

Running stand-alone MATLAB applications on INFN-GRID

Author: M.D'Amato

To build stand-alone C/C++ applications the **MATLAB**, the **MATLAB Compiler**, a **C/C++ Compiler**, **MATLAB C/C++ Math Library** and **MATLAB C/C++ Graphics Library** must be installed on your system. Therefore the stand-alone applications can run without the help of the MATLAB interpreter without install it on the system. However, stand-alone applications require the run-time shared libraries.

The source code for a stand-alone C/C++ application consists of either M-files alone or a combination of M-file, MEX-file and C or C++ source code files.

The MATLAB Compiler, when invoked with the appropriate option flags, translates the input M-files into C/C++ source code suitable for your own stand-alone applications. After compiling this C/C++ source code, the resulting object files are also linked to the MATLAB C/C++ Math Library.

Building UNIX applications with MATLAB Compiler

In the MATLAB command window type:

```
> mbuild -setup
```

The *setup* option creates a user-specific options file for your ANSI C or C++ compiler.

There is MATLAB code for an example, *hello.m*, included in the <matlab>/extern/examples/compiler directory, where <matlab> is the MATLAB root directory.

Hello.m is the following m-file ascii:

```
function hello
% This is the hello, world function written in M code
% $Revision: 1.1 $
%
    fprintf(1,'Hello, World\n' );
```

The command “*mcc*” is the MATLAB command that invokes the MATLAB Compiler; type the following at the MATLAB prompt:

```
> mcc -m hello.m          for translating a M-file in C
```

```
[> mcc -p hello.m        for translating a M-file in C++]
```

When you execute this command it creates the following files: two C-file *hello.c* and *hello_main.c*, one header file *hello.h* and stand-alone C application *hello*.

The MATLAB compiler automatically invokes *mbuild* under certain conditions. In particular, *mcc -m* invokes *mbuild* to perform compilation and linking.

The flag *-m* generates a C stand-alone application. It is equivalent to:

```
-W main -L C -t -T link:exe -h libmmfile.mlib
```

where:

- W type : controls the generation of function wrappers (type=mex, main, simulink, ecc.);
- L language : specifies output target language (language=C, C++);
- t : translates M code to C/C++ code ;
- T target : specifies output stage (target=compile:bin,link:bin, where bin=mex,exe,lib);
- h : automatically, finds and compiles helper functions included in the source M-file;
- lmmfile.mlib : links to this shared library whenever needed.

You can run the *mcc* command from either the MATLAB command prompt (MATLAB mode) or the UNIX command line (stand-alone mode).

Stand-alone application: type *hello* at the UNIX prompt. The application should run and display the message:

```
> Hello, World
```

There are two possibilities:

1. to execute the stand-alone application on the same PC in which the MATLAB interpreter is installed;
2. to execute the stand-alone application on another PC without the help of the MATLAB interpreter.

In the first case before running the stand-alone application, you must tell the system where the API and C/C++ shared libraries reside, by setting the `LD_LIBRARY_PATH` environment variable:

```
# setenv LD_LIBRARY_PATH <matlab>/extern/lib/<arch>:$LD_LIBRARY_PATH
```

where `<matlab>` is the MATLAB root directory and `<arch>` is your architecture (i.e. alpha, glnx86, sol2, ecc).

In the second case the stand-alone application and the run-time shared libraries must to be copied on the other PC. The MATLAB C Math library includes all the MATLAB run-time libraries required by stand-alone applications prepackaged into a single, self-extracting archive file:

```
<matlab>/extern/lib/<arch>/mglinstaller
```

On UNIX system, users run the MATLAB Math and Graphics Run-Time Library Installer by executing the *mglinstaller* at the system prompt. This program extracts the libraries from the archive and installs them in a directory specified by the user. By default, the installer puts the files in the current directory and creates two directories: `./bin` and `./toolbox`.

In `./bin/<arch>` directory are installed MATLAB Math and Graphics libraries and shared libraries. Instead `./toolbox/matlab/<dir>` contains a set the specific MATLAB functions.

The environment variable have to set also in this case.

On all UNIX platforms, the compiler library is shipped as shared object files (with extension `.so`) or shared libraries (`.sl`). Any compiler-generated, stand-alone application must also find the C/C++ libraries such as specified with the library path environment variable (`SHLIB_PATH`, `LIBPATH`, or `LD_LIBRARY_PATH`).

Running stand-alone application jobs on INFN-GRID

A typical JDL file (*matlab.jdl*) needed to run a stand-alone C application job over the INFN-GRID is shown below:

```
Executable="matlab_env.sh";
StdOutput="matlab-out";
StdError="matlab-err";
OutputSandbox={"output", "matlab-out", "matlab-err"};
InputSandBox={"hello", "mglinstaller.sh", "matlab_env.sh"};
```

The executable *matlab_env.sh* is the following script:

```
#!/bin/sh
#
#sets the permissions of the files
/bin/chmod a+x mglinstaller.sh
/bin/chmod a+x hello
#execute MATLAB Math and Graphics Run-Time Library Installer
$PWD/mglinstaller.sh
#set the LD_LIBRARY_PATH environment variable
export LD_LIBRARY_PATH=$PWD/bin/glnx86:$LD_LIBRARY_PATH
#execute the stand-alone C applitation
$PWD/hello > output
exit
```

You must change the name to the file *mglinstaller* into *mglinstaller.sh* since the system does not recognise the type of file.

Since the compiler C or C++ is already installed on all UNIX nodes of GRID, no particular requirements have to be specified in the jdl file. The library path environment variable is already configured in GRID nodes to find the C/C++ libraries.