

[skimpydog-gentoo](#)

Main

Posted by: skimpydog

Posted on : 2010/12/26 12:37:42

So my new project as of lately is skimpydog-gentoo.

This is Gentoo, 32bit - built how I like it.

When asked "what is the purpose of skimpydog-Gentoo?": it is to promote safe computing practices, and to introduce others to Linux - A free Operating System. I did NOT write this, I put it together, and worked out bugs - I created a few scripts to assist with ease of use.

You may download, and watch the installation video as a demonstration of the installation:

[AVI](#) ~215MB

[Flash](#) ~72MB

[iPhone/iPod Touch/Android/PSP](#) ~136MB

[iPhone 4](#) ~177MB

When asked about Linux (by a stranger), I normally have to follow up with: "There's nothing wrong about not running Windows, I use Microsoft Windows as well".

An example use of Linux: What does the iPhone, iPod, iPad, and Android run? Your computer can run it much better.

I am updating the tar.bz2 (compressed files) as often as I can, as well as the installation instructions - but not too often as my goal is stability on an Intel ChipSet without using proprietary drivers/code; and attempting to keep things in layman's terms.

- However complex computers may be, the more familiar You become with the terminology the better off you will be. Computers are not going away anytime soon.

Built for an Intel based architecture, this is fully bloated with:

```
#gentoo-sources-2.6.28-r6
#xorg-server-1.7.7-r1
#mesa-7.7.1
#Gnome-2.30.2 built with hal support
#Compiz-Fusion-0.8.6
#cairo-dock-1.5.4.2
#enlightenment-1.0.2
#supertux
#superkart
#extreme-tuxracer
#python-2.6.5
#mythtv-0.23.1_p25496
```

#lirc
#gphoto2
#dialup support

Works right out of the box with dualhead nVidia older cards, like GeForceFX 5600. "eselect set opengl nvidia". You may use "eselect opengl list" to see what is available:

```
X11 # eselect opengl list Available OpenGL implementations: [1] nvidia * [2] xorg-x11
```

With Intel Based Video cards i810, i915 (I assume newer chipsets as well; just not tested), use "eselect set opengl xorg-x11", and modify /etc/Xorg/xorg.conf by using one of the examples for Hardware Video acceleration:

```
/etc/X11 # ls xorg.conf* xorg.conf      xorg.conf.example      xorg.conf.org xorg.conf.DualDead  
xorg.conf.nvidia.TwinView xorg.conf.solo.intel
```

Click "Read More" below...

If you intend to attempt to install this, first you must be a member in order to download. This is setup this way to prevent stupid s#!t thanks to bots.

If "I", feel you are a bot, you will be denied access. Use a valid email.

If you need quick access, you should contact me. Other than this I have to verify the email. My contact information is on this site.

This compilation of software does include license restrictions, which all of the licensing is included, and this is 100% free of cost. I have agreed to the respective owners licensing agreements, and therefore you must as well. I have no licensing over the software's included. I have only added to what others have built in effort to have a stable 32bit Operating System.

These licenses are for Java, nVidia Drivers, Flash, and maybe a few other things. I'm not charging for this, and I'm not trying to distribute code illegally. I'm passing along the same version I downloaded from them, under the same terms of their Agreements.

Here's the shakedown:

[skimpydog-gentoo](#)

So far this has been tested on 6 different machines, and all with Intel based Video, and other.

I have run this with a Windows VirtualBox which gets turned on and off all of the time running for over 60 days. It has been quite stable so far.

Noted bugs:

I have not had any issues excluding how hardware is added to the system.

Since there are different ID's with every machines NIC (MAC Address), I assume it continues to count upward - such as eth0, eth1, eth2, eth3, eth4, I have already set the NIC's to zero, however just beware of it - keep trying the interfaces until you get the right one, I'm working on having all of these set proper.

This can be the same with CDROMS, and other hardware.

So be sure to look in /dev for how the kernel is loading devices to verify how the device is being bound, as well as "dmesg | less" to see what has been found. Adding several hardware types should be as simple as loading a kernel module.