

User Guide for Gadget-4 installation and run on Linux and Windows

Thomas Raveau, Ari'el University, Ramat HaGolan St 65, Israel

Abstract

An explicit, childish, and complete guide to help users install and use Gadget-4 (Volker Springel), and visualize their snapshots (Splash, Daniel Price) on Linux and Windows systems. We use WSL (Windows Subsystem Linux) to incorporate Linux (Debian) into our Windows system. Windows 11 has been used for this tutorial, and works for Windows 10 as well but with different bug fixes. Ignore the WSL section if you are only interested in Linux. You will find in section 2 the installation of WSL and Debian. In section 3 the configuration of Debian and the installation of a first bunch of basics libraries needed for section 4. In section 4 a more complex installation of additional libraries. In section 5 the installation and the use of Gadget-4. In section 6 the installation and the use of Splash. Section 7 is a gallery to show few examples (not exhaustive) of what you can expect to visualize. This guide includes some tips for a better user experience, and enlight useful commands for beginner.

1 Introduction

As far as I know, there is no complete guide installation yet for Windows, though there are many reasons that may motivate you to run Gadget-4 on your Windows system. You might have a computer that you have been using for many years with Windows and you do not want to reboot your entire system, transfer data twice and reinstall all your software. You may also be unable to find or install the equivalent specific Windows software you really need on Linux. Perhaps you are simply too used to Windows to consider switching completely to Linux. Finally, you might prefer Windows for the interface, and how quickly you can navigate in the folders, copy, erase, or edit files. On my side, I am not allergic to Linux but I grew up with Windows, and playing in a hybrid way with the Windows interface and the Linux commands in the console, is optimized for me and make me efficient.

Inspired by the usefull guide of Hideki Tanimura for Linux, I was motivated to write my own guide because it was too short and incomplete for the beginner I was, facing so many issues and difficulties, moreover with my stubborn will to do it on Windows. This guide should allow you to do in 2 hours what I did in 3 months.

The guide is written almost like a script with indentation, which makes it very intuitive and visual. Each line corresponds to a thing to do, although sometimes I write a comment. It can be a physical action such as going in a directory, download something or write a command. Commands start with a small letter, and you will always find the instruction *Type* above, asking you to write something. Indentation spaces show that the instruction is still active, that the directory is still the same, that we go further in the process, or used to add a clarification. Punctuation is very important, but the guide should be punctuation doubt free. The guide is case sensitive. Italics are reserved for comments when I especially want to designate or enlight a specific in-computer file or action, if I did not, it is because I was more global.

Unplug your brain, copy, paste, and have fun.

2 WSL

In Windows:

Open Microsoft Store.

Search for:

WSL

Install Windows Subsystem for Linux.

Search for:

Debian

Install Debian.

Open Windows Powershell.

Type:

wsl --update

wsl --shutdown

Find Debian by scrolling in your Start menu program list. Create an easy shortcut that fit to you to open Debian fast. My advise : pin it in your Start menu shortcut.

Open CMD (Windows command processor).

You should be in the default path:

C:\Users*user_name*>

Type:

echo. > .wslconfig

Open C:\Users*user_name*\.wslconfig

Type:

[wsl2]

memory=32GB

processors=24

Note : Only the half of the memory will/can be used. Memory and processors numbers depend on your configuration. They do not state what will be allocated permanently, but what is the maximum WSL can use through your next requests in term of resources. The resources will be shared between WSL and Windows, that's why no more than the half of the memory can be allocated.

3 Debian

In Windows, open Debian console.

Type for the first and the last time:

your_nickname

It will be basically the name of your folder.

your_password

your_password

Any further password request (mostly *sudo*) will be *your_password* .

Type:

sudo apt update

sudo apt upgrade

sudo apt-get install build-essential

sudo apt-get install gfortran

sudo apt install plocate

sudo apt install dnf

sudo apt install apt-file

sudo apt-get install pkg-config

sudo apt install vim

sudo apt install git

```

sudo apt-get install libgsl0-dev
sudo apt-get install libgsl-dev
sudo apt-get install zlib1g-dev
sudo apt install libc6-i386
sudo apt-get install libcairo2-dev
sudo apt-get install libfftw3-dev libfftw3-doc
sudo apt-get install libfreetype6-dev
sudo apt install libtiff-dev
sudo cp -r freetype2/freetype /usr/include/freetype
sudo cp freetype2/ft2build.h /usr/include/ft2build.h
vim .vimrc
    i    ; just press i to enter in edit mode.
        set mouse==a
        echap    ; just press echap to quit edit mode.
    :x    ; just press : then x then enter to save and leave. :q! would quit without saving.

