

# APT NASIL

Yazan:  
**Gustavo Noronha Silva**  
<kov (at) debian.org>

Çeviren:  
**Murat Demirten**  
<murat (at) debian.org>

Güncelleyen:  
**Oğuz Yarıntepe**  
<oguzy (at) comu.edu.tr>

Ocak 2006

## Özet

Bu belge Debian paket yönetim uygulaması APT hakkında genel kullanım bilgileri vermeyi amaçlamaktadır. APT'nin amacı Debian kullanıcılarının hayatını kolaylaştırmak ve sistemin yönetiminin daha iyi, anlaşılır şekilde yapılabilmesini sağlamaktır. APT Debian dağıtımı olarak kullanıcıları daha çok desteklemek amacıyla, Debian projesi için geliştirilmiştir.

## Konu Başlıkları

<b>1. Başlarken</b>	4
<b>2. Temel Yapılandırma</b>	4
2.1. /etc/apt/sources.list dosyası	4
2.2. APT'nin yerel olarak kullanımı	5
2.3. sources.list dosyası için en iyi yansıya karar verme	5
2.4. sources.list dosyasına CD-ROM ekleme	6
<b>3. Paketlerin yönetimi</b>	7
3.1. Paket listesini güncelleme	7
3.2. Paket kurma	7
3.3. Paket kaldırma	8
3.4. Paket güncelleme	9
3.5. Yeni bir sürüme güncelleme	10
3.6. Kullanılmayan paket dosyalarını temizleme	12
3.7. APT ile dselect kullanımı	12
3.8. Karışık bir sistem nasıl kurulur?	13
3.9. Sadece belirli bir Debian versiyonuna sahip paketleri güncelleme	14
3.10. Belirli bir sürüme sahip paketlerin kurulu olarak kalmasını sağlama	14
<b>4. Yardımcı araçlar</b>	16
4.1. Kendi derlediğim paketleri nasıl kuracağım?	16
4.2. Kullanılmayan yerelleştirme dosyalarını kaldırma	17
4.3. Güncellenebilir paketleri nasıl öğrenebilirim?	18
<b>5. Paketler hakkında bilgi toplama</b>	18
5.1. Paket isimlerini keşfetme	18
5.2. Paket adlarını bulmak için dpkg kullanma	20
5.3. Programları anında kurma	20
5.4. Bir dosyanın hangi pakete ait olduğunu bulma	21

5.5. Paketlerdeki deęişikliklerden haberdar olma . . . . .	21
<b>6. Kaynak paketlerle alıřma . . . . .</b>	<b>22</b>
6.1. Kaynak paketleri indirme . . . . .	22
6.2. Kaynak paketleri derlemek iin gerekli paketler . . . . .	22
<b>7. Hatalarla bařa ıkma . . . . .</b>	<b>23</b>
7.1. Genel hatalar . . . . .	23
7.2. Nereden yardım bulabilirim? . . . . .	24
<b>8. Hangi Linux daęıtımları APT destekliyor? . . . . .</b>	<b>24</b>
<b>9. Teřekkürler . . . . .</b>	<b>24</b>
<b>10. Bu belgenin yeni sürümleri . . . . .</b>	<b>25</b>

**Geçmiş**

1.8.12	Nisan 2006	NBB
Çevirinin lisansı özgün belgenin lisansı ile aynı (GPL) yapıldı: <a href="http://cvs.debian.org/ddp/manuals.sgml/apt-howto/apt-howto.tr.sgml?rev=1.8\&amp;root=debian-howto\&amp;view=log">http://cvs.debian.org/ddp/manuals.sgml/apt-howto/apt-howto.tr.sgml?rev=1.8\&amp;root=debian-howto\&amp;view=log</a>		
1.8.11	Ocak 2006	OY
Özgün belgenin 1.8.11 sürümüne güncellendi.		
1.8.3	Aralık 2002	MD
İlk çeviri.		

**Geçmiş**

1.8.11	Ağustos 2005	GNS
--------	--------------	-----

**Yasal Açıklamalar**

Bu belge çevirisinin, *APT NASIL*, 1.8.11 sürümünün **tefif hakkı © 2006 Oğuz Yarımtepe**'ye, 1.8.3 sürümünün çevirisinin **tefif hakkı © 2003 Murat Demirten**'e ve özgün belgenin **tefif hakkı © 2002 Gustavo Noronha Silva**'ya aittir. Bu çeviriyi, Free Software Foundation tarafından yayınlanmış bulunan **GNU Genel Kamu Lisansı**<sup>(B2)</sup>'nin 2. ya da daha sonraki sürümünün koşullarına bağılı kalarak kopyalayabilir, dağıtabilir ve/veya değiştirebilirsiniz. Bu Lisansın özgün kopyasını <http://www.gnu.org/copyleft/gpl.html> adresinde bulabilirsiniz.

