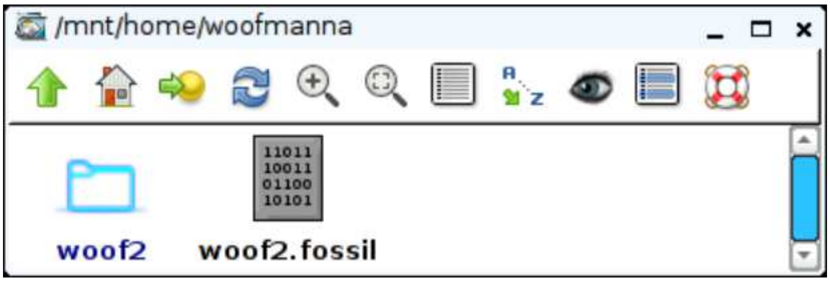
I am putting together this tutorial on woof2. I would also suggest that you use the puppy for which you want to build with woof2.

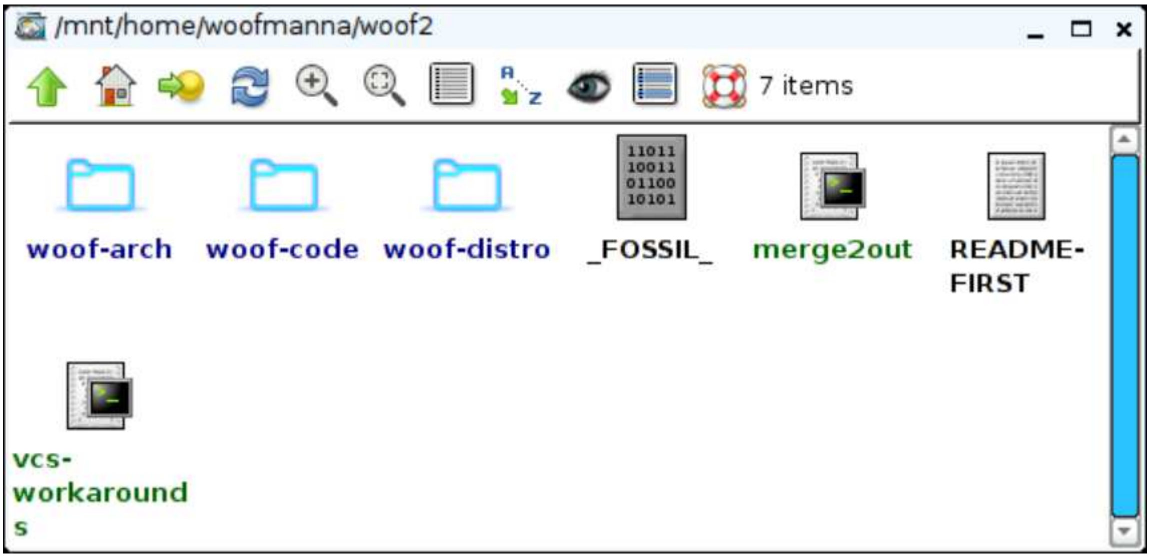
I strongly suggest that if you have never used woof2 or woof befor,e to build the original puppy first to get a feel on what everything does.

This does require firefox webbrowser.

1) first go here http://bkhome.org/fossil/woof2.cgi/wiki?name=Getting+Started+with+Woof and read about getting started with woof2.

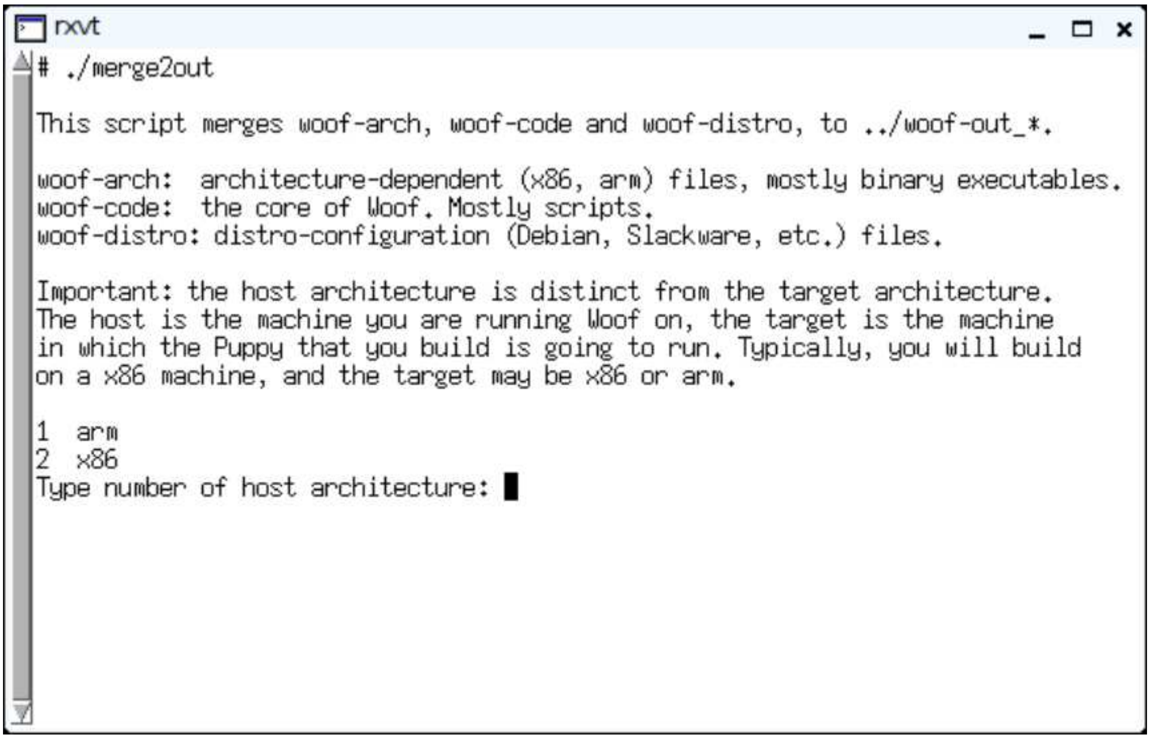
2) After you have created your folder and before you did merge2out this is what you should have. Make sure to read all the readme files before you start anything. #2png



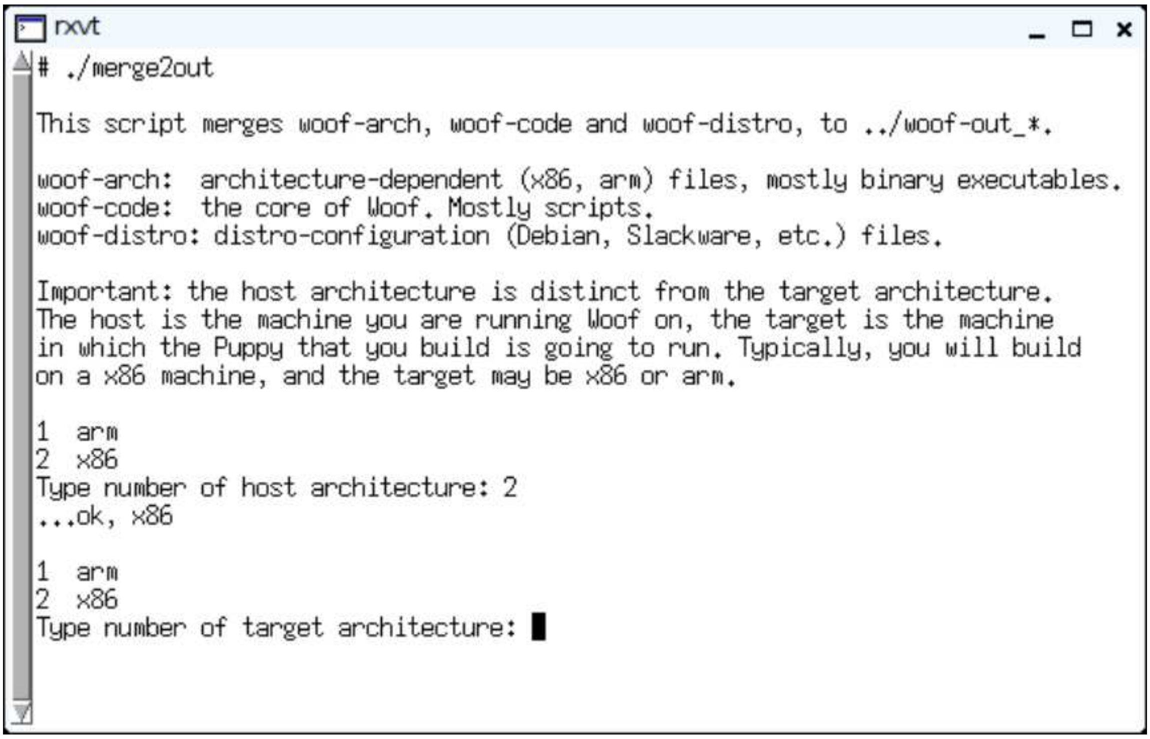


The next 8 pictures shows what you can choose for creating your Puppy, whether it be arm or x86.

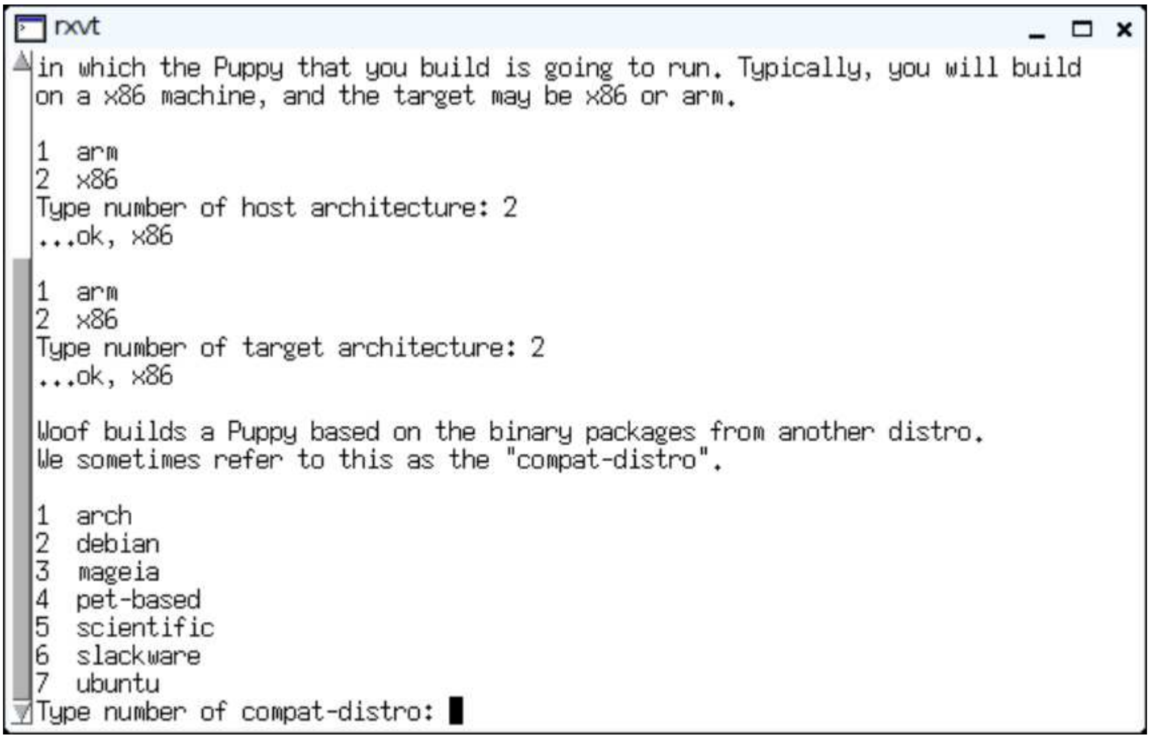
It will also ask about what linux distro you want to build from.



