

# Install and use GHDL and GtkWave

---

## Table of Contents

---

### Install and use GHDL and GtkWave

- Table of Contents
- GHDL
- GtkWave
- To install, Get the necessary programs and tools
  - GHDL:
  - GtkWave
- Install the tools
  - Install GNAT
  - Install LLVM
  - Install GHDL
  - GtkWave
  - The end
- Tools Documentation
  - GHDL
  - GtkWave

## GHDL

---

GHDL is an open-source simulator for the VHDL hardware language. GHDL is not an interpreter; it allows you to analyze and elaborate sources to generate machine code from your design. Native program execution is the only way for high speed simulation.

Features as given on the official [GHDL Github](#) page.

- GHDL fully supports the 1987, 1993, 2002 versions of the IEEE 1076 VHDL standard, and the latest 2008 revision
- Partial support of [PSL](#).
- By using a code generator ([LLVM](#), [GCC](#) or, [x86\\_64/i386](#) only, a built-in one), it is much faster than any interpreted simulator. It can handle very large designs, such as [leon3/grlib](#).
- GHDL runs on [GNU/Linux](#), [Windows](#) and [macOS](#), both on `x86` and on `x86_64`.
- GHDL can write waveforms to a [GHW](#), [VCD](#) or FST file. Combined with a [GUI](#)-based [waveform viewer](#) and a good text editor, GHDL is a very powerful tool for writing, testing and simulating code.
- Supported third party projects: [VUnit](#), [OSVVM](#), [cocotb](#) (through the [VPI interface](#)), ...

## GtkWave

---

GTKWave is a fully featured [GTK+](#) based [waveform viewer](#) for Unix, Win32, and Mac OSX which reads LXT, LXT2, VZT, FST, and GHW files as well as standard Verilog VCD/EVCD files and allows their viewing.

## To install, Get the necessary programs and tools

---

This is a guide to install both tools on Ubuntu\_18.04 and possibly next coming releases of Ubuntu. Download GHDL and GtkWave and download tools and programs necessary to install and run both tools.

## GHDL:

- This guide describes the GHDL - LLVM installation, thus download the latest pre-build GHDL-LLVM version. At time of writing this article it was: [ghdl-20181129-ubuntu14-llvm-3.8](#)
- Other tools necessary to install GHDL-LLVM are:
  - GNAT - ADA compiler: The latest community version at the time of writing was/is **gnat-community-2018-20180528-x86\_64-linux-bin**. Get it [here](#).
  - LLVM - Low Level Virtual Machine: In the GHDL documentation pages version 5.0 is the most recent version that can be used, but it's running without any issue with newer versions too. This guide uses version 7.0. This tool can directly installed using apt. See below how to install LLVM.

## GtkWave

- Download GtkWave from [here](#).
- Just click the big [Download] button and the latest version for the used operating system is selected and downloaded. It is also possible to select the file manually. To pursue this path, click on the download page the [Files] tab 1/3 (down the page). Select there the version of the tool to download and save it onto your system.
- At time of writing this article the latest version Linux Ubuntu it was "**gtkwave-3.3.98.tar.gz**".

## Install the tools

---

### Install GNAT

- Assumed is that the earlier downloaded files are stored in */Downloads/Ghdl*
- Create under */Downloads/Vhdl* a folder for unzipping of the compiler.
- Unzip the downloaded file using the Ubuntu "Archive Manager".
- Run the *doinstall* script. and install the ADA compiler under */opt/Gnat*. Be aware that installing software in */opt* requires one to be sudo or root.
- Add following line to your *.bashrc* file.

```
# GNAT GPL (ADA compiler) for GHDL
export PATH=$PATH:/opt/Gnat/2018/bin
```

### Install LLVM

- Browse to this page: <https://apt.llvm.org/>
- On the page go to the Ubuntu section and copy the two text lines for version 7 on Bionic (18.04)
  - This are the lines:

```
deb http://apt.llvm.org/bionic/ llvm-toolchain-bionic-7 main
deb-src http://apt.llvm.org/bionic/ llvm-toolchain-bionic-7 main
```

- Press the windows key and start typing **Software & Updates**. AS soon as some letters are typed, icons will appear. Click the [Software & Update] icon and select the [Other Software] tab. Click [Add] and enter the first "deb ..." line in the new popup. Click [Add Source] and provide the sudo(root) password.  
Add the second "deb ..." line the same way.

- Hit [Close] and allow a new scan of the repositories.

- Install the stable version of all key packages

- Lower on the page with repository archive information find how to install all key packages or just read on here.
- Open a terminal (Right click the desktop and select [Open Terminal]).
- Type or better copy, one by one, following lines in the terminal:

```
■ wget -O - https://apt.llvm.org/llvm-snapshot.gpg.key | sudo apt-key add -
```

and provide the sudo password to install the archive signature.

