Deploying Matlab applications (windows)

This short tutorial is made using Matlab R2008b (Matlab compiler v4.9) and relies on the use of `deploytool`. This tutorial might not be suited for older versions of Matlab (deploytool was introduced in version 2006b).

To Compile or Not to Compile

There are several cases where compiling your code can be advantageous:

- If you want to distribute your code to people that have no Matlab installed,
- If you want to run a large number of Matlab jobs simultaneously on a computing cluster, you are usually limited by the number of licenses available. Working with a compiled standalone version, not requiring a Matlab license, provides a nice workaround.
- If your Matlab is using your institute's license server, it might happen that no licenses are available anymore. If you have your Matlab code and you are to run

several paparmatric studies, then a standalone version might bring help, since you are not dependent anymore on the availability of the licenses on the license server. This might require some minor re-thinking and re-writing of your program to let it behave as a general-purpose program (read some parameters, read the input data and start the calculations).

After creating your component with the MATLAB Compiler, you can distribute, or deploy, it to others so that they can use it on their machines, independent of MATLAB.

The deployment process requires that you:

- Package the necessary components depending on the type of generated application.
- Distribute them to your end user.
- Have the end users install them on their systems. During this phase of the installation process, the end users run MCRInstaller *once* on their target machine, that is, the machine where they will run the application or library.
  On Windows, MCRInstaller is a self-extracting executable that installs thenecessary components to run your application. On UNIX, MCRInstaller is a ZIP file.

The compiling/packaging process

Step 0:

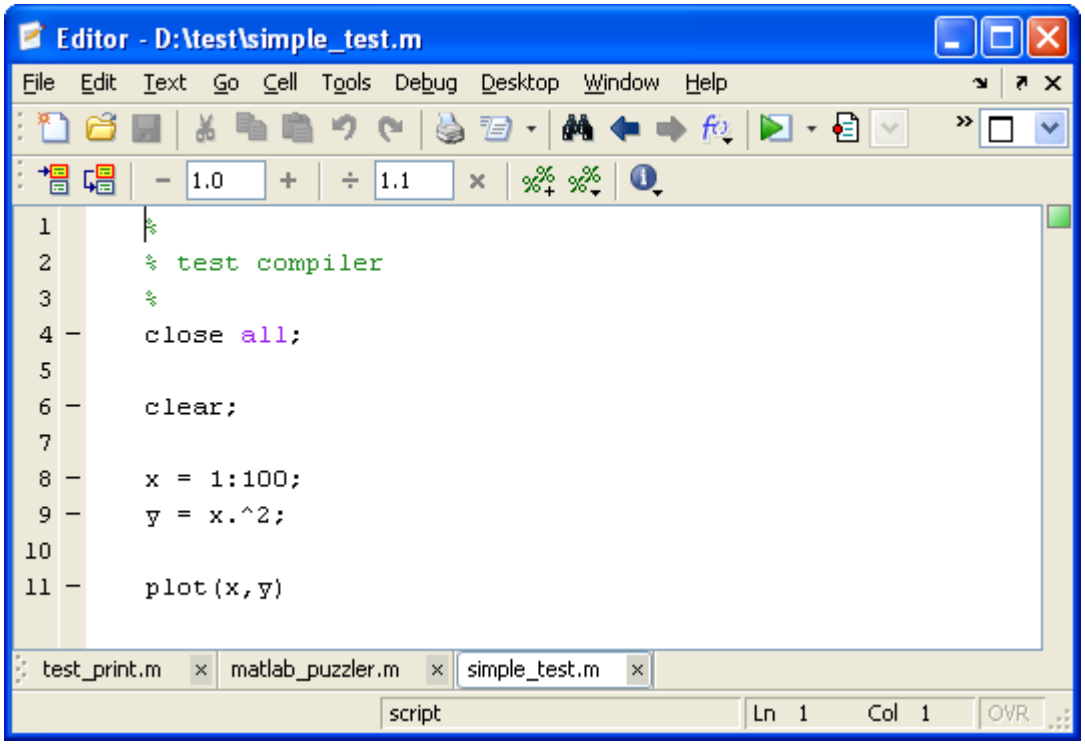
Before using the Matlab Compiler for the first time, you must run the setup command on `mbuild` to configure your C/C++ compiler to work with the Matlab Compiler. (The Matlab Compiler requires a supported ANSI C or C++ compiler installed on your system.). This setup step is only to be performed once.

```
>> mbuild -setup


Please choose your compiler for building standalone MATLAB
applications:


Would you like mbuild to locate installed compilers [y]/n?
Select a compiler:
[1] Lcc C version 2.4.1 in C:\PROGRAM FILES\MATLAB71\sys\lcc
```

```
[2] Microsoft Visual C/C++ version 7.1 in C:\Program
Files\Microsoft Visual Studio .NET 2003


[0] None
```

Depending on the compilers installed on your system, you can select one. The Lcc compiler is installed by default when installing the Compiler Toolbox.

