The pkgsrc guide

Documentation on the NetBSD packages system

(2007/09/18)

Alistair Crooks

agc@NetBSD.org

Hubert Feyrer

hubertf@NetBSD.org

The pkgsrc Developers

The pkgsrc guide: Documentation on the NetBSD packages system

by Alistair Crooks, Hubert Feyrer, The pkgsrc Developers

Published 2007/09/18 08:17:21

Copyright © 1994-2007 The NetBSD Foundation, Inc

pkgsrc is a centralized package management system for Unix-

Table of Contents

4.1.5. Checking for security vulnerabilities in

17.5. The *fetch* phase

List of Tabababababbe9g071(a)-1a

Chapter 1.
What is pkgsrc?

1.1. Introduction

Program

The piece of software to be installed which will be constructed from all the files in the distfile by the actions defined in the corresponding package.

1.3.1. Roles involved in pkgsrc

pkgsrc users

change files that are managed by CVS, later updates will try to merge your changes with those that have

Chapter 3.

3.3.2. FreeBSD

To bootstrap using icc, assuming the default icc installation directory:

```
CC=      cc
CXX=     CC
CPP=     cc -E
CXXCPP= CC -E
```

Note: The CPP

Chapter 4.

CDROM (depending on which medium you are using), and the ones that have vulnerabilities and therefore are considered insecure to install without thinking before.

```
% make clean-depends
```

Taking the figlet utility as an example, we can install it on ou

Chapter 5.
Configuring pkgsrc

Chapter 6.
Creating binary packages

```
# ln -s ../disk1/cvs .
   # ln -s cvs/src-2.0 src
```

Each image will contain `README`, `COPYING`, and `bin/myscript` in their root directories.

Chapter 8.
Directory layout of the installed files

Chapter 9.
Frequently Asked Questions

This section contains hints, tips & tricks on special things in pkgsrc that we didn’t find a better place for

II. The pkgsrc developer’s guide

Chapter 10.
Creating a new pkgsrc package from scratch

6. In many cases the package is not yet ready to build. You can fi

```
MAINTAINER=     rillig@NetBSD.org
HOMEPAGE=       http://cvs.nvu.com/
COMMENT=        Web Authoring System

# url2pkg-marker (please do not remove this line.)
.include "../../mk/bsd.pkg.mk"
```

and since a perl wrapper is automatically installed in the “tools” phase, I need to build the package from scratch.

```
$ bmake clean
===> Cleaning for nvu-1.0
$ bmake
[...]
*** /tmp/roland/pkgsrc/www/nvu/work.bacc/.tools/bin/make is not \
```

Chapter 11.
Package components - files, directories and contents

Whenever you’re preparing a package, there are a number of fil

- Add `MANCOMPRESSED`

The general rule to follow is: instead of checking for the operating system the application is being built

Chapter 14.
Buildlink methodology

Buildlink is a framework in pkgsrc that controls what headers and libraries are seen by a package's

```
BUILDLINK_API_DEPENDS.foo+=   foo>=1.1.0
.include "../../category/foo/buildlink3.mk"
```

3. It should be written to allow multiple inclusion. This is *very* important and takes careful attention to `Makefile` coding.

15.5.1. Disabling shell registration

The automatic registration of shell interpreters can be disabled by the administrator by setting the `PKG_REGISTER_SHELLS` environment variable to `NO`.

Chapter 16.
Options handling

Many packages have the ability to be built to support different sets of features. `bsd.options.mk` is a

```
PKG_SUGGESTED_OPTIONS=          wibble-foo
PKG_OPTIONS_LEGACY_VARS+=       WIBBLE_USE_OPENLDAP:ldap
PKG_OPTIONS_LEGACY_OPTS+=       foo:wibble-foo

.include "../../mk/bsd.prefs.mk"

# this package was previously named wibble2
.if defined(PKG_OPTIONS.wibble2)
PKG_LEGACY_OPTIONS+=            ${PKG_OPTIONS.wibble2}
PKG_OPTIONS_DEPRECATED_WARNINGS+= \
        "Deprecated variable PKG_OPTIONS.wibble2 used, use ${PKG_OPTIONS_VAR} instead."
.endif

.include "../../mk/bsd.options.mk"

# Package-specific option-handling

###
### FOO support
###
.if !empty(PKG_OPTIONS:Mwibble-foo)
CONFIGURE_ARGS+=    --enable-foo
.endif

###
### LDAP support
###
.if !empty(PKO support
```

Options that enable features specific to one package, where it's unlikely that another (unrelated) package has the same (or a similar) optional feature, should use a name prefixed with *pkgname*-

PKGPATH

This is a pathname relative to

```
DISTFILES+=     foo-file.tar.gz
SITES.foo-file.tar.gz= \
http://www.somewhere.com/somehow/ \
http://www.somewhereelse.com/mirror/somehow/
```

When actually fetching the distfiles, each item from `MASTER_SITES` or `SITES.*` gets the name of each

```
MASTER_SITES=   ${MASTER_SITE_GNU:=subdirectory/name/}
MASTER_SITES=   ${MASTER_SITE_SOURCEFORGE:=project_name/}
```

Note the trailing slash after the subdirectory name.

17.5.2. How are the files fetched?

The *fetch*

```
INSTALL_LIB_DIR
```

bin-install

Chapter 18.
Tools needed for building or running

The USE_TOOLS definition is used both internally by pkgsrc and also for indi

Chapter 19.
Making your package work

19.1. General operation

Packages that are built with recommendations ignored may not be uploaded to ftp.NetBSD.org by

19.5.1.3. C preprocessor macros to identify the compiler

```
GCC          __GNUC__ (major version), __GNUC_MINOR__
MIPSpro      _COMPILER_VERSION (0x741 for MIPSpro 7.41)
SunPro       __SUNPRO_C (0x570 for Sun C 5.7)
SunPro C++   __SUNPRO_CC (0x580 for Sun C++ 5.8)
```

19.5.2. How to handle compiler bugs

19.6.6. Packages installing perl modules

Chapter 21.
Submitting and Committing

```
(bindings/x.y/x.y.z/sources
```

`do_cleanup()`

This function cleans everything up after the test has been ru

Chapter 26.

```
250-
250-The official FIGlet web page is:
250- http://www.figlet.org/
250-
250-If you have questions, please mailto:info@figlet.org. If you want to
```