Introduction

This guide leads you through all the necessary steps to setup your own Taskserver to sync your Taskwarrior-tasks.

Please follow the steps carefully and note all things you do differently.

Preparation – Backup Your Data

Let’s reinforce a good habit and make a backup copy of your data first. Here is a very easy way to backup your data:

```
$ cd ~/.task
$ tar czf task-backup-$(date +'%Y%m%d').tar.gz *
```

Now move that file somewhere safe. All software contains bugs, so make regular backups.

Attention!

This is not only due to a good habit, we will modify your data, so this backup is highly recommended.

Preparation – Choose A Machine

A suitable machine to run your Taskserver is one that is always available. If you have such a machine, or have access to a hosted machine, that is ideal.

If your machine is not continuously available, it can still be a suitable Taskserver because the sync mechanism doesn't require continuous access. When a client cannot sync, it simply accumulates local, unpropagated changes until it can sync.

A laptop is a poor choice for a Taskserver host.

Preparation – Choose A Port

By default, Taskserver uses port 53589. You can choose any port you wish, provided it is unused. If you choose a port number that is under 1024, then Taskserver must run as root, which is not recommended.

Preparation – User/Group

Ideally you will create a new user and group solely to run the Taskserver. This helps you keep the data secure from other users on the machine, as well as controlling the privileges of Taskserver.

Preparation – Firewall

Depending on what devices you use to access your server, you may need to configure the firewall to allow incoming TCP/IP traffic on your chosen port.

Installation

Installing Taskserver from a binary package is the simplest option, but you will need to refer to your package manager's documentation and procedures for doing this.

Take a look at the Download page for examples. Generally there are too many package managers to make a complete list with instructions here.

Most importantly, for now, Taskserver is a new product, and there are very few packages available. It is expected that this situation will change soon. When it does, this page will be updated.

Meanwhile, there is installation from either git or tarball.

Installation – Introduction

Installing Taskserver from a tarball is a matter of downloading the tarball, extracting it, satisfying dependencies and building the server.

Installation – Dependencies (general)

Before building the software, you will need to satisfy the dependencies by installing the following:

- GnuTLS (ideally version 3.2 or newer)
- `libuuid`
- CMake (2.8 or newer)
- `make`
- A C++ Compiler (GCC 4.7 or Clang 3.0 or newer)

Note that some OSes (Darwin, FreeBSD ...) include `libuuid` functionality in libc, check the following slides for more detailed instructions.

You don't necessarily need the latest version of all components, but it is a good idea if you can. GnuTLS is a security component, and as such, it is very important that it is current.

Using GnuTLS version 2.12.x is neither adequately secure, nor production quality.

Installation – Dependencies (OS)

We have detailed instructions for the following operating operating systems (click on the name to continue):

- CentOS
- Debian
- Fedora
- openSUSE
- Ubuntu
- Windows with Cygwin
- Mac OS X

In case you can add your operating system of choice, please send an email to support@taskwarrior.org (Thank you!).

Installation – CentOS

Before building the software, you will need to satisfy the dependencies by installing the following:

```
$ sudo yum install gcc-c++
$ sudo yum install gnutls-devel
$ sudo yum install libuuid-devel
$ sudo yum install cmake
$ sudo yum install make
```

Continue with Download.

Installation – Debian

Before building the software, you will need to satisfy the dependencies by installing the following:

```
$ sudo apt install gcc-c++
$ sudo apt install gnutls-devel
$ sudo apt install libuuid-devel
$ sudo apt install cmake
$ sudo apt install make
```

Continue with Download.

Installation – Fedora

Before building the software, you will need to satisfy the dependencies by installing the following:

```
$ sudo dnf install gcc-c++
$ sudo dnf install gnutls-devel
$ sudo dnf install libuuid-devel
$ sudo dnf install cmake
$ sudo dnf install make
```

Continue with Download.

Installation – openSUSE

Before building the software, you will need to satisfy the dependencies by installing the following:

```
$ sudo zypper install gcc-c++
$ sudo zypper install gnutls-devel
$ sudo zypper install libuuid-devel
$ sudo zypper install cmake
$ sudo zypper install make
```

Continue with Download.

Installation – Ubuntu

Before building the software, you will need to satisfy the dependencies by installing the following:

```
$ sudo apt install gcc-c++
$ sudo apt install gnutls-devel
$ sudo apt install libuuid-devel
$ sudo apt install cmake
$ sudo apt install make
```

Continue with Download.