A very simple program is used to demonstrate the compiler/deploytool

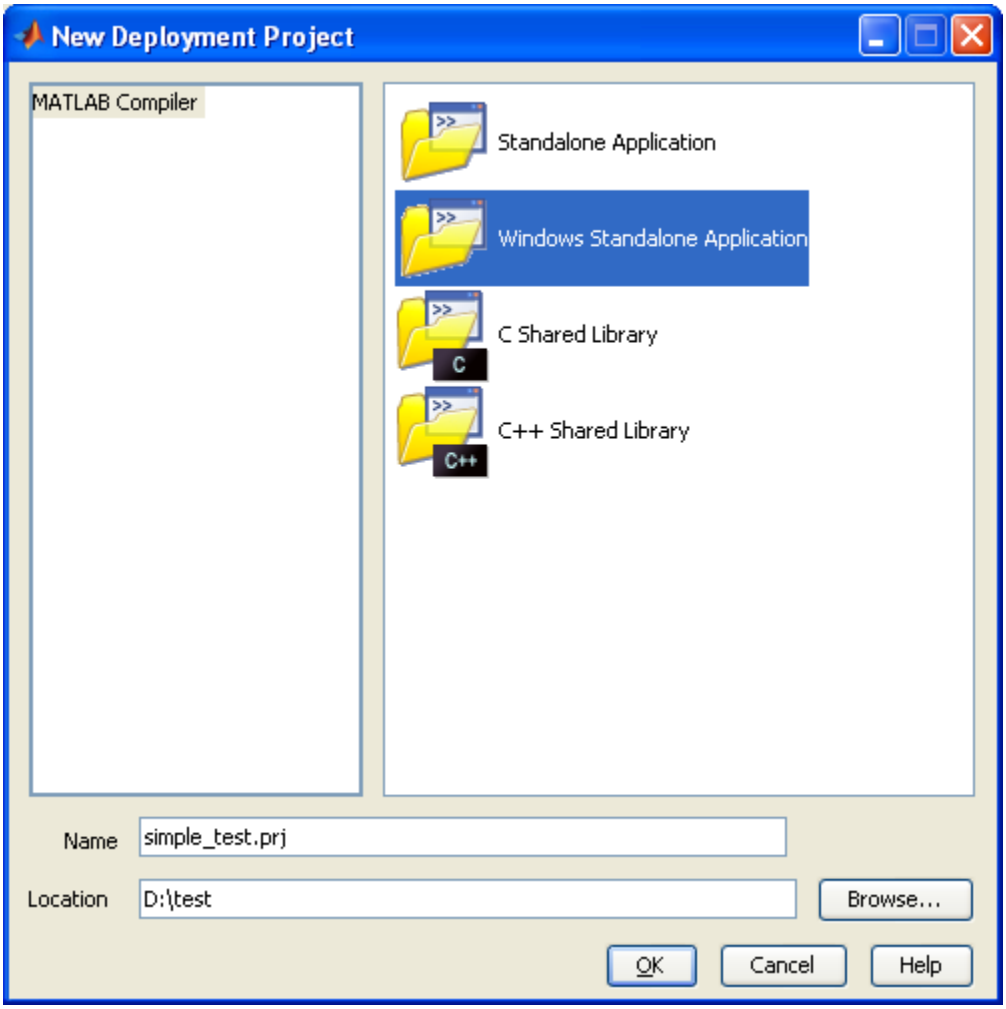


Compiling the code

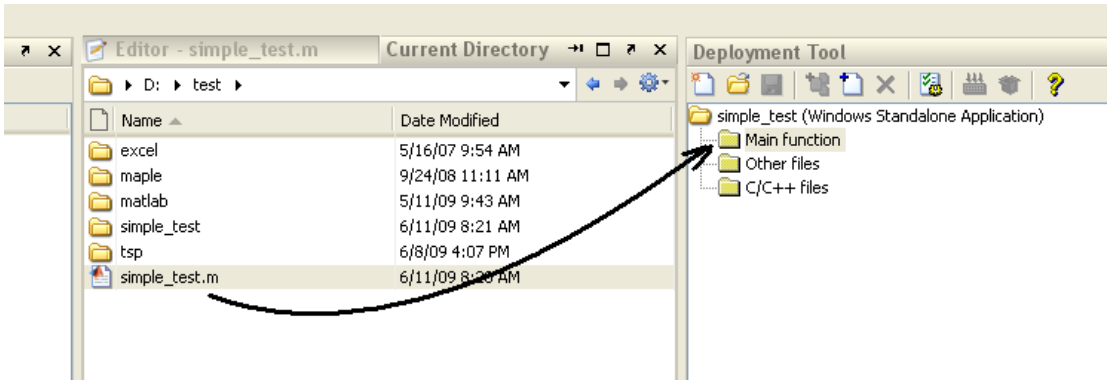
Remark: older version of the compiler might have problems when compiling script M-files Instead, they must be converted into function M-files. Normally, this is simply a case of wrapping the main section of code within a function.

- type `deploytool` to open the Deployment Tool window.
  The Deployment Tool opens as a dockable window in the MATLAB desktop, and a menu labeled **Project** is added to the MATLAB menu bar.

