

**Project: IVlonOSX -
Icarus Verilog
Interactive for
Mac OS X**

Document: Build Instructions

Author: Felix Bertram

Date: 2003 sep 08

Scope

The *IVlonOSX* effort is a binary distribution of the *Icarus Verilog Interactive* simulator frontend, combined with the *Icarus Verilog* and *GHDL* VHDL simulators, shrinked-wrapped as a convenient MacOSX installer package. Find further information on these projects on the following homepages:

- <http://ivi.sourceforge.net/>
- <http://www.icarus.com/eda/verilog/>
- <http://ghdl.free.fr/>

This document gives instructions how to build the IVlonOSX package. The tools may be build independently from each other, therefore the build steps are described individually for each tool.

Prerequisites

Mac OS X

IVI has been built and tested on Mac OS X 10.2 *Jaguar*. The port maintainer is currently using

- Version 10.2.6

Other versions may work as well, however no attempts will be made to support non-recent OS versions. Instead, the maintainer will continue to update his system to the latest available release.

Developer Tools

The *Developer Tools* contain the GNU C-Compiler, along with all supporting tools and required libraries. The Developer Tools are updated on a regular basis, the port maintainer is currently using

- Version 2.1 (December 2002 Developer Tools)

The most recent Developer Tools may be downloaded here:

- <http://developer.apple.com/tools/>

GNAT

In order to compile *GHDL*, the *GNAT* Ada Compilers are required. The Gnat packages seem to be a little bit behind Apple's progress on the developer tools. For use with the December 2002 Developer Tools, the *gnat-gcc-3.1-1252-jaguar.dmg.gz* package is required. At the time of this writing, this version is referred to as the *Wavefront*.

The Gnat package may be downloaded here:

- <http://www.macada.org/>

Please note, that installing GNAT will update your gcc to version 1252 as well. The port maintainer has not experienced any incompatibilities due to this, but this could be a potential source of problems with *ProjectBuilder*.

Fink

While the Developer Tools provide a complete software development environment, a project like *IVlonOSX* requires a few more tools and libraries. The most convenient way of installing and maintaining these, is to use the *Fink* package manager. Fink may be downloaded here:

- <http://fink.sourceforge.net/>

You should install Fink to `/sw` (default). Once that Fink is up and running, you will need to install the following additional packages:

- `autoconf2.5` 2.57-1
- `automake1.7` 1.7.4-1
- `dlcompat` 20030522-1
- `dlcompat-dev` 20030522-1
- `dlcompat-shlibs` 20030522-1
- `gperf` 2.7.2-2

See the Fink documentation on how to install additional packages. For a graphical frontend to Fink check *FinkCommander*, which may be downloaded here:

- <http://finkcommander.sourceforge.net/>

Tcl/Tk

IVI was compiled with the *Tcl/Tk* frameworks for Aqua

- Version 8.4.4

It is highly recommended, to install Tcl/Tk from the *Batteries-Included* binary distribution. This package may be downloaded here:

- <http://www.maths.mq.edu.au/~steffen/tcltk/TclTkAqua/>

Build IVI

Download

First step is to download the IVI source code from the *SourceForge* project site:

- http://sourceforge.net/project/showfiles.php?group_id=53425

You will need the following archive:

- IVI Project
<http://prdownloads.sourceforge.net/ivi/ivi-0.3-pre-20030906.tar.gz?download>

Please note the following:

- The exact name of the IVI package changes with each update. Don't worry about this, just use the latest version you can find.
- You will not need download or install the *Tcl/Tk/Tix* package, as this was replaced by the binary Tcl/Tk distribution already described in the previous section.
- You will not need the *Icarus Verilog* package right now, please refer to the Icarus Verilog paragraph for further instructions.

