

# Creare un port per FreeBSD

*GUFICon #5 Milano, 2/3 Ottobre 2004*

Filippo Natali

`filippo.natali@widestore.net`

# I ports di FreeBSD

- Offrono un sistema semplice di gestione dei processi di installazione e aggiornamento delle applicazioni
- Comprendono oltre 11000 applicazioni suddivise in diverse categorie
- E' possibile installare le applicazioni partendo dal sorgente oppure utilizzando pacchetti precompilati
- L'albero dei ports viene costantemente aggiornato
- Chiunque puo' contribuire creando un port o adottando un port non mantenuto

# Struttura di un port

- Makefile** Contiene informazioni sul port (maintainer, categoria, commento, etc) e le istruzioni necessarie per lo scaricamento dei sorgenti, la compilazione e l'installazione del port
- distinfo** Contiene la dimensione e l'hash MD5 dei files da scaricare (DISTFILES)
- pkg-descr** Descrizione sintetica del port, puo' contenere un riferimento all'autore o alla pagina web dell'applicazione
- pkg-plist** Lista dei files e delle directories installati dal port
- pkg-message** Informazioni aggiuntive sull'installazione o utilizzo del port, in genere viene stampato dopo l'installazione
- files** Directory che contiene patch o altri files necessari al port\*
- pkg-[de]install** Scripts lanciati prima e dopo la [de]installazione del package binario\*
- pkg-req** Script lanciato del package per determinare se debba essere [de]installato\*

\*non obbligatori

# Come si crea un port(1)

- Individuare un applicazione che non sia compresa nei ports, forse la parte piu difficile :)
- Scaricare il sorgente e scompattarlo in una directory di prova
- Tener traccia di ogni modifica effettuata per compilare/installare l'applicazione e (se possibile) creare una lista dei files/directory installati
- Creare la directory del port ed il Makefile, mettendo eventuali patch (create con diff(1)) nella directory files
- Creare distinfo con "#make makesum" (verranno scaricati i DISTFILES se non presenti)
- Creare pkg-plist con i dati del punto 2. oppure tramite uno script (semi)automatico

# Come si crea di un port(2)

- Creare pkg-descr
- Testare il port
- Ri-testare il port, probabilmente ci e' sfuggito qualcosa
- Inviare un problem report per fare includere il port nel tree ufficiale
- **Mantenerlo aggiornato**

# Makefile(1)

```
# New ports collection makefile for: nc6
# Date created: 30 May 2002
# Whom: Filippo Natali <filippo@widestore.net>
#
# $FreeBSD$
#

PORTNAME= nc6
PORTVERSION= 0.5
PORTREVISION= 3
CATEGORIES= net ipv6
MASTER_SITES= ftp://ftp.deepspace6.net/pub/ds6/sources/nc6/ \
http://ftp.deepspace6.net/pub/ds6/sources/nc6/

MAINTAINER= filippo@widestore.net
COMMENT= Netcat clone with IPv6 support
```

# Makefile(2)

```
MAN1= nc6.1
USE_ICONV= yes
GNU_CONFIGURE= yes
CONFIGURE_ENV= CPPFLAGS="-I${LOCALBASE}/include" \
                LDFLAGS="-L${LOCALBASE}/lib"
CONFIGURE_TARGET= --target=${MACHINE_ARCH}-unknown-freebsd${OSREL}

.if !defined(WITHOUT-NLS)
USE_GETTEXT= yes
PLIST_SUB+= NLS=""
.else
CONFIGURE_ARGS+=--disable-nls
PLIST_SUB+= NLS="@comment "
.endif

.include <bsd.port.mk>
```

# Makefile - dipendenze

**LIB\_** Specifica le shared libraries necessarie al programma, ad es:

```
LIB_DEPENDS= jpeg.9:${PORTSDIR}/graphics/jpeg:install
```

**RUN\_** Eseguibili o files necessari a runtime:

```
RUN_DEPENDS= ${ZOOPEBASEDIR}/${ZOOPEPRODUCTDIR}CMFDefault/__init__.py: \  
              ${PORTSDIR}/www/zope-cmf \  
              pdftohtml:${PORTSDIR}/textproc/pdftohtml
```

**BUILD\_** Eseguibili o files necessari per il build del port, specificati come in RUN\_DEPENDS

**FETCH\_** Eseguibili o files necessari per scaricare i DISTFILES.

**EXTRACT\_** Eseguibili o files necessari per scompattare i DISTFILES, da utilizzare se il formato dell'archivio non e' gzip (default), zip (USE\_ZIP) o bzip2 (USE\_BZIP2)

**PATCH\_** Eseguibili o files necessari per applicare patch.

**DEPENDS** Da utilizzare se la dipendenza non ricade nei casi sopra elencati

# Makefile - altre variabili(1)

**DISTNAME** Nome del port o del file dell'applicazione, utilizzato per generare WRKSRC e DISTFILES (default: `${PORTNAME}-${PORTVERSION}`)

**EXTRACT\_SUFX** Suffisso degli archivi che contengono l'applicazione (default: `.tar.gz`)

**DISTFILES** Nome del file (o dei files) che contengono l'applicazione (default: `${DISTNAME}${EXTRACT_SUFX}`)

**WITH[OUT]\_\*** Variabili utilizzate per selezionare opzioni di compilazione/installazione, standard (definite in `bsd.*.mk`) o specifiche del port. Normalmente si controlla solo se le variabili sono definite.