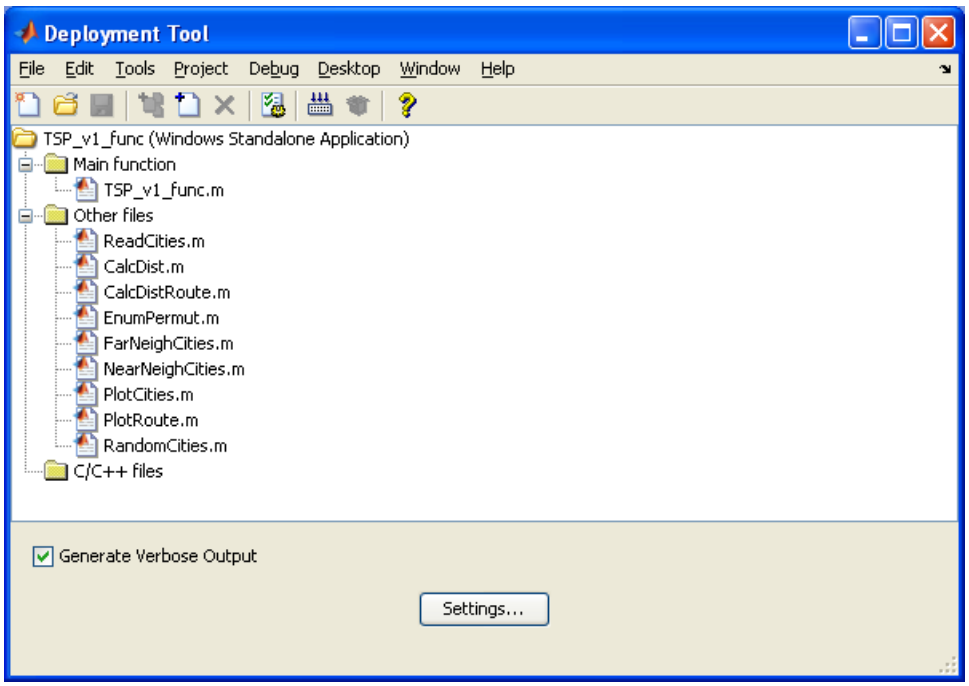
- Create a new project:
    - In the Deployment Tool toolbar, click the New Project icon.
    - In the New Deployment Project dialog box, select (**Windows**) **Standalone Applications**, and enter the following settings (The Matlab Compiler deployment target “Windows Standalone Application” was new with Matlab R2007b. Older versions only had the “Standalone Application” option - the difference being the new Windows Standalone Application does not start a command console (dos box)):
        - In the **Name** field, enter the project name.
        - In the **Location** field, enter the name of your work directory
        - Click **OK**.



- The MATLAB Compiler displays the project folder in the Deployment Tool window. The folder contains three folders, which are empty.
    - Drag the main routine of your Matlab program to the **Main function** folder.
    - If other files are included in your programming project, drag then to the **Other files** folder.

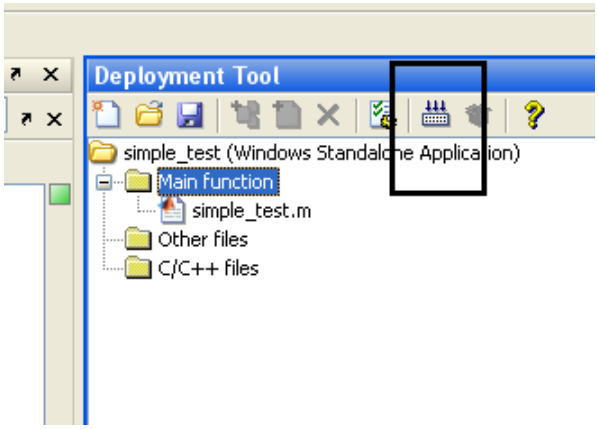


In the screendump below, an example is shown with the main program (TSP_v1_func.m) and all the functions that are called by this main function are placed under 'Other files')

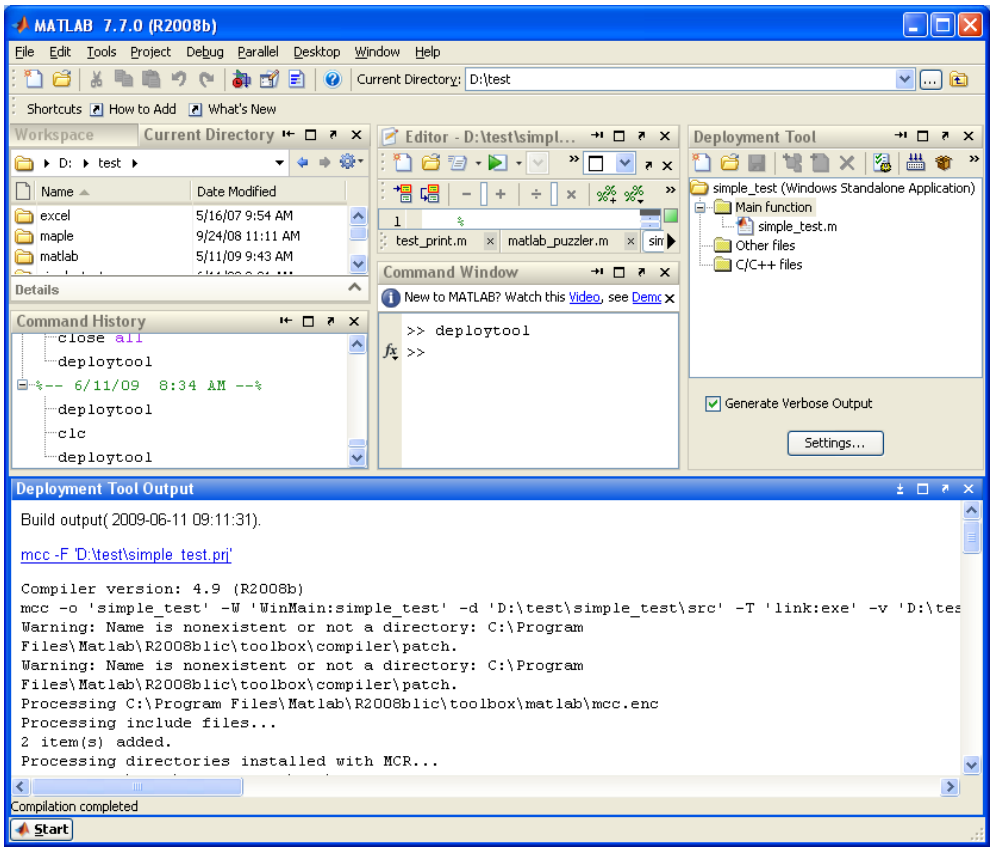


- Build the application as follows:

