

1 MERLIN installation

The following instructions are valid for installing MERLIN under a Unix system.

1. Download the latest MERLIN package from <http://merlin.cs.uoi.gr>. The corresponding file will be of the form `merlin-x.y.tar.gz` where `x.y` is the MERLIN version.
2. Uncompress and untar the file
`tar xvfz merlin-x.y.tar.gz`
A directory named `merlin-x.y` will be created.
3. Change into the newly created directory
`cd merlin-x.y`
4. Edit `Makefile.inc` and provide appropriate values for the variables `DESTDIR` and `FOPTIONS` (found in the first few lines of `Makefile.inc`).

`DESTDIR` is the destination directory where the MERLIN binaries and accompanying files will be installed. You must specify `DESTDIR` using a full path. Note that this directory must be different from the directory where you unpacked the MERLIN sources. Also note that if installing MERLIN outside your home directory you must have super-user privileges. Example:

```
DESTDIR=/usr/local/merlin
```

`FOPTIONS` is a string of options that must be passed to the Fortran compiler during the installation process. This is a good place to pass appropriate optimization options to the compiler. Example:

```
FOPTIONS=-O3
```

In addition the following options may be set:

`LINKOPTIONS`: Options that must be passed to the linker during installation.

`TYPE`: This specifies whether MERLIN will be compiled using `REAL` or `DOUBLE PRECISION` arithmetic.

`MXV`: The maximum number of optimization parameters MERLIN will handle.

`MXT`: The maximum number of terms in a sum-of-squares objective function.

`MCLBUF`: The maximum size (in bytes) of a compiled MCL program.

`MCLMEM`: The maximum size (in words) of memory available to MCL programs.

The default values provided for `DESTDIR` and `FOPTIONS` are adequate for most systems.

Note for Cygwin users: You must set `F77=g77` and uncomment `EXESUFFIX=.exe` in `Makefile.inc`.

5. Build the package
`make`
6. Install the package
`make install`

2 Running MERLIN

In order to compile the user written subprograms with the rest of the MERLIN package one must use the `run-merlin` script. After installation, the `run-merlin` script as well as other binaries are located in `DESTDIR/bin`. Hence one needs to add this directory to its Unix `PATH`. To add `DESTDIR/bin` to your Unix path and assuming that `DESTDIR` is set to `/usr/local/merlin`:

- Users of `csch` and `tcsh` add the following line in the file `.cshrc`
`set path = ($path /usr/local/merlin/bin)`
- Users of `sh` add the following lines in the file `.profile`
`PATH=$PATH:/usr/local/merlin/bin`
`EXPORT PATH`
- Users of `bash` add the following lines in the file `.bashrc`
`PATH=$PATH:/usr/local/merlin/bin`
`EXPORT PATH`

Alternatively you can invoke the `run-merlin` script using a full path. For example:
`/usr/local/merlin/bin/run-merlin funmin.f`

The `run-merlin` script accepts one or more file names as arguments that can be of the following types:

- Files ending in `.d`
These are assumed to be files that must be processed by the MERLIN preprocessor before they can be compiled. The MERLIN preprocessor will use the definitions file `DEFS` in the current directory, or if no such file exists, the one used for the installation of the MERLIN package (found in `DESTDIR/files`). After preprocessing the files, the script will compile them and link them with the rest of the MERLIN package.
- Files ending in `.f`
These are assumed to be files containing Fortran-77 code that must be compiled.
- Files ending in `.o`
These are assumed to be object files (already compiled).

In addition the `run-merlin` script recognizes the following environment variables.

- `MERLIN_F77`
This is the name of the compiler that will be used to compile the user written code.
- `MERLIN_FOPTIONS`
Options that will be used when compiling the user written code. The default is to use `-c` (compile only-do not link) and any other flags that were specified when editing `Makefile.inc`. Note that if you set this environment variable you must include the `-c` compiler flag.

- **MERLIN_LDOPTIONS**

Options that will be passed to the linker when building the final MERLIN executable. Here you can specify any libraries required by the user written subprograms.