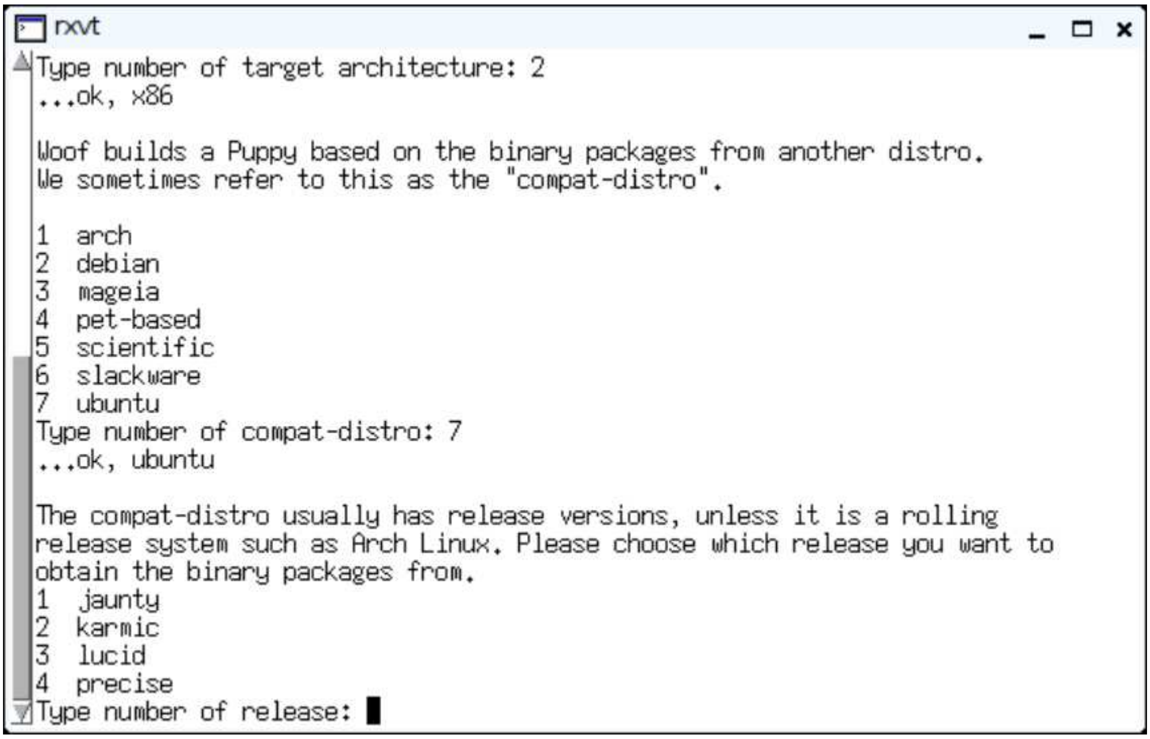
```
# ./merge2out

This script merges woof-arch, woof-code and woof-distro, to ../woof-out_*.

woof-arch:  architecture-dependent (x86, arm) files, mostly binary executables.
woof-code:  the core of Woof. Mostly scripts.
woof-distro: distro-configuration (Debian, Slackware, etc.) files.

Important: the host architecture is distinct from the target architecture.
The host is the machine you are running Woof on, the target is the machine
in which the Puppy that you build is going to run. Typically, you will build
on a x86 machine, and the target may be x86 or arm.

1  arm
2  x86
Type number of host architecture: 
```



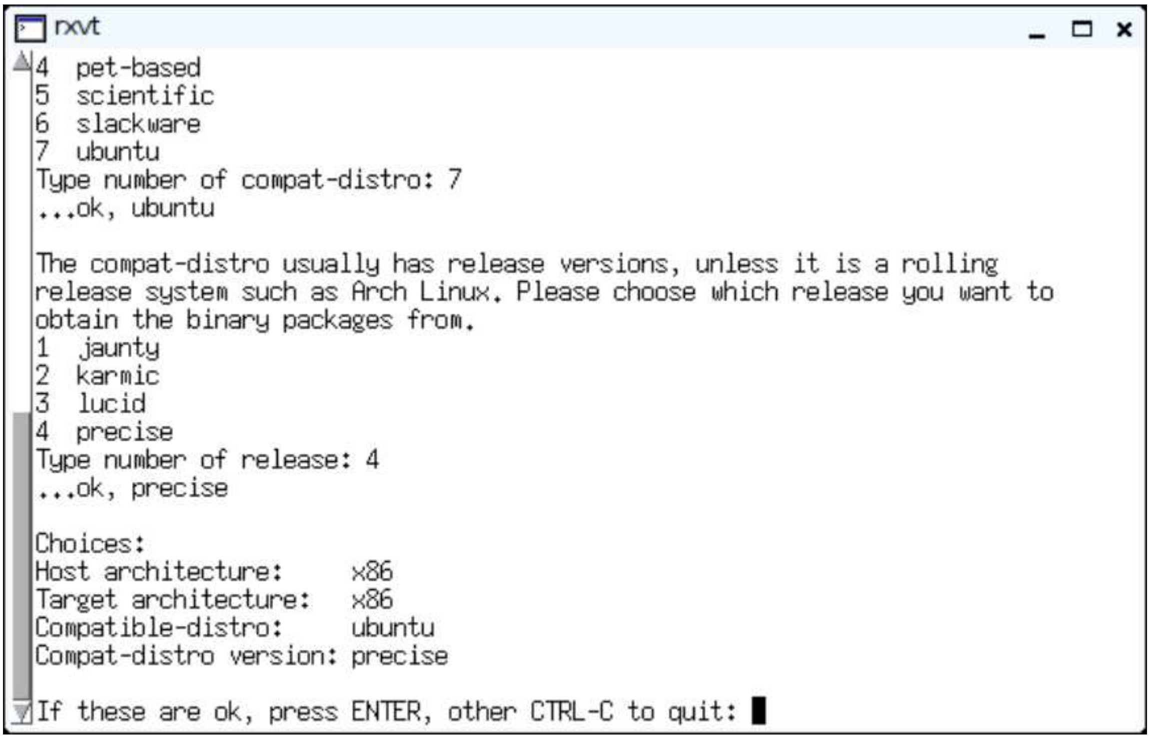
```
# ./merge2out

This script merges woof-arch, woof-code and woof-distro, to ../woof-out_*.

woof-arch:  architecture-dependent (x86, arm) files, mostly binary executables.
woof-code:  the core of Woof. Mostly scripts.
woof-distro: distro-configuration (Debian, Slackware, etc.) files.

Important: the host architecture is distinct from the target architecture.
The host is the machine you are running Woof on, the target is the machine
in which the Puppy that you build is going to run. Typically, you will build
on a x86 machine, and the target may be x86 or arm.

1  arm
2  x86
Type number of host architecture: 2
...ok, x86

1  arm
2  x86
Type number of target architecture: 
```



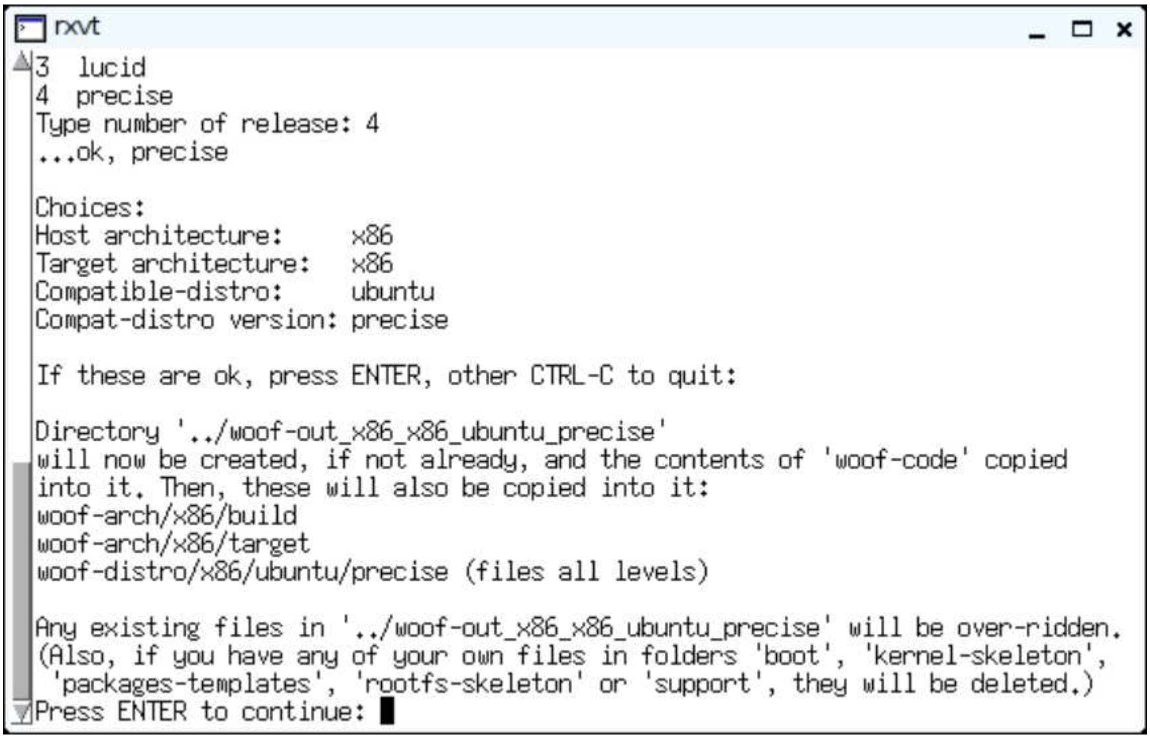
```
in which the Puppy that you build is going to run. Typically, you will build
on a x86 machine, and the target may be x86 or arm.

1  arm
2  x86
Type number of host architecture: 2
...ok, x86

1  arm
2  x86
Type number of target architecture: 2
...ok, x86

Woof builds a Puppy based on the binary packages from another distro.
We sometimes refer to this as the "compat-distro".

1  arch
2  debian
3  mageia
4  pet-based
5  scientific
6  slackware
7  ubuntu
Type number of compat-distro:
```



