

Universal Linux Environment Installer

Description

This is an environment installer for Linux that will setup a clean installation of Linux ready to use. It consists of a series of smaller install scripts and a GUI that gives the user an overview of the different components and an easy way of choosing which to install.

As of now, it is not completely automated and input may still be required from the user under the various installations. Please finish up all external installation procedures before closing the GTG installer, as it will clean up all downloaded installation files when it quits.

Supported Distro's

- Ubuntu
- Open SUSE (work in progress)

How To Use

Copy/Clone the `install_script` folder to your home folder.

Open a terminal and navigate to the `install_script` folder and run the script.

```
cd install_script  
./run
```

First, the script will update your system and install Java to run the installer GUI.

Follow the instructions by the installer

When Java is installed you will be greeted by the welcome screen



Rovsing

Welcome to the Rovsing GTG Linux installer

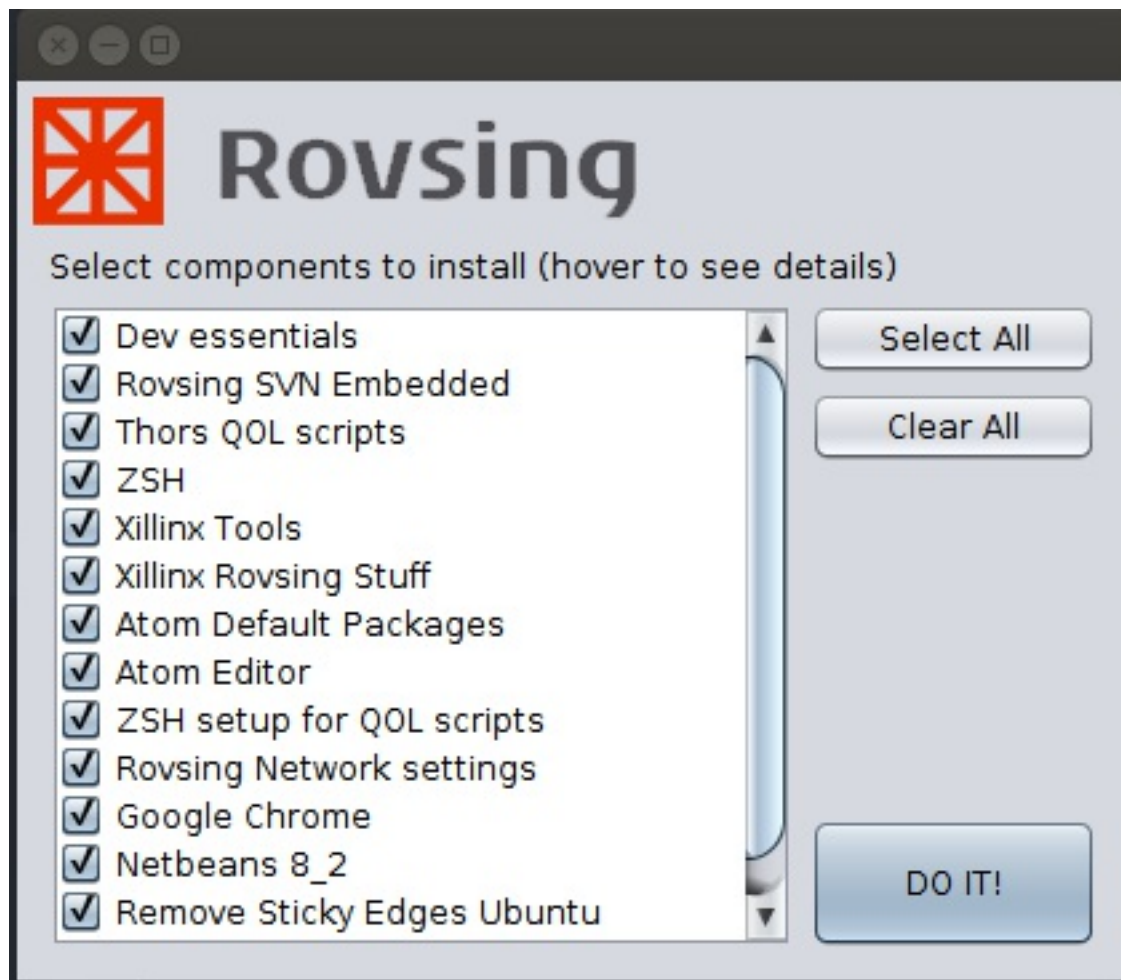
This installer will install any
required programs and configure your
environment so it's ready to use. Enjoy 😊

Linux username

Linux password

[Next](#)

Make sure the username is correct and type your `sudo` password in the password field.
The installer will handover the password to the scripts that needs it.



Each install script is represented as a component on the selection screen.

Select the components that you want to install.

If the install script has a description it will be shown as a tooltip, that will be shown when you hover your mouse over its component.

Some components may depend on each other (should be clear by the naming), but for now it is not possible to see dependencies (future improvements).

When you have selected the components to install click the "DO IT" button and the installer will run through the selected scripts and install/setup the components.



