



pkgsrc



A Framework for Portable Software Management

What is Pkgsrc?

Pkgsrc (*Package Source*) is a software management framework originally developed for NetBSD. It has since been ported to several other operating systems, including Linux, Solaris, AIX, OSF/1, DragonflyBSD and even MS Windows with Interix! It is a well ordered hierarchy of directories and Makefiles, used to install software and all its dependencies from source or binary packages. Additionally, several programs are available to automate maintenance tasks.

With pkgsrc, you can consolidate your whole software management process in heterogenous networks.

Supported Operating Systems

Supported operating systems include, but are not limited to:

- AIX
- BSD/OS
- Bluewall Linux
- Darwin PowerPC
- Debian GNU/Linux
- DragonFlyBSD
- Fedora Core
- FreeBSD
- IRIX
- Interix
- NetBSD
- OSF/1
- OpenBSD
- PC-BSD
- ROOT Linux
- Red Hat Enterprise Linux
- Slackware Linux
- Solaris
- OpenSUSE Linux

Why do you need Pkgsrc?

On modern computers, several hundred software applications are installed. All this software has to be maintained – e.g. updated, checked for security holes, compiled from source and so on. With the increasing dependencies between software packages, the amount of work to be done tends to explode. Doing all that work by hand is almost impossible – especially if one has to maintain different operating systems on different architectures and in different versions.

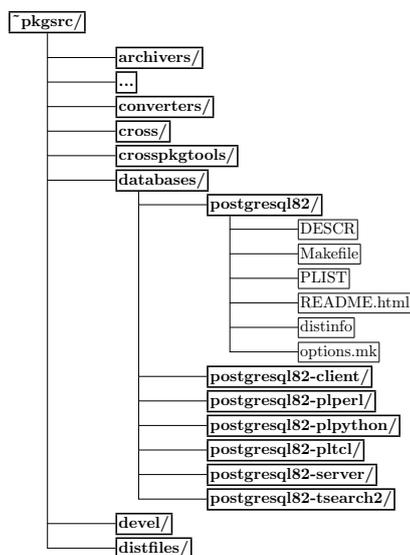
To install software by hand, one needs to know the whole required toolchain – including several versions of *configure*, several hundred versions of *Make* and so on. No one wants to spend a whole workday just to figure out which build toolchain is required to install that piece of software.

With pkgsrc, all these tasks can be automated and consolidated – even in a heterogenous network with different operating systems and architectures!

Separate the Packages from the OS

Some operating systems or distributions mix up configuration files for the base OS with those of the packages. This makes maintenance much harder than it already is.

Pkgsrc separates the operating system from the packages – everything has its own place under `/usr/pkg/`. From man pages to configuration files and binaries – `/usr/pkg/` holds a well ordered hierarchy of directory to be used exclusively by packages.



```

[ /usr/pkgsrc/databases ]
[ phpmgadmin ] Set of PHP-scripts to adminstrate MySQL over the WWW
[ phppgadmin ] Set of PHP-scripts to administer PostgreSQL over the..
[ postgresql80 ] Robust, next generation, object-relational DBMS
[ postgresql80... ] PostgreSQL database client programs
[ postgresql80... ] PL/Perl procedural language for the PostgreSQL backe..
[ postgresql80... ] PL/Python procedural language for the PostgreSQL bac..
[ postgresql80... ] PL/Tcl procedural language for the PostgreSQL backen..
[ postgresql80... ] PostgreSQL database server programs
[ postgresql81 ] Robust, next generation, object-relational DBMS
[ postgresql81... ] PostgreSQL database client programs
[ postgresql81... ] PL/Perl procedural language for the PostgreSQL backe..
[ postgresql81... ] PL/Python procedural language for the PostgreSQL bac..
[ postgresql81... ] PL/Tcl procedural language for the PostgreSQL backen..
[ postgresql81... ] PostgreSQL database server programs
[ postgresql81... ] Tsearch2 contrib module for fulltext indexing in Pos..
[ postgresql82 ] Robust, next generation, object-relational DBMS
[ postgresql82... * ] PostgreSQL database client programs
[ postgresql82... ] PL/Perl procedural language for the PostgreSQL backe..
[ postgresql82... ] PL/Python procedural language for the PostgreSQL bac..
[ postgresql82... ] PL/Tcl procedural language for the PostgreSQL backen..
[ postgresql82... * ] PostgreSQL database server programs
[ postgresql82... * ] Tsearch2 contrib module for fulltext indexing in Pos..
[ pxtools ] Paradox database export/info utilities
[ py-IndexedCa... ] Indexed Catalog extension for standalone ZODB
[ py-PgSQL ] Python DB-API 2.0 compliant interface to PostgreSQL
[ py-ZODB ] Standalone distro of Zope Object Database
[ py-bdb-xml ] Python wrapper for Berkeley DB XML C++ API

[ret] enter [/] search [n] next [o] other cmds
[<-] back [q] quit [f] pkgfind [I] installed packages
  
