

Installation of Cadfaelbrew on Xubuntu Linux 16.04 LTS

F.Mauger

2016-08-09

This note reports how I have built and installed Cadfaelbrew on my Xubuntu Linux 16.04 LTS (xenial) system.
Links:

- DocDB 3738
- <https://github.com/SuperNEMO-DBD/brew>

System

- OS: Xubuntu Linux 16.04 LTS
- Architecture: x86_64
- Processors: 4 x Intel(R) Core(TM) i7-3540M CPU @ 3.00GHz

Requirements

Installed packages

```
$ doxygen --version
1.8.11
```

```
$ g++ --version
g++ (Ubuntu 5.4.0-6ubuntu1~16.04.1) 5.4.0 20160609
```

```
$ make --version
GNU Make 4.1
```

```
$ git --version
git version 2.7.4
```

System dependencies

```
$ LANG=C sudo apt-get install \
    build-essential \
    curl \
    git \
    m4 \
    ruby \
    texinfo \
    libbz2-dev \
    libcurl4-openssl-dev \
    libexpat-dev \
    libncurses-dev \
    zlib1g-dev \
    libx11-dev \
    libxpm-dev \
```

```
libxft-dev \
libxext-dev \
libpng-dev \
libjpeg-dev \
libxmu-dev \
libgl1-mesa-dev \
libglu1-mesa-dev
```

Disk storage

In this report, the `/opt` directory is used to host the Cadfaelbrew build/installation tree. At least 6 GB must be available.

```
$ LANG=C df /opt
Filesystem      1K-blocks      Used Available Use% Mounted on
/dev/sdall       34471692  3851676   28845888  12% /opt
```

You may also choose the `${HOME}` directory.

Installation

1. Prepare working directory:

```
$ sudo chmod 1777 /opt # Allow installation in /opt by a standard user
$ export SNSW_BASE_DIR=/opt/sw/SuperNEMO-DBD
$ mkdir -p ${SNSW_BASE_DIR}
```

Cache and temporary directory used during the building of Cadfaelbrew formulae:

```
$ mkdir -p /opt/var/cache/Homebrew
$ export HOMEBREW_CACHE=/opt/var/cache/Homebrew
$ mkdir -p /opt/var/tmp
$ export HOMEBREW_TEMP=/opt/var/tmp
```

2. Download the software:

```
$ cd ${SNSW_BASE_DIR}
$ git clone https://github.com/SuperNEMO-DBD/brew.git ./Cadfaelbrew
$ export PATH="${SNSW_BASE_DIR}/Cadfaelbrew/bin:${PATH}"
$ export MANPATH="${SNSW_BASE_DIR}/Cadfaelbrew/share/man:${MANPATH}"
$ export INFOPATH="${SNSW_BASE_DIR}/Cadfaelbrew/share/info:${INFOPATH}"
$ which brew
/opt/sw/SuperNEMO-DBD/Cadfaelbrew/bin/brew
```

3. Bootstrap:

```
$ brew update
==> Tapping supernemo-dbd/cadfael
Clonage dans '/opt/sw/SuperNEMO-DBD/Cadfaelbrew/Library/Taps/supernemo-dbd/homeb.
remote: Counting objects: 359, done.
remote: Total 359 (delta 0), reused 0 (delta 0), pack-reused 359
Réception d'objets: 100% (359/359), 83.84 KiB | 0 bytes/s, fait.
Résolution des deltas: 100% (219/219), fait.
Vérification de la connectivité... fait.
Tapped 14 formulae (41 files, 242.6K)
==> Pinned supernemo-dbd/cadfael
Already up-to-date.
```

```
$ brew cadfael-bootstrap
...
==> Bootstrap of toolchain complete, installed formulae
zlib 1.2.8
xz 5.2.2
readline 6.3.8_1
gdbm 1.12
sqlite 3.13.0
makedepend 1.0.5
pkg-config 0.29.1_1
gettext 0.19.8.1
ninja 1.7.1
curl 7.50.1
gnu-getopt 1.1.6
sphinx-doc 1.4.5
libidn 1.33
openssl 1.0.2h_1
libxml2 2.9.4
gpatch 2.7.5
ncurses 6.0_1
unzip 6.0_2
python 2.7.12_1
bzip2 1.0.6_1
cmake 3.6.1
git-flow-avh 1.9.1
patchelf 0.9_1
Bootstrap of CadfaelBrew complete under
```

```
/opt/sw/SuperNEMO-DBD/Cadfaelbrew
```

To use the programs and libraries supplied by Cadfael you can:

1. (Recommended) Use brew's setup facility to start a new shell session with the environment correctly configured:

```
$ /opt/sw/SuperNEMO-DBD/Cadfaelbrew/bin/brew sh
```

This starts a new shell with PATH and other environment variables set correctly. Just exit the shell to return to your original session.