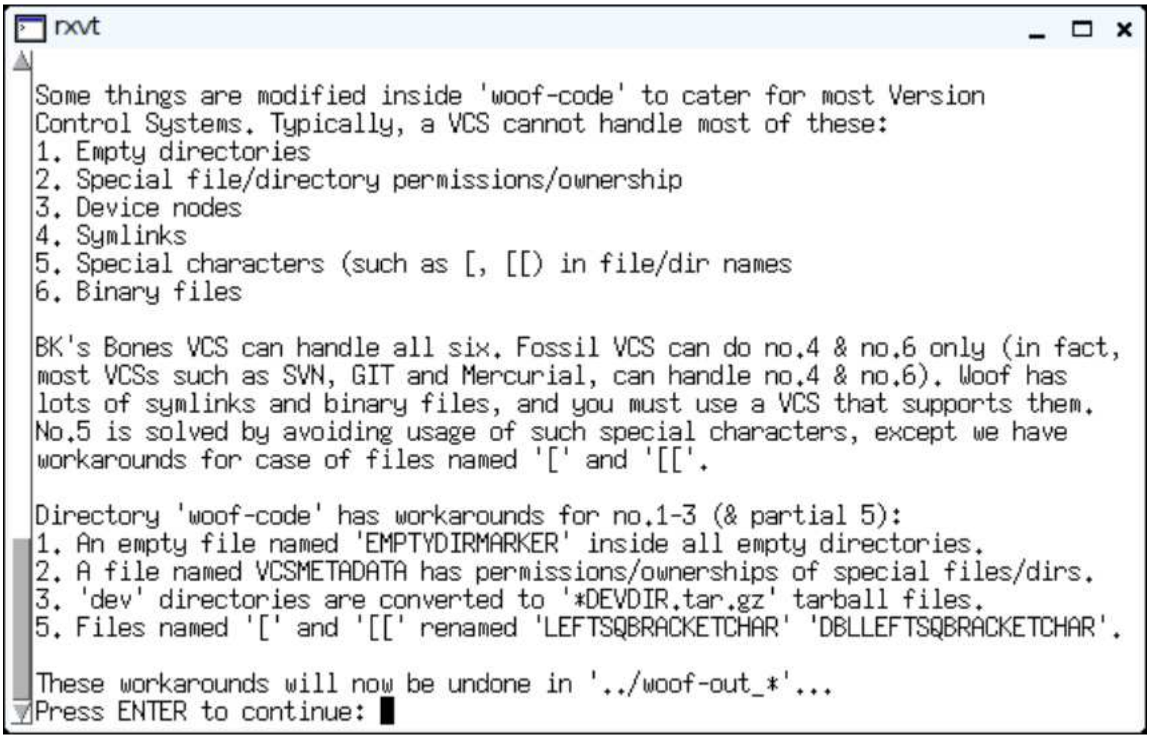
```
Type number of target architecture: 2
...ok, x86

Woof builds a Puppy based on the binary packages from another distro.
We sometimes refer to this as the "compat-distro".

1  arch
2  debian
3  mageia
4  pet-based
5  scientific
6  slackware
7  ubuntu
Type number of compat-distro: 7
...ok, ubuntu

The compat-distro usually has release versions, unless it is a rolling
release system such as Arch Linux. Please choose which release you want to
obtain the binary packages from.
1  jaunty
2  karmic
3  lucid
4  precise
Type number of release:
```



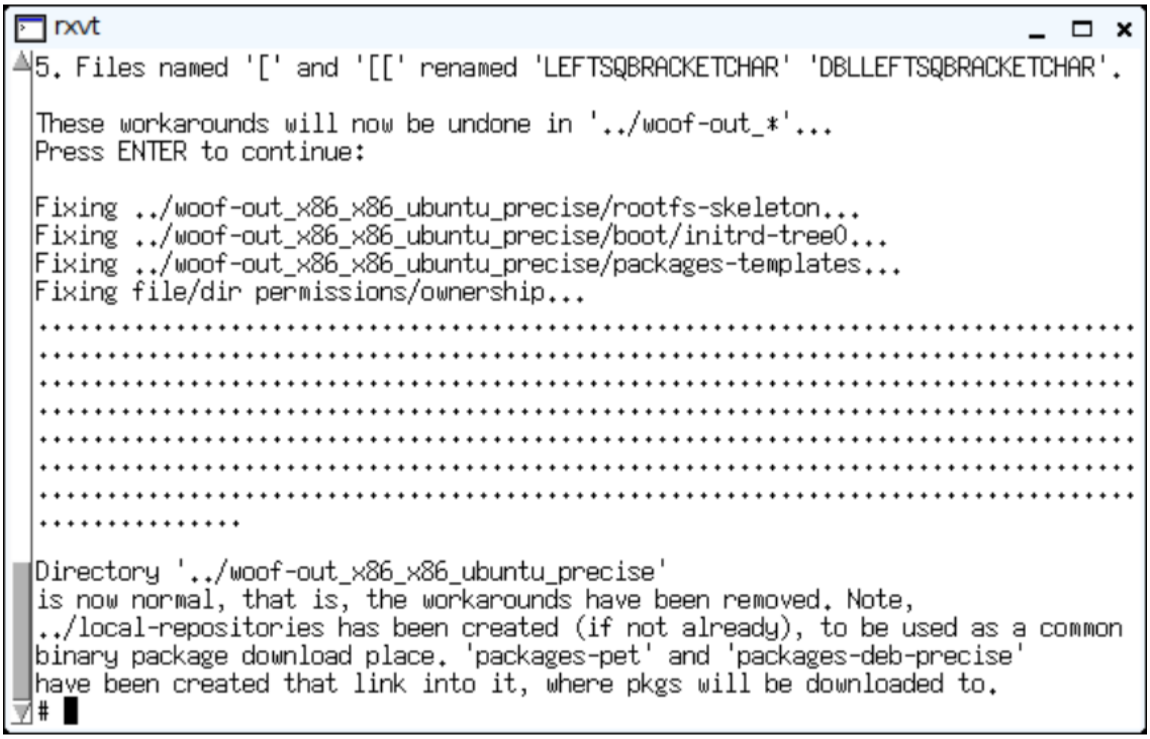
```
4  pet-based
5  scientific
6  slackware
7  ubuntu
Type number of compat-distro: 7
...ok, ubuntu

The compat-distro usually has release versions, unless it is a rolling
release system such as Arch Linux. Please choose which release you want to
obtain the binary packages from.
1  jaunty
2  karmic
3  lucid
4  precise
Type number of release: 4
...ok, precise

Choices:
Host architecture:     x86
Target architecture:   x86
Compatible-distro:     ubuntu
Compat-distro version: precise

If these are ok, press ENTER, other CTRL-C to quit:
```



```
3  lucid
4  precise
Type number of release: 4
...ok, precise

Choices:
Host architecture:     x86
Target architecture:   x86
Compatible-distro:     ubuntu
Compat-distro version: precise

If these are ok, press ENTER, other CTRL-C to quit:

Directory '../woof-out_x86_x86_ubuntu_precise'
will now be created, if not already, and the contents of 'woof-code' copied
into it. Then, these will also be copied into it:
woof-arch/x86/build
woof-arch/x86/target
woof-distro/x86/ubuntu/precise (files all levels)

Any existing files in '../woof-out_x86_x86_ubuntu_precise' will be over-ridden.
(Also, if you have any of your own files in folders 'boot', 'kernel-skeleton',
 'packages-templates', 'rootfs-skeleton' or 'support', they will be deleted.)
Press ENTER to continue:
```



```
Some things are modified inside 'woof-code' to cater for most Version
Control Systems. Typically, a VCS cannot handle most of these:
1. Empty directories
2. Special file/directory permissions/ownership
3. Device nodes
4. Symlinks
5. Special characters (such as [, [[) in file/dir names
6. Binary files

BK's Bones VCS can handle all six. Fossil VCS can do no.4 & no.6 only (in fact,
most VCSs such as SVN, GIT and Mercurial, can handle no.4 & no.6). Woof has
lots of symlinks and binary files, and you must use a VCS that supports them.
No.5 is solved by avoiding usage of such special characters, except we have
workarounds for case of files named '[' and '[['.

Directory 'woof-code' has workarounds for no.1-3 (& partial 5):
1. An empty file named 'EMPTYDIRMARKER' inside all empty directories.
2. A file named VCSMETADATA has permissions/ownerships of special files/dirs.
3. 'dev' directories are converted to '*DEVDIR.tar.gz' tarball files.
5. Files named '[' and '[[' renamed 'LEFTSQBRACKETCHAR' 'DBLLEFTSQBRACKETCHAR'.

These workarounds will now be undone in '../woof-out_*'...
Press ENTER to continue: █
```



```
5. Files named '[' and '[[' renamed 'LEFTSQBRACKETCHAR' 'DBLLEFTSQBRACKETCHAR'.

These workarounds will now be undone in '../woof-out_*'...
Press ENTER to continue:

Fixing ../woof-out_x86_x86_ubuntu_precise/rootfs-skeleton...
Fixing ../woof-out_x86_x86_ubuntu_precise/boot/initrd-tree0...
Fixing ../woof-out_x86_x86_ubuntu_precise/packages-templates...
Fixing file/dir permissions/ownership...
.................................................................................
.................................................................................
.................................................................................
.................................................................................
.................................................................................
.................................................................................
.................................................................................
...............

Directory '../woof-out_x86_x86_ubuntu_precise'
is now normal, that is, the workarounds have been removed. Note,
../local-repositories has been created (if not already), to be used as a common
binary package download place. 'packages-pet' and 'packages-deb-precise'
have been created that link into it, where pkgs will be downloaded to.
# █
```

