

Nasıl Debian Paketi Yapılır?

(Howto Build Debian Package?)

Gelişim:

Sürüm 1.0 Aralık 2006 Ozgur Karatas <ozgur at ozgurkaratas dot com>

Özet:

Bu belgede evrensel bir işletim sistemi olan Debian GNU/Linux için nasıl .deb paketi hazırlanacağı anlatılmıştır. Belgenin amacı okuyucuları .deb paketinin oluşumu hakkında bilgilendirmek ve debian paket depolarını zenginleştirmeye teşvik etmektir.

Telif Hakkı:

Bu belgenin; *Nasıl Debian Paketi Yapılır? 1.0* sürümünün telif hakkı Ozgur Karatas'a aittir. Bu belgeyi, Free Software Foundation tarafından yayınlanmış bulunan GNU Özgür Belgeleme Lisansının 1.1 ya da daha sonraki sürümünün koşullarına bağlı olarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz.

Linux, Linus Torvalds adına kayıtlı bir ticarî isimdir.

Başlarken:

Debian bilgisayarınız için özgür bir işletim sistemidir. Bir işletim sistemi bilgisayarınızın çalışmasını sağlayan bir dizi temel program ve araçtan oluşmuş bir yazılım topluluğudur. Debian Linux çekirdeğini (işletim sisteminin çekirdeği) kullanır, ancak temel işletim sistemi araçlarının çoğu GNU projesinden gelmektedir; bu yüzden GNU/Linux olarak adlandırılmıştır.

Debian GNU/Linux salt bir işletim sisteminden daha fazlasını sağlar. Paket depolarında 15490 paket önceden derlenmiş ve kurulumu hazır olarak gelmektedir.

APT Nedir?

Debian işletim sistemini diğer linux dağıtımlarından ayıran ve popüler kılan araçlardan birisi APT'dir. Açılımı Advanced Package Tool'dur. APT'nin amacı Debian kullanıcılarının hayatını kolaylaştırmak ve sistemin yönetiminin daha iyi, anlaşılır şekilde yapılabilmesini sağlamaktır. APT, Debian geliştiricileri tarafından, Debian projesi için geliştirilmiştir.

Önce .tar.gz vardı. Kullanıcılar GNU/Linux sistemlerinde kullandıkları her programı derlemek zorundaydılar. Debian geliştirimi sırasında, kurulu paketlerin yönetimini sağlayacak bir sistemin zorunluluğu hemen görüldü. Bu amaçla geliştirilen sisteme **dpkg** adı verildi. Sonuçta GNU/Linux dünyası, RedHat'in kendi paket yönetim sistemini geliştirmesinden biraz önce, dpkg ile tanıştı.

Hemen ardından GNU/Linux sisteminin geliştiricilerinin kafasında yeni bir ikilem oluştu. Paketlerin hızlı, pratik ve verimli bir yöntemle kurulabilmesini, paket bağımlılıkları yönetimi ve paket güncellemeleri sırasında konfigürasyon dosyalarının güncellenmesini sağlayacak bir araca ihtiyaç vardı. Yine Debian projesi kapsamında ilk ürün ortaya çıktı, APT (Advanced Packaging Tool). Apt daha sonra Connectiva tarafından rpm paketler ile kullanılmak üzere port edildi ve bazı dağıtımlar tarafından da kullanılmaya başlandı.

Aptitude Nedir?

Aptitude kabiliyet anlamına gelmektedir. Apt için kullanılan birçok komut aynı şekilde aptitude için de geçerlidir. **man aptitude** komutu ile hakkında detaylı bilgi alınabilir.

1. Gerekli Paketlerin Kurulumu:

Bir debian paketi hazırlamak için bazı programlara ihtiyaç vardır. Öncelikle sisteminizde bu paketlerin kurulu olduğundan emin olun. Kurulu değil ise aşağıdaki komutla gerekli programları debian depolarından çekerek kurabilirsiniz.

aptitude install build-essential binutils fakeroot lintian debhelper dh-make devscripts

2. Paket Seçimi:

Bir program için debian paketi (.deb) oluşturmadan önce birkaç noktaya dikkat etmelisiniz.