BU BELGE "ÜCRETSİZ" OLARAK RUHSATLANDIĞI İÇİN, İÇERDİĞİ BİLGİLER İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGEYİ "OLDUĞU GİBİ", AŞIKAR VEYA ZIMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BİLGİNİN KALİTESİ İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATALI BİLGİDEN DOLAYI DOĞABİLECEK OLAN BÜTÜN SERVİS, TAMİR VEYA DÜZELTME MASRAFLARI SİZE AİTTİR.

İLGİLİ KANUNUN İCBAR ETTİĞİ DURUMLAR VEYA YAZILI ANLAŞMA HARİCİNDE HERHANGİ BİR ŞEKİLDE TELİF HAKKI SAHİBİ VEYA YUKARIDA İZİN VERİLDİĞİ ŞEKİLDE BELGEYİ DEĞİŞTİREN VEYA YENİDEN DAĞITAN HERHANGİ BİR KİŞİ, BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİLERİN BAŞKA BİLGİLERLE UYUMSUZ OLMASI) YÜZÜNDEN OLUŞAN GENEL, ÖZEL, DOĞRUDAN YA DA DOLAYLI HERHANGİ BİR ZARARDAN, BÖYLE BİR TAZMİNAT TALEBİ TELİF HAKKI SAHİBİ VEYA İLGİLİ KİŞİYE BİLDİRİLMİŞ OLSA DAHİ, SORUMLU DEĞİLDİR.

Tüm telif hakları aksi özellikle belirtilmediği sürece sahibine aittir. Belge içinde geçen herhangi bir terim, bir ticari isim ya da kuruma itibar kazandırma olarak algılanmamalıdır. Bir ürün ya da markanın kullanılmış olması ona onay verildiği anlamında görülmemelidir.

## 1. Başlarken

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, Red Hat'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 yapılandırma dosyalarının güncellenmesini sağlayacak bir araca ihtiyaç vardı. Gene Debian projesi kapsamında ilk ürün ortaya çıktı: APT (Advanced Packaging Tool – Gelişmiş Paket Yönetim Aracı). APT daha sonra Connectiva tarafından RPM paketler ile kullanılmak üzere port edildi ve bazı dağıtımlar tarafından da kullanılmaya başlandı.

Bu belgede Connectiva APT uyarlaması olan `apt-rpm`'den bahsedilmemekle birlikte, belgeye eklenti amacıyla bu konuda da bir şeyler gönderebilirsiniz.

Bu belge bir sonraki Debian sürümü olacak `Sarge` temel alınarak yazılmıştır.

## 2. Temel Yapılandırma

### 2.1. `/etc/apt/sources.list` dosyası

APT, paketlerin bulunduğu kaynaklara nasıl erişebileceğinin bilgisini `/etc/apt/sources.list` dosyasında saklar.

Dosya içindeki girdilerin biçimi aşağıdaki gibidir:

```
deb http://site.http.org/debian sürüm bölüm1 bölüm2 bölüm3
deb-src http://site.http.org/debian sürüm bölüm1 bölüm2 bölüm3
```

Elbette yukarıda verdiğimiz örnek kullanılabilir değildir. Her satırın ilk kelimesi mutlaka `deb` veya `deb-src` olmak zorundadır. Bu ifadeler arşivin türünü belirler: derlenmiş ve kullanıma hazır duruma getirilmiş, çalıştırılabilir paketler (`deb`) veya programın asıl kaynak kodu + Debian kontrol dosyası (`.dsc`) ve programı 'debianlaştırmak' için gerekli değişiklikleri içeren `diff.gz`.

Öntanımlı Debian `sources.list` dosyasının biçimi genellikle aşağıdaki gibi olacaktır:

```
# See sources.list(5) for more information, especialy
# 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
```

En temel Debian kurulumu için ihtiyaç duyulan satırlar bunlardır. Birinci `deb` satırı resmi Debian arşivini gösterirken, ikincisi ABD dışı arşivini ve üçüncüsü ise Debian güvenlik güncellemelerini içeren arşivi göstermektedir.

Sondaki iki satır yorum haline getirilmiş olup (satır başındaki `#` karakteri ile) `apt-get` tarafından yok-sayılmaktadır. Buradaki `deb-src` satırları Debian kaynak paketlerini göstermektedirler. Eğer sıklıkla program kaynak kodlarını test veya yeniden derleme amaçlı olarak indiriyorsanız satırları aktif hale getirmelisiniz.

`/etc/apt/sources.list` dosyası çeşitli tiplerde satırlar içerebilir. APT uygulaması `http`, `ftp`, `file` (yerel dosyalar, örneğin bağlı durumdaki bir ISO9660 dosya sistemi) ve `ssh` arşiv türlerini tanıyabilmektedir.