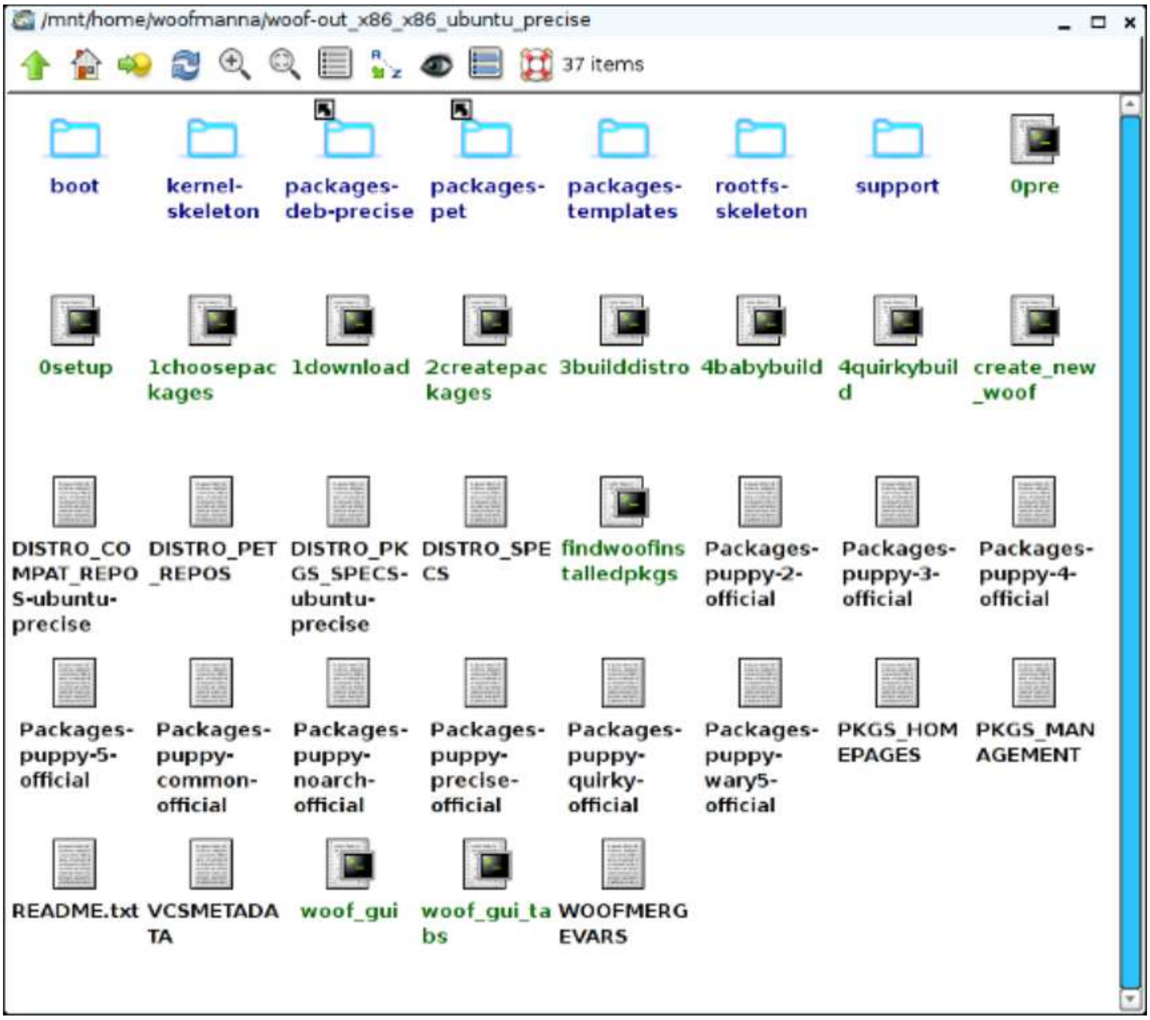
Now we will take a look at the gui and the tabs that are there and what they are for. I will try and give ideas on what you can do to change things once you get comfortable with woof2.

Before we look at the gui let us look at the folders and files that will be there. Again you want to go into all the folders and read any of the Readme files first before you do anything.

They will give you information that will help with your build.

You can go into the folder rootfs-skeleton and add or even remove some things that you don't want in your build that is in by default. ONLY do this if you know what you are doing.

If you want to make a splash-screen image than go to the boot folder and find boot-dialog and there you will get the information to do so.


/mnt/home/woofmanna/woof-out_x86_x86_ubuntu_precise
37 items
boot
kernel-skeleton
packages-deb-precise
packages-pet
packages-templates
rootfs-skeleton
support
0pre
0setup
1choosepackages
1download
2createpackages
3builddistro
4babybuild
4quirkybuild
create_new_woof
DISTRO_COMPAT_REPOS-ubuntu-precise
DISTRO_PET_REPOS
DISTRO_PKGS_SPECS-ubuntu-precise
DISTRO_SPECS
findwoofinstalledpkgs
Packages-puppy-2-official
Packages-puppy-3-official
Packages-puppy-4-official
Packages-puppy-5-official
Packages-puppy-common-official
Packages-puppy-noarch-official
Packages-puppy-precise-official
Packages-puppy-quirky-official
Packages-puppy-wary5-official
PKGS_HOMEPAGES
PKGS_MANAGEMENT
README.txt
VCSMETADATA
woof_gui
woof_gui_tabs
WOOFMERGEVARS

Now we will take a look at the gui. Here we will take a look at the first four tabes. These pictures are self explaining.

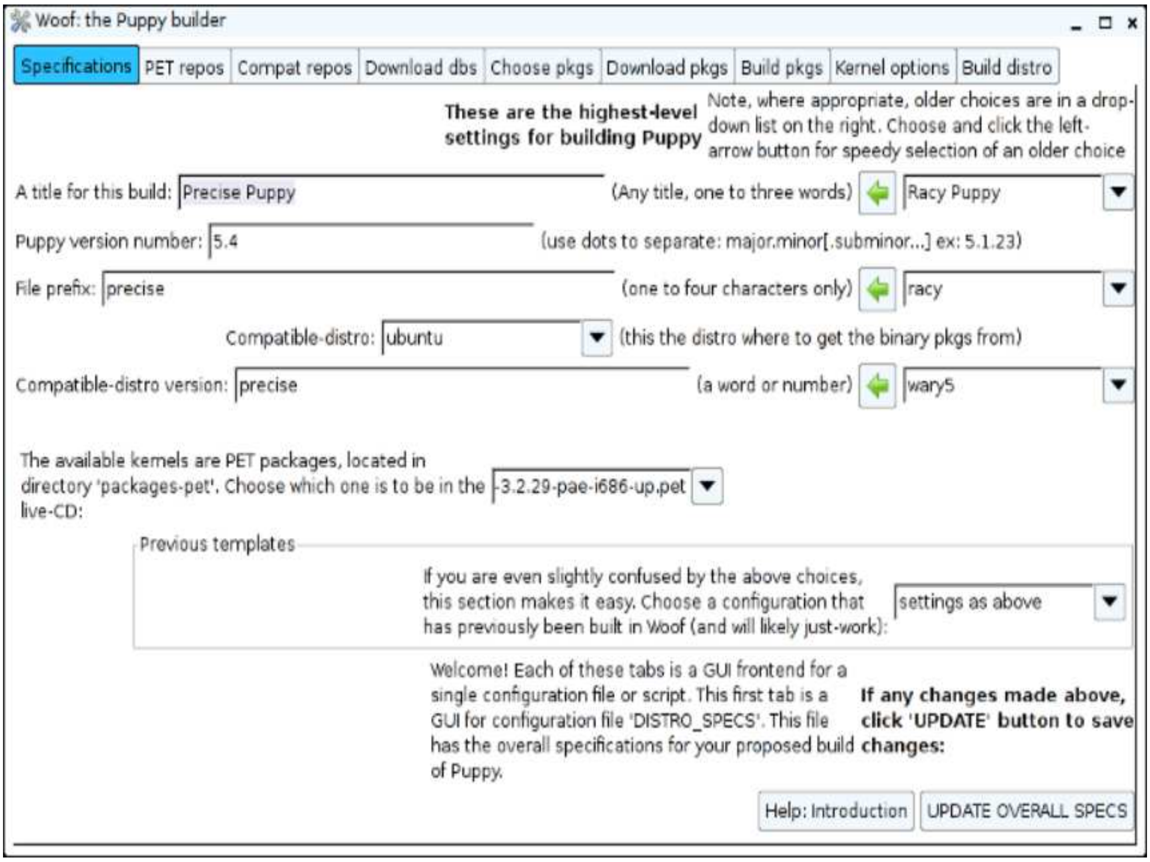
1) The Specification tab here you can name what your puppy will be called and the number of your build. Also you can choose your kernel as well.

2) The Pet Repos tab you don't have to update this when you first download woof2 for it will

already be up to date.

3) The Compact Repo tab the same applies here as well.

4) The Download dbs tab here you will need to update the compact distro database and only these get updated.



These are the highest-level settings for building Puppy

Note, where appropriate, older choices are in a drop-down list on the right. Choose and click the left-arrow button for speedy selection of an older choice

A title for this build: Precise Puppy (Any title, one to three words) Racy Puppy

Puppy version number: 5.4 (use dots to separate: major.minor[.subminor...] ex: 5.1.23)

File prefix: precise (one to four characters only) racy

Compatible-distro: ubuntu (this the distro where to get the binary pkgs from)

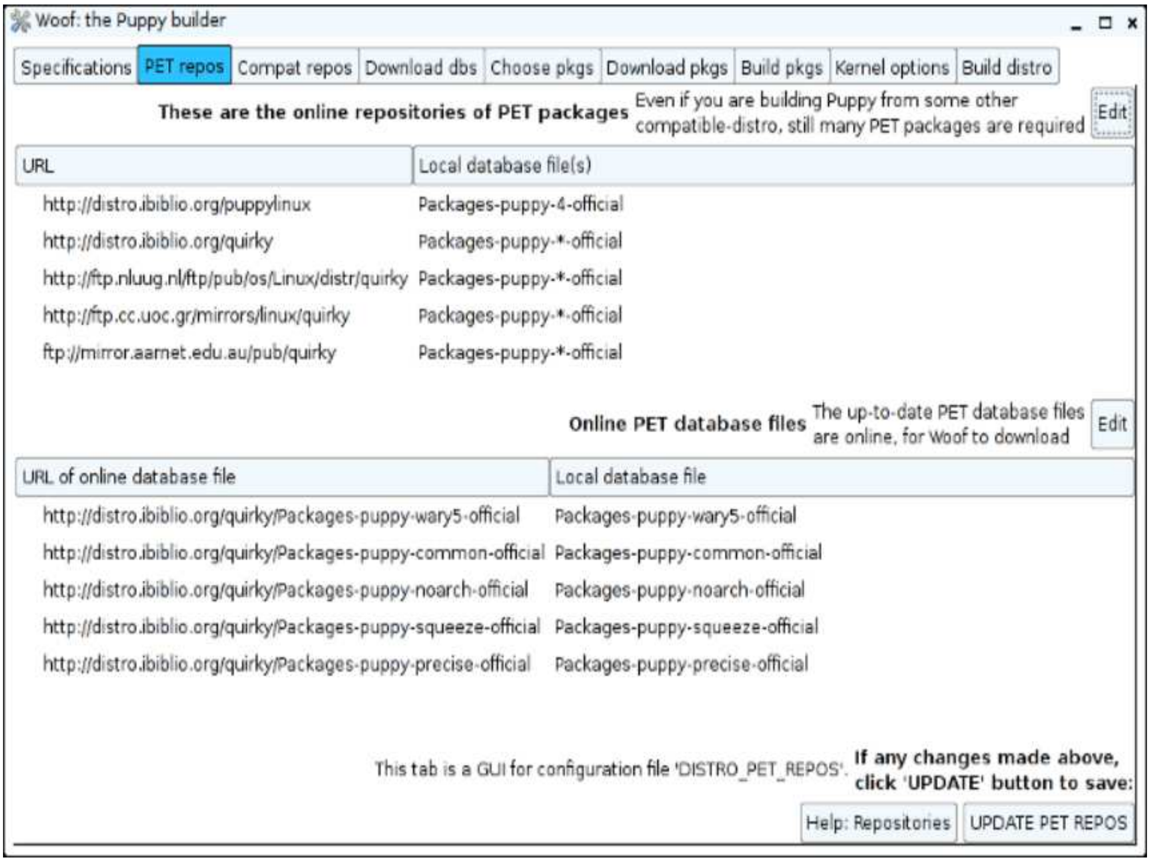
Compatible-distro version: precise (a word or number) wary5

