OxOOL community edition compiling HOWTO

If you want to compile the latest version of the OxOffice Online community edition on GitHub, please refer to this document for instructions on how to set up the compilation environment and the compilation process for the OxOffice Online community edition.

- OxOOL v4 community edition compiling on Ubuntu 20.04 HOWTO
- OxOOL v5 community edition compiling on Rocky Linux 8 HOWTO
- OxOOL v5 community edition compiling on Ubuntu 24.04 HOWTO

OxOOL v4 community edition compiling on Ubuntu 20.04 HOWTO

1. First thing first, install a Ubuntu 20.04 server! Can it be run on Ubuntu 20.04 desktop? Yes, but why bother? :-)
2. Upgrade the system:

```
sudo apt update
sudo apt upgrade -y
```

3. Make sure you install OpenSSH and Vim (I'm a Vim guy! All right, but remember to install your favorite editor, okay?) and a couple of tools:

```
sudo apt install vim openssh-server net-tools curl git -y
```

4. Import OSSII public key

```
curl http://www.oxoffice.com.tw/deb/OSSII.key | sudo apt-key add
```

5. Set up OxOOL repository

```
cd /etc/apt/sources.list.d/
sudo wget http://www.oxoffice.com.tw/deb/oxool-community-v4-focal.list
```

6. Install software packages necessary for compiling OxOOL (use option apt install -y if you don't like to be asked everytime)

```
sudo apt update
sudo apt install oxoffice* libpoco-dev libgumbo-dev
sudo apt install build-essential libsqlite3-dev libcurl4-openssl-dev libcppunit-dev libcap-dev
libcap2-bin libgit2-dev libtool libpng-dev automake m4 wget curl autoconf pkg-config openssl
libgumbo-dev ccache fontconfig libfontconfig1-dev libpam-dev
sudo apt install devscripts debhelper dh-systemd dh-exec
sudo apt install libodbc1 libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5
```

```
sudo apt install hunspell
sudo apt install python3-polib python3-lxml
```

7. Install and setup Node.js

```
cd ~
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
sudo apt install -y nodejs
```

8. Clone OxOOL-community code from GitHub and switch to branch `v4`

```
cd ~
git clone https://github.com/OSSII/oxool-community.git
cd oxool-community
git switch v4
```

9. For debug and developing, run autogen.sh directly WITHOUT running configure

```
cd oxool-community
./autogen.sh
```

It will enable debug options so that you can test and debug by running `make run`.

10. For installing, you can build deb packages directly: `debuild -b -uc -us` The built deb packages will be one level up the compiling directory:

```
cd ..
ls oxool_4.x.x-1.community_amd64.deb
```

The generated deb files can be then installed in the system.

OxOOL v5 community edition compiling on Rocky Linux 8 HOWTO

Please install Rocky Linux 8 first, and then execute the following commands to update the system and install the necessary software packages (note that this is based on logging in as a regular user, and it is not recommended to use the root account during compilation).

```
sudo dnf config-manager --set-enabled powertools
sudo dnf upgrade -y
sudo dnf install vim net-tools unzip wget -y
sudo wget -P /etc/yum.repos.d http://www.oxoffice.com.tw/rpm/el/oxool-community-v5-el8.repo
sudo dnf install epel-release -y
sudo dnf upgrade -y
sudo dnf install make git yum-utils libpcap libcap-devel libgit2-devel libzstd-devel poco-devel ccache python3-devel python3-polib python3-lxml -y
sudo dnf install libtool gcc-c++ fontconfig-devel cppunit-devel pam-devel rpm-build -y
sudo dnf install oxoffice* poco* gumbo* -y
```

Disable SELinux

```
vim /etc/selinux/config # vim □□□□□□□□□□□□□□□□
# □□ SELinux=enforcing □□□□ enforcing □□ disabled □ permissive □□□□□
```

Install node.js

```
sudo dnf module enable nodejs:18 -y
sudo dnf install nodejs -y
```

Clone source code of OxOffice Online v5 Community

```
git clone https://github.com/OSSII/oxool-community.git
cd oxool-community
git switch v5
```

Compile and package OxOffice Online v5 Community Edition

```
./autogen.sh
make dist
rpmbuild -tb oxool-x.x.x.tar.gz
```

The compiled RPM software package will be placed in the following path

```
~/rpmbuild/RPMS/x86_64
```

The RPM software package here can be used for installation.

OxOOL v5 community edition compiling on Ubuntu 24.04 HOWTO

1. First thing first, install a Ubuntu 24.04 server! Can it be run on Ubuntu 24.04 desktop? Yes, but why bother? :-)
2. Upgrade the system:

```
sudo apt update
sudo apt upgrade -y
```

3. Make sure you install OpenSSH and Vim (I'm a Vim guy! All right, but remember to install your favorite editor, okay?) and a couple of tools:

```
sudo apt install vim openssh-server net-tools curl git -y
```

4. Import OSSII public key

```
sudo curl -o /etc/apt/keyrings/OSSII.asc http://www.oxoffice.com.tw/deb/OSSII.key
```

5. Set up OxOOL repository

```
sudo curl -o /etc/apt/sources.list.d/oxool-community-v5-noble.list
http://www.oxoffice.com.tw/deb/oxool-community-v5-noble.list
```

6. Install software packages necessary for compiling OxOOL (use option apt install -y if you don't like to be asked everytime)

```
sudo apt update
sudo apt install oxoffice* libpoco-dev libgumbo-dev
sudo apt install -y build-essential libsqlite3-dev libcurl4-openssl-dev libcppunit-dev libcap-dev libcap2-bin libgit2-dev libtool libpng-dev automake m4 wget autoconf pkg-config openssl libgumbo-dev ccache fontconfig libfontconfig1-dev libpam-dev
sudo apt install -y devscripts debhelper dh-exec
sudo apt install -y libpcre16-3 libpcre3-dev libpcre32-3 libpcrecpp0v5
```

```
sudo apt install -y hunspell
sudo apt install -y python3-polib python3-lxml
```

7. Install and setup Node.js

```
cd ~
curl -fsSL https://deb.nodesource.com/setup_20.x -o nodesource_setup.sh
sudo -E bash nodesource_setup.sh
sudo apt install -y nodejs
```

8. Clone OxOOL-community code from GitHub and switch to branch `v5`

```
cd ~
git clone https://github.com/OSSII/oxool-community.git
cd oxool-community
git switch v5
```

9. For debug and developing, run autogen.sh directly WITHOUT running configure

```
cd oxool-community
./autogen.sh
```

It will enable debug options so that you can test and debug by running `make run`.

10. For installing, you can build deb packages with following command. The built deb packages will be one level up the compiling directory:

```
./autogen.sh
debuild -b -uc -us
cd ..
ls oxool*.deb
```

The generated deb files can be then installed in the system.