Esempi: `WITH_OPTIMIZED_CFLAGS`, `WITHOUT_GUI`, `WITH_GTK2`

In genere vengono controllate con:

```
.if defined(NOMEVARIABILE)
...
.endif
```

# Makefile - altre variabili(2)

**OPTIONS** Permette di impostare le variabili WITH\_\* attraverso un menu autogenerato, ad es:

```
OPTIONS= EXAMPLES "install example files" off \  
          HTMLMAN "install man pages in HTML format" off
```

genera un menu che permette di impostare interattivamente WITH\_EXAMPLES e WITH\_HTMLMAN (off di default).

Le opzioni impostate da menu vengono memorizzate in /var/db/ports/[nomeport]/options per successivi builds e possono essere modificate con "#make config".

Le seguenti variabili vengono utilizzate in presenza di licenze restrittive:

**NO\_PACKAGE** Impedisce la creazione del package

**NO\_CDROM** Impedisce l'inclusione nei CDROMs di FreeBSD

**RESTRICTED** Impedisce il mirroring dei DISTFILES.

# Targets principali

**fetch** Se i DISTFILES non sono presenti in DESTDIR, verranno scaricati da MASTER\_SITES attraverso il comando FETCH e copiati in DESTDIR. Viene eseguito un controllo di integrità (MD5) e dimensione dei DISTFILES (distinfo).

**extract** I files vengono estratti nella sottodirectory WRKDIR (default: work)

**patch** Se sono stati definiti dei PATCHFILES, vengono scaricati ed applicati. Vengono poi applicati (in ordine alfabetico) tutti i files denominati patch-\* all'interno di PATCHDIR (default: files)

**configure** Se esiste, viene eseguito scripts/configure. Se è definito HAS\_CONFIGURE o GNU\_CONFIGURE, viene eseguito WRKSRC/configure. Se è definito USE\_IMAKE, viene eseguito XMKMF

**build** viene eseguito il build del port all'interno di WRKSRC. Se e' definito USE\_GMAKE, viene utilizzato GNU Make, altrimenti viene utilizzato il Make di sistema.

- E' possibile ridefinire ciascun target (ad es: do-install) e definire azioni da compiere prima o dopo ciascun target (ad es: pre-install, post-install).
- Prima di eseguire ciascun target, vengono verificate e soddisfatte eventuali dipendenze.

# pkg-descr

Netcat 6 is a netcat clone with IPv6 support.  
It is designed to be tiny, extensible, and elegant,  
and it makes use of the new extended BSD socket APIs.

Author: Mauro Tortonesi <mauro@deepspace6.net>

WWW: <http://www.deepspace6.net>

# Creazione di pkg-plist(1)

Se i files e le directory create sono molte, e' possibile creare pkg-plist con i seguenti comandi:

Creare una dir temporanea contenente il tree in cui verranno installato il port e le relative dipendenze:

```
# mkdir /var/tmp/nomeport
# mtree -U -f /etc/mtree/BSD.tipo\_di\_port.dist -d -e -p \
/var/tmp/nomeport
# make depends PREFIX=/var/tmp/nomeport
```

Creare un file contenente il tree appena installato:

```
# (cd /var/tmp/port-name && find -d * -type d) | sort > OLD-DIRS
```

Creare pkg-plist vuoto:

```
# touch pkg-plist
```

Installare il port nella dir temporanea ed aggiungere i files a pkg-plist:

```
# make install PREFIX=/var/tmp/nomeport
# (cd /var/tmp/port-name && find -d * \! -type d) | sort > pkg-plist
```

# Creazione di pkg-plist(2)

Aggiungere le directory create dal port:

```
$(cd /var/tmp/port-name && find -d * -type d) | sort \
| comm -13 OLD-DIRS - | sort -r | sed -e 's#^#@dirrm #' >> pkg-plist
```

Se una directory e' condivisa di piu' ports, potrebbe contenere files che non appartengono al port corrente, quindi per evitare che @dirrm dia errore cercando di rimuovere una dir non vuota si utilizza:

```
@unexec rmdir %D/share/locale/it 2>/dev/null || true
```

Alternativamente e' possibile utilizzare lo script /usr/ports/Tools/scripts/plist. Dopo aver installato il port in una dir temporanea come in precedenza, lanciare:

```
# /usr/ports/Tools/scripts/plist -Md -m \
/etc/mtree/BSD.tipo_di_port.dist \
/var/tmp/tipo_di\port > pkg-plist
```

In entrambi i casi pkg-plist deve essere controllato e sistemato "a manina".

# Creazione di pkg-plist(3)

A volte e' necessario includere elementi in pkg-plist in base alle opzioni con cui e' stato compilato il port. Per farlo basta utilizzare %%VAR%%, dove VAR e' il nome della variabile.