Source Tree

The following source tree layout is recommended:

- ◆ IVIonOSX-root
 - ◆ IVI
 - ◆ ivi-0.3-pre-20030906
 - ◆ MacOSX
 - ◇ build
 - ◇ Icarus
 - ◇ GHDL

The IVI, Icarus, and GHDL folders are all arranged on the same hierarchy level. The build directories are buried inside these, to keep the object files clearly separated. The readily-build application will be created in the */Simulator/IVI* folder.

Build Steps

To build IVI, just double-click the *build.command* script located in the MacOSX folder. This will configure, build, and install IVI as required.

To cleanup all temporary files, simply remove the *build* directory located in the MacOSX folder.

The following listing shows the contents of the *build.command* script. Please note that this listing is likely to be outdated, please refer to the source archive for the most recent listing.

```
#!/bin/sh

macosx=`dirname $0`
cd ${macosx}

cd ..
ivi_src=`pwd`
ivi=`basename ${ivi_src}`

cd ${macosx}

ivi_obj=${macosx}/build
ivi_inst=/Simulator/IVI

ivi_app=${ivi_inst}/IVI.app
ivi_home=${ivi_app}/Contents/MacOS/ivi_home
fink=/SW

CC="cc -no-cpp-precomp"
export CC

CPATH=${ivi_home}/include
export CPATH

LIBRARY_PATH=${ivi_src}/src/common:${ivi_home}/lib
export LIBRARY_PATH

MACOSX_DEPLOYMENT_TARGET="10.2"
export MACOSX_DEPLOYMENT_TARGET

echo "-----"
echo "App Package"
echo "-----"

mkdir ${ivi_inst}

rm -rf ${ivi_app}
cp -r src/IVI_app ${ivi_app}

sed s/@RELEASE@/${ivi}/g src/IVI_app/Contents/Info.plist \
  > ${ivi_app}/Contents/Info.plist

echo "-----"
echo "autoconf"
echo "-----"

cd ${ivi_src}
./autoconf.sh
```

```
echo "-----"
echo "configure"
echo "-----"

cd ${macosx}
mkdir build
cd build
${ivi_src}/configure \
  --enable-release \
  --prefix=${ivi_home} \
  --with-tcl=/Library/Frameworks/Tcl.framework \
  --with-tclinclude=/Library/Frameworks/Tcl.framework/Headers \
  --with-tk=/Library/Frameworks/Tk.framework \
  --with-tkinclude=/Library/Frameworks/Tk.framework/Headers \
  || exit 1

echo "-----"
echo "make"
echo "-----"

make || exit 1

echo "-----"
echo "install"
echo "-----"

make install || exit 1

# -----
# copy additional files
# -----

echo ${ivi} > ${ivi_inst}/VERSION.txt

# move examples out of application package
mv -f ${ivi_home}/examples ${ivi_inst}/examples

# add copy of GPL
cp ${ivi_src}/src/ivi_ui/tcl/License.txt ${ivi_inst}

# add credits
cp ${ivi_src}/src/ivi_ui/tcl/Credits.txt ${ivi_inst}

# add IVI release notes
cp ${ivi_src}/ReleaseNotes.txt ${ivi_inst}

# add Mac OS X readme
cp ${macosx}/Readme-OSX.txt ${ivi_inst}

# add Mac OS X documentation
cp ${macosx}/IVIonOSX.pdf ${ivi_inst}

echo "-----"
echo "done"
echo "-----"
```

Build Icarus Verilog

Download

First step is to download the *Icarus* source code. For the time being, *IVI* requires a patched version of *Icarus Verilog*, which is provided on *IVI*'s SourceForge page.

- http://sourceforge.net/project/showfiles.php?group_id=53425

You will need the following archive:

- Icarus Verilog
<http://prdownloads.sourceforge.net/ivi/ivi-verilog-0.7-20030427-prepatched.tar.gz?download>

Please note the following:

- The exact name of the *IVI* package changes with each update. Don't worry about this, just use the latest version you can find.