Installation – Windows

Before building the software, you will need to satisfy the dependencies by installing the following:

Start the Cygwin GUI and install the following packages and their dependencies.

- `GnuTLS`
- `libuuid`
- `CMake`
- `make`
- `gcc-c++`

Continue with Download.

Installation – Mac OS X

Before building the software, you will need to satisfy the dependencies by installing the following:

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

We expect you to have Homebrew installed on your Mac.

```
$ brew install cmake
$ brew install git
$ brew install gnutls
```

Continue with Download.

Installation – Download

The next step is to obtain the code. This means getting the Task Server 1.0.0 (or newer) source tarball. You should check for the latest stable release here:

http://taskwarrior.org/download/

You can download the tarball with `curl`, as an example of just one of many ways to download the tarball.

```
$ curl -O http://taskwarrior.org/download/taskd-latest.tar.gz
```

Installation – Build

Expand the tarball, and build the Taskserver.

```
$ tar xzf taskd-latest.tar.gz
$ cd taskd-latest
$ cmake -DCMAKE_BUILD_TYPE=release .
...
$ make
...
```

Installation – Build Again

If you ever want to build the software again, do some cleanup.

```
$ cd taskd-latest
$ make clean
...
$ rm CMakeCache.txt
...
```

Installation – make install

Now install Taskserver. This copies files into the right place, and installs man pages.

```
$ sudo make install
...
```

Installation – Verify installation

Run the `taskd` command to verify that the server is installed, and the location is in your $PATH.

You should see something like this:

```
$ taskd

Usage: taskd -v|--version
      taskd -h|--help
      taskd diagnostics
      taskd validate <JSON | file>
      taskd help [<command>]

Commands run only on server:
      taskd add [options] org <org>
      taskd add [options] group <org> <group>
      taskd add [options] user <org> <user>
      taskd config [options] [--force] [<name> [<value>]]
      taskd init [options]
      taskd remove [options] org <org>
      taskd remove [options] group <org> <group>
      taskd remove [options] user <org> <user>
      taskd resume [options] org <org>
      taskd resume [options] group <org> <group>
      taskd resume [options] user <org> <user>
      taskd server [options] [--daemon]
      taskd status [options]
...
```

Installation – from Git-Repository

Installing Taskserver from git is a matter of cloning the git repository and building the server.

The same dependencies as for installation from tarball apply. We have detailed instructions for the following operating operating systems (click on the name to continue), afterwards come back to this slide (23) or continue with cloning the repository:

- CentOS
- Debian
- Fedora
- openSUSE
- Ubuntu
- Windows with Cygwin
- Mac OS X

Installation – Cloning the repository

Now clone the repository like this:

```
$ git clone https://git.tasktools.org/scm/tm/taskd.git taskd.git
...
```

Use stable!

It is highly recommended that you build the stable version. This involves simply moving on to the next step, build.

Only under special circumstances should you build the unstable development version.

Installation – Special Circumstances (1)

The unstable development version is at no point guaranteed to work or even compile. The only time it does stabilize is right at the end of the development cycle, and in that case, you should wait until the release, so you are running a supported version.

The stable version is always merged to the `master` branch, which is the default branch, so ordinarily nothing needs to be done. To build an unstable branch, first determine which branch by looking at the available branches:

```
$ cd taskd.git
$ git branch -a
* master
  remotes/origin/1.1.0
  remotes/origin/1.1.1
  remotes/origin/1.2.0
  remotes/origin/HEAD -> origin/master
  remotes/origin/master
```

Installation – Special Circumstances (2)

The convention we use is that `master` represents the stable release. The numbered branches represent the latest development (1.2.0, the ’highest’ branch number, ending in ’.0’) and a patch branch (1.1.1, ending in a non-zero number).

Patch branches are reserved for emergency releases, so in this example you would choose to build 1.2.0 as the latest development branch like this:

```
$ git checkout 1.2.0
Branch 1.2.0 set up to track remote branch 1.2.0 from origin.
Switched to a new branch ’1.2.0’
```

Installation – Build from Git

Now build the Taskserver.

```
$ cd taskd.git
$ cmake -DCMAKE_BUILD_TYPE=release .
...
$ make
...
```