

My github: <https://github.com/Billour/hello>

My launchpad : <https://launchpad.net/~billour/hello>

Version: 20230518

1. Get the source code of that Debian package.
2. Add an executable bash script called “testing.sh”
 - Executing testing.sh will echo “this is a test from YuTai(Billour) Ou ” on the console.
3. Please echo string “this is a test from YuTai(Billour) Ou” during the Debian package installation time.
4. Host all of your change to a public git repository (e.g. launchpad git, Github or GitLab)
5. Dput the source change to your ppa.
6. Install your package from your ppa.
7. Past the message printed by the installation.
 - The string “this is a test from YuTai(Billour) Ou” is expected.
8. Past the executing result of `dpkg -S testing.sh; testing.sh`
 - The string “this is a test from {your name}” is expected.
9. Record things that you experienced for this task on this mail as a document.
 - Image the steps is to guide people who did not experience this task to finish this task.

Please submit the following result in a PDF file

1. **Get the source code of that Debian package.**

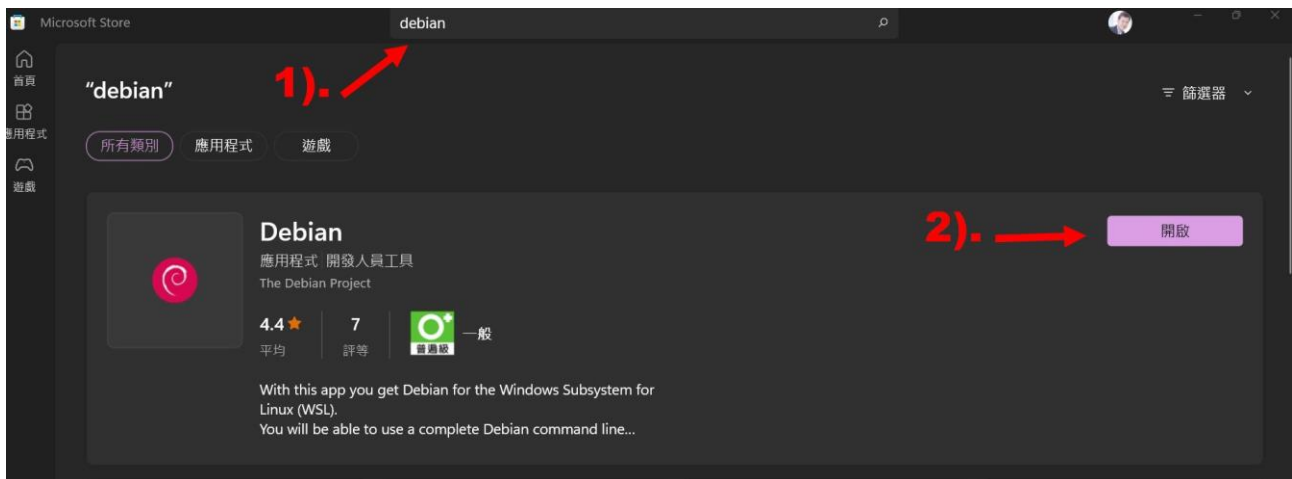
Using the Windows 11 and Searching ‘debian’ on the search bar of **Microsoft Store**.

- 1). Searching ‘debian’.
- 2). Install the Debian and open it.
- 3). Typing “**sudo apt-get update**”, to update the last version.
 - * It’s “sudo(space)apt-get(space)update”[p12]
 - * When you type your password on the console, this is normal that you can’t see any letters of the alphabet.[p13]

Just like the following the picture.

A terminal window showing the command prompt [sudo] password for billourcode: with the cursor at the end of the line.

- 4).



[p11]

And you will see the following:

```

選取 billourcode@AMD-desktop: ~
Windows 子系統 Linux 版現已在 Microsoft Store 中推出！
您可以執行 'wsl.exe --update' 進行升級或瀏覽 https://aka.ms/wslstorepage
從 Microsoft Store 安裝 WSL 將會為您提供最新的 WSL 更新，且更快速。
如需詳細資訊，請瀏覽 https://aka.ms/wslstoreinfo

billourcode@AMD-desktop:~$ sudo apt-get update

```

[p12]

```

billourcode@AMD-desktop:~$ sudo apt-get update
[sudo] password for billourcode:

```

[p13]

Getting the source code of a Debian package : Here's a step-by-step tutorial to help you through the process:

1). Install the required tools

Make sure you have the necessary tools installed on your system. Open a terminal and run the following command to install them:

```

sudo apt update
sudo apt install build-essential

```

2). Enable source repositories

Debian provides source repositories alongside the binary repositories. You need to enable these repositories to access the source code. Open the `/etc/apt/sources.list` file in a text editor with root privileges:

```

sudo vim /etc/apt/sources.list

```

Uncomment or add the `deb-src` line(s) for your desired Debian version. It should look similar to this:

```

deb-src http://archive.ubuntu.com/ubuntu/ focal main restricted
deb-src http://security.ubuntu.com/ubuntu/ focal-security main restricted

```

Save the file and exit the text editor.

3). Update the package lists

To ensure you have the latest package information, update the package lists by running the following command:

```
sudo apt update
```

4). Navigate to the source code directory

After the extraction is complete, navigate to the directory containing the source code. The directory name typically matches the package name. Use the cd command to change to the appropriate directory.

2. Add an executable bash script called "testing.sh"

Executing testing.sh will echo "this is a test from YuTai(Billour) Ou " on the console.

To create a bash script and modify the Debian package installation configuration. Here's an example:

1). Create the "testing.sh" script

Open a text editor and create a file named "testing.sh" with the following contents:

```
#!/bin/bash
echo "This is a test from YuTai(Billour) Ou"
exit 0
```

Save the file and make it executable by running the following command in the terminal:[p21]

```
sudo chmod +x testing.sh
```

"sudo(space)chmod(space)+x(space)testing.sh"

*When you'd like to use the command 'chomd', you should have to use 'sudo' model.[p21]

If you only type 'chmod +x testing.sh', you will get an error message '-bash:chomd: command not found'.

```
billourcode@AMD-desktop:~$ chomd +x testing.sh
-bash: chomd: command not found
billourcode@AMD-desktop:~$ sudo chmod +x testing.sh
[sudo] password for billourcode:
billourcode@AMD-desktop:~$ ls
testing.sh
```

[p21]

When you finish the above, you can use the command './testing.sh' to execute it.[p22]

```
billourcode@AMD-desktop:~$ ./testing.sh
This is a test from YuTai(Billour) Ou
billourcode@AMD-desktop:~$
```

[p22]

3. **Please echo string “this is a test from YuTai(Billour) Ou” during the Debian package installation time.**

Modify the Debian package installation configuration open the terminal and navigate to the directory, ‘~’ where you want to create your Debian package. Create a new directory (e.g., "hello") and enter it:

1). Command ‘mkdir hello’, and navigate into the directory ‘cd hello’ and create a new directory, it name ‘DEBIAN’, it like the picture 31 [p31]. The file tree will like the picture 32.

2). Using the command ‘vim control’ to create a file as named control on the ‘~/hello/DEBIAN’. If you can’t understand archival Hierarchy, please to refer the picture 32 [p32]. In the editor and add following content as same as picture 33 [p33] and Save the file.

3). Inside the ‘~/hello/DEBIAN/’ directory, create a new file named ‘postinst’ using the following command, ‘vim postinst’ in the editor and add following content:

```
#!/bin/bash
echo "This is a test from YuTai(Billour) Ou"
exit 0
```

4) Save the postinst file and make it executable. After you executed the command, there will be to like the picture on the console, the picture 34 [p34].

```
sudo chmod +x postinst
```

5). Build the Debian package. Assuming you have the necessary tools installed, you can use the dpkg-deb command to build the Debian package. In the terminal, run the following command, ‘sudo dpkg-deb –build ./’ from the "hello" directory. If you are in a wrong path of directory, you will get an error from dpkg-deb on console, just like the picture 34 [p34]. So I recommend you use the absolute path to build the package ‘hello’, ‘~/hello/’ the success picture will be like the picture 35 [p35]. The picture 36 [p36] show the right path and directory, the path is the key of success to build the package.

```
dpkg-deb --build ~/hello/
```

6). This will create a Debian package file with the extension .deb in the current directory, the picture 36 [p36], there will be the 'hello.deb' it will show red color on the console.