The available kernels are PET packages, located in directory 'packages-pet'. Choose which one is to be in the live-CD: -3.2.29-pae-i686-up.pet

If you are even slightly confused by the above choices, this section makes it easy. Choose a configuration that has previously been built in Woof (and will likely just-work): settings as above

Welcome! Each of these tabs is a GUI frontend for a single configuration file or script. This first tab is a GUI for configuration file 'DISTRO_SPECS'. This file has the overall specifications for your proposed build of Puppy.

If any changes made above, click 'UPDATE' button to save changes:



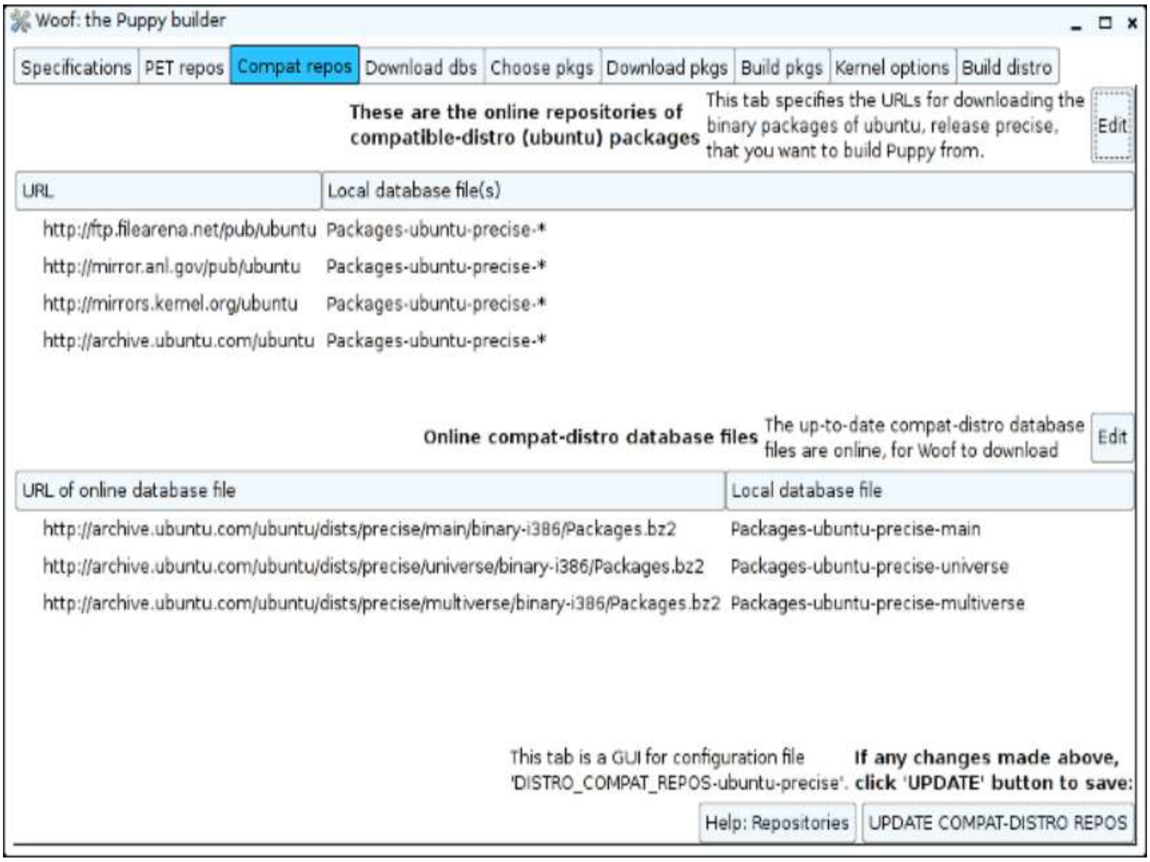
These are the online repositories of PET packages Even if you are building Puppy from some other compatible-distro, still many PET packages are required

URL	Local database file(s)
http://distro.ibiblio.org/puppylinux	Packages-puppy-4-official
http://distro.ibiblio.org/quirky	Packages-puppy-*-official
http://ftp.nluug.nl/ftp/pub/os/Linux/distr/quirky	Packages-puppy-*-official
http://ftp.cc.uoc.gr/mirrors/linux/quirky	Packages-puppy-*-official
ftp://mirror.aarnet.edu.au/pub/quirky	Packages-puppy-*-official

Online PET database files The up-to-date PET database files are online, for Woof to download

URL of online database file	Local database file
http://distro.ibiblio.org/quirky/Packages-puppy-wary5-official	Packages-puppy-wary5-official
http://distro.ibiblio.org/quirky/Packages-puppy-common-official	Packages-puppy-common-official
http://distro.ibiblio.org/quirky/Packages-puppy-noarch-official	Packages-puppy-noarch-official
http://distro.ibiblio.org/quirky/Packages-puppy-squeeze-official	Packages-puppy-squeeze-official
http://distro.ibiblio.org/quirky/Packages-puppy-precise-official	Packages-puppy-precise-official

This tab is a GUI for configuration file 'DISTRO_PET_REPOS'. **If any changes made above, click 'UPDATE' button to save:**



These are the online repositories of compatible-distro (ubuntu) packages

This tab specifies the URLs for downloading the binary packages of ubuntu, release precise, that you want to build Puppy from.

URL	Local database file(s)
http://ftp.filearena.net/pub/ubuntu	Packages-ubuntu-precise-*
http://mirror.anl.gov/pub/ubuntu	Packages-ubuntu-precise-*
http://mirrors.kernel.org/ubuntu	Packages-ubuntu-precise-*
http://archive.ubuntu.com/ubuntu	Packages-ubuntu-precise-*

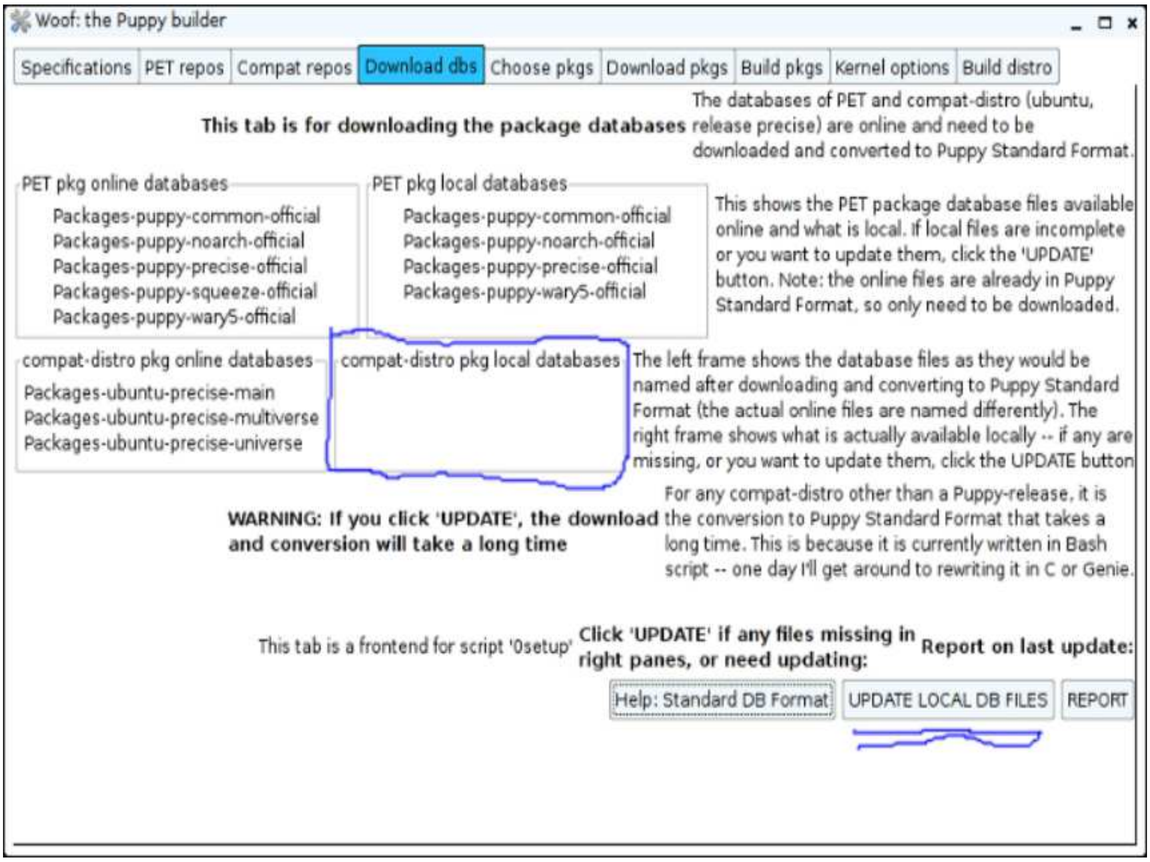
Online compat-distro database files

The up-to-date compat-distro database files are online, for Woof to download

URL of online database file	Local database file
http://archive.ubuntu.com/ubuntu/dists/precise/main/binary-i386/Packages.bz2	Packages-ubuntu-precise-main
http://archive.ubuntu.com/ubuntu/dists/precise/universe/binary-i386/Packages.bz2	Packages-ubuntu-precise-universe
http://archive.ubuntu.com/ubuntu/dists/precise/multiverse/binary-i386/Packages.bz2	Packages-ubuntu-precise-multiverse

This tab is a GUI for configuration file 'DISTRO_COMPAT_REPOS-ubuntu-precise'.