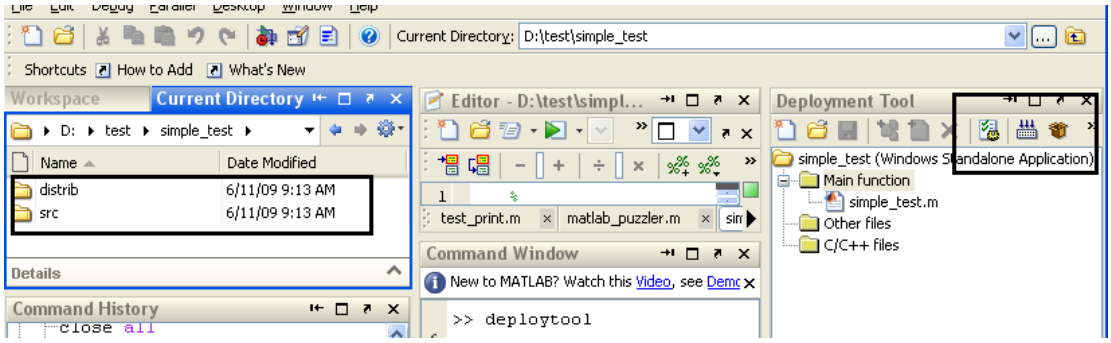
- In the Deployment Tool toolbar, click the Build Project icon.



- The build process begins, and a log of the build appears in the **Deployment Tool Output** pane. The status of the process is displayed in the status bar at the bottom of the output pane. The **Deployment Tool Output** pane is dockable; by default it appears across the bottom of the MATLAB desktop.

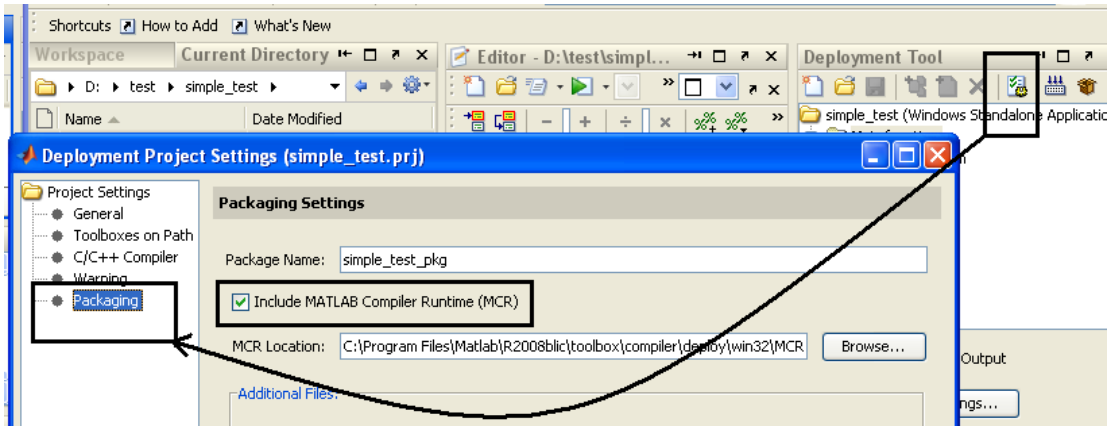


- The MATLAB Compiler product puts the files that are needed for the application in two newly created subdirectories, **src** and **distrib**, in the directory.

