

## Notes on compiling ESP-r for Windows

20 Jan 2020

Jon W. Hand, ESRU, Glasgow, Scotland

These notes are intended for ESP-r developers who wish to work with a 'native version' of ESP-r on computers running W7/W8/W10. There is also an alternative approach to running ESP-r on Windows computers via Cygwin (discussed elsewhere).

The Windows port of ESP-r uses the standard source distribution as well as the same GNU compiler suite as all other versions of ESP-r. To enable this compilation process and provide access to underlying operating system services we use MSYS2 which provides basic shell, make, and install utilities needed to support the Windows port of the GNU gcc, g++ and gfortran compilers. Reading this whole document prior to touching the keyboard is a REALLY GOOD IDEA.

### Setting up MSYS2 and compilers

Information about msys2 (minimal system for Windows) can be found at [sourceforge.net/projects/msys2/](http://sourceforge.net/projects/msys2/). 32bit and 64bit installers and instructions are found at [msys2.org](http://msys2.org). Other instructions can be found at <https://github.com/msys2/msys2/wiki/MSYS2-installation>. Run the installer appropriate to your PC. When you are given the option to run msys2 – say yes. A new folder will appear on your C drive named either msys64 or msys32. A command window (with an M in the upper left corner) will start up and that is where you will be issuing most of your commands. Software within msys2 is managed by a package management tool called pacman. Its command line arguments are used to identify what is to be installed, updated or removed and it checks with a remote repository to determine dependencies for each 'package'. Pacman was ported from Linux and works similar to package management such as apt-get on Linux.

Pacman is also used to do the initial setup of msys2. To begin issue the following commands in the msys2 command window and press return:

```
pacman -Syuu
```

When these tasks complete you will need to shutdown msys2 by typing *exit*.

#### *Restarting msys2*

To complete the install of msys2 will require starting msys2. The sequence can also be used after the install process is complete. Go to the Windows Start button and open up the *All programs* tab and look for an entry with MSYS in the name e.g. MSYS2 64bit. Open this up and select MSYS2 Shell (you might also create a shortcut to this).

In the restarted msys2 command window give the commands:

```
pacman -Syu
```

This scans and recommends packages which can be updated. If it finds any agree to them and when that task completes do another restart of msys2. The next step installs software needed to compile applications from source. In the re-started command window give the following commands:

```
pacman -S nano make gettext base-devel subversion git net-utils
pacman -S libxml2-devel libxslt-devel
pacman -S gcc-fortran
```

Next you need the compilers that can create native Windows executables. If your computer is 64bit do the following:

```
pacman -S mingw-w64-x86_64-toolchain
pacman -S mingw-w64-x86_64-gtk2
```

If your computer is 32bit do the following:

```
pacman -S mingw-w64-i686-toolchain
pacman -S mingw-w64-i686-gtk2
```

Several of these include multiple packages, use the *return* key to signal that you want them all. You might have to use the *return* key a couple of times to get all of the dependencies.

## Windows & msys2

Inside the C:\msys64 or C:\msys32 folder are a number of subfolders. Use Windows Explorer to have a look. The folder you are mostly interested in will be either C:\msys64\home or C:\msys32\home. If your Windows login name is fred then there will be a folder C:\msys64\home\fred or C:\msys32\home\fred. When you are working within the msys2 command window this is your location. Since Windows can view and copy and delete files within the msys2 folders you can use Explorer if you find that more convenient for file operations.

To view or edit the files acquired/generated with *ESP-r* a text editor is required – Notepad++ is a better choice than Wordpad. Avoid notepad. Notepad++ is free and can open multiple files at the same time and is designed to work with ASCII text files. A web search will easily locate an installer for Notepad++.

To conclude setting up msys2 there are a number of folders which will be convenient to have in the 'home' folder. The following commands will take you to your home folder and then create folders for scripts, for Models and for the source code you will be working with.

```
cd
mkdir bin
mkdir Models
mkdir Src
```

The installer should include mingw32\_shell or mingw64\_shell batch files in the application selection. If not find them in C:\msys64\ . This starts a command window

with an interpreter similar to bash and sh on a Linux computer. It may look a bit like a DOS window, but it has a more powerful syntax. The command window deals with text based commands and can run text based applications and scripts. For example the pattern matching tools grep and awk are available. A subset of the standard commands you would find on a Linux computer are included (look in the C:\msys64\mingw64\bin).