`/etc/apt/sources.list` dosyasında deęişiklik yaptıktan sonra **apt-get update** komutunu alıřtırmayı unutmayınız. Bu komutla APT'nin dosyada belirtmiř olduęunuz arřivlerdeki gncel paket listesini edinmesini saęlamıř olursunuz.

## 2.2. APT'nin yerel olarak kullanımı

Bazen APT tarafından kurulmasını istedięiniz onlarca paketiniz olabilir. Bunların hepsini birden sisteminizdeki kopyasından kurulumunu isterseniz ařaęıdaki adımları izlemelisiniz.

ncelikle bir dizin oluřturup ierisinde sizdeki `.deb` paketlerini atınız. rneęin:

```
# mkdir /root/debs
```

Paketlerin `control` dosyalarında belirtilen tanımlamalarının zerine kendi tanımlamalarınızı `override` dosyası kullanarak yapabilirsiniz. Bu dosya ierisinde paketle birlikte gelen bazı seenekleri ařaęıdaki gibi tanımlayabilirsiniz:

### *paket ncelik blm*

*paket*, paketin ismini belirtir; *ncelik* deęeri sırasıyla dřk, orta ve yksek anlamında `low`, `medium` ve `high` olabilir; *blm* ise paketin ait olduęu blm gsterir. Dosya adının mutlaka byle olması gerekmez, bařka bir dosya kullanıp **dpkg-scanpackages**'a parametre olarak verebilirsiniz. Eęer bir `override` dosyası yazmak istemiyorsanız **dpkg-scanpackages** programını aęırırken `/dev/null`'u kullanabilirsiniz.

Halen `/root` dizini ierisindeyken ařaęıdaki komutu alıřtırın:

```
# dpkg-scanpackages debs dosya | gzip > debs/Packages.gz
```

Yukarıdaki satırda *dosya* olarak girilen dosya, `override` dosyasıdır. Bu komut APT tarafından kullanılmak zere, paketler hakkında eřitli bilgiler ieren `Packages.gz` dosyasını retir. Paketleri kullanmak iin son olarak ařaęıdaki satırı `sources.list` dosyanıza ekleyin:

```
deb file:/root debs/
```

Bu eklentiden sonra artık APT uygulamasını yeni arřivinle birlikte kullanabilirsiniz. Ayrıca isterseniz kaynak paketler iin de yerel bir arřiv oluřturabilirsiniz. Bunun iin izlemeniz gereken yol yukarıdaki ile hemen hemen aynıdır. Dikkat etmeniz gereken nokta, `.orig.tar.gz`, `.dsc` ve `.diff.gz` dosyalarını da dizin ierisine kopyalamanız ve `Packages.gz` dosyası yerine `Sources.gz` dosyasını oluřturmanız gerektięidir. Kullanacaęınız program ise **dpkg-scansources** olacaktır.

```
# dpkg-scansources debs | gzip > debs/Sources.gz
```

Dikkat ettiyseniz **dpkg-scansources** uygulaması parametre olarak bir `override` dosyasına ihtiya duymamaktadır. `sources.list` dosyanıza eklemeniz gereken satır ise ařaęıdadır:

```
deb-src file:/root debs/
```

## 2.3. sources.list dosyası iin en iyi yansıya karar verme

### *netselect, netselect-apt*

Yeni Debian kullanıcılarının kararsız oldukları noktalardan biri `sources.list` dosyasına hangi Debian yansısını eklemeleri gerektięidir. En iyi yansıya karar vermek iin pek ok yntem mevcuttur. Deneyimli kullanıcılar bir betik programı yazarak yansılar ile aradaki iletiřim hızını lebilirler. Bu iři sizin iin byk lde yapacak bir paket mevcuttur: **netselect**.

**netselect** paketini bildiğimiz yöntemle kuralım:

```
# apt-get install netselect
```

Parametre vermeden programı çalıştırdığımızda bir yardım sayfası gözükecektir. Programı birbirinden boşluk karakteri ile ayrılmış yansı adresleriyle çalıştırırsanız geriye bir skor ve yansı adresi döndürecektir. Buradaki değer yaklaşık olarak **ping** zamanı, ilgili yansıya kaç adımda ulaşılabilindiği bilgisini dikkate alır ve yaklaşık indirme zamanıyla ters orantılıdır (yani, en düşüğü en iyisidir). Geri dönen alan adı en düşük skora sahip olanıdır. Eğer parametre olarak verdiğiniz tüm yansılara ait skorları görmek isterseniz programı **-vv** seçeneği ile çalıştırabilirsiniz.

```
# netselect ftp.debian.org http.us.debian.org ftp.at.debian.org download.unesp.br ftp.d
365 ftp.debian.org.br
#
```

Bu çıktının anlamı, **netselect** programına parametre olarak verdiğimiz yansılar arasından en uygununun **