If any changes made above, click 'UPDATE' button to save:



This tab is for downloading the package databases The databases of PET and compat-distro (ubuntu, release precise) are online and need to be downloaded and converted to Puppy Standard Format.

Packages-puppy-common-official
Packages-puppy-noarch-official
Packages-puppy-precise-official
Packages-puppy-squeeze-official
Packages-puppy-wary5-official

Packages-puppy-common-official
Packages-puppy-noarch-official
Packages-puppy-precise-official
Packages-puppy-wary5-official

This shows the PET package database files available online and what is local. If local files are incomplete or you want to update them, click the 'UPDATE' button. Note: the online files are already in Puppy Standard Format, so only need to be downloaded.

Packages-ubuntu-precise-main
Packages-ubuntu-precise-multiverse
Packages-ubuntu-precise-universe

The left frame shows the database files as they would be named after downloading and converting to Puppy Standard Format (the actual online files are named differently). The right frame shows what is actually available locally -- if any are missing, or you want to update them, click the UPDATE button

WARNING: If you click 'UPDATE', the download and conversion will take a long time For any compat-distro other than a Puppy-release, it is the conversion to Puppy Standard Format that takes a long time. This is because it is currently written in Bash script -- one day I'll get around to rewriting it in C or Genie.

This tab is a frontend for script '0setup' **Click 'UPDATE' if any files missing in right panes, or need updating:**

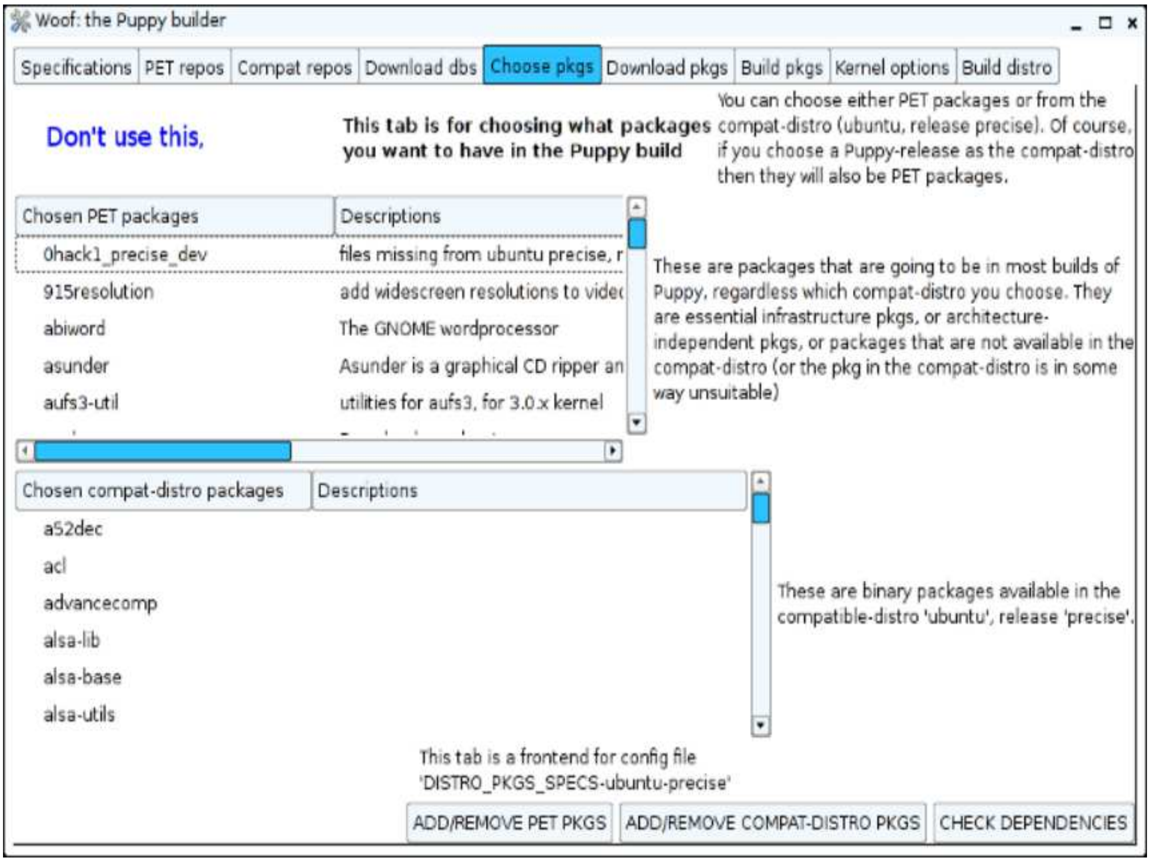
Now on to the next three tabs

1) Choose pkgs tab this one I don't touch the reason is this unless you know what pkgs need to be in puppy, when you add pkgs that will be the only ones that will be in the build.

I will get into how to add pkgs after we go through the gui.

2) Download pkgs tab here there really is nothing that needs to be said, except that if a pkge doesn't get downloaded you will have to get it from the web. Either form ibiblio or archive site of the compact distro that you are using for your build.

3) Build pkgs tab here it takes all the pkgs and unpacks them and makes them ready for the build.



Don't use this,

This tab is for choosing what packages you want to have in the Puppy build

You can choose either PET packages or from the compat-distro (ubuntu, release precise). Of course, if you choose a Puppy-release as the compat-distro then they will also be PET packages.

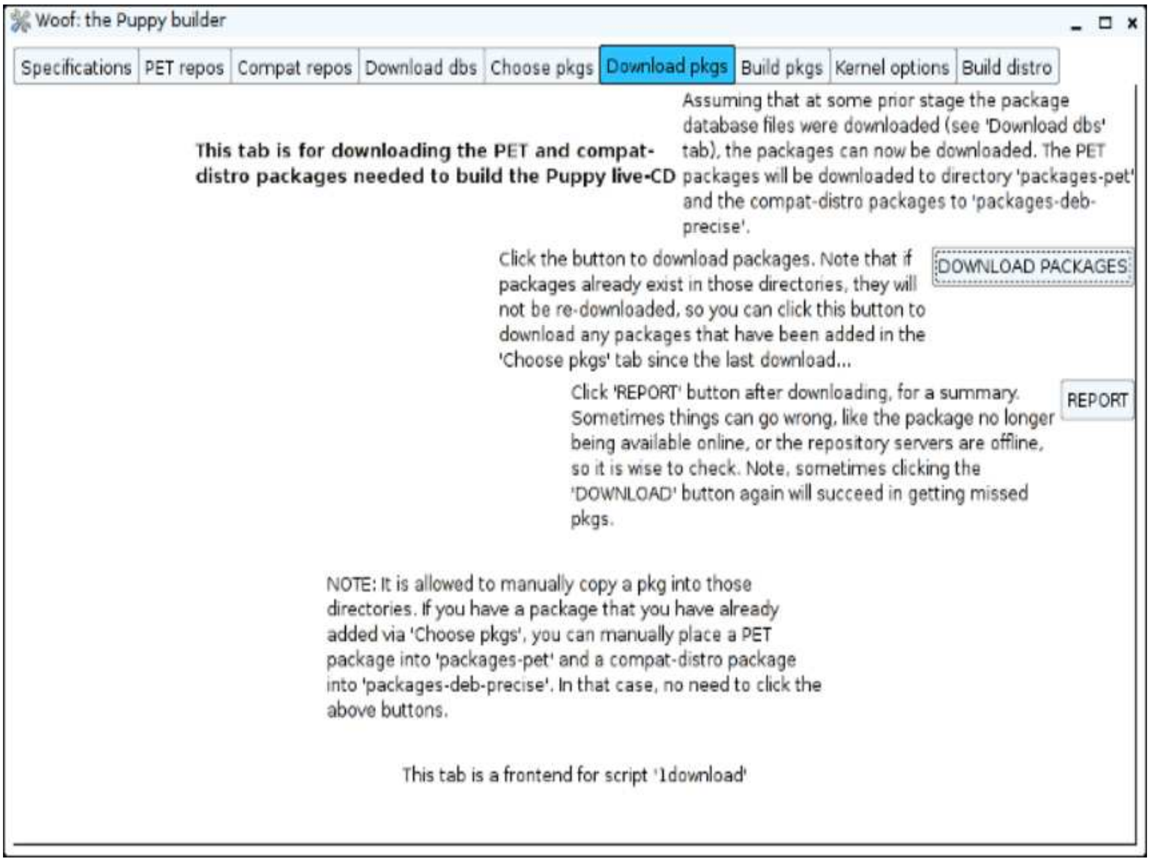
Chosen PET packages	Descriptions
0hack1_precise_dev	files missing from ubuntu precise, r
915resolution	add widescreen resolutions to videc
abiword	The GNOME wordprocessor
asunder	Asunder is a graphical CD ripper an
aufs3-util	utilities for aufs3, for 3.0.x kernel

These are packages that are going to be in most builds of Puppy, regardless which compat-distro you choose. They are essential infrastructure pkgs, or architecture-independent pkgs, or packages that are not available in the compat-distro (or the pkg in the compat-distro is in some way unsuitable)

Chosen compat-distro packages	Descriptions
a52dec	
acl	
advancecomp	
alsa-lib	
alsa-base	
alsa-utils	

These are binary packages available in the compatible-distro 'ubuntu', release 'precise'.