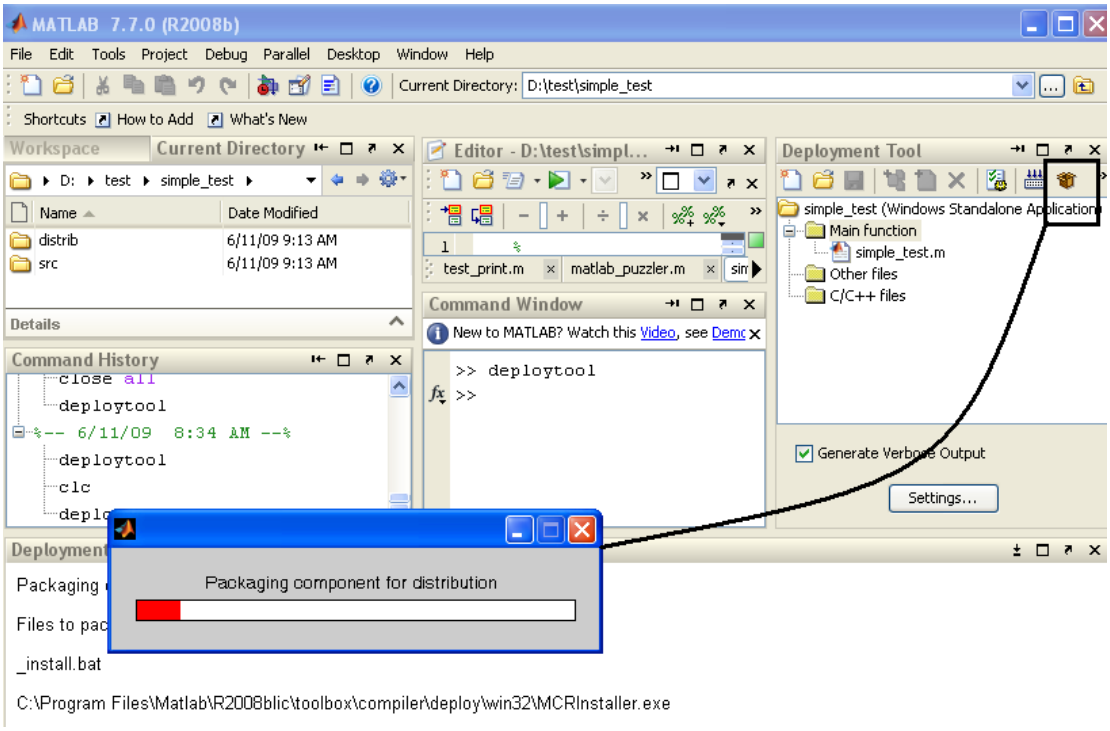


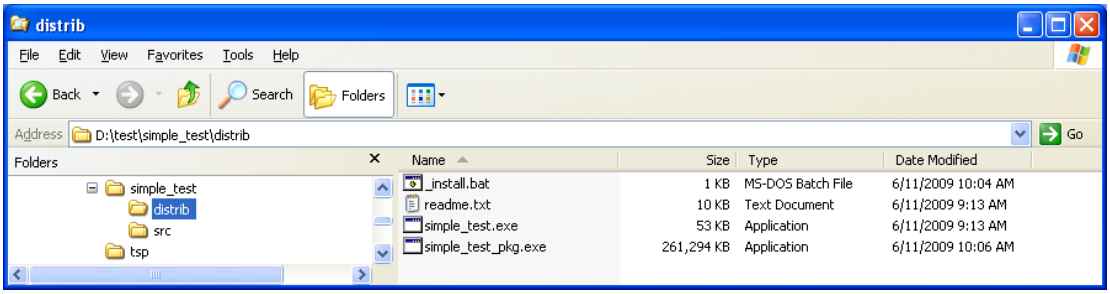
Package the code

- Package the application so it can run on machines that do not have MATLAB installed.
  - Check the settings to verify if the **Include MATLAB® Compiler™ Runtime (MCR)** option is selected (in the Packaging pane of Project Settings) or not. Deselect the option if you do not want to include the MCR in your package.
  - In the Deployment Tool toolbar, click the Package Project icon.



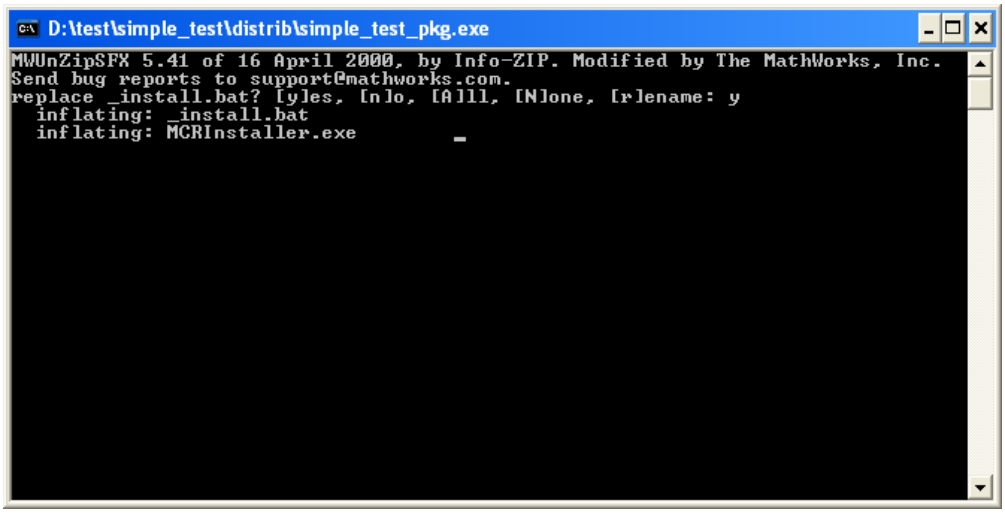
The MATLAB Compiler product creates a package in the distrib subdirectory. On Windows, the package is a self-extracting executable, and on platforms other than Windows, it is a .zip file.