7). Install the package, use the 'apt' to install the 'hello.deb'. You can also use the path '/home/billourcode/hello.deb', it's as same as './hello.deb', the picture 36 [p36], [p37], [p38].

```
sudo apt install ~/hello.deb
```

```
sudo apt install /home/billourcode/hello.deb
```

8). Uninstall the package, use the command 'dpkg' to uninstall the 'hello.deb', it'll be like as same as the picture 310 [p310].

```
sudo dpkg -P hello
```

The set of commands.

```
mkdir hello
cd hello
mkdir DEBIAN
cd DEBIAN
vim control
vim posinst
sudo chmod +x postinst
dpkg-deb --build ~/hello/
```

The executed picture on the console.

```
~/hello$ mkdir DEBIAN
~/hello$ cd DEBIAN/
~/hello/DEBIAN$ vim control
```

[p31]

The tree view of the hello directory.

```
billourcode@AMD-desktop:~/hello$ tree ~/hello
/home/billourcode/hello
├── DEBIAN
│   ├── control
│   └── postinst
```

[p32]

* If your computer don't have the 'tree' command and please install it on your computer.

```
sudo apt install tree
```

2). Using the command 'vim control' to create a file as named control on the '~/'hello/DEBIAN'. If you can't understand archival Hierarchy, please to refer the picture 32 [p32]. In the editor and add following content as same as picture 33 [p33].

```
Package: hello
Version: 1.0
Architecture: all
Essential: no
Priority: optional
Depends:
Maintainer: YuTai(Billour) Ou
Description: Take Home Test from Billour Canonical.
```

[p33]

4) Save the postinst file and make it executable. After you executed the command, there will be to like the picture on the console, the picture 34 [p34].

```
billourcode@AMD-desktop:~$ mkdir hello
billourcode@AMD-desktop:~$ cd hello
billourcode@AMD-desktop:~/hello$ touch postinst
billourcode@AMD-desktop:~/hello$ ls
postinst
billourcode@AMD-desktop:~/hello$ vim postinst
billourcode@AMD-desktop:~/hello$ sudo chmod +x postinst
[sudo] password for billourcode:
billourcode@AMD-desktop:~/hello$ ls
postinst
billourcode@AMD-desktop:~/hello$ sudo dpkg-deb --build ./
dpkg-deb: error: failed to open package info file './DEBIAN/control' for reading: No such file or directory
billourcode@AMD-desktop:~/hello$
```

[p34]

5). Build the Debian package, '~/'hello/', the success picture will be like the picture 35 [p35].

```
billourcode@AMD-desktop:~/hello/DEBIAN$ cd ..
billourcode@AMD-desktop:~/hello$ cd ..
billourcode@AMD-desktop:~$ dpkg-deb --build ~/hello/
dpkg-deb: building package 'hello' in '/home/billourcode/hello.deb'.
billourcode@AMD-desktop:~$
```

[p35]

The picture 36 [p36] show the right path and directory, the path is the key of success to build the package.


```
billourcode@AMD-desktop:~/hello/DEBIAN$ cd /home/billourcode/
billourcode@AMD-desktop:~$ ls
hello  hello.deb  testing.sh
```

[p36]

How to Install the 'hello.deb'

```
sudo apt install ./hello.deb
```

You also use the path '/home/billourcode/hello.deb', it's as same as './hello.deb'

```
sudo apt install /home/billourcode/hello.deb
```

```
:~$ sudo apt install /home/billourcode/hello.deb
```

[p37]

When you've finished it.

```
billourcode@AMD-desktop:~$ sudo apt install /home/billourcode/hello.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'hello' instead of '/home/billourcode/hello.deb'
The following NEW packages will be installed:
  hello
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/776 B of archives.
After this operation, 0 B of additional disk space will be used.
Get:1 /home/billourcode/hello.deb hello all 1.0 [776 B]
Selecting previously unselected package hello.
(Reading database ... 11927 files and directories currently installed.)
Preparing to unpack /home/billourcode/hello.deb ...Unpacking hello (1.0) ...
Setting up hello (1.0) ...
this is a test from YuTai(Billour) Ou.
```

[p38]

When you've finished it.

```
billourcode@AMD-desktop:~$ sudo apt install ./hello.deb
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'hello' instead of './hello.deb'
hello is already the newest version (1.0).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

[p39]

How to uninstall the 'hello.deb'

```
sudo dpkg -P hello
```

```
billourcode@AMD-desktop:~$ sudo dpkg -P hello
(Reading database ... 11927 files and directories currently installed.)
Removing hello (1.0) ...
```

[p310]

4. Host all of your change to a public git repository (e.g. launchpad git, Github or GitLab)

My github: <https://github.com/Billour/hello>

1). Install Git:

Open the terminal on your Linux Debian system.

Run the following command to install Git:

```
sudo apt update  
sudo apt install git
```

2). Configure Git:

Set your name and email for Git using the following commands:

```
sudo git config --global user.name "Your Name"  
sudo git config --global user.email "Your.email"
```

3). Create a new repository on GitHub:

- Open your web browser and go to <https://github.com/>.
- Log in to your GitHub account (create one if you don't have it).
- Click on the '+' button in the top-right corner and select "New repository".
- Enter a repository name (e.g., 'hello'), add an optional description, and choose other settings as needed.
- Click on the "Create repository" button.

4). Initialize Git repository:

- Go back to your terminal.
- Navigate to the project directory where your 'hello' project is located using the cd command. For example:

```
cd ~/hello/
```

- Run the following command to initialize a new Git repository:

```
git init
```

5). Add files to the repository:

Use the following command to add all files and directories in the 'hello' project to the repository:

```
git add .
```

* “git(space)add(space).”

6). Commit the changes:

Committing captures a snapshot of the project at that point in time. Run the following command to commit the changes:

```
git commit -m "Initial commit"
```

7). Add the remote repository:

In the GitHub repository you created earlier, copy the repository's URL. It should be something like <https://github.com/your-username/hello.git>.

Back in the terminal, run the following command to add the remote repository:

```
git remote add origin https://github.com/Billour/hello.git
```

Note: If you are the owner of the github, `git@github.com:Billour/hello.git`, you can use the SSH key, and the tutorial how to upload git

<https://www.youtube.com/watch?v=Vlmlj680d3o&t=6s>

8). Push the code to GitHub:

Finally, push the committed changes to GitHub using the following command:

```
git push -u origin master
```

You may need to enter your GitHub username and password to authenticate.

9). To download a GitHub repository using git clone, follow these steps:

Open your terminal or command prompt.

Navigate to the directory where you want to clone the repository.

Execute the following command:

```
git clone https://github.com/Billour/hello.git
```

5. **Dput the source change to your ppa.**

Search PPAs matching:

https://launchpad.net/ubuntu/+ppas?name_filter=hello

Use the dput command.

1). Install dput

Make sure you have dput installed on your system. If you don't have it installed, run the following command:

```
sudo apt install dput
```

2). Configure dput

You need to create a configuration file for dput that specifies your PPA. Open a terminal and create a new file called `~/.dput.cf`:

```
vim ~/.dput.cf
```

Add the following lines to the file, replacing your-ppa-name with the name of your PPA:

```
[ppa]  
fqdn = ppa.launchpad.net
```

```
method = ftp
incoming = ~billour/ubuntu/hello/
login = anonymous
allow_unsigned_uploads = 0
```

Save the file and exit the text editor, vim.

3). Prepare the source package

Create a source package of the changes you want to upload. Make sure the package has a “.changes” file, which is generated by running the `debuild` command in the root directory of the package source. For example:

```
cd hello/
debuild -S -sa
```

4). Upload the source package

Use the “dput” command to upload the source package to your PPA. Run the following command, replacing “hello_source.changes” with the name, hello, of your source package:

```
dput ppa hello
```

This will upload the source package to your PPA and make it available for building and distribution.

6. Install your package from your ppa.

To install a package named 'hello' from a Personal Package Archive (PPA) on Ubuntu Linux, you can follow these steps:

1). Open a terminal on your Debian/Ubuntu system.

Add the PPA to your system by running the following command:

```
sudo add-apt-repository ppa:billour/hello
```

Replace <ppa_name> with the name of the PPA and <ppa_repository> with the repository you want to add. For example, if the PPA is ppa:myppa/hello, you would run:

```
sudo add-apt-repository ppa:billour/hello
```

Enter your password when prompted and press Enter to confirm adding the PPA.

2). Update the package lists on your system by running:

```
sudo apt update
```

This command will refresh the available packages and their versions.

3). Once the package lists are updated, you can install the 'hello' package using the following command:

```
sudo apt install hello
```

Confirm the installation by entering 'y' when prompted.

After the installation is complete, you can start using the 'hello' package as per its documentation or by executing the associated command.

7. **Past the message printed by the installation.**

The string “this is a test from YuTai(Billour) Ou” is expected.