Remember - in the MSYS command terminal use Linux commands. Applications you start from within the MSYS command syntax are generally restricted to the executables within the C:\msys64\bin folder.

## Setting environment variables

You will need to prepend to the Path environment variable C:\esp-r\bin;C:\msys64\mingw64\bin

These paths assume that you are going to compile ESP-r and install the executables and databases and example models into C:\esp-r. This avoids spaces in the paths to the files.

ESP-r needs a place to put new models as well as the results files generated by the simulator. If you do not specify otherwise the default is C:\Documents and Settings\username (yuck). Some people prefer to define a new environment variable `HOME_PATH = C:\esp-r`

## Locals

ESP-r uses a lot (~1 million lines) of Fortran (as well as C and C++). Fortran does not understand international character sets and it expects that real numbers use a period as a decimal place and that commas are used to separate items in a line. If your locale (France, Germany, Japan etc.) uses a **comma as a decimal point** then ESP-r models **quickly become corrupted**. Choose something like a USA or UK or C locale.

## Getting an ESP-r distribution

Access to the ESP-r source distribution is typically via the ESRU ESP-r web site: a version control system called git. You need a git client and one of the pacman commands above installed one within the MSYS2 environment. There is a script that does the following with little interaction (`download_and_install_most_current_ESP-r.bash`).

The following commands do the steps manually. In the Msys2 command window issue the following commands (you can cut and paste from this document into the Msys2 command window) we use `ESPrGit` as the folder name for the source distribution:

```
cd
cd Src
mkdir ESPrGit
```

```
cd ESPrgit
git clone --recursive https://github.com/ESP-rCommunity/ESP-rSource.git
git clone --recursive https://github.com/ESP-rCommunity/ESP-rDatabases.git
git clone --recursive https://github.com/ESP-rCommunity/ESP-rModels.git
git clone --recursive https://github.com/ESP-rCommunity/ESP-rDoc.git
```

Then use rsync to get the contents of ESP-rDatabases, ESP-rModels, and ESP-rDoc in place before a full install can be carried out. Here are the steps assuming a folder ESP-rMaster is created:

```
mkdir ESP-rMaster
rsync -av ESP-rSource/ ESP-rMaster/
rsync -av ESP-rDatabases/ ESP-rMaster/data/
rsync -av ESP-rDoc/ ESP-rMaster/doc/
rsync -av ESP-rModels/ ESP-rMaster/models/
```

In ESP-rMaster will be everything you need to fully build ESP-r.

## Compiling commands

You can build ESP-r executables as console applications that can be run within a DOS or MSYS2 command window. You can also build ESP-r executables as graphic applications which can be run within a DOS or MSYS2 command window or (with the -mode text command line option) force the graphic application to run in text mode within a MSYS2 command window.

ESP-r as a graphical application is based on the GTK graphics libraries which were installed via the earlier pacman commands. Use the normal Install script (fast to compile but slow to run) or the faster executing Install\_o1 script.

```
./Install -d C:/ESRU --gcc4 --debug
Or
./Install_o1 -d C:/ESRU -gcc4 --debug
```

There are a few choices in the script. Choose text or graphic, whether or not to include XML2 facilities and say yes to creating databases and example models if this is the first time you are compiling. The compilation process takes several minutes.

You should then find a number of exe files have been placed in C:\ESRU\esp-r\bin. Although the PATH update you did earlier should allow ESP-r executables to find the dll files needed you might find Windows complain about specific dll files not being found. You might have to manually copy a number of dll files from /mingw64/bin or /usr/bin to C:/ESRU/esp-r/bin from within the msys2 command window. Use the following as a guide :

```
cd /mingw64/bin
cp libgcc_s_seh-1.dll libwinpthread-1.dll C:/ESRU/esp-r/bin
cp libgfortran-3.dll libquadmath-0.dll C:/ESRU/esp-r/bin
cp libgdk-win32-2.0-0.dll libcairo-2.dll C:/ESRU/esp-r/bin
cp libfontconfig-1.dll libexpat-1.dll C:/ESRU/esp-r/bin
cp libfreetype-6.dll libbz2-1.dll C:/ESRU/esp-r/bin
cp libharfbuzz-0.dll libglib-2.0-0.dll C:/ESRU/esp-r/bin
cp libintl-8.dll libiconv-2.dll libpcre-1.dll C:/ESRU/esp-r/bin
```

```
cp libpng16-16.dll zlib1.dll libpixmap-1-0.dll C:/ESRU/esp-r/bin
cp libgdk_pixbuf-2.0-0.dll libgio-2.0-0.dll C:/ESRU/esp-r/bin
cp libgmodule-2.0-0.dll libgobject-2.0-0.dll C:/ESRU/esp-r/bin
cp libgobject-2.0-0.dll libffi-6.dll C:/ESRU/esp-r/bin
cp libpango-1.0-0.dll libpangocairo-1.0-0.dll C:/ESRU/esp-r/bin
cp libpangoft2-1.0-0.dll libpangowin32-1.0-0.dll C:/ESRU/esp-r/bin
cp libgtk-win32-2.0-0.dll libatk-1.0-0.dll C:/ESRU/esp-r/bin
```

