

Debian Repositories

A Debian repository is a set of Debian packages organized in a special directory tree which also contains a few additional files containing indexes and checksums of the packages. By adding a repository to his **/etc/apt/sources.list** file, a user can easily view and install all the packages available in it just like the packages contained in Debian.

Debian Repositories

A repository can be both online and offline (for example on a CD-ROM), although the former is the more common case.



Terminology

distributions

The three Debian distributions: *stable*, *testing*, and *unstable*.

index files

The *Packages.gz* and *Sources.gz* files.

How Repositories Work

A repository consists of at least one directory with some DEB packages in it, and two special files: `Packages.gz` for the binary packages, and `Sources.gz` for the source packages.

If your repository is listed correctly in `sources.list` (more on that later), `apt-get` will fetch the `Packages.gz` index if the binary packages are listed (with the `deb` keyword) and `Sources.gz` if the sources are listed (with the `deb-src` keyword).

How Repositories Work

Packages.gz contains the name, version, size, the short and the long description, and the dependencies of each package, plus some additional information which is not of interest for us. All that information is listed (and used by) the Debian package managers such as dselect or aptitude.

Sources.gz contains the name, version and the build dependencies (the packages needed to build) of each package (plus some information which is not of interest for us, too); that information is used by apt-get source and similar tools.

How Repositories Work

Thus, once you have set up your repository, you can list and install all of your packages together with the ones in Debian; if you update a package, it'll be upgraded when the user runs

apt-get update

apt-get upgrade

and every user will be able to easily see a short description and other important information about your packages.

How Repositories Work

Repositories can offer different packages for each supported distribution and for each of the (currently eleven) supported architectures; apt will automatically fetch the right one for a user's machine, without him even having to know about all the different architectures. It also allows you to group your packages into components, just as Debian's packages are divided into main, non-free and contrib. So, especially if your software is cross-platform, you'll love package repositories.

A Standard Debian Repository

```
(your repository root)
|
+-dists
|
| -stable
| | -main
| | | -binary-alpha
| | | -binary-arm
| | +-source
| | -contrib
| | | -binary-alpha
| | | -binary-arm
| | +-source
| +-non-free
| | -binary-alpha
| | -binary-arm
| | +-source
|
| -testing
| | -main
| | | -binary-alpha
| ..
|
+-unstable
| -main
| | -binary-alpha
| ..
```

Supported architectures



Contents-alpha.diff/	18-Apr-2007 20:11	-	Contents-alpha.gz	18-Apr-2007 20:04	11M
Contents-amd64.diff/	18-Apr-2007 20:11	-	Contents-amd64.gz	18-Apr-2007 20:07	12M
Contents-arm.diff/	18-Apr-2007 20:11	-	Contents-arm.gz	18-Apr-2007 20:11	11M
Contents-hppa.diff/	19-Apr-2007 08:46	-	Contents-hppa.gz	15-Apr-2007 08:02	11M
Contents-hurd-i386.diff/	19-Apr-2007 08:46	-	Contents-hurd-i386.gz	15-Apr-2007 08:03	9.1M
Contents-i386.diff/	19-Apr-2007 08:46	-	Contents-i386.gz	15-Apr-2007 08:07	12M
Contents-ia64.diff/	15-Apr-2007 20:11	-	Contents-ia64.gz	15-Apr-2007 20:05	11M
Contents-m68k.diff/	15-Apr-2007 20:11	-	Contents-m68k.gz	15-Apr-2007 20:07	11M
Contents-mips.diff/	15-Apr-2007 20:11	-	Contents-mips.gz	15-Apr-2007 20:10	11M
Contents-mipsel.diff/	16-Apr-2007 08:16	-	Contents-mipsel.gz	16-Apr-2007 08:08	11M
Contents-powerpc.diff/	16-Apr-2007 08:16	-	Contents-powerpc.gz	16-Apr-2007 08:12	12M
Contents-s390.diff/	16-Apr-2007 08:16	-	Contents-s390.gz	16-Apr-2007 08:15	11M
Contents-sparc.diff/	16-Apr-2007 20:09	-	Contents-sparc.gz	16-Apr-2007 20:03	11M
contrib/	10-Aug-2006 15:20	-	Contents-udeb.gz	17-Apr-2007 20:03	41K
main/	10-Mar-2007 14:09	-	Release	18-Apr-2007 20:13	78K
non-free/	10-Aug-2006 15:20	-	Release.gpg	18-Apr-2007 20:13	189

Debian/NSLU2



LINKSYS
A Division of Cisco Systems, Inc.