Source Tree

The following source tree layout is recommended:

- ◆ IVIonOSX-root
 - ◇ IVI
 - ◆ Icarus
 - ◇ ivi-verilog-20030815-prepatched
 - ◇ build
 - ◇ GHDL

The *IVI*, *Icarus*, and *GHDL* folders are all arranged on the same hierarchy level. The build directories are buried inside these, to keep the object files clearly separated. The readily-build simulator will be created in the */Simulator/Icarus* folder.

Build Steps

The following listing shows the contents of the port maintainer's *sbuidl.command* script. This script will configure, build and install Icarus as required. Please note, this listing is likely to be outdated, but the principle should be clear.

```
#!/bin/sh

macosx=`dirname $0`
cd ${macosx}

fink=/sw
ivl_pfx=/Simulator/Icarus
ivl_src=`pwd`/ivi-verilog-20030815-prepatched

echo "-----"
echo "setup"
echo "-----"

mkdir ${ivl_pfx}
mkdir ${ivl_pfx}/include
mkdir ${ivl_pfx}/lib
cp ${fink}/include/dlfcn.h ${ivl_pfx}/include/dlfcn.h
cp ${fink}/lib/libdl.0.dylib ${ivl_pfx}/lib/libdl.dylib

CC="cc -no-cpp-precomp"
export CC

CPATH=${ivl_pfx}/include
export CPATH

LIBRARY_PATH=${ivl_pfx}/lib
export LIBRARY_PATH

echo "-----"
echo "autoconf"
echo "-----"

cd ${ivl_src}
sh autoconf.sh

echo "-----"
echo "configure"
echo "-----"

cd ..
mkdir build
cd build

${ivl_src}/configure --prefix=${ivl_pfx} || exit 1

echo "-----"
echo "make"
echo "-----"

make || exit 1
```



```
echo "-----"
echo "install"
echo "-----"

make install || exit 1

# -----
# fix Icarus build problems (v0.7, 2003vii28)
# -----

ranlib ${ivl_pfx}/lib/libvpi.a
ranlib ${ivl_pfx}/lib/libveriususer.a

# -----
# copy additional files
# -----

echo `${ivl_pfx}/bin/iverilog -V` | \
  sed -e 's/.*version //' | \
  sed -e 's/ Copyright.*//' > ${ivl_pfx}/VERSION.txt

cp -f ${ivl_src}/COPYING      ${ivl_pfx}
cp -f ${ivl_src}/QUICK_START.txt ${ivl_pfx}
cp -f ${ivl_src}/README.txt   ${ivl_pfx}
cp -f ${ivl_src}/BUGS.txt     ${ivl_pfx}

echo "-----"
echo "done"
echo "-----"
```

Build GHDL

Download

First step is to download the *GHDL* source code from the GHDL home page, and the GCC source code from a GCC mirror:

- <http://ghdl.free.fr/>
- <http://gcc.gnu.org/mirrors.html>

You will need the following archives:

- GHDL
<http://ghdl.free.fr/ghdl-0.7.tar.bz2>
- GCC
[gcc-3.3.tar.bz2](http://gcc.gnu.org/mirrors.html)

Source Tree

The following source tree layout is recommended:

- ◆ IVIonOSX-root
 - ◇ IVI
 - ◇ Icarus
 - ◆ GHDL
 - ◇ build
 - ◆ gcc-3.3
 - ◆ gcc
 - ◇ vhdl

The IVI, Icarus, and GHDL folders are all arranged on the same hierarchy level. The build directories are buried inside these, to keep the object files clearly separated. The readily-build simulator will be created in the */Simulator/GHDL* folder.