```

Note : i and :x steps will not be written anymore in this tutorial. :x and :q! are commands, so remember to quit edit mode before writting those commands.

4 Libraries

Debian installed, you now have a new root folder on your computer. Its location should be as high as your main disk C: , which is the highest. Its name is Linux/Debian , and the Windows path associated is \\wsl.localhost\Debian . When you open the Debian console, you will be by default in the path \home\your_nickname\ , which is equivalent to the Windows path \\wsl.localhost\Debian\home\your_nickname\ , and any further your_nickname call, will refer to that folder.

From Windows, open *Debian* folder, then *home* folder, then *your_nickname* folder without forgetting about making its shortcut on the desktop first.

Download each link and think about taking the latest version, and put it in the folder.

```

https://www.mpich.org/downloads/
    mpich-4.1.2.tar.gz
https://fftw.org/download.html
    fftw-3.3.10.tar.gz
https://www.open-mpi.org/software/hwloc/v2.9/
    hwloc-2.9.3.tar.gz
https://ftp.gnu.org/gnu/gsl/
    gsl-2.7.tar.gz
https://mirror.sobukus.de/files/src/hdf5/
    hdf5-1.14.3.tar.gz

```

Note : GSL has already been installed with libgsl0-dev. Therefore, the installation of GSL is useless, will not even been read by Gadget-4, and could be ignored. For now, I let the lines if they are usefull in the futur. Yet, next lines about GSL in the next section are mandatory.

Open Debian console.

Type:

```

tar xvfz mpich-4.1.2.tar.gz
tar xvfz fftw-3.3.10.tar.gz
tar xvfz hwloc-2.9.3.tar.gz
tar xvfz gsl-2.7.tar.gz

```

```

tar xvfz hdf5-1.14.3.tar.gz
mkdir lib
cd mpich-4.1.2
./configure --prefix=/home/your_nickname/lib/mpich/4.1.2
make
make install
cd ..
cd fftw-3.3.10
./configure --prefix=/home/your_nickname/lib/fftw/3.3.10 --enable-openmp --enable-float
make
make install
cd ..
cd hwloc-2.9.3
./configure --prefix=/home/your_nickname/lib/hwloc/2.9.3
make
make install
cd ..
cd gsl-2.7
./configure --prefix=/home/your_nickname/lib/gsl/2.7
make
make install
cd ..
cd hdf5-1.14.3
./configure --prefix=/home/your_nickname/lib/hdf5/1.14.3
make
make install
cd ..

```

```
vim .bashrc
```

Take care of the version and insert at the very end:

```

export GSL_HOME=${HOME}/lib/gsl/2.7
export FFTW3_HOME=${HOME}/lib/fftw/3.3.10
export HDF5_HOME=${HOME}/lib/hdf5/1.14.3
export MPICH_HOME=${HOME}/lib/mpich/4.1.2
export HWLOC_HOME=${HOME}/lib/hwloc/2.9.3

```

The next one is on a single line:

```

export LD_LIBRARY_PATH=${GSL_HOME}/lib:${FFTW3_HOME}/lib:${HDF5_HOME}/lib:
${MPICH_HOME}/lib:${HWLOC_HOME}/lib:$LD_LIBRARY_PATH

```

The next one is on a single line:

```

export PATH=${GSL_HOME}/bin:${usr}/include/freetype2:${FFTW3_HOME}/bin:
${HDF5_HOME}/bin:${MPICH_HOME}/bin:${HWLOC_HOME}/bin:$PATH
export FITS_DIR=/usr/local/lib
export HDF5ROOT=~/lib/hdf5/1.14.3/
export SPLASH_DIR=$HOME/splash
export PATH=$PATH:$SPLASH_DIR/bin
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$SPLASH_DIR/giza/lib
export LD_LIBRARY_PATH="/usr/local/lib:$LD_LIBRARY_PATH"
export LD_LIBRARY_PATH="{HOME}/lib/hdf5/1.14.3/lib:$LD_LIBRARY_PATH"
alias gsplash='splash -f gadget_hdf5'
alias python='python3'

```

```
source ~/.bashrc
```

5 Gadget4

From Windows:

Download Gadget-4, and put it in your *your_nickname* folder.

```
https://gitlab.mpcdf.mpg.de/vrs/gadget4
gadget4-master.tar.gz
```

Download the data needed for the example of simulations provided by Gadget-4:

```
https://wwwmpa.mpa-garching.mpg.de/gadget4/example_ics.tar
example-ics.tar
```

Unzip it (WinRAR), and put the folder *ExampleICs* in the Windows path:

```
\\wsl.localhost\Debian\u\vrs\Simulations\ICs\
```

Create yourself the necessary folders if they do not exist yet.