LINKSYS NSLU2 Network Storage Link for USB2.0 Disk Drives 1 10/100 RJ-45 Ethernet port, 2 USB 2.0 ports - Retail



Standards	USB 1.0/1.1/2.0, IEEE 802.3, IEEE 802.3u
Port / Connector	1 10/100 RJ-45 Ethernet port, 2 USB 2.0 ports
Data Transmission Speed	10/100 M
LEDs	Status, Ethernet, USB1,USB2
Dimensions	5.1" x 3.6" x 0.8"
Weight	0.34 lbs.

Quickly add your USB 2.0 storage devices to your network
Built-in 10/100 Fast Ethernet port for throughput up to 200Mbps
Simple Web-based user interface for fast configuration from anywhere on the network

Debian/NSLU2 is the name given to the official Debian distribution for ARM that runs on the NSLU2, thanks to the efforts of the nslu2 linux developers to get the necessary kernel patches and utilities accepted upstream by the kernel maintainers and by the Debian Project.

Debian Repositories

The free packages go into main; the non-free ones into non-free, and the free ones which depend on non-free ones into contrib. Debian currently supports 11 architectures; I've omitted most of them for the sake of brevity.

Each binary-* directory contains a Packages.gz and an optional Release file; each source directory contains a Sources.gz and an optional Release file. Notice that the packages do not have to be in the same directory as the index files, because the index files contain paths to the individual packages; in fact, they could be anywhere else in the repository. This makes it possible to create **pools**.

Debian Repositories

You are free to create as many distributions and components and to call them as you wish; the ones I used in the example are just the ones used in Debian. You could, for example, create the distributions `current` and `beta` (instead of `stable`, `testing` and `unstable`), and the components `foo`, `bar`, `baz` and `qux` (instead of `main`, `contrib` and `non-free`).

While you are free to call the components as you want, it is generally a good idea to use the standard Debian distributions, because that's what Debian users expect.

Using a Repository

Each line has the following syntax:

```
deb|deb-src uri distribution [component1]  
[component2] [...]
```

The uri is the URI of the root of the repository, such as `ftp://ftp.yoursite.com/debian`, `http://yoursite.com/debian`, or, for local files, `file:///home/joe/my-debian-repository`. The trailing slash is optional.

For automatic repositories, you must specify one distribution and one or more components; the distribution must not end in a slash.

Repository Example: /etc/sources.list

```
deb ftp://sunsite.cnlab-switch.ch/mirror/debian/ unstable main contrib non-free
deb-src ftp://sunsite.cnlab-switch.ch/mirror/debian/ unstable main contrib non-free

deb file:///home/aisotton/rep-exact binary/
deb-src file:///home/aisotton/rep-exact source/
```

Standard Repositories

```
# See sources.list(5) for more information, especially
# Remember that you can only use http, ftp or file URIs
# CDRoms are managed through the apt-cdrom tool.
deb http://http.us.debian.org/debian stable main contrib non-
  free
deb http://non-us.debian.org/debian-non-US stable/non-US main
  contrib non-free
deb http://security.debian.org stable/updates main contrib non-
  free

# Uncomment if you want the apt-get source function to work
#deb-src http://http.us.debian.org/debian stable main contrib
  non-free
#deb-src http://non-us.debian.org/debian-non-US stable/non-US
  main contrib non-free
```

Drawbacks of Standard Repository Mirrors

Download speed (dialup, DSL, Cable, ...)

Mirrors get overloaded --> dialup speed with cable or DSL.

What is considered “stable”, “testing”, “unstable”, “experimental” is outside your control.

Creating a local mirror

Various packages such as **apt-move** can create a mirror of a Debian repository on your local LAN to allow transfer speeds about 10x cable modem.