```
If [ -d “/usr/share/oem-workaround/” ]; then
```

*It’s “**If(space)[(space)-d(space)“/usr/share/oem-workaround/”(space)];(space)then**”

Important: The red color of letters of the alphabet.

- I. “/usr/share/oem-workaround/” # / It’s a directory.
- II. [**(space)-d(space)“/usr/share/oem-workaround/”]; # You have to (space)**
- III. [-d “/usr/share/oem-workaround/”]**;** #**;** Note have to add ;

```
If test -f “$PATH_USR”; then  
    sudo rm $PATH_USR  
fi
```

In vim, you can use **:set nu** to show line numbers.

```
1 #!/bin/bash  
2  
3 PATH_SH_USR="/usr/bin/tes  
4  
5 sudo touch $PATH_SH_USR  
6 sudo echo '#!/bin/bash' >  
7 sudo echo 'echo \"this is  
8 sudo echo 'exit 0' >> $PA  
9  
10 exit 0  
11
```

*Linux bash Permission denied: When you get the error message, you can do the following steps to solve the issue.

```
:/usr/bin/test.sh: Permission denied
```

```
chmod u+x test.sh  
chmod 777 test.sh
```

8. **Past the executing result of `dpkg -S testing.sh; testing.sh`**

The string “this is a test from {your name}” is expected.

```
#!/bin/bash  
#Author:  
#Date : 20230515  
#Version: 0.1  
#Email: billourcode@gmail.com  
  
PATH_USR='/usr/share/oem-workaround/timesyncd.conf'  
PATH_ETC='/etc/systemd/timesyncd.conf'  
  
PATH_SH_USR='/usr/bin/test.sh'  
PATH_T='test.sh'  
  
MY_MSG='this is a test from YuTai(Billour) Ou'  
  
{  
    # + replace_file Start  
    if test -f "$PATH_USR"; then  
        sudo rm $PATH_USR  
    fi  
  
    if [ ! -d "/usr/share/oem-workaround/" ]; then  
        sudo mkdir "/usr/share/oem-workaround/"  
    fi  
  
    sudo touch $PATH_USR  
    echo "+ replace_file=$PATH_USR"  
  
    #END of + replace_file  
  
    # + taget Start
```

```
sudo touch $PATH_ETC  
echo "+ target=$PATH_ETC"
```

```
# END of + taget
```

```
echo "+ echo $MY_MSG"
```

```
echo $MY_MSG  
echo "+ exit 0"
```

```
echo "Processing triggers for man-db (2.9.1-1)..."  
echo "Processing triggers for install-infor(6.7.0.dfsg.2-5)..."
```

```
echo " ";
```

```
#dpkg Start---  
if test -f "$PATH_T"; then  
    sudo rm $PATH_T  
fi
```

```
if test -f "$PATH_SH_USR"; then  
    sudo rm $PATH_SH_USR  
fi
```

```
#sudo touch $PATH_T  
sudo echo '#!/bin/bash' > $PATH_T  
#sudo chmod u+x $PATH_T  
sudo echo 'echo hello: \usr/bin/test.sh ' >> $PATH_T  
sudo echo 'echo this is a test from YuTai(Billour) Ou ' >> $PATH_T  
sudo echo 'exit 0' >> $PATH_T  
sudo chmod u+x $PATH_T  
sudo cp $PATH_T $PATH_SH_USR  
sudo chmod u+x $PATH_SH_USR  
#delete the temp file  
sudo rm $PATH_T
```

```
#Exec Result
```

```
sudo dpkg -S {$PATH_SH_USR}; $PATH_SH_USR
```

```
#sudo $PATH_SH_USR
```

```
#END---

    exit 0
} || {

    exit 9
}
```

9. **Record things that you experienced for this task on this mail as a document.**

Image the steps is to guide people who did not experience this task to finish this task.

Thank you for you watch this article.

About the Author Billour's Resume (歐育漆):

2023 First-Place Winner, National Yunlin University of Science & Technology,

Department of Electrical Engineering, Master Degree.

2022 Awesome Article Author, The IT Help ithome Forum.

2022 Google Technical Lead, Job Interview Invitation.

2022 FPGA Embedded System Acceleration Programming Design Certificate.

2006 First-Place Winner, The 2nd DSP (Digital Signal Processing) Creative Design Contest,
Taiwan(Organized by the Ministry of Education, Taiwan)

2005 Database Programming Design And Implementation(RDBMs)

2005 Level B technician for Computer Hardware Fabrication