

## Linux Setup

The SpinWorks executable is in MSIL code and runs on Windows machines under the .NET environment. On Linux and MacOS, the code can run under the open source “mono” .NET runtime environment. In principle, the same executable could be used for Windows and Linux. In practice, it has turned out to be more convenient to have separate builds in order to more easily handle difference in the file systems, libraries, etc. Three different test builds of the Linux version of SpinWorks are usually available:

- A build using MS Visual Studio on Windows
- A build using MonoDevelop on Windows
- A build using MonoDevelop on Linux (openSUSE 11.1)

In principle, all three should work identically. In practice, you may find that one works better than the others on your system.

Due to differences in the Windows and Linux run times and file structures, there are a number of changes in the installation procedure for Linux. There are many Linux variants but following is a general guide:

1. Ensure that you have a recent mono version installed on your computer.
2. Move the SpinWorks executable (e.g. SW\_317a\_MDlinux.exe) and the run script (spinworks) to a location used for locally installed software. Something like /usr/local/bin would be typical. Edit the run script to reflect the correct name and path to the SpinWorks executable. Ensure that the SpinWorks executable and the run script have execute permission at the appropriate user level (usually world). It is recommended that there be no blanks in the path to the SpinWorks executable. This would typically be true in most Linux systems.
3. Start SpinWorks by giving the spinworks command. If everything is well, SpinWorks should start within a few seconds.
4. In the Options menu select the start-up options dialog and set the external module path to the same location as the SpinWorks executable above (e.g. /usr/local/bin). In the same dialog set the scratch folder to someplace convenient in your home directory. I usually use something like /home/marat/Temp. The default data folder can be almost anything you chose, but something like /home/marat/NMRdata. Ensure that you have write permission in this folder. ***I strongly suggest that you avoid having any blanks in the file path.***
5. If you intend to use the simulation modules they will have to be compiled.
  - a. Ensure that you have a suitable C compiler installed. I usually use gcc.
  - b. Go to the distribution directory for the simulation routine. This would be sim\_main for NUMMRIT. If you are using other than the gcc compiler edit the Makefile to reflect the compiler that you are using.
  - c. Give the make command. Pay attention to any error messages. If there are no errors, an executable (e.g. sim\_main.exe) will be generated. Move

the executable to the folder used in step 1, and ensure that it has the correct permissions.