- Install LLVM

```
apt-get install libllvm-7-ocaml-dev libllvm7 llvm-7 llvm-7-dev  
llvm-7-doc llvm-7-examples llvm-7-runtime
```

- Install Clang & co

```
apt-get install clang-7 clang-tools-7 clang-7-doc libclang-common-  
7-dev libclang-7-dev libclang1-7 clang-format-7 python-clang-7
```

- Libfuzzer

```
apt-get install libfuzzer-7-dev
```

- lldb

```
apt-get install lldb-7
```

- lld (linker)

```
apt-get install lld-7
```

- libc++

```
apt-get install libc++-7-dev libc++abi-7-dev
```

- OpenMP

```
apt-get install libomp-7-dev
```

- To make use of this latest version of LLVM tools some settings need to be done. In the terminal continue typing or copying and executing following lines one after the other.sudo update-

```
alternatives --install /usr/bin/clang clang /usr/bin/clang-7 100
sudo update-alternatives --install /usr/bin/clang++ clang++
/usr/bin/clang++-7 100
sudo update-alternatives --install /usr/bin/clang-apply-replacements
clang-apply-replacements /usr/bin/clang-apply-replacements-7 100
sudo update-alternatives --install /usr/bin/clang-check clang-check
/usr/bin/clang-check-7 100
sudo update-alternatives --install /usr/bin/clang-query clang-query
/usr/bin/clang-query-7 100
sudo update-alternatives --install /usr/bin/clang-tidy clang-tidy
/usr/bin/clang-tidy-7 100
sudo update-alternatives --install /usr/bin/scan-build scan-build
/usr/bin/scan-build-7 100
sudo update-alternatives --install /usr/bin/scan-view scan-view
/usr/bin/scan-view-7 100
```

The above lines set a permanent link for the system to use the newly installed LLVM tools instead of possible pre-installed or earlier installed versions.

## Install GHDL

In the assumption that GNU-ADA and LLVM are installed and the downloaded GHDL .tgz file is uncompressed in */Downloads/Ghd/Ghdl* , do following to install GHDL.

- Open a terminal window and change directory (cd) to the */opt* folder.  
Create a new directory called */Ghdl* (*sudo mkdir Ghdl*).
- Change the owner and group of the directory (*sudo chown Ghdl & sudo chgrp Ghdl*) = your user name. This is done to make later modifications and additions easier.
- change directory to the folder used by the uncompressed GHDL files,  
*cd /home/Downloads/Vhdl/Ghdl*.
- Run the terminal command:

```
./configure --with-llvm-config=/usr/lib/llvm-7/bin/llvm-config --
prefix=/opt/Ghdl
```

- When the previous run is done, type *make* and let run again.
- Run *make install* when previous run has finished.
- The tool should now be installed in */opt/Ghdl*, showing three folders (bin, include, lib).
- Add the path */opt/Ghdl/bin* to the systems path by editing the *.bashrc* file.

```
# GHDL simulator
export GHDL_ROOT="/opt/Ghdl"
export PATH=$PATH:$GHDL_ROOT/bin
```

- Change directory to */opt/Ghdl/bin* and create a symbolic link for ghdl.

```
sudo ln -s /opt/Ghdl/bin/ghdl /usr/bin/ghdl
```

## GtkWave

This is a similar install procedure as for GHDL, do following:

- Open a terminal window and change directory (cd) to the */opt* folder.  
Create a new directory called */GtkWave* (*sudo mkdir GtkWave*).
- Change the owner and group of the directory (*sudo chown GtkWave & sudo chgrp GtkWave*). = your user name. This is done to make later modifications and additions easier.
- Change directory to the folder used for the uncompressed files, *cd /home/Downloads/Vhdl/GtkWave*.
- Run the terminal command: *./configure --prefix=/opt/GtkWave*
  - The above command should do the job when the configure script can find the Tcl/Tk tools installed on the machine. The tool must be able to find following files: *tclConfig.sh* and *tkConfig.sh*. When the configure script cannot find one of the files check if tcl and tk are installed. When not install both and try again. When tcl and tk are installed run the script with following options:
  - *./configure --prefix=/opt/GtkWave --with-tcl=/opt/ActiveTcl/lib/tcl8.6 --with-tk=/opt/ActiveTcl/lib/tk8.6* (replace the path */opt/ActiveTcl/lib/tcl8.6* (tk8.) by the appropriate path on your system).
  - Another thing where the configuration script can trip over are the xz-utils (zip utilities). To avoid problems run the command:  
*./configure --prefix=/opt/GtkWave --with-tcl=/opt/ActiveTcl/lib/tcl8.6 --with-tk=/opt/ActiveTcl/lib/tk8.6 --disable-xz*
- When the previous run is done, type *make* and let run again
- Type *su* and provide the root or sudo password then type *make install* and let run again until finished.
- Check the install by running: *make installcheck*.
- The tool should now be installed in */opt/GtkWave*, showing three folders (bin, lib, share).
- Add the path */opt/GtkWave/bin* to the systems path by editing the *.bashrc* file.

```
# GtkWave viewer
export GTKWAVE_ROOT="/opt/GtkWave"
export PATH=$PATH:$GTKWAVE_ROOT/bin
```

- Change directory to */opt/GtkWave/bin* and create a symbolic link for ghdl.
  - *sudo ln -s /opt/GtkWave/bin/gtkwave /usr/bin/gtkwave*

## The end

Both tools are installed.

Under Linux/Ubuntu opening a terminal and typing:

- ghdl  
Shows **ghdl:error: missing command, try ghdl --help**
- gtkwave  
Shows an empty black pop-up screen normally used for waveforms.

The tools are installed and functional.

## Tools Documentation

---

### GHDL

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

## GtkWave

- <http://gtkwave.sourceforge.net/gtkwave.pdf>
- VCD writer: <http://pyvcd.readthedocs.io/en/latest/index.html>

Marc Defossez

20 Jan 2019