

# Installation Notes

October 1, 2015

## 1 Background

TQGG (pronounce like Jacuzzi) is a general purpose interactive program that creates 2-dimensional unstructured grids. The long name is Triangle Quadrilateral Grid Generation and it was developed from the program Trigrd and its later incarnation GridGen.

For Linux and OSX, the GUI uses openMotif and is compiled with gcc and gfortran. For Windows, the GUI uses the QUICKWIN routines in Visual Fortran (a combination of Visual Studio and the Intel fortran compiler). When initially developed 30 odd years ago, the program was written in ancient fortran and used a command line interface. With the advent of more modern GUIs, the graphical interface is written in C and has been separated from the fortran computational code. Hence, this is a mixed-language program.

The source code is in a git repository. The first step is to install git (a revision control program) on the target system. This is freely available software and may be obtained at

<http://git-scm.com/downloads>

Once git is installed, create a base directory in your user space for the TQGG source; ie, MyPrograms. Change directory to the directory that has just been created. Then type

```
git clone https://github.com/rrusk/TQGG.git
```

which will create a directory ./TQGG and install the code there. In the next sections, specific instructions are given for building the program with different operating systems.

## 2 Linux

### 2.1 System Requirements

Linux systems require the openMotif development environment, gcc, and gfortran. Normally, gcc is available with the code development environment, and gfortran and Motif are available through the software manager for the particular flavor of Linux being used. If netCDF is desired, the netCDF libraries can be installed with the software manager. The fortran interface is also needed and can be downloaded from the netCDF download site. The fortran files should be built with gfortran to be compatible with the other code. Building the interface is simple and well documented. We have used version 4.2.1. The software has been tested on several versions of openSUSE and Linux Mint.

### 2.2 Building

On Linux, just type 'make TQGG' in the directory TQGG and the executable will be built and put into the subdirectory 'bin'. Then typing 'bin/TQGG' will execute the program. There are test files in the directory 'demodata'. Typing 'make clean' will remove the object and module object files.

There are 4 flavours of TQGG that can be built by using one of the following names in the make command:

**TQGG** The normal run version.

**TQGGdbg** The debug version with traceback and bounds checking.

**TQGGnc** The run version with netCDF support.

**TQGGncdbg** The debug version with netCDF support and traceback and bounds checking.

Run the program by copying it from subdirectory bin to somewhere on \$PATH or run it from bin.

## 3 Apple OSX

### 3.1 System Requirements

OSX like Linux requires the openMotif development environment, gcc, and gfortran. These are not normally installed on the proprietary OSX machines nor is there a package manager. OpenMotif can be installed from several sources on the web, gfortran is available from the GNU web site, and gcc (GNU or clang version) comes with OSX development environment depending on OSX version. You will need to install Apple's XCode for the compilers. It is interesting getting all this software to dance together, but the result is worth it. The TQGG code was developed on a Macbook with OSX 10.6, XCode 4, and openMotif 2.1.32-22i and with OSX 10.10.5, XCode 7, and openMotif 2.3.4.

## 3.2 Building

### 3.2.1 OSX 10.6.8 (and earlier?)

This was the first attempt to port the program from linux to a macbook. After searching around and taking a few missteps, the following procedure was successful.

1. Install XCode from Apple. This provides gcc.
2. Install or update to the latest version of XQuartz (The X11 system used by OSX).
3. Install gfortran from <https://gcc.gnu.org/wiki/GFortranBinariesMacOS>.
4. Download the openMotif 2.1.32 libraries and include files from <http://www.ist-inc.com/motif/download/index.html>
5. Build TQGG by typing 'make -f Makefile-mac TQGG', where TQGG can have one of the forms described earlier.

Run the program by copying it from subdirectory bin to somewhere on \$PATH or run it from bin.

### 3.2.2 OSX 10.10.5

Somewhere between OSX 10.6.8 and 10.10.5, the available download for openMotif 2.1.32-22i no longer functions properly and openMotif must be rebuilt. However, others have made this a simple task. In all, the necessary software is provided by following these steps:

1. Install XCode and Command Line Tools from Apple. This provides gcc (clang) and the tools for rebuilding openMotif.
2. Install or update to the latest version of XQuartz (The X11 system used by OSX).
3. Install gfortran from <https://gcc.gnu.org/wiki/GFortranBinariesMacOS>.
4. Install Homebrew for package management from <http://brew.sh/>. Also see <https://github.com/Homebrew/homebrew>.
5. Build the openMotif libraries and include files using a one-line command with Homebrew.  
See <https://gist.github.com/steakknife/60a39a32ae84e12238a2>
6. Create a symbolic link so the compiler and linker can find the openMotif include files and libraries.  
`sudo ln -s /usr/local/Cellar/openmotif/2.3.4 /usr/OpenMotif`
7. Build TQGG by typing 'make -f Makefile-mac TQGG', where TQGG can have one of the forms described earlier.

Run the program by copying it from subdirectory bin to somewhere on \$PATH or run it from bin.

## 4 MS Windows

### 4.1 System Requirements

The Windows version requires Visual Studio and a fortran compiler that includes the QUICKWIN libraries. These libraries are fortran callable subroutines that access a simplified version of the WIN API. The program was developed using Win XP SP2, Visual Studio 2005, and Intel fortran compiler 2011.8. Visual Studio 2008 has also been used with 64-bit XP where the project file is automatically updated to a new format.

### 4.2 Building

The project file TQGG.vfproj can be found in './TQGGWS\_vf'. Open this file in Visual Studio and there will be options to build the 32-bit debug and release versions, and the 64-bit debug and release versions.

Unlike the Linux and Mac versions, all the dialog boxes in Windows are modal; ie, the dialog must be closed before any other operations can commence. The only solution to this is to program the GUI directly with the WIN API and not use QUICKWIN.

There have been issues in running these executables on later versions of Windows (7 and possibly 8). Apparently, there is some security issue with the later versions such that the dialog boxes will not accept any input so the program hangs. This may not happen with later versions of VS and QUICKWIN that are built for the later versions of Windows.