```

pkg_select, a curses based interface for pkgsrc

An extract of the pkgsrc hierarchy



pkgsrc



A Framework for Portable Software Management

Precompiled Binary Packages or Compilation from Source?

No matter how you prefer your software – pkgsrc will help you. If you want installation exclusively from precompiled binary packages, just adjust one environment variable and pkgsrc will use binaries. A lot of enthusiasts from all over the world compile software with pkgsrc for different operating systems and different platforms. Those packages are uploaded to <ftp://ftp.pkgsrc.org> and freely available to all pkgsrc users. Compiling from source is as easy as using binary packages. Pkgsrc will automatically solve all dependencies, fetch the required software and install it. Pkgsrc can also be used to create binary packages. Those packages can be installed on other machines. So it is possible to build packages for slower machines on a dedicated build server. Even crosscompiling of packages can be done on selected architectures!

One File to Rule them All!

The complete build process is controlled via one single text file! In `/etc/mk.conf`, all required environment variables can be adjusted. Those variables determine global options, like which compiler and which directories should be used during compilation or where the man pages of packages shall be installed. In addition to the global options, package specific options can be used to configure packages, e.g. which dependencies or which version should be used.

How do I use Pkgsrc?

Fetch pkgsrc from pkgsrc.org via CVS or FTP and extract it on your hard drive. Pkgsrc is a hierarchy of directories and files, ordered by categories. To install a package, simply enter the specific directory and type *make install clean clean-depends* – pkgsrc will do the rest for you!

Which Make-Targets do Exist?

- `make fetch`: fetch sources
- `make checksum`: check the sources' integrity
- `make install-depends`: Install all dependencies
- `make extract`: extract sources
- `make patch`: apply patches to the source files
- `make configure`: configure the sources
- `make build`: compile sources
- `make install`: install and register packages
- `make package`: create a binary package from sources
- `make clean`: clean the working directories

- `make clean-depends`: clean the working directories of dependencies
- `make deinstall`: deinstall package
- `make replace`: replace package with newer version
- `make update`: update package and dependencies
- `make fetch-list`: generate a shell script to fetch sources via ftp
- `make show-options`: show available packages

Additional Programs

- `pkg_admin` perform various administrative tasks
- `pkg_add`, `pkg_delete` install and upgrade software packages
- `pkg_chk` check, and optionally update, installed packages
- `pkg_tarup` tars up installed software packages
- `pkgfind` search for packages by keyword
- `pkg_view` install packages in a chroot
- `cdpack` create CDs with binary packages, solves dependencies
- `pkg_select` a text based user interface for pkgsrc

Security

Like the NetBSD project, Pkgsrc emphasizes security. All source files and patches are signed both with a SHA1 and RipeMD160 checksum. Both checksums are checked before the sources are extracted, and the install process will stop if the checksums do not match.

In addition to checksums, OpenPGP signatures can be used to check the integrity of source files. With this procedure enabled, it is almost impossible to plant trojan horses in packages.

To find insecure packages easily, the pkgsrc project maintains a vulnerability list. This text file contains a list of insecure packages and their vulnerabilities. With the program *audit_packages*, the list can be fetched and automatically compared to the installed packages. This task can be automated with cron and can improve your systems security dramatically.

Installation as User

With pkgsrc, a user without root-privileges can install packages into his home directory.

Comprehensive Documentation

The whole system is documented. The pkgsrc guide is freely available at <http://www.NetBSD.org/Documentation/pkgsrc/> or as `pkgsrc/doc/pkgsrc.txt`.

www.pkgsrc.org