Build Steps

Before compiling GHDL, a few adjustments need to be made.

vhdl/Makefile.in

edit vhdl/Makefile.in as follows:

```
GRT_LINUX_OBJS=i386.o linux.o
GRT_PTHREAD_OBJS=pthread.o
GRT_ADD_OBJS= $(GRT_PTHREAD_OBJS) grt-cbinding.o grt-cvpi.o
```

The following listing shows the contents of the port maintainer's *sbuild.command* script. This script will configure, build and install Icarus as required. Please note, this listing is likely to be outdated, but the principle should be clear.

```
#!/bin/sh

root=`dirname $0`
cd ${root}

gcc=gcc-3.3
srcdir=${root}/${gcc}
ghdlldir=${root}/ghdl-0.7
objdir=${root}/build
pfx=/Simulator/GHDL
fink=/sw

echo "-----"
echo "merge GHDL into GCC"
echo "-----"

cp -rf ${ghdlldir}/vhdl ${srcdir}/gcc

echo "-----"
echo "setup"
echo "-----"

CC="cc -no-cpp-precomp"
export CC

mkdir ${objdir}
cd ${objdir}

mkdir ${pfx}/include
mkdir ${pfx}/lib
cp ${fink}/include/dlfcn.h ${pfx}/include/dlfcn.h
cp ${fink}/lib/libdl.0.dylib ${pfx}/lib/libdl.dylib

echo "-----"
echo "configure"
echo "-----"

${srcdir}/configure \
  --enable-languages=vhdl \
  --prefix=${pfx} || exit 1
```

```

echo "-----"
echo "make"
echo "-----"

make || exit 1

echo "-----"
echo "install"
echo "-----"

make install || exit 1

# -----
# fix GHDL build problems
# -----

echo "@/libgrt.a" > ${pfx}/lib/gcc-lib/powerpc-apple-
darwin6.6/3.3/vhdl/lib/grt.lst
echo "@/system.o" >> ${pfx}/lib/gcc-lib/powerpc-apple-
darwin6.6/3.3/vhdl/lib/grt.lst
echo "${pfx}/lib/libdl.dylib" >> ${pfx}/lib/gcc-lib/powerpc-apple-
darwin6.6/3.3/vhdl/lib/grt.lst
cp /usr/lib/ada/system.o ${pfx}/lib/gcc-lib/powerpc-apple-
darwin6.6/3.3/vhdl/lib

# -----
# copy additional files
# -----

echo `${pfx}/bin/ghdl --version` | \
  sed -e 's/GHDL //' | \
  sed -e 's/ \[.*/' > ${pfx}/VERSION.txt

cp -f ${srcdir}/COPYING ${pfx}
cp -f ${srcdir}/COPYING.lib ${pfx}
cp -f ${srcdir}/MAINTAINERS ${pfx}
cp -f ${srcdir}/README ${pfx}
cp -f ${srcdir}/FAQ ${pfx}
cp -f ${srcdir}/BUGS ${pfx}
cp -f ${srcdir}/GNATS ${pfx}

echo "-----"
echo "done"
echo "-----"

```

Installation

General

IVIonOSX installs to the /Simulator folder. This location has been chosen to provide easy access to the command line tools, namely Icarus and GHDL, and to avoid any interference with existing tools.

Pathes

In case use of Icarus and GHDL from the Terminal window is required, the followi pathes need to be added to the PATH variable:

- /Simulator/GHDL/bin
- /Simulator/Icarus/bin

Preferences

IVI creates the following preference file

- ~/.ivi/ivi.conf

The most important settings here are:

- vlib_dirs
- inc_dirs

The *vlib_dirs* setting allows to set up additional search pathes for modules librarie. Usually this parameter will be used to include your vendor-specific libraries.

The *inc_dirs* setting allows to set up additional search pathes for include files.

The following snippet shows an example ivi.conf file:

```
section WaveWidget {  
}  
  
section IVerilog {  
    current vlib_dirs "/Users/fbertram/Verilog/unisims"  
    current inc_dirs "/Users/fbertram/Verilog/inc"  
    current inc_dirs "[current inc_dirs] /home/ballance/this"  
}
```

Feedback

The *IVI* and *IVlonOSX* projects depend on volunteers participating in the project. If you have any feedback on IVI, the build procedure, or the installation, if you have created documentation, examples, or any other useful material, please send your input to either Matthew Ballance or me.

Thank you,
best regards

Felix Bertram
fbertram@users.sourceforge.net