Compiling

- **Install the following packages before starting the compilation:**
  - CodeBlocks IDE (http://www.codeblocks.org/).
  - CMake (https://cmake.org/)

- **Install the following packages before starting the compilation if you are building on Windows:**
  - TDM GCC compiler (http://tdm-gcc.tdragon.net/). Choose the 32bit or 64bit compiler.
  - MSYS (http://www.mingw.org/wiki/msys).
  - Run MSYS and type: gcc -v. Check COLLECT_GCC path. This should be something like c:\TDM-GCC-64\bin\gcc.exe.
  - Setup CodeBlocks IDE to use the previously installed TDM-32 bit or TDM-64 bit compiler. Open CodeBlocks and go to [settings]->[compiler]->[toolchain executables]. Change [compiler installation directory] path to TDM GCC compiler installation folder like C:\TDM-GCC-64 for 64 bit build and choose auto detect.
  - Install NSIS (https://nsis.sourceforge.io/Download). This is for creating installation package.

- **Install Xcode and Xcode command line tools in you are building on MACOSX.**

- **Install on Linux(Ubuntu):**
  - Synaptic Package Manager.
  - Install the following packages using synaptic package manager:
    - Gnu c++ compiler: build-essential
    - Gtk+ development libraries: libgtk-3-dev
    - Libudev: libudev-dev
    - OpenGL development libraries: libgl1-mesa-dev

- **Extract sources to the directory of your choosing.**

- **Create build directory. This can be the same or different directory from your sources.**

- **First step (Windows): Configure and generate ’MinGW makefiles’ using Cmake.**
- **First step (Linux and Mac): Configure and generate ’Unix makefiles’ using Cmake.**
  - Make sure you enter the correct build type for CMAKE_BUILD_TYPE variable (this can be **Debug** or **Release**), after performing ’Configure’ step.
  - Select ’Configure’ step two times.
  - Check that all SCOPEFUN variables are properly set after ’Configure’ step. There should be no red colored configuration variables or errors in cmake prior to generating choosen build system scripts.
  - Choose ’Generate’.
  - Open MSYS (Windows), go to your build folder and type: mingw32-make package. This command will build libraries in the ScopeFun source folder.
  - Run terminal (Linux and Mac), go to your build folder and type: make package

- **Second step (Windows): Configure and generate ’CodeBlocks-MinGW Makefiles’ using Cmake.**
- **Second step (Linux and Mac): Configure and generate ’CodeBlocks-Unix Makefiles’ using Cmake.**
  - Change ’Where to build the binaries’ folder in Cmake (this must be different folder as in first step).
  - Make sure you enter the correct build type for CMAKE_BUILD_TYPE variable (this can be **Debug** or **Release**), after performing ’Configure’ step.
  - Select ’Configure’ step two times.
  - Check that all SCOPEFUN variables are properly set after ’Configure’ step. There should be no red colored configuration variables or errors in cmake prior to generating choosen build system scripts.
  - Choose ’Generate’.
  - Open generated project with Codeblocks and select target ’sfScope’
  - Go to Project options-> Build targets tab: change ’Execution working dir’ to project source folder
  - You can now build the project

- **Open CodeBlocks IDE and compile or run MSYS command shell.**

  - **Build executable files(Windows):** got to build directory and type: mingw32-make -f makefile
  - **Build installer package(Windows):** got to build directory and type: mingw32-make package
  - **Build source files package(Windows):** got to build directory and type: mingw32-make package_source
  - **Build executable files(Linux and Mac):** got to build directory and type: make -f makefile
  - **Build installer package(Linux and Mac):** got to build directory and type: make package
  - **Build source files package(Linux and Mac):** got to build directory and type: make package_source

- **Questions ?** Go to www.scopefun.com where you can join a forum or contact us by e-mail.