2. Set the following environment variables either directly in your shell's .rc file or through the configuration mechanism of your choice (e.g. Environment Modules)

```
PATH="/opt/sw/SuperNEMO-DBD/Cadfaelbrew/bin:$PATH"
MANPATH="/opt/sw/SuperNEMO-DBD/Cadfaelbrew/share/man:$MANPATH"
INFOPATH="/opt/sw/SuperNEMO-DBD/Cadfaelbrew/share/info:$INFOPATH"
```

In both cases that should be all that's needed, though certain use cases may also required the dynamic loader or Python path to be set. This is to be reviewed.

```
$ brew ls
bzip2  gdbm          gnu-getopt  libxml2      ninja      pkg-config  sphinx-doc  xz
cmake  gettext           gpatch     makedepend  openssl    python      sqlite
```

curl git-flow-avh libidn ncurses patchelf readline unzip

4. Installation of third-party software:

```
$ brew sh
$ LANG=C tree ./Cadfaelbrew/Library/Taps/supernemo-dbd/homebrew-cadfael
...
$ brew search boost
$ brew install supernemo-dbd/cadfael/boost
$ brew install supernemo-dbd/cadfael/camp
$ brew install supernemo-dbd/cadfael/clhep
$ brew install supernemo-dbd/cadfael/xerces-c
$ brew install supernemo-dbd/cadfael/geant4 --with-opengl-x11
$ brew install supernemo-dbd/cadfael/root5
```

Setup

In Bash (~/.bashrc):

```
export SNSW_BASE_DIR="/opt/sw/SuperNEMO-DBD"
function do_cadfaelbrew_setup()
{
    if [ -n "${CADFAELBREW_INSTALL_DIR}" ]; then
        echo "WARNING: Cadfaelbrew is already setup !" >&2
        return 1
    fi
    export CADFAELBREW_INSTALL_DIR="${SNSW_BASE_DIR}/Cadfaelbrew"
    if [ -n "${MANPATH}" ]; then
        export MANPATH="${CADFAELBREW_INSTALL_DIR}/share/man:${MANPATH}"
    else
        export MANPATH="${CADFAELBREW_INSTALL_DIR}/share/man"
    fi
    if [ -n "${INFOPATH}" ]; then
        export INFOPATH="${CADFAELBREW_INSTALL_DIR}/share/info:${INFOPATH}"
    else
        export INFOPATH="${CADFAELBREW_INSTALL_DIR}/share/info"
    fi
    mkdir -p /opt/var/cache/Homebrew
    export HOMEBREW_CACHE=/data/var/cache/Homebrew
    mkdir -p /opt/var/tmp
    export HOMEBREW_TEMP=/opt/var/tmp
    ${SNSW_BASE_DIR}/Cadfaelbrew/bin/brew sh
    return 0;
}
alias brewsh='do_cadfaelbrew_setup'
```

Test

Enter a dedicated Cadfaelbrew shell:

```
$ brewsh
```

Testing CLHEP (brew version) :

```
$ which clhep-config
/opt/sw/SuperNEMO-DBD/Cadfaelbrew/bin/clhep-config
```

Testing GSL (brew version) :

```
$ which gsl-config
/opt/sw/SuperNEMO-DBD/Cadfaelbrew/bin/gsl-config
$ gsl-config --prefix
/opt/sw/SuperNEMO-DBD/Cadfaelbrew/Cellar/gsl/1.16
```

Testing Root (brew version) :

```
$ which root
/opt/sw/SuperNEMO-DBD/Cadfaelbrew/bin/root
$ root
...
root [0] .q
```

Leave the dedicated Cadfaelbrew shell:

```
$ exit
```