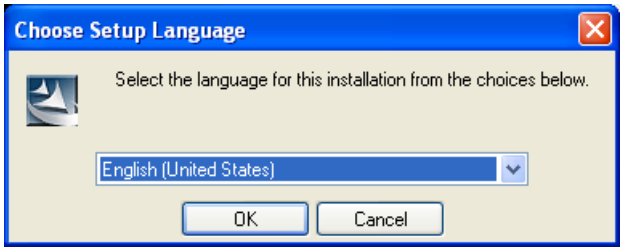


Deploying a Matlab Application

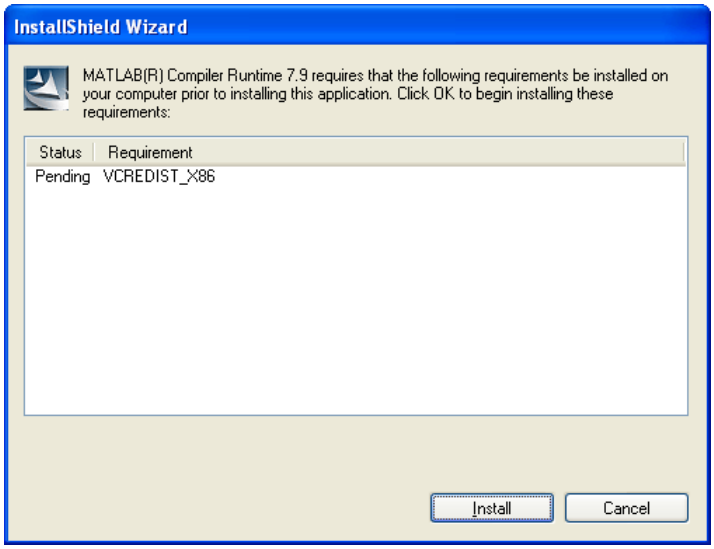
- Double click on the exe-file (in this example it will be simple_test_pkg.exe (the MCR installer was also included in the package).



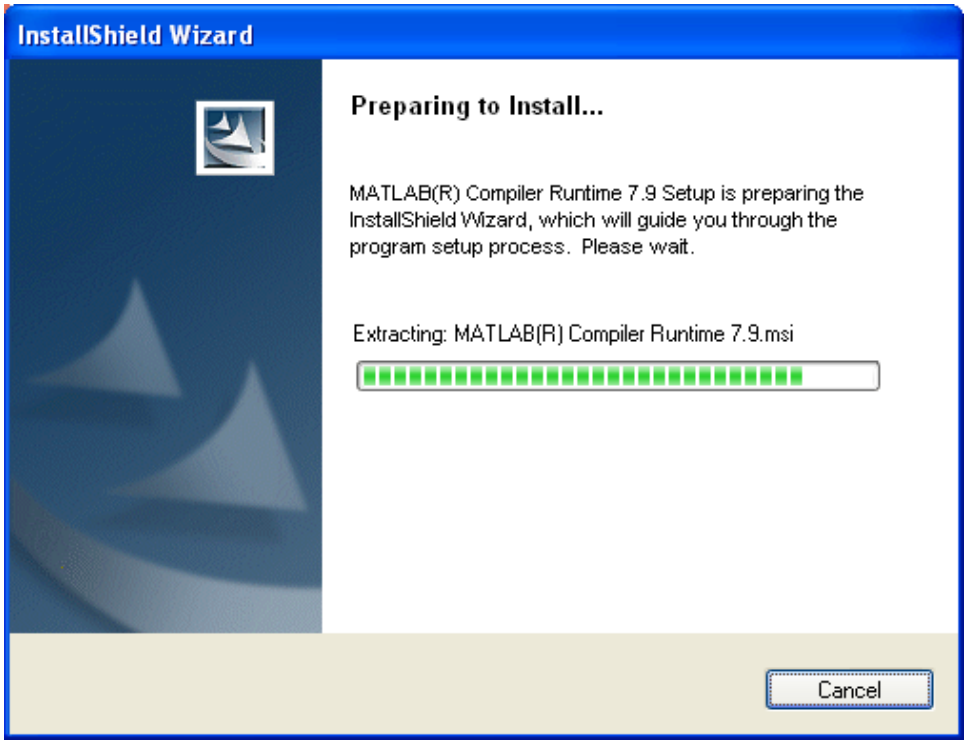
- The command window will appear, and the exe-file will inflate the files contained.
- The installation wizard will start with the language window

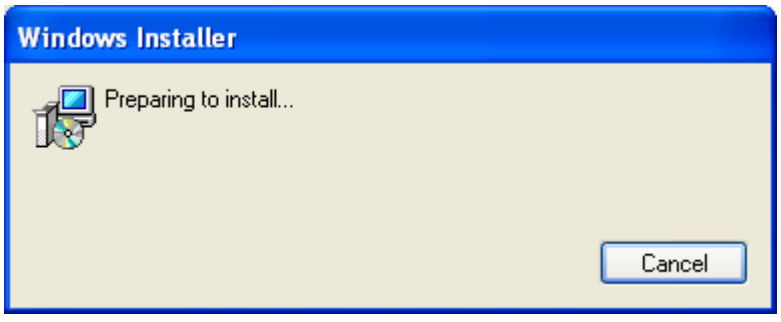


- Select the language
- Additional software might be necessary, click Install to install these components.