If you want to use ESP-r for production work driven by scripts within the `msys2` command window you need only to add `-mode text` to the command invocation. If you want to work from a DOS command window you will need to compile a pure-text based application. Some users do an initial compile as pure-text and include the databases and example models. Then they issue a *make clean* command from within the 'src' folder and do a second `./Install` for the graphic version of the executables.

Note: the ESP-r Install script should detect that the compilation is being done on a Native Windows computer, if it does not select native windows from the options listed. When you are finished using MSYS type 'exit' in the MSYS command window.

## Using ESP-r

Assuming your Install task worked you will now have an ESP-r distribution on your computer. The native windows version of ESP-r is a graphic application which is normally invoked from within Windows rather than from within MSYS. The exception would be scripted production work or if you wanted to use the GNU debugger `gdb` and this should be run from within MSYS and from the folder with the source code and executable.

There are many ways to use these executables and example models and one pattern which works well for many users is listed below:

- a) Create a folder for simulation models. Good places are `C:\Esru\Models` or `C:\Users\your_name\Models`
- b) Create an association between model configuration files (\*.cfg) and the `esp-r.cmd` file found in `C:\ESRU\esp-r\bin`. One way to do this is to use Windows explorer and find the training folder e.g. `C:\ESRU\esp-r\training` and locate a model folder and then its `cfg` folder and right click on a file ending in `.cfg`. Associate it with the `esp-r.cmd` file and then try clicking on the `cfg` file to see if the ESP-r project manager starts up.
- c) Make a habit of opening up existing models by using Windows Explorer to go into the `cfg` folder of the model and clicking on the model `cfg` file. This will ensure that all of your work is 'local' to the model.
- d) copy the `esp-r.cmd` file into the Models folder (from step a). You can click on this if you want to create a new model in your Models folder.

## Hints

A few hints for use of ESP-r on Windows machines:

- path lengths in ESP-r are limited and to ensure that model file names are not truncated you should place models in a location such as C:\Esru\Models or C:\Users\user\_name\Models would be good locations.
- Note that when you select existing models from the project model exemplars list you have an option to specify where the model will be placed. Avoid paths with spaces and long paths.
- Enable an association between model .cfg files and the C:\Esru\esp-r\esp-r.cmd file. Then you can use Windows Explorer to click on model cfg files and automatically start the ESP-r Project Manager. You can also copy the esp-r.cmd file to a folder where you want to create new models (e.g. c:\Esru\Models) and then you can click on that cmd file to start up ESP-r.
- ESP-r assumes that real numbers use a . to mark decimal places and that a comma is a token separator. It gets seriously confused in locales where decimals are defined with commas. Please switch locales **before** running ESP-r.
- ESP-r assumes that so-called corporate databases can not be overwritten by normal users of ESP-r. On Windows machines this protection does not exist and can cause potential problems. Make a backup of the distribution databases and climate folders just to be safe.
- The Windows version lacks some facilities available on other operating systems (e.g. click-on-bitmap, select via mouse). When the unexpected happens we urge you to forward comments to Jon Hand (jon@esru.strath.ac.uk)
- ESP-r evolves so use svn or git commands to update your distribution. More information on subversion and git commands can be found in the ESP-r Developers Guide download site for updates.
- ESP-r has few 'undo' options. So for your own protection make regular backups of your work. How to make backups? Use a tool such as WinZip or 7-zip and right click on the top level folder of the model to create a compressed archive. Do this often. If possible give the compressed archive a name that helps you remember which version of the model is held in the archive.
- If you want to transfer a model to a different computer or a computer with a different operating system then you may need to use one of the scripts found in the source code distribution bin folder such as Linux\_model\_to\_dos or dos\_model\_to\_linux.
- Join the ESP-r community and send your questions to esp-r@lists.strath.ac.uk!