Using apache or another web server, your local repository will be used instead of (or in preference to) a master internet repository.

apt-move

The local mirror machine does ordinary apt-get update/upgrade from <http://us.debian.org>, for example.

Apt-move mirror will populate the local repository from `/var/cache/apt/archives` first and then fetch non-installed packages from the `us.debian.org` to avoid downloading the same package twice.

apt-move

A current snapshot of the Debian archive (about 16000 packages) should take about 10G disk space.

My archive (old versions not purged) is 94G

Why older versions are good

```
aptsh> install python
Reading package lists... Done
Building dependency tree... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
```

Since you only requested a single operation it is extremely likely that the package is simply not installable and a bug report against that package should be filed.

The following information may help to resolve the situation:

The following packages have unmet dependencies:

```
python: Depends: python2.4 (>= 2.4.4) but 2.4.4~c1-1 is to be installed
           Depends: python-minimal (= 2.4.4-1) but 2.4.3-11 is to be installed
```

E: Broken packages

```
Generating and mapping caches...
```

Why older versions are good

```
aptsh> show python
Package: python
Priority: standard
Section: python
Installed-Size: 608
Maintainer: Matthias Klose <doko@debian.org>
Architecture: all
Source: python-defaults
Version: 2.4.4-1
Replaces: python2.3 (<= 2.3.2-6), python-xmlbase, python-base
Provides: python-email, python-xmlbase
Depends: python2.4 (>= 2.4.4), python-minimal (= 2.4.4-1)
Suggests: python-doc, python-tk, python-profiler
Conflicts: python2.3 (<< 2.3.5-14), python2.1 (<= 2.1.2), python-xmlbase, python-csv, python-bz2, python-
base, python-central (<< 0.5.5)
Filename: pool/main/p/python-defaults/python_2.4.4-1_all.deb
Size: 141162
MD5sum: 73a2cb88018dc0097a5a2ab4132aa2b7
Description: An interactive high-level object-oriented language (default version)
Python, the high-level, interactive object oriented language,
includes an extensive class library with lots of goodies for
network programming, system administration, sounds and graphics.
.
This package is a dependency package, which depends on Debian's default
Python version (currently v2.4).
```

Half way point!