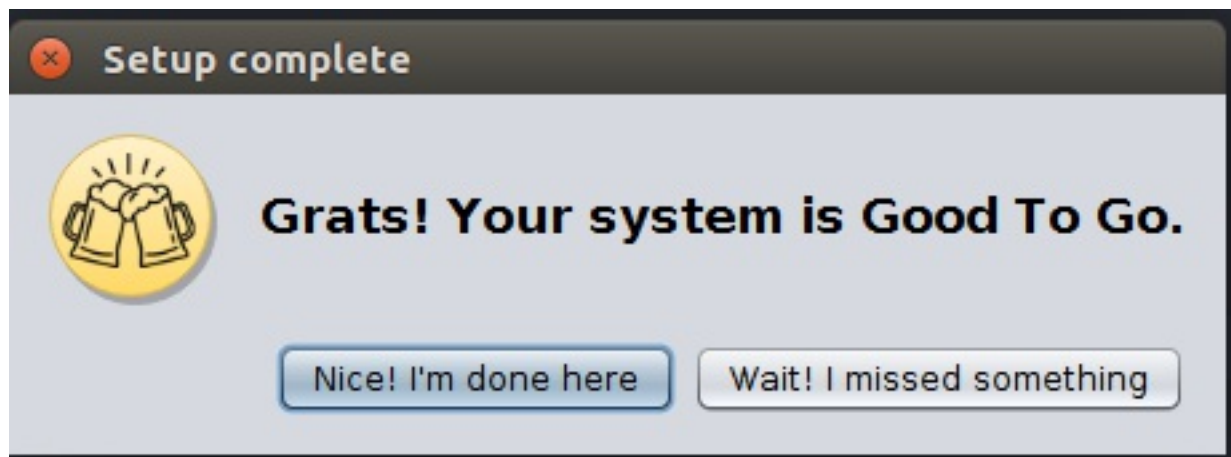
The progress window will show the overall progress of the installations and print the output from the scripts.

It may seem like nothing is happening but have patience. The installer is not multithreaded at the moment (because of dependencies) and some scripts may be downloading stuff in the background.

Some scripts may be calling external installers that will open windows and require user interactions.

*** Please finish all external installations before closing the GTG installer ***

Be aware that the scripts assume everything are installed at default locations.



When all the scripts have been run and the installer is finished you have the option to go back if you missed something or finish the installer. Be aware that the installer will cleanup the download folder when it quits.

If you don't want to run the GUI you can call each install script individually but they assume to be run from the `install_script` root folder and that argument 2 and 3 are the username and password respectively. I.e

```
cd install_script
./scripts/<script to call> <path to script> <username> <password>
```

Customization

How It Works

The GUI will scan all files in the script folder. The file name (file extensions are ignored) will be shown as an install component in the list. I.e “File Name.sh” will be shown as “File Name”. No special characters are allowed (“/,”). Dots are considered as the file extensions delimiter and everything after the dot is not considered part of the item name.

The GUI will scan each file for a description. A description is defined as:

```
if [ "$1" == "-h" ]; then
    echo "This is a description"
    exit 0
fi
```

The description must be the first thing in the script after the shebang (`#!/bin/bash`) else it will be

ignored.

At the moment it is not possible to define dependencies but it may be included at a later point.

All scripts is called through the `entry_point_script` located at `resources/worker_scripts/script_entry_point`.

At the moment the entry point is not used for anything but it may be used to differentiate between how scripts should be called (or anything else that should be handled before each install script is called)

Stuff that is too cumbersome to download or must be a specific version is placed in the `archive` folder.

The GUI will use the PNG image located in `resources/company/` folder as the logo and name for the installer.

It should have the dimensions 226x48 pixels and have the company name as the file name (the png extension will be ignored).

If there is no logo in the folder a default name and logo is used.

The GUI source code is included and can be edited but no documentation is currently available for that.

Things To Consider (quirks and novelties)

When a description is found by the GUI it will call the script with the `-h` argument as the first parameter. Please consider what you put in the description and remember to exit else the script will be run before the user has a chance to choose it.

Adding New Scripts

Each script should be a self-winding recipe to install one item or setup one thing.

To add a new script just make a new file in the script folder and give it a meaningful name. (File extensions are ignored by the GUI). Add the shebang on the first line:

```
#!/bin/bash
```

(As of the time of writing bash makes out to be the most reliable interpreter)

**** The script will be called with the following parameters ****

- `$1` ** ~> Absolute path to the script **
- `$2` ** ~> User name **
- `$3` ** ~> User password **

Add a description that will be shown as a tooltip in the GUI or be printed when the script is called from the command line with the `-h` parameter.

```
if [ "$1" == "-h" ]; then
    echo "This is a description"
    exit 0
fi
```

A number of helper functions are available in the `resources/worker_scripts/helpers` file. To use them add this to your script:

```
. resources/worker_scripts/helpers
```

It is recommended to use the helper functions as they have been tested to work with the GUI and will call the functions with the required arguments and place files at the correct locations.

If you need to download or make temporary files place them in the `resources/downloads/` folder as it will be cleaned every time the GUI quits. The path to the folder is defined in the `helpers` as

```
$DOWNLOAD_PATH
```

But remember files here will be deleted :-)

For `sudo` commands use the following command to pass the password to it

```
echo $3 | sudo -S
```

If you have something used by many different install scripts place it in the `resources/worker_scripts/` folder and source it from there (just like the `helpers` script). If you place it in the script folder it will show up as a component in the GUI.

If you need to `echo` into a file with `sudo` do this:

```
echo $3 | sudo -S bash -c 'echo "text to write with ""$VARIABLE_FROM_SCRIPT"" >> file/path'
```