Alcune variabili sono standard, come ad esempio %%OSREL%%, %%PERL\_VER%%, %%PORTDOCS%%. Altre possono essere specificate nel Makefile tramite PLIST\_SUB=VAR=VALUE. Ad esempio per rendere opzionale il native language support si puo' utilizzare:

Nel Makefile:

```
.if !defined(WITHOUT-NLS)
...
PLIST_SUB+=      NLS=" "
.else
...
PLIST_SUB+=      NLS="@comment "
.endif
```

in pkg-plist:

```
%%NLS%%share/locale/de/LC_MESSAGES/whois.mo
%%NLS%%share/locale/el/LC_MESSAGES/whois.mo
...
```

# Package naming convention

I nomi dei packages devono rispettare la seguente forma:

**[language[\_region]]-name[[-]compiled.specifics]-version.numbers**

Il nome del package e' definito da alcune variabili (ridefinibile nel Makefile) nel seguente modo:

**`${PKGNAMEPREFIX}${PORTNAME}${PKGNAME_SUFFIX}-${PORTVERSION}`**

In particolare devono essere rispettate le seguenti regole:

- L'identificativo del linguaggio deve essere di due lettere e impostato in PKGNAMEPREFIX
- La prima lettera del nome deve essere minuscola, il resto puo' contenere lettere maiuscole
- Se il port puo' essere compilato con impostazioni di default hardcoded, devono essere specificati come -compiled.specifics in PKGNAME\_SUFFIX
- La versione puo' essere composta da caratteri alfanumerici separati da punti e non puo' contenere il carattere -

Altre variabili che (se definite) fanno parte del nome del package sono PORTREVISION e PORTEPOCH, ad esempio:

Applicazione	PORTNAME	PORTVERSION	PORTREVISION	PACKAGE
nc6-0.5	nc6	0.5	3	nc6-0.5_3

# Testare il port

Una volta creato il port, e' bene testarlo prima di inviare un pr per l'inclusione nel tree. Supposto che il port venga compilato e installato correttamente, verificare:

- che pkg-plist contenga tutti i files installati dal port e solamente quelli.
- che vengano rimosse tutte le directories create dal port
- che il port possa essere reinstallato piu' volte con make reinstall

La sequenza raccomandata per testare un port e' la seguente:

- 1 make install
- 2 make package
- 3 make deinstall
- 4 pkg\_add nomepacchetto
- 5 make deinstall
- 6 make reinstall
- 7 make package

devel/portlint e' il tool solitamente utilizzato per testare la correttezza del port.

# Il problem report

I nuovi ports (o le modifiche a quelli esistenti) devono essere sottoposti ai committers affinché vengano inclusi nel tree ufficiale. A tale scopo è necessario creare un problem report ed inviarlo al Problem Report Management System (GNATS). Di tutto questo si occupa l'utility `send-pr(1)`:

`Synopsis: [NEW PORT] categoria/nomeport`

`Priority: low`

`Category: ports`

`Class: change-request`

`Description: descrizione breve del port`

`Fix: <risultato di shar `find portdirectory`>`

Nel caso si debba aggiornare un port esistente:

`Synopsis: [MAINTAINER UPDATE] categoria/nomeport`

`Class: maintainer-update`

`Fix: <diff e/o files da includere/cancellare>`

- Se i files da allegare superano i 20k, devono essere compressi e codificati con `uuencode(1)`
- **NON** allegare la `directory work`, i `distfiles` o il `package!!!!`

# Alcuni suggerimenti

- Utilizzare il piu' possibile macro e variabili di `bsd.port.mk`
- Lavorare sempre all'interno di `WRKDIR`
- Non aggiungere nulla dopo `.include <bsd.port.mk>`, se necessario utilizzare `bsd.port.pre.mk/bsd.port.post.mk`
- Usare tab per indentare
- Rispettare il `filesystem`, in particolare il sistema base
- Rispettare variabili importanti (`CC`, `CXX`, `CFLAGS`, `LDFLAGS`) evitando di ridefinirle
- Eliminare tutto quello che non serve e solamente quello (**attenzione ai files di configurazione!**)

# Riferimenti

- FreeBSD Porter's Handbook  
[http://www.freebsd.org/doc/en\\_US.ISO8859-1/books/porters-handbook](http://www.freebsd.org/doc/en_US.ISO8859-1/books/porters-handbook)
- /usr/ports/Mk/\*, in particolare bsd.port.mk
- Ogni port e' un esempio...
- <http://www.freshports.org>
- Le mailing lists di FreeBSD (freebsd-ports@, cvs-ports@...)
- The FreeBSD Port Building Cluster: <http://pointyhat.freebsd.org>
- The FreeBSD Port Distfile Survey: <http://people.freebsd.org/fenner/portsurvey>
- The FreeBSD Ports Monitoring System: <http://portsmon.firepipe.net>
- Current FreeBSD problem reports: <http://www.freebsd.org/cgi/query-pr-summary.cgi>