Why older versions are good

```
taz10:/mnt/three/debian/pool/main/p/python2.4# ls
```

```
idle-python2.4_2.4.2+2.4.3c1-1_all.deb  python2.4-dev_2.4.3-4_i386.deb          python2.4-minimal_2.4.2-2_i386.deb
idle-python2.4_2.4.2-2_all.deb          python2.4-dev_2.4.3-5_i386.deb          python2.4-minimal_2.4.3-1_i386.deb
idle-python2.4_2.4.3-1_all.deb          python2.4-dev_2.4.3-7_i386.deb          python2.4-minimal_2.4.3-3_i386.deb
idle-python2.4_2.4.3-3_all.deb          python2.4-dev_2.4.3-8_i386.deb          python2.4-minimal_2.4.3-4_i386.deb
idle-python2.4_2.4.3-4_all.deb          python2.4-dev_2.4.3-9_i386.deb          python2.4-minimal_2.4.3-5_i386.deb
idle-python2.4_2.4.3-5_all.deb          python2.4-dev_2.4.4-1_i386.deb          python2.4-minimal_2.4.3-7_i386.deb
idle-python2.4_2.4.3-7_all.deb          python2.4-dev_2.4.4~c1-1_i386.deb        python2.4-minimal_2.4.3-8_i386.deb
idle-python2.4_2.4.3-8_all.deb          python2.4-doc_2.4.2+2.4.3c1-1_all.deb    python2.4-minimal_2.4.3-9_i386.deb
idle-python2.4_2.4.3-9_all.deb          python2.4-doc_2.4.2-2_all.deb            python2.4-minimal_2.4.4-1_i386.deb
idle-python2.4_2.4.4-1_all.deb          python2.4-doc_2.4.3-1_all.deb            python2.4-minimal_2.4.4~c1-1_i386.deb
idle-python2.4_2.4.4~c1-1_all.deb        python2.4-examples_2.4.2+2.4.3c1-1_all.deb python2.4-tk_2.4.2+2.4.3c1-1_i386.deb
python2.4-dbg_2.4.2+2.4.3c1-1_i386.deb  python2.4-examples_2.4.2-2_all.deb        python2.4-tk_2.4.2-2_i386.deb
python2.4-dbg_2.4.2-2_i386.deb           python2.4-examples_2.4.3-1_all.deb        python2.4-tk_2.4.3-1_i386.deb
python2.4-dbg_2.4.3-1_i386.deb           python2.4-examples_2.4.3-3_all.deb        python2.4-tk_2.4.3-3_i386.deb
python2.4-dbg_2.4.3-3_i386.deb           python2.4-examples_2.4.3-4_all.deb        python2.4_2.4.2+2.4.3c1-1_i386.deb
python2.4-dbg_2.4.3-4_i386.deb           python2.4-examples_2.4.3-5_all.deb        python2.4_2.4.2-2_i386.deb
python2.4-dbg_2.4.3-5_i386.deb           python2.4-examples_2.4.3-7_all.deb        python2.4_2.4.3-1_i386.deb
python2.4-dbg_2.4.3-7_i386.deb           python2.4-examples_2.4.3-8_all.deb        python2.4_2.4.3-3_i386.deb
python2.4-dbg_2.4.3-8_i386.deb           python2.4-examples_2.4.3-9_all.deb        python2.4_2.4.3-4_i386.deb
python2.4-dbg_2.4.3-9_i386.deb           python2.4-examples_2.4.4-1_all.deb        python2.4_2.4.3-5_i386.deb
python2.4-dbg_2.4.4-1_i386.deb           python2.4-examples_2.4.4~c1-1_all.deb     python2.4_2.4.3-7_i386.deb
python2.4-dbg_2.4.4~c1-1_i386.deb        python2.4-gdbm_2.4.2+2.4.3c1-1_i386.deb   python2.4_2.4.3-8_i386.deb
python2.4-dev_2.4.2+2.4.3c1-1_i386.deb  python2.4-gdbm_2.4.2-2_i386.deb          python2.4_2.4.3-9_i386.deb
python2.4-dev_2.4.2-2_i386.deb           python2.4-gdbm_2.4.3-1_i386.deb          python2.4_2.4.4-1_i386.deb
python2.4-dev_2.4.3-1_i386.deb           python2.4-gdbm_2.4.3-3_i386.deb          python2.4_2.4.4~c1-1_i386.deb
python2.4-dev_2.4.3-3_i386.deb           python2.4-minimal_2.4.2+2.4.3c1-1_i386.deb
```

Why older versions are good

```
taz10:/mnt/three/debian/pool/main/p/python2.4# ls -al python2.4_2*
```

```
-rw-r--r-- 1 root root 2730120 Mar 28 2006 python2.4_2.4.2+2.4.3c1-  
1_i386.deb  
-rw-r--r-- 1 root root 2718742 Nov 20 2005 python2.4_2.4.2-2_i386.deb  
-rw-r--r-- 1 root root 2762388 Mar 30 2006 python2.4_2.4.3-1_i386.deb  
-rw-r--r-- 1 root root 2769736 Apr 22 2006 python2.4_2.4.3-3_i386.deb  
-rw-r--r-- 1 root root 2763108 Jun 7 19:02 python2.4_2.4.3-4_i386.deb  
-rw-r--r-- 1 root root 2758048 Jun 9 12:17 python2.4_2.4.3-5_i386.deb  
-rw-r--r-- 1 root root 2758208 Jun 14 18:47 python2.4_2.4.3-7_i386.deb  
-rw-r--r-- 1 root root 3011410 Jul 30 12:02 python2.4_2.4.3-8_i386.deb  
-rw-r--r-- 1 root root 3015492 Oct 1 19:17 python2.4_2.4.3-9_i386.deb  
-rw-r--r-- 1 root root 2848366 Oct 19 18:48 python2.4_2.4.4-1_i386.deb  
-rw-r--r-- 1 root root 2848106 Oct 12 18:49 python2.4_2.4.4~c1-1_i386.deb
```

