Taskwarrior Installation from Source

Dirk Deimeke – Taskwarrior Academy

Preparation for December, 9, 2016

Taskwarrior Installation
Dirk Deimeke
dirk@deimeke.net

Contents

1 Preface

Our advice is that you install Taskwarrior from source since all of our documentation refers to paths which some packaged versions of Taskwarrior do not install Taskwarior and scripts to. The information provided in this document will guide you through the necessary steps to compile Taskwarrior from source.

2 Preparation

If your system is not listed here, we like to support you in installing to your device. If you managed to do that on your own, we like to hear from you as well.

Please send an email to dirk@deimeke.net (German or English).

In general you need a build environment, consisting of

- `GnuTLS+` (ideally version 3.2 or newer)
- `libuuid`
- `CMake+` (2.8 or newer)
- `GNU make`
- `GCC` version 4.7 or `Clang` version 3.3 or newer versions
  In general, you need a compiler supporting `C++11`.

2.1 CentOS, Fedora, openSUSE

```
yum install gnutls-devel libuuid-devel cmake gcc-c++ # or clang
# or
dnf install gnutls-devel libuuid-devel cmake gcc-c++ # or clang
# or
zypper install gnutls-devel libuuid-devel cmake gcc-c++ # or clang
```

2.2 Debian, Ubuntu

```
apt install libgnutls28-dev uuid-dev cmake g++ # or clang
```

2.3 macOS

Install Xcode from Apple, via the AppStore, launch it, and select from some menu that you want the command line tools.

In more recent versions of macOS, the command line tools have to be installed via `xcode-select --install` and Git will be installed as well, no need to do this via Homebrew.

With Homebrew or MacPorts install the necessary packages:

```
brew install cmake git gnutls
```

Darwin, FreeBSD . . . include the neccessary `libuuid` (mentioned above) functionality in libc.

3 Getting the source

3.1 Stable – current released version

3.1.1 tarball

Get the tarball with your download program of choice, extract it and change to the resulting directory.

```
> curl -O https://taskwarrior.org/download/task-2.5.1.tar.gz
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  882k  100  882k    0     0   185k      0  0:00:04  0:00:04 --:--:--  278k

> tar xzf task-2.5.1.tar.gz

> cd task-2.5.1
```

3.1.2 Git – master branch

Clone the repository from our repository server at `git.tasktools.org` – the master branch equals the current stable version – and change to the new directory.

```
> git clone https://git.tasktools.org/scm/tm/task.git task.git
Cloning into 'task.git'...
remote: Counting objects: 55986, done.
remote: Compressing objects: 100% (13215/13215), done.
remote: Total 55986 (delta 45388), reused 52425 (delta 42516)
Receiving objects: 100% (55986/55986), 25.46 MiB | 1.19 MiB/s, done.
Resolving deltas: 100% (45388/45388), done.
Checking connectivity... done.

> cd task.git
```

3.2 Developer Version – and versions greater or equal 2.6.0

3.2.1 Git – developer branch

As above clone from our repository server and change to the cloned directory.

Afterwards checkout the development branch (currently 2.6.0), initialize and update the submodule containing `libshared.git`, which includes the commonly used functions across all our projects.

All members of the team use the developer version. It is expected to work, but due to the fact that this is work in progress it could break any time.

```
> git branch --list
* 2.6.0
  master

> git checkout 2.6.0
Branch 2.6.0 set up to track remote branch 2.6.0 from origin.
Switched to a new branch '2.6.0'

> git submodule init
Submodule 'src/libshared' (https://git.tasktools.org/scm/tm/libshared.git)
    registered for path 'src/libshared'

> git submodule update
Cloning into 'src/libshared'...
remote: Counting objects: 2443, done.
remote: Compressing objects: 100% (1642/1642), done.
remote: Total 2443 (delta 1843), reused 1018 (delta 796)
Receiving objects: 100% (2443/2443), 391.76 KiB | 398.00 KiB/s, done.
Resolving deltas: 100% (1843/1843), done.
Checking connectivity... done.
Submodule path 'src/libshared': checked out '
    ce5c3414de56a2d1390893bbdc46e7116c38cd90'
```

3.2.2 Updating the Git repository

Please follow the following commands if you like to stay on top of development and want to rebuild Taskwarrior with a more recent version of the sourcecode.

```
> git clean -dfx

> git pull --recurse-submodules=yes

> git submodule update
```

4 Compile and run

Please note that the installation to standard directories – beneath `/usr/local` – needs root access on the system. For a custom directory installation, no root access is necessary.

