KB
Sans titre.xcf 191 KB
Command Window
Generating file "C:\Users\Legrand\Documents\MATLAB\readme.txt".
>> mcc -mv demo_mcc
Compiler version: 4.15 (R2011a)
Processing C:\Program Files\MATLAB\R2011a\toolbox\matlab\moc.enc
Processing C:\Users\Legrand\Documents\MATLAB\demo_mcc.fig
Processing C:\Program Files\MATLAB\R2011a\toolbox\matlab\guide\guideopts.fig
Processing C:\Program Files\MATLAB\R2011a\toolbox\matlab\winfun\actxcontrolselect.fig
Processing C:\Program Files\MATLAB\R2011a\toolbox\matlab\winfun\actxcontrolcreateproperty.
Processing include files...
2 item(s) added.
Processing directories installed with MCR...
The file mocExcludedFiles.log contains a list of functions excluded from the CTF archive.
2 item(s) added.
Generating MATLAB path for the compiled application...
Created 41 path items.
Begin validation of MEX files: Wed Jun 13 14:35:54 2012
Validating 'C:\Program Files\MATLAB\R2011a\toolbox\compiler\componentinfo.mexw64'.
Found MATLAB file 'C:\Program Files\MATLAB\R2011a\toolbox\compiler\componentinfo.m'.
MEX file
'C:\Program Files\MATLAB\R2011a\toolbox\compiler\componentinfo.mexw64'
is valid. It contains 'mexFunction.'
End validation of MEX files: Wed Jun 13 14:35:54 2012
Parsing file "C:\Users\Legrand\Documents\MATLAB\demo_mcc.m"
(Referenced from: "Compiler Command Line").
Parsing file "C:\Program Files\MATLAB\R2011a\toolbox\compiler\deploy\deployprint.m"
(Referenced from: "Compiler Command Line").
Parsing file "C:\Program Files\MATLAB\R2011a\toolbox\compiler\deploy\printdlg.m"
(Referenced from: "Compiler Command Line").
Deleting 1 temporary MEX authorization files.
Removing: 'C:\Users\Legrand\AppData\Local\Temp\mathworks_tmp_7004_27347_7004.auth'.
Generating file "C:\Users\Legrand\Documents\MATLAB\demo_mcc_1\readme.txt".
Workspace
Name Value
Command History
-[xlswrite('Classeur2.xlsx')];
-[f,T]=xlsread('Classeur2.xlsx');
T
[f,TextContentAndNamesOfCharts]=xls
ismean(f)
R=randin(4096);
figure;imagesc(R)
exp(log(100000000000000000000000000000000))
clear all
mcc -mv exemple_compilation
13/06/2012 14:22 -->
guide
mcc -mv demo_mcc

```

Figure: Matlab command window

Some files are created inside the current directory.

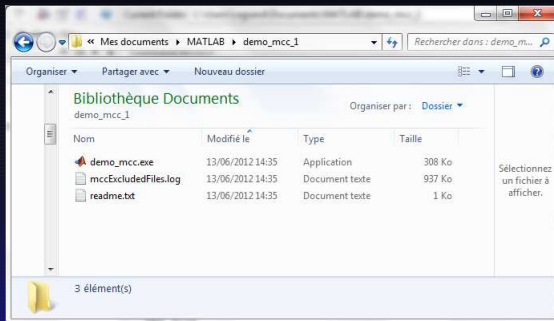
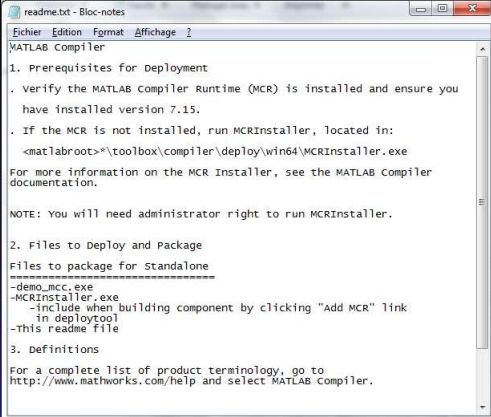


Figure: Current directory

The content of the readme file



```
readme.txt - Bloc-notes
Eichier  Edition  Format  Affichage  ?
MATLAB Compiler
1. Prerequisites for Deployment
. verify the MATLAB Compiler Runtime (MCR) is installed and ensure you
  have installed version 7.15.
. If the MCR is not installed, run MCRInstaller, located in:
  <matlabroot>*\toolbox\compiler\deploy\win64\MCRInstaller.exe
For more information on the MCR Installer, see the MATLAB Compiler
documentation.

NOTE: You will need administrator right to run MCRInstaller.

2. Files to Deploy and Package
Files to package for Standalone
-----
-demo_mcc.exe
-MCRInstaller.exe
  -include when building component by clicking "Add MCR" link
  in deploytool
-This readme file

3. Definitions

For a complete list of product terminology, go to
http://www.mathworks.com/help and select MATLAB Compiler.
```

Figure: Content of the readme file

The content of the mccExcludedFiles.log file

```

Fichier Edition Format Affichage ?
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\ctr_lobsolete\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\ctr_lobsolete\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\ctr_lobsolete\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\ctr_lobsolete\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.

Introduction

Matlab Compiler

Compiler Commands

Passing Arguments to and from a Standalone Application

Deployment process

Prerequisites for Deployment and Files to Deploy

Very important

Exemple

Build a graphical interface

With a call to an external function

Use the compiler

Files

Readme file

mccExcludedFiles.log file

Run it

blabla