

# Generic Download and Install Instructions

## Contents

<b>1</b>	<b>If you use Pacman</b>	<b>1</b>
1.1	Security . . . . .	2
<b>2</b>	<b>If you don't use Pacman</b>	<b>2</b>
2.1	Downloading the source tarball . . . . .	2
2.1.1	Checking the signature . . . . .	2
2.2	Installing from source . . . . .	3
<b>3</b>	<b>Getting the development history</b>	<b>3</b>

This documents how to get and install most of my software. Several methods are covered, depending on the tools you have available. In the rest of this document, I assume you want to download and install a package named `$pkgname`.

There are basically two methods, depending on whether you use Arch Linux<sup>1</sup> (more precisely, its package manager, Pacman<sup>2</sup>) or not.

## 1 If you use Pacman

Most of my publicly-available software have been packaged for `x86_64` and can be directly downloaded and installed from my public arch repository.

You first have to add my repository to Pacman. Append:

```
[svasey-public]
Server = http://public-repo.svasey.org/os/x86_64
```

to `/etc/pacman.conf`.

Then to download and install the latest version of `$pkgname`:

```
$ pacman -Sy $pkgname
```

That should do it ! Pay attention to any message that is printed at installation time.

---

<sup>1</sup><http://www.archlinux.org/>

<sup>2</sup><https://wiki.archlinux.org/index.php/Pacman>

## 1.1 Security

At present (version 3.3.3), Pacman does not implement package signing, which makes this method potentially vulnerable to Man-in-the-middle attacks<sup>3</sup>, and could also lead to problems if my server is compromised.

However, all my packages are signed with my GPG key<sup>4</sup>, so one workaround for you would be to manually download the package file and verify the signature, and then install it with Pacman. To do that, browse [http://public-repo.svasey.org/os/x86\\_64/](http://public-repo.svasey.org/os/x86_64/) and download your package and its `.sig` file. Then call:

```
$ gpg --verify $sigfile
```

Where `$sigfile` is the signature file. If the signature has been verified correctly, you can then install the package:

```
$ pacman -U $pkgfile
```

See the without Pacman method (section 2) for more on signature verification.

## 2 If you don't use Pacman

You should be aware that those instructions are not well-tested. Do not hesitate to send me feedback<sup>5</sup> if something goes wrong.

### 2.1 Downloading the source tarball

Download the source tarball:

```
$ wget http://public-repo.svasey.org/src/${pkgname}.tar.gz
```

#### 2.1.1 Checking the signature

I then advise you to check its digital signature<sup>6</sup>. If you do not care, feel free to skip that section.

First, import my GPG key<sup>7</sup>, and make sure you trust it. There is a lot of documentation on how to do this on the web.

Then, download the signature:

```
$ wget http://public-repo.svasey.org/src/${pkgname}.tar.gz.sig
```

And verify it:

```
$ gpg --verify ${pkgname}.tar.gz.sig
```

---

<sup>3</sup>[http://en.wikipedia.org/wiki/Man-in-the-middle\\_attack](http://en.wikipedia.org/wiki/Man-in-the-middle_attack)

<sup>4</sup><http://certs.svasey.org/gpg-svasey-pubkey.asc>

<sup>5</sup>[http://svasey.org/about\\_en.html#contact-information](http://svasey.org/about_en.html#contact-information)

<sup>6</sup>[http://en.wikipedia.org/wiki/Digital\\_signature](http://en.wikipedia.org/wiki/Digital_signature)

<sup>7</sup><http://certs.svasey.org/gpg-svasey-pubkey.asc>

## 2.2 Installing from source

Extract the tarball and cd to the resulting directory:

```
$ tar -xf ${pkgname}.tar.gz
$ cd $pkgname
```

After doing an `ls`, you should see several documentation files, like e.g `README`, `INSTALL` and `COPYING`. Read the `INSTALL` file, install the dependencies, and see if it contains something unusual or if it just points to those instructions. Most of the time it will just point here. If it doesn't, follow the instructions from `INSTALL` instead of those.

There should also be an `install` file. To install the package, run (as root):

```
$ ./install $DESTDIR $PREFIX
```

Where `$DESTDIR` and `$PREFIX` can be omitted, especially if you do not know what they are (good default values will be chosen). `$DESTDIR` is the file system root where the package should be installed and defaults to `/`. `$PREFIX` is the prefix where most of the program files will be installed. It defaults to `/usr/local`.

If there is a file `install.post`, you can have a look at `INSTALL` and read what it tells you to do with that file, then run:

```
$ ./install.post $DESTDIR $PREFIX
```

## 3 Getting the development history

Most of my projects have a `git`<sup>8</sup> repository. If you are a developer you can also get the source from there. To clone it, run:

```
$ git clone git://git.svasey.org/${pkgname}.git
```

---

<sup>8</sup><http://git-scm.com/>