- Hazırladığınız paketin debian depolarında olup olmadığını kontrol edin.
[apt-cache search paket-adi](#)
- Paketleyeceğiniz programın ne tür bir lisansa sahip olduğunu kontrol edin.
Genel olarak GNU\GPL lisansına sahip olması gerekmektedir.
- Programın sistem için bir güvenlik problemi oluşturmadığından emin olun.
- Programın yazarı ile bağlantı kurun ve debian depolarına girmesi için debian geliştiricilerine bilgi verin. Genellikle debian-devel@lists.debian.org mail listesinden temas kurabilirsiniz.

3. Paketimizi hazırlamaya başlayalım:

Öncelikle kafanızın karışmaması için home dizininiz altında yeni bir dizin oluşturun.

```
root@ozgur:/home/ozgur# cd /home/ozgur/  
root@ozgur:/home/ozgur# mkdir package  
root@ozgur:/home/ozgur# cd package/
```

Ardından .deb paketini hazırlayacağınız programın kaynak kodunu bu dizin altında açın.

```
root@ozgur:/home/ozgur/package# ls  
rsyslog-1.12.2.tar.gz  
root@ozgur:/home/ozgur/package# tar -zxvf rsyslog-1.12.2.tar.gz  
root@ozgur:/home/ozgur/package# cd rsyslog-1.12.2
```

Genellikle tüm programların kaynak kodu ile birlikte INSTALL ve README dosyaları gelmektedir. Siz programın ne olduğunu, nasıl çalıştığını bilerseniz de bu dosyaları okumak için biraz zaman ayırmanız yararınıza olacaktır.

Bir program **./configure make** ve **make install** komutları ile sisteminize kurulabilir. Fakat **./configure** seçeneği için çeşitli parametreler vardır. Bilgi edinmek için **./configure --help** komutunu kullanabilirsiniz.

4. Geliştirici Bilgileri:

Programınızın .deb paketini oluşturmadan önce paket ismine ve sürüm numarasına dikkat ediniz. Ayrıca paket oluştururken bazı paketleyici bilgilerine de ihtiyaç duyulmaktadır.

Bunun aşağıdaki komutlar ile bilgilerinizi export etmelisiniz.

```
export DEBEMAIL="ozgur@ozgurkaratas.com"  
export DEBFULLNAME="Ozgur Karatas"
```

Ve bu adımdan sonra sihirli komutumuz olan `dh_make` komutunu kullanıyoruz.

```
root@ozgur:/home/ozgur/package/rsyslog-1.12.2# dh_make
```

```
Type of package: single binary, multiple binary, library, kernel module or cdfs?  
[s/m/l/k/b]
```

`dh_make` komutundan sonra sizden yukarıdaki gibi bazı bilgiler istenecektir. Burada paketinizin tipini seçip enter tuşuna bastıktan sonra aşağıdaki gibi bir çıktı karşınıza gelecektir. Bilgiler onaylıyorsanız enter tuşuna basınız.

```
Maintainer name : Ozgur Karatas  
Email-Address  : ozgur@ozgurkaratas.com  
Date           : Thu, 7 Dec 2006 15:40:48 +0200  
Package Name   : rsyslog  
Version        : 1.12.2  
License        : blank  
Type of Package : Single  
Hit <enter> to confirm:
```

Bu adımdan sonra bir üst dizinde `.orig` uzantılı bir dizin oluşacaktır.

```
root@ozgur:/home/ozgur/package# ls  
rsyslog-1.12.2 rsyslog-1.12.2.orig rsyslog-1.12.2.tar.gz
```

Ayrıca bulunduğunuz dizinde `debian` isminde yeni bir dizin de oluşacaktır ve program ile ilgili tüm `debian` paket bilgileri bu dizin altındaki dosyalarda tutulmaktadır.

```
root@ozgur:/home/ozgur/package/rsyslog-1.12.2# cd debian/  
root@ozgur:/home/ozgur/package/rsyslog-1.12.2/debian# ls  
changelog cron.d.ex      emacsen-remove.ex  manpage.sgml.ex  postrm.ex   rsyslog-default.ex  
compat  dirs      emacsen-startup.ex  manpage.xml.ex   preinst.ex  rsyslog.doc-base.EX  
control docs      init.d.ex          menu.ex          perm.ex        rules  
copyright emacsen-install.ex  manpage.1.ex     postinst.ex     README.Debian  watch.ex  
root@ozgur:/home/ozgur/package/rsyslog-1.12.2/debian#
```