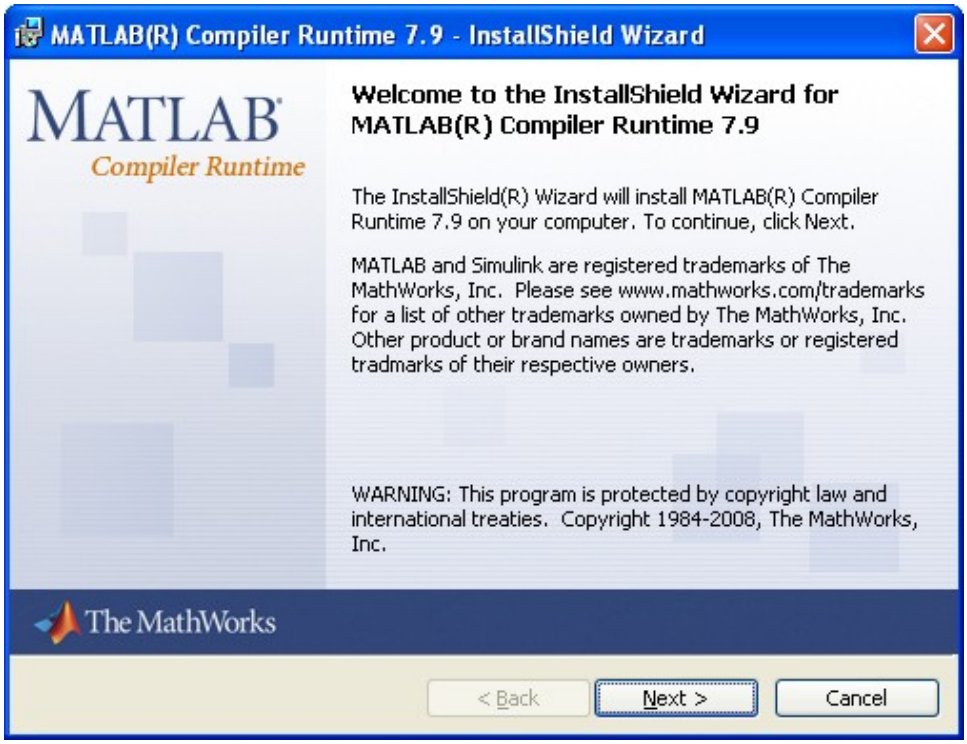


- Click Install to continue
- Several installer windows will appear

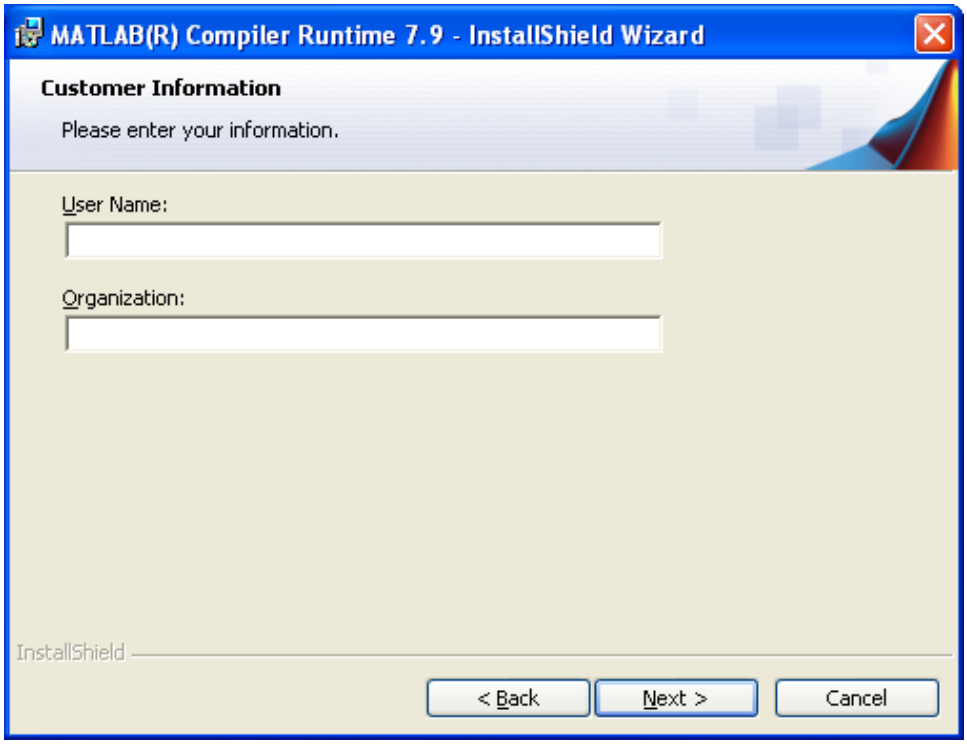




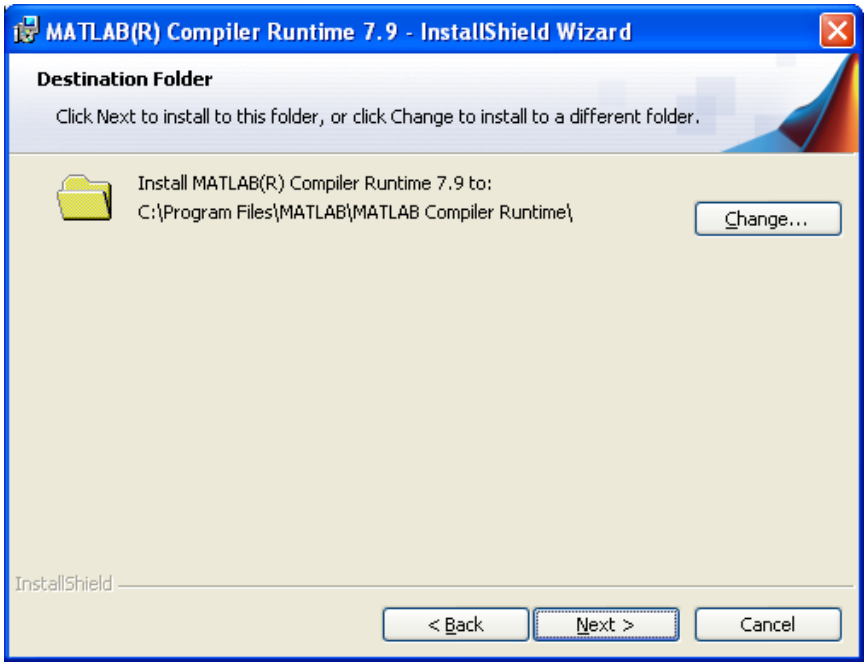
- The actual installation starts with

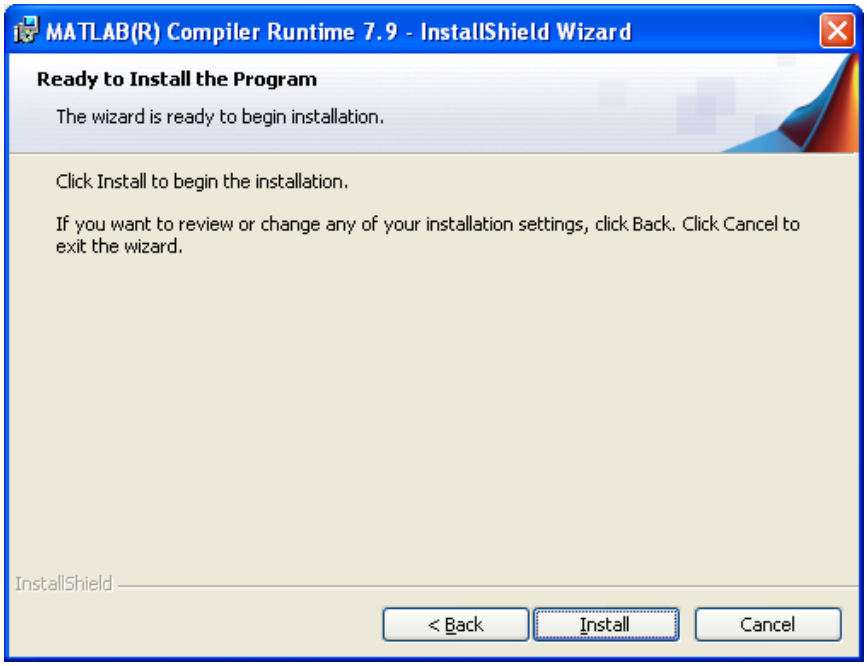


- Fill out your name and organization

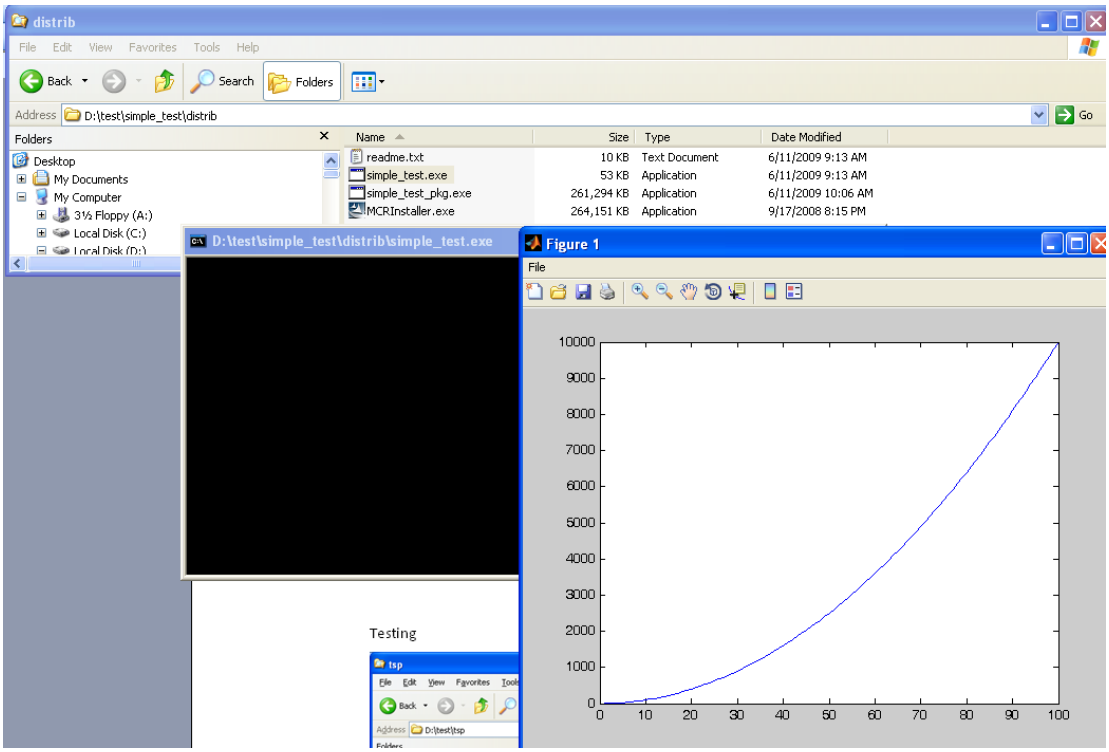


- You can keep the default settings





- When the installation is finished, a test can be ran. Doublick on the .exe of your program (it might take some time before the execution actually starts.

- 

Tips:

- For a full description of how to use the matlab compiler, please refer to Mathworks' Matlab Compiler Users Guide.
- check http://blogs.mathworks.com/loren/category/deployment/ for useful information