If you use a version of `GCC` lower than 4.9 (for example on CentOS), you will see a lot of warnings, Taskwarrior will be compiled nevertheless.

4.1 Build and install

`CMake` is used to create a `Makefile` and afterwards Taskwarrior is built before it gets installed.

The installation to a custom directory adds a location – in this case `~/MYDIR` – to the Parameters of `CMake`.

Please note there is a dot "." as last parameter to cmake.

```
> cmake -DCMAKE_BUILD_TYPE=release .
# or cmake -DCMAKE_INSTALL_PREFIX=~/MYDIR -DCMAKE_BUILD_TYPE=release .
-- Configuring C++11
-- System: Linux
-- Looking for SHA1 references
-- Found SHA1 reference: 4da25ddd
-- Looking for GnuTLS
-- Looking for libuuid
-- Found libuuid
```


```
-- Configuring cmake.h
-- Configuring man pages
-- Configuring done
-- Generating done
-- Build files have been written to: /home/dirk/task.git

> make -j 4 # if '4' is the number of cores of your machine
Scanning dependencies of target libshared
Scanning dependencies of target task
Scanning dependencies of target columns
Scanning dependencies of target commands
[  0%] Building CXX object src/CMakeFiles/libshared.dir/libshared/src/FS.cpp.o
[  1%] [  3%] Building CXX object src/CMakeFiles/task.dir/CLI2.cpp.o
Building CXX object src/columns/CMakeFiles/columns.dir/Column.cpp.o
[  4%] Building CXX object src/commands/CMakeFiles/commands.dir/Command.cpp.o
...
Building CXX object src/CMakeFiles/task_executable.dir/main.cpp.o
[100%] Built target lex_executable
Linking CXX executable task
Linking CXX executable calc
[100%] Built target task_executable
[100%] Built target calc_executable

> sudo make install
# No sudo needed for a custom directory installation
[  6%] [ 31%] Built target libshared
Built target task
[ 54%] Built target columns
[ 98%] Built target commands
[ 98%] [ 98%] Built target calc_executable
[100%] Built target task_executable
Built target lex_executable
Install the project...
...
-- Installing: /usr/local/share/doc/task/scripts/add-ons
-- Up-to-date: /usr/local/share/doc/task/scripts/add-ons/README
-- Up-to-date: /usr/local/share/doc/task/scripts/add-ons/update-holidays.pl
```

4.2 Setting the environment for a custom directory installation

You do not need this step if you installed Taskwarrior to the default location.

After compiling and installing to your custom directory, it makes sense to set some environment variables and maybe add (persist) the settings in your local `.bashrc` or `.zshrc` or whatever shell you use.

```
export PATH=${PATH}${PATH:+:}~/MYDIR/bin
export MANPATH=${MANPATH}${MANPATH:+:}~/MYDIR/share/man
```

4.3 First test

If everything went well, you should see an output like the following when running `task diagnostics`. Just confirm the message that a configuration file is missing with *yes*.

```
> task diagnostics
A configuration file could not be found in

Would you like a sample /home/dirk/.taskrc created, so Taskwarrior can proceed?
     (yes/no) yes

task 2.6.0
   Platform: Linux

Compiler
    Version: 4.8.5 20150623 (Red Hat 4.8.5-4)
       Caps: +stdc +stdc_hosted +LP64 +c8 +i32 +l64 +vp64 +time_t64
 Compliance: C++11

Build Features
      Built: Nov 28 2016 12:34:32
     Commit: 4da25ddd
      CMake: 2.8.11
    libuuid: libuuid + uuid_unparse_lower
  libgnutls: 3.3.8
 Build type: release

Configuration
       File: /home/dirk/.taskrc (found), 1466 bytes, mode 100664
       Data: /home/dirk/.task (found), dir, mode 40755
```


```
    Locking: Enabled
         GC: Enabled
    $EDITOR: vim
     Server:
         CA: -
Certificate: -
        Key: -
      Trust: strict
    Ciphers: NORMAL
      Creds:

Hooks
     System: Enabled
   Location: /home/dirk/.task/hooks
             (-none-)

Tests
      $TERM: xterm-256color (144x62)
       Dups: Scanned 0 tasks for duplicate UUIDs:
             No duplicates found
 Broken ref: Scanned 0 tasks for broken references:
             No broken references found
```

5 Reference

Check the cheat sheet:

`/usr/local/share/doc/task/task-ref.pdf` or `~/MYDIR/share/doc/task/task-ref.pdf`

If anything went wrong don't hesitate to contact me dirk@deimeke.net.