Why older versions are good

```
taz10:/var/log# grep ' installed python' dpkg.log.1
2006-11-02 09:35:26 status installed python-minimal 2.4.4-1
2006-11-02 09:35:26 status installed python 2.4.4-1
2006-11-02 09:35:26 status installed python-central 0.5.9
2006-11-02 09:35:27 status installed python-numeric 24.2-7
2006-11-14 09:16:37 status installed python-central 0.5.10
2006-11-16 07:48:38 status installed python-support 0.5.5
2006-11-25 10:34:29 status installed python-central 0.5.12
2006-11-25 10:34:30 status installed python-support 0.5.6
```

Local Repositories: Pros

We can back out to a previous version easily:

```
dpkg -i python2.4_2.4.3-1_i386.deb
```

Note that 'stable', 'testing', and 'unstable' may point to three different versions of the same package. You can also select which version you think is the 'best'.

Local Repositories: Pros

You can create your own packages.

You can have packages of things that the standard repositories don't have, due to license/patent issues.



Local Repositories: Cons

Bad versions that should be purged hang around.



Debootstrap

NAME

debootstrap - Bootstrap a basic Debian system

SYNOPSIS

```
debootstrap [OPTION...] SUITE TARGET [MIRROR [SCRIPT]]
```

```
debootstrap [OPTION...] --second-stage
```

DESCRIPTION

debootstrap bootstraps a basic Debian system of SUITE (eg, sarge, etch, sid) into TARGET from MIRROR by running SCRIPT. MIRROR can be an http:// URL or a file:/// URL.

Debootstrap can be used to install Debian in a system without using an installation disk but can also be used to run a different Debian flavor in a chroot environment. This way you can create a full (minimal) Debian installation which can be used for testing purposes.

Debootstrap

```
taz6:/tmp# time debootstrap sid ./tstrap
http://taz10:9080/debian
I: Retrieving Release
I: Retrieving Packages
I: Validating Packages
I: Resolving dependencies of required packages...
I: Resolving dependencies of base packages...
I: Found additional base dependencies: update-inetd
I: Checking component main on http://taz10:9080/debian...
I: Retrieving adduser
I: Validating adduser
I: Retrieving apt
I: Validating apt
I: Retrieving apt-utils
I: Validating apt-utils
I: Retrieving aptitude
I: Validating aptitude
...
```

Debootstrap

...

```
I: Retrieving whiptail
I: Validating whiptail
I: Retrieving zlib1g
I: Validating zlib1g
I: Extracting base-files...
I: Extracting base-passwd...
I: Extracting bash...
I: Extracting bsutils...
I: Extracting coreutils...
```

...

Debootstrap

...

```
I: Extracting tar...
I: Extracting tzdata...
I: Extracting util-linux...
I: Extracting zlib1g...
I: Installing core packages...
I: Unpacking required packages...
I: Unpacking base-files...
I: Unpacking base-passwd...
I: Unpacking bash...
```

...

```
I: Unpacking tar...
I: Unpacking tzdata...
I: Unpacking util-linux...
I: Unpacking zlib1g...
I: Configuring required packages...
I: Configuring sysv-rc...
I: Configuring tzdata...
I: Configuring gcc-4.1-base...
I: Configuring libpam-runtime...
```

...

Debootstrap

...

```
I: Configuring sysvinit...
I: Configuring debconf-i18n...
I: Configuring debconf...
I: Unpacking the base system...
I: Unpacking adduser...
I: Unpacking apt...
I: Unpacking apt-utils...
```

...

```
I: Unpacking vim-tiny...
I: Unpacking wget...
I: Unpacking whiptail...
I: Configuring the base system...
I: Configuring gpgv...
I: Configuring gettext-base...
I: Configuring module-init-tools...
```

...

Debootstrap

...

```
I: Unpacking vim-tiny...  
I: Unpacking wget...  
I: Unpacking whiptail...  
I: Configuring the base system...  
I: Configuring gpgv...  
I: Configuring gettext-base...  
I: Configuring module-init-tools...
```

...

```
I: Configuring sysklogd...  
I: Configuring tasksel...  
I: Base system installed successfully.
```

```
real    1m31.601s  
user    0m38.778s  
sys     0m13.353s  
taz6:/tmp# du -h -s tstrap/  
162M    tstrap/
```

Debootstrap uses

- 1) Base of an installation (needs kernel, config. Files, etc.)
- 2) chroot environment (builds, UML, jails, ...)