5. Paket ile ilgili Debian Dosyaları:

Dilerseniz yeni oluşan debian/ dizini altındaki dosyaların ne işe yaradığını beraber öğrenelim.

control dosyası:

Paket ile ilgili çeşitli bilgileri içerir. içeriği aşağıdaki gibidir.

```
root@ozgur:/home/ozgur/package/rsyslog-1.12.2/debian# cat control
Source: rsyslog
Section: main
Priority: optional
Maintainer: Ozgur Karatas <ozgur@ozgurkaratas.com>
Build-Depends: debhelper (>= 5)
Standards-Version: 3.7.2
```

```
Package: rsyslog
Architecture: any
Depends: ${shlibs:Depends}, ${misc:Depends}
Description: <insert up to 60 chars description>
<insert long description, indented with spaces>
```

Source satırı: Programınızın adını belirteceğiniz satırdır.

Section satırı: Programınızın lisansına göre hangi bölüme ait olduğunu belirleyen satırdır.

Maintainer satırı: Paketi hazırlayan kişi bilgilerini içeren satırdır.

Build-Depends satırı: Bağımlılıkların belirlendiği satırdır.

Depends satırı: Bu satır oldukça önemlidir. Paketinizin bağımlılıklarını bu değerle belirlersiniz.

Description satırı: Paket hakkında bilgi girebileceğiniz satırdır.

copyright dosyası:

Bu dosya ise programın lisansı hakkında bilgi içermektedir ve default içeriği ise aşağıdaki gibidir.

```
root@ozgur:/home/ozgur/package/rsyslog-1.12.2/debian# cat copyright
This package was debianized by Ozgur Karatas <ozgur@ozgurkaratas.com> on
Thu, 7 Dec 2006 15:40:48 +0200.
```

It was downloaded from <fill in http/ftp site>

Upstream Author: <put author(s) name and email here>

Copyright: <put the year(s) of the copyright, and the names of the
copyright holder(s) here>

License:

<Put the license of the package here>

The Debian packaging is (C) 2006, Ozgur Karatas <ozgur@ozgurkaratas.com> and
is licensed under the GPL, see `/usr/share/common-licenses/GPL'.

changelog dosyası:

Bu dosya ise programınızın seyir defteri yol haritası gibidir. Programın kaynağından bağımsız birşeyler yaptı iseniz veya bazı bug'ları giderdi iseniz bu dosyanın içerisine ekleyebilirsiniz.

rules dosyası:

Bir linux programı kaynak koddan kurulurken ./configure make ve make install komutları işletilir demiştik. Bu komutlar işletilirken program ile beraber gelen **Makefile** dosyasına göre kurulum tamamlanır. Rules dosyası da debian paketiniz için **Makefile** dosyasıdır. Hazırladığınız Debian paketi dpkg ile kurulurken bu dosyadaki bilgilere göre kurulur ve benim programıma göre içeriği aşağıdaki gibidir.

```
root@ozgur:/home/ozgur/package/rsyslog-1.12.2/debian# cat rules
```

```
#!/usr/bin/make -f
```

```
# *- makefile *-
```

```
# Sample debian/rules that uses debhelper.
```

```
# This file was originally written by Joey Hess and Craig Small.
```

```
# As a special exception, when this file is copied by dh-make into a
```

```
# dh-make output file, you may use that output file without restriction.
```

```
# This special exception was added by Craig Small in version 0.37 of dh-make.
```

```
# Uncomment this to turn on verbose mode.
```

```
#export DH_VERBOSE=1
```

```
CFLAGS = -Wall -g
```

```
ifneq (,$(findstring noopt,$(DEB_BUILD_OPTIONS)))
```

```
    CFLAGS += -O0
```

```
else
```

```
    CFLAGS += -O2
```

```
endif
```

```
configure: configure-stamp
```

```
configure-stamp:
```

```
    dh_testdir
```

```
    # Add here commands to configure the package.
```

```
    touch configure-stamp
```

```
build: build-stamp
```

```
build-stamp: configure-stamp
```

```
    dh_testdir
```

```
    # Add here commands to compile the package.
```

```
    $(MAKE)
```

```
#docbook-to-man debian/rsyslog.sgml > rsyslog.1
```

```
touch $@
```

```
clean:
```

```
dh_testdir
```

```
dh_testroot
```

```
rm -f build-stamp configure-stamp
```

```
# Add here commands to clean up after the build process.
```

```
-${MAKE} clean
```

```
dh_clean
```

```
install: build
```

```
dh_testdir
```

```
dh_testroot
```

```
dh_clean -k
```

```
dh_installdirs
```

```
# Add here commands to install the package into debian/rsyslog.
```

```
$(MAKE) install DESTDIR=$(CURDIR)/debian/rsyslog
```

```
# Build architecture-independent files here.
```

```
binary-indep: build install
```

```
# We have nothing to do by default.
```

```
# Build architecture-dependent files here.
```

```
binary-arch: build install
```

```
dh_testdir
```

```
dh_testroot
```

```
dh_installchangelogs
```

```
dh_installdocs
```

```
dh_installexamples
```

```
# dh_install
```

```
# dh_installmenu
```

```
# dh_installdebconf
```

```
# dh_installogrotate
```

```
# dh_installemacsen
```

```
# dh_installpam
```

```
# dh_installmime
```

```
# dh_python
```

```
# dh_installinit
```

```
# dh_installcron
```

```
# dh_installinfo
```

```
dh_installman
dh_link
dh_strip
dh_compress
dh_fixperms
# dh_perl
# dh_makeshlibs
dh_installdeb
dh_shlibdeps
dh_gencontrol
dh_md5sums
dh_builddeb
```

binary: binary-indep binary-arch

.PHONY: build clean binary-indep binary-arch binary install configure

Bu dosyadaki parametreleri programınızın nasıl kurulacağını belirlemek için değiştirebilirsiniz. Bizim için gerekli olan önemli dosyalar bunlardır. Diğer dosyaların üzerinde pek fazla durmayacağız.

[README.Debian](#): Beni Oku dosyası

[conffiles.ex](#): Eğer program kurulurken eski ayar dosyanızı korumak istiyorsanız bu dosyayı kullanın.

[cron.d.ex](#): cron işlemlerini bu dosyadan yapabilirsiniz.

[dirs](#): Kurulumla beraber oluşmayan, daha sonra oluşması gereken dizinleri belirlemek için kullanılır.

[docs](#): Programınız ile beraber belgeler de varsa bu dosya ile belirleyin.

[emacsen-*.ex](#): Programınız kurulurken Emacs dosyasına ihtiyaç duyuyorsa bu dosya ile belirleyin.

[init.d.ex](#): Sistem açılışında programınız da çalışsın istiyorsanız bu dosyayı düzenleyin.

NOT:

Bir sonraki adıma geçmek için, eğer ihtiyacınızın olmadığını düşündüğünüz bir dosya varsa bunu silmeli ve dosyaların .ex uzantılarını silmelisiniz. Burada ex (example) örnek dosya olduğunu göstermektedir.

6. Paketimizi inşa edelim:

Buraya kadar herşeyi anladı iseniz artık programınızın debian paketini hazırlayabilirsiniz. Bunun için aşağıdaki komutu çalıştırın.

```
dpkg-buildpackage -rfakeroot
```

Burada dikkat edilmesi gereken bir konu da Maintainer olarak export etmiş olduğunuz mail adresi için GPG oluşturmuş olmanızdır.

```
export DEBEMAIL="ozgur@ozgurkaratas.com"
```

dpkg; paket oluştururken gpg bilgilerinizi arayacaktır. [gpg --list-keys](#) komutu ile listeleyebilirsiniz.

Herşey yolunda gitti ise programınız için Debian paketi bir üst dizinde kurulmaya hazırdır. [dpkg -i paket adı](#) komutu ile paketi kurup, test ederek gözden geçirebilirsiniz.