Go back to your Debian console.

Type:

```
tar xvfz gadget4-master.tar.gz
mv gadget4-master gadget4
cd gadget4
cp Template-Makefile.systype Makefile.systype
vim Makefile.systype
  Modify the line:
    #SYSTYPE="hydra"
    SYSTYPE="hydra"
vim Makefile
  Modify the line (around line 97):
    PYTHON = python
    PYTHON = python3
cd buildsystem
rm Makefile.path.hydra
vim Makefile.path.hydra
  GSL_INCL = -I${GSL_HOME}/include
  GSL_LIBS = -L${GSL_HOME}/lib
  FFTW_INCL = -I${FFTW3_HOME}/include
  FFTW_LIBS = -L${FFTW3_HOME}/lib
  HDF5_INCL = -I${HDF5_HOME}/include
  HDF5_LIBS = -L${HDF5_HOME}/lib
  HWLOC_INCL = -I${HWLOC_HOME}/include
  HWLOC_LIBS = -L${HWLOC_HOME}/lib
```

At this point, thinking about restart the computer might be a good idea.

```
sudo apt update
sudo apt upgrade
```

For advanced users:

A standart run consist in copying the file *Template-Config.sh* into *Config.sh* , and to edit it as your needs. Then edit your *param.txt* file (you can find a template in *examples* folder) as your needs.

Type:

```
make
```

It permits to create the executable *Gadget4* based on *Config.sh* . If you want to do another simulation with another *Config.sh* , you should have a different folder for each configuration, or erase the executable *Gadget4* and the directory *build* to *make* again.

```
./Gadget4 param.txt
```

The run start. By editing the output directory line in *param.txt* , you can launch several

different runs with different parameters from the same initial folder.

For us:

Type:

```
./make-examples.sh
cd examples
cd DM-L50-N128
```

Edit *param.txt* as your needs.

For a fast check (less than 2 hours), modify:

```
TimeBegin 0.015625
TimeBegin 0.955625
NSample 128
NSample 16
GridSize 128
GridSize 16
```

For a full run (around 10 hours or more), do not modify *param.txt* .

For a weak computer, decrease *NSample* and *GridSize* without need to set it as low as 16.

```
./Gadget4 param.txt
```

Think about visiting the official website, which will give much more informations about the execution and Gadget4 overall.

<https://wwwmpa.mpa-garching.mpg.de/gadget4/>

6 SPLASH

Download fitsio and put it in your *your_nickname* folder.

<http://heasarc.gsfc.nasa.gov/FTP/software/fitsio/cfitsio-4.3.0.tar.gz>

In your Debian console:

Type:

```
tar xf cfitsio-4.3.0.tar.gz
cd cfitsio-4.3.0
make
sudo make install
cd ..
git clone https://github.com/danieljprice/giza.git
cd giza
./configure
make
sudo make install
cd ..
git clone https://github.com/danieljprice/splash.git
cd splash
make HDF5=yes FITS=yes SYSTEM=gfortran
sudo make install
cd ..
```

For advanced users:

You now can go in any output directory from your Gadget4 simulations, where are stored snapshot files,

and read them with one of the two commands.

Type:

```
gsplash snapshot_*.hdf5  
gsplash snapshot_*.hdf5
```

For us:

Type:

```
cd gadget4  
cd examples  
cd output  
gsplash snapshot_00*.hdf5
```

Type *2* then *1* then *3* then *enter* several times in a row.

What you see right now is the position of dark matter particles (x,y), where the color shows the depth z.

You can by yourself run all the others examples. If it is possible, render the density and not the depth.

Enjoy and take nice screenshots!

Note : Once the graph is open, keep your cursor above the graph and press h. You will find a list of interactive commands on your console. Press space to watch the next snapshot, press b to comeback. Press enter for a nice graph window border. Press r to refresh. Left click twice for a zoom in a square. Press a to refresh color label. Graph is sensitive to the position of your cursor. Sometimes, spam is necessary to get the graph reacting. Press q to close the graph. Type q to exit Splash.

Think about visiting the official website, which will give much more informations about the execution and Splash overall.

<https://splash-viz.readthedocs.io/en/latest/index.html>

7 Gallery