This tab is a frontend for config file 'DISTRO_PKGS_SPECS-ubuntu-precise'



This tab is for downloading the PET and compat-distro packages needed to build the Puppy live-CD

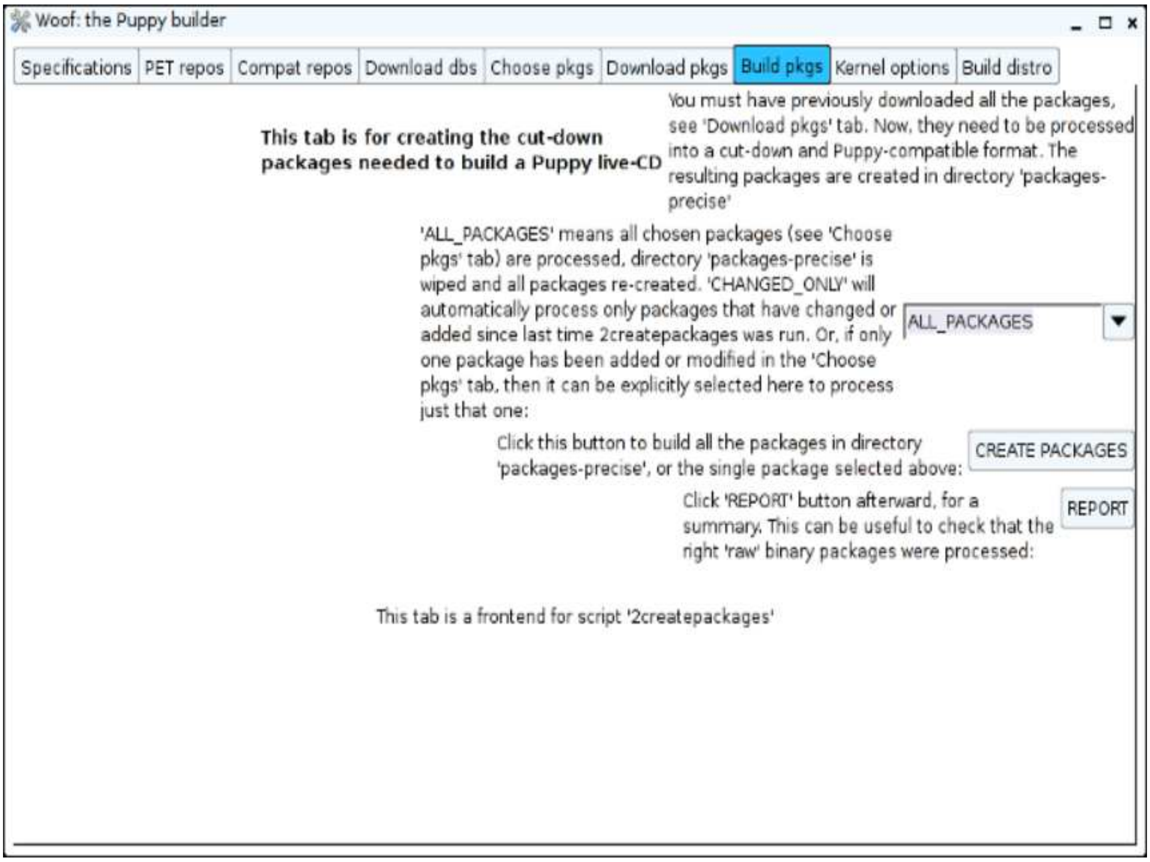
Assuming that at some prior stage the package database files were downloaded (see 'Download dbs' tab), the packages can now be downloaded. The PET packages will be downloaded to directory 'packages-pet' and the compat-distro packages to 'packages-deb-precise'.

Click the button to download packages. Note that if packages already exist in those directories, they will not be re-downloaded, so you can click this button to download any packages that have been added in the 'Choose pkgs' tab since the last download...

Click 'REPORT' button after downloading, for a summary. Sometimes things can go wrong, like the package no longer being available online, or the repository servers are offline, so it is wise to check. Note, sometimes clicking the 'DOWNLOAD' button again will succeed in getting missed pkgs.

NOTE: It is allowed to manually copy a pkg into those directories. If you have a package that you have already added via 'Choose pkgs', you can manually place a PET package into 'packages-pet' and a compat-distro package into 'packages-deb-precise'. In that case, no need to click the above buttons.

This tab is a frontend for script '1download'



This tab is for creating the cut-down packages needed to build a Puppy live-CD

You must have previously downloaded all the packages, see 'Download pkgs' tab. Now, they need to be processed into a cut-down and Puppy-compatible format. The resulting packages are created in directory 'packages-precise'

'ALL_PACKAGES' means all chosen packages (see 'Choose pkgs' tab) are processed, directory 'packages-precise' is wiped and all packages re-created. 'CHANGED_ONLY' will automatically process only packages that have changed or added since last time 2createpackages was run. Or, if only one package has been added or modified in the 'Choose pkgs' tab, then it can be explicitly selected here to process just that one:

Click this button to build all the packages in directory 'packages-precise', or the single package selected above:

Click 'REPORT' button afterward, for a summary. This can be useful to check that the right 'raw' binary packages were processed:

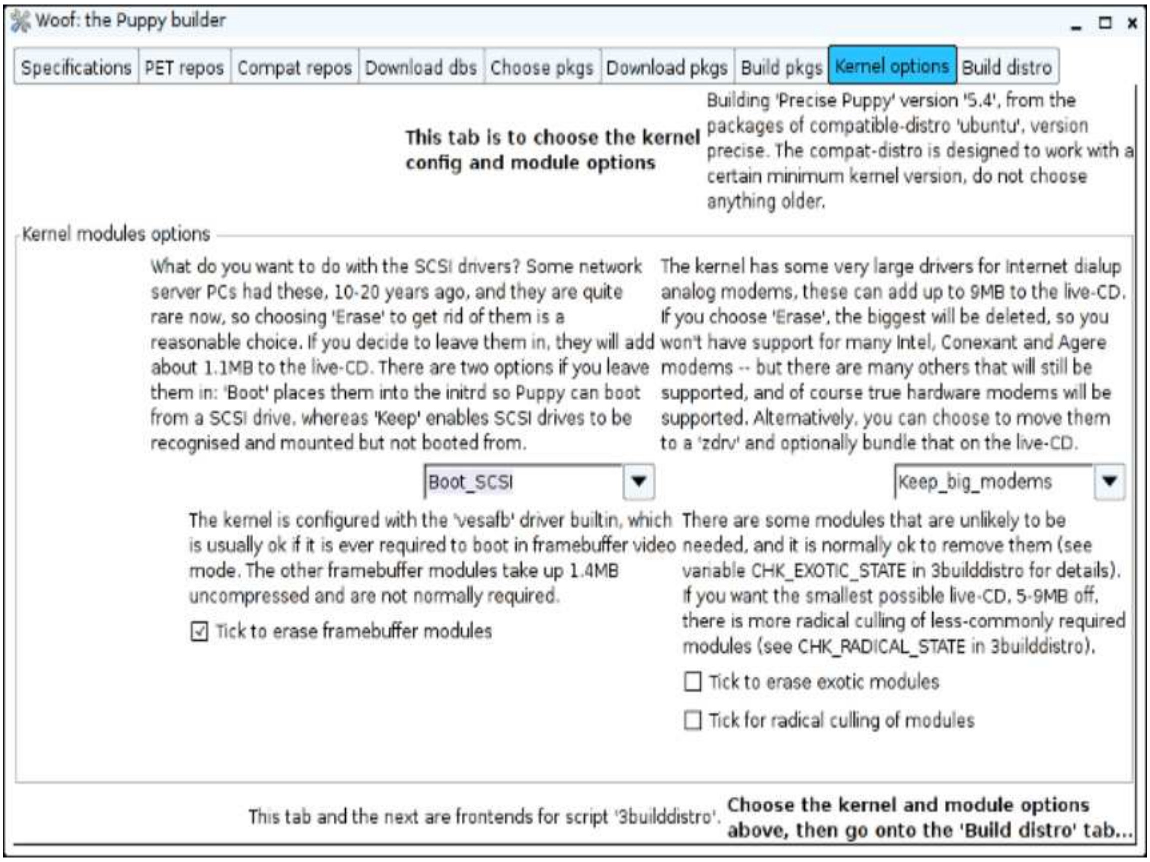
This tab is a frontend for script '2createpackages'

Are last two tabs.

1) The kernel tab here to you can choose the kernel and what module option you want.

2) The Build tab. it says it all; this will build your puppy. It will take you through a series of questions throughout the build process that you will need to answer.

I will post these if they are really needed.



This tab is to choose the kernel config and module options

Building 'Precise Puppy' version '5.4', from the packages of compatible-distro 'ubuntu', version precise. The compat-distro is designed to work with a certain minimum kernel version, do not choose anything older.

Kernel modules options

What do you want to do with the SCSI drivers? Some network server PCs had these, 10-20 years ago, and they are quite rare now, so choosing 'Erase' to get rid of them is a reasonable choice. If you decide to leave them in, they will add about 1.1MB to the live-CD. There are two options if you leave them in: 'Boot' places them into the initrd so Puppy can boot from a SCSI drive, whereas 'Keep' enables SCSI drives to be recognised and mounted but not booted from.

Boot_SCSI ▼

The kernel has some very large drivers for Internet dialup analog modems, these can add up to 9MB to the live-CD. If you choose 'Erase', the biggest will be deleted, so you won't have support for many Intel, Conexant and Agere modems -- but there are many others that will still be supported, and of course true hardware modems will be supported. Alternatively, you can choose to move them to a 'zdrv' and optionally bundle that on the live-CD.

Keep_big_modems ▼

The kernel is configured with the 'vesafb' driver builtin, which is usually ok if it is ever required to boot in framebuffer video mode. The other framebuffer modules take up 1.4MB uncompressed and are not normally required.