Building a smaller repository

If we look in
`/tmp/tstrap/var/cache/apt/archives`, we will
find the deb files used for the minimal system

Building a smaller repository

```
taz6:/tmp/tstrap/var/cache/apt/archives# ls
adduser_3.100_all.deb
apt-utils_0.6.46.3_i386.deb
apt_0.6.46.3_i386.deb
aptitude_0.4.4-1_i386.deb
base-files_4_i386.deb
base-passwd_3.5.11_i386.deb
bash_3.1dfsg-7_i386.deb
bsdmainutils_6.1.5_i386.deb
bsdutils_1%3a2.12r-15_i386.deb
coreutils_5.97-5.2_i386.deb
cpio_2.6-17_i386.deb
cron_3.0pl1-99_i386.deb
cyrus-sasl2-doc_2.1.22.dfsg1-5_all.deb
debconf-i18n_1.5.9_all.deb
debconf_1.5.9_all.deb
    2006.11.14+dfsg-1_i386.deb
debian-archive-keyring_2006.11.22_all.deb
debianutils_2.17.4_i386.deb
dhcp3-client_3.0.4-10_i386.deb
dhcp3-common_3.0.4-10_i386.deb
diff_2.8.1-11_i386.deb
    4_i386.deb
dmidecode_2.8-2_i386.deb
dpkg_1.13.24_i386.deb
libnewt0.52_0.52.2-8_i386.deb
libopencdk8_0.5.9-2_i386.deb
libpam-modules_0.79-4_i386.deb
libpam-runtime_0.79-4_all.deb
libpam0g_0.79-4_i386.deb
libpopt0_1.10-3_i386.deb
libreadline5_5.2-1_i386.deb
libsasl2-2_2.1.22.dfsg1-5_i386.deb
libsasl2_2.1.22.dfsg1-5_i386.deb
libselinux1_1.32-3_i386.deb
libsepol1_1.14-2_i386.deb
libsigc++-1.2-5c2_1.2.7-2_i386.deb
libsigc++-2.0-0c2a_2.0.17-2_i386.deb
libslang2_2.0.6-4_i386.deb
libss2_1.39+1.40-WIP-
libssl0.9.8_0.9.8c-4_i386.deb
libstdc++6_4.1.1-20_i386.deb
libtasn1-3-bin_0.3.6-2_i386.deb
libtasn1-3_0.3.6-2_i386.deb
libtext-charwidth-perl_0.04-
libtext-iconv-perl_1.4-3_i386.deb
libtext-wrapi18n-perl_0.06-5_all.deb
```

Building a smaller repository

dselect_1.13.24_i386.deb
e2fslibs_1.39+1.40-WIP-2006.11.14+dfsg-1_i386.deb
2006.11.14+dfsg-1_i386.deb
e2fsprogs_1.39+1.40-WIP-2006.11.14+dfsg-1_i386.deb
ed_0.2-20_i386.deb
findutils_4.2.28-1_i386.deb
gcc-4.1-base_4.1.1-20_i386.deb
gettext-base_0.16.1-1_i386.deb
gnupg_1.4.5-3_i386.deb
gpgv_1.4.5-3_i386.deb
grep_2.5.1.ds2-6_i386.deb
groff-base_1.18.1.1-12_i386.deb
gzip_1.3.5-15_i386.deb
1_i386.deb
hostname_2.93_i386.deb
ifupdown_0.6.8_i386.deb
info_4.8.dfsg.1-4_i386.deb
initscripts_2.86.ds1-36_i386.deb
iptables_1.3.6.0debian1-5_i386.deb
iputils-ping_3%3a20020927-3.1_i386.deb
klogd_1.4.1-20_i386.deb
laptop-detect_0.12.1_i386.deb
libacl1_2.2.41-1_i386.deb
libattr1_2.4.32-1_i386.deb
libblkid1_1.39+1.40-WIP-2006.11.14+dfsg-1_i386.deb
libusb-0.1-4_2%3a0.1.12-2_i386.deb
libuuid1_1.39+1.40-WIP-
libwrap0_7.6.dbs-11_i386.deb
login_1%3a4.0.18.1-5_i386.deb
logrotate_3.7.1-3_i386.deb
lsb-base_3.1-22_all.deb
makedev_2.3.1-83_all.deb
man-db_2.4.3-5_i386.deb
manpages_2.39-1_all.deb
mawk_1.3.3-11_i386.deb
mktemp_1.5-2_i386.deb
module-init-tools_3.3-pre3-
modutils_2.4.27.0-6_i386.deb
mount_2.12r-15_i386.deb
nano_2.0.1-1_i386.deb
ncurses-base_5.5-5_all.deb
ncurses-bin_5.5-5_i386.deb
net-tools_1.60-17_i386.deb
netbase_4.27_all.deb
netcat_1.10-32_i386.deb
openbsd-inetd_0.20050402-3_i386.deb
passwd_1%3a4.0.18.1-5_i386.deb

Building a smaller repository

```
libbz2-1.0_1.0.3-6_i386.deb  
libc6_2.3.6.ds1-8_i386.deb  
libcap1_1%3a1.10-14_i386.deb  
libcomerr2_1.39+1.40-WIP-2006.11.14+dfsg-1_i386.deb  
libconsole_1%3a0.2.3dbs-65_i386.deb  
libdb4.2_4.2.52+dfsg-1_i386.deb  
libdb4.3_4.3.29-6_i386.deb  
libdb4.4_4.4.20-8_i386.deb  
libdevmapper1.02_2%3a1.02.12-1_i386.deb  
libgcc1_1%3a4.1.1-20_i386.deb  
libgcrypt11_1.2.3-2_i386.deb  
libgdbm3_1.8.3-3_i386.deb  
libgnutls13_1.4.4-3_i386.deb  
libgpg-error0_1.4-2_i386.deb  
libldap-2.3-0_2.3.29-1_i386.deb  
libldap2_2.1.30-13.2_i386.deb  
liblocale-gettext-perl_1.05-1_i386.deb  
liblzo1_1.08-3_i386.deb  
liblzo2-2_2.02-2_i386.deb  
libncurses5_5.5-5_i386.deb  
libncursesw5_5.5-5_i386.deb  
perl-base_5.8.8-6.1_i386.deb  
procps_1%3a3.2.7-3_i386.deb  
readline-common_5.2-1_all.deb  
sed_4.1.5-1_i386.deb  
sysklogd_1.4.1-20_i386.deb  
sysv-rc_2.86.ds1-36_all.deb  
sysvinit-utils_2.86.ds1-36_i386.deb  
sysvinit_2.86.ds1-36_i386.deb  
tar_1.16-2_i386.deb  
tasksel-data_2.58_all.deb  
tasksel_2.58_all.deb  
tcpd_7.6.dbs-11_i386.deb  
traceroute_1.4a12-21_i386.deb  
tzdata_2006p-1_all.deb  
update-inetd_4.27-0.2_all.deb  
util-linux_2.12r-15_i386.deb  
vim-common_1%3a7.0-164+1_i386.deb  
vim-tiny_1%3a7.0-164+1_i386.deb  
wget_1.10.2-2_i386.deb  
whiptail_0.52.2-8_i386.deb  
zlib1g_1%3a1.2.3-13_i386.deb
```

Reprepro

Reprepro is a package that can take a collection of deb packages and build a repository.

If we delete the packages in bold such as **dselect**, and then build a repository using reprepro, then use debootstrap on that repository, the resulting directory is reduced from 166M to 129M.



Apt-get clean

Once packages are installed, apt-get clean will remove the deb packages from /var/cache/apt/archives.

This reduces our install sizes:

$166\text{M} - 40\text{M} \text{ *.deb} = 126\text{M}$

$129\text{M} - 34\text{M} \text{ *.deb} = 95\text{M}$

System Test

Running updates from an external vendor (Microsoft, Redhat, Debian, Ubuntu, IBM, ...) on a live system violates the concept of “system test”.

By packaging updates in our own system tested “distro” we can provide “selective fix” capability.

Bleeding Edge

One can run the so-called “unstable” distro from a custom repository knowing that a bad update can easily be reverted.