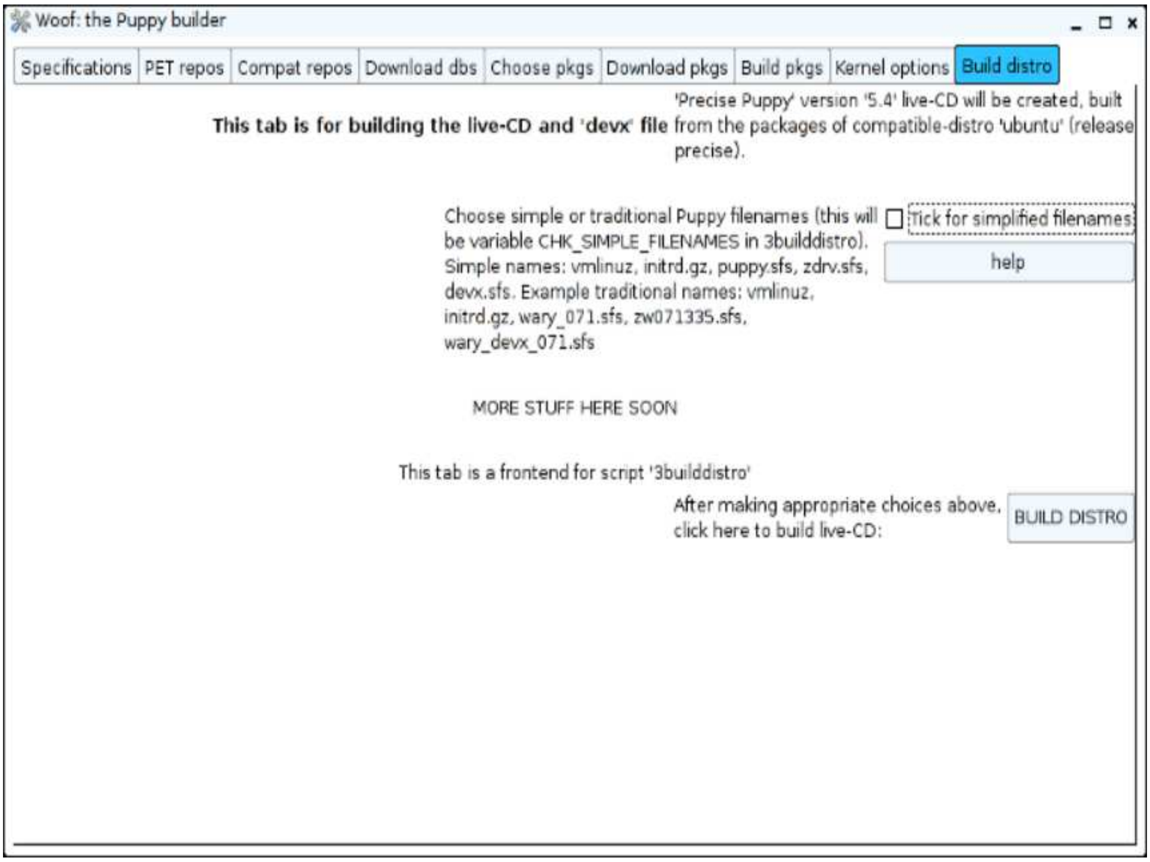
☑ Tick to erase framebuffer modules

There are some modules that are unlikely to be needed, and it is normally ok to remove them (see variable CHK_EXOTIC_STATE in 3builddistro for details). If you want the smallest possible live-CD, 5-9MB off, there is more radical culling of less-commonly required modules (see CHK_RADICAL_STATE in 3builddistro).

☐ Tick to erase exotic modules

☐ Tick for radical culling of modules

This tab and the next are frontends for script '3builddistro'. **Choose the kernel and module options above, then go onto the 'Build distro' tab...**



This tab is for building the live-CD and 'devx' file 'Precise Puppy' version '5.4' live-CD will be created, built from the packages of compatible-distro 'ubuntu' (release precise).

Choose simple or traditional Puppy filenames (this will be variable CHK_SIMPLE_FILENAMES in 3builddistro). Simple names: vmlinuz, initrd.gz, puppy.sfs, zdrv.sfs, devx.sfs. Example traditional names: vmlinuz, initrd.gz, wary_071.sfs, zw071335.sfs, wary_devx_071.sfs

□ Tick for simplified filenames

MORE STUFF HERE SOON

This tab is a frontend for script '3builddistro'

After making appropriate choices above, click here to build live-CD:

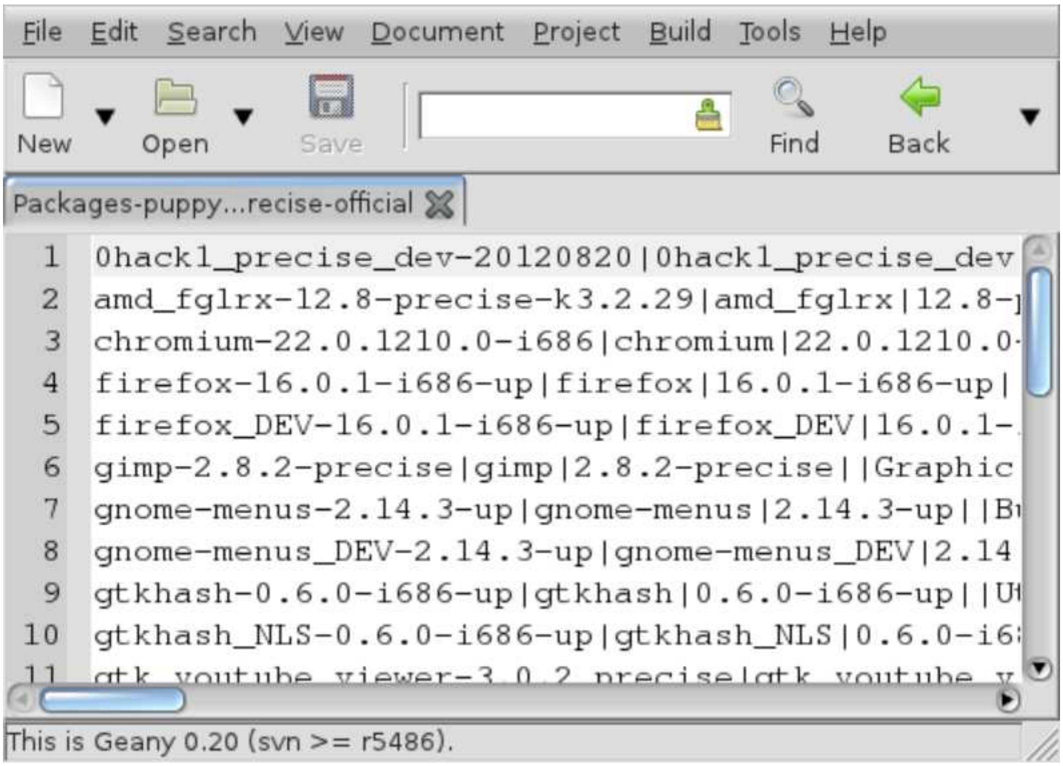
So how does one add different pkgs to the build that are not a part of the puppy that they want.

1) in the puppy pkgs is where you can add the pet that you want to add. I would already have it downloaded and put in the packages-name of puppy folder. You can get the specs on the pet when you unpack it.

You can unpack by renaming it as a tar.gz file, inside will be the pet specifications that you will need to put in the puppy-pkgs-spec. You do this so that when woof checks for packages to be used it will find what you just added.

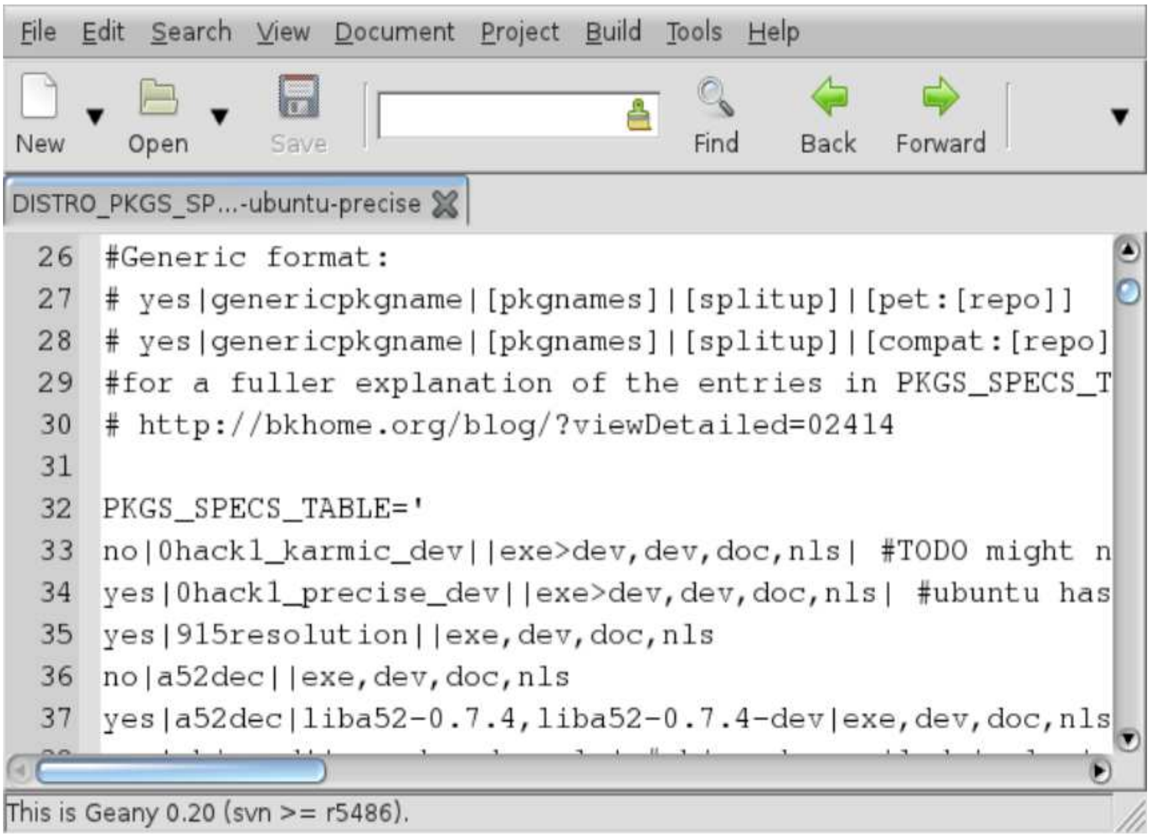
But wait you will need to add to the Distro-packages as well.

2) Distro-packages this is where you will say yes for the package you want to add, you can just add the name in. This is also where you can say no to the pkgs you don't want in as well. I would remove the folders as well if they where downloaded before.



```
0hackl_precise_dev-20120820|0hackl_precise_dev
amd_fglrx-12.8-precise-k3.2.29|amd_fglrx|12.8-
chromium-22.0.1210.0-i686|chromium|22.0.1210.0
firefox-16.0.1-i686-up|firefox|16.0.1-i686-up|
firefox_DEV-16.0.1-i686-up|firefox_DEV|16.0.1-
gimp-2.8.2-precise|gimp|2.8.2-precise||Graphic
gnome-menus-2.14.3-up|gnome-menus|2.14.3-up||B
gnome-menus_DEV-2.14.3-up|gnome-menus_DEV|2.14
gtkhash-0.6.0-i686-up|gtkhash|0.6.0-i686-up||U
gtkhash_NLS-0.6.0-i686-up|gtkhash_NLS|0.6.0-i6
gtk_youtube_viewer-3.0.2_precise|gtk_youtube_v
```

This is Geany 0.20 (svn >= r5486).



File Edit Search View Document Project Build Tools Help
New
Open
Save
Find
Back
Forward
DISTRO_PKGS_SP...-ubuntu-precise

```
#Generic format:
# yes|genericpkgname|[pkgnames]|[splitup]|[pet:[repo]]
# yes|genericpkgname|[pkgnames]|[splitup]|[compat:[repo]
#for a fuller explanation of the entries in PKGS_SPECS_T
# http://bkhome.org/blog/?viewDetailed=02414

PKGS_SPECS_TABLE='
no|0hack1_karmic_dev||exe>dev,dev,doc,nls| #TODO might n
yes|0hack1_precise_dev||exe>dev,dev,doc,nls| #ubuntu has
yes|915resolution||exe,dev,doc,nls
no|a52dec||exe,dev,doc,nls
yes|a52dec|liba52-0.7.4,liba52-0.7.4-dev|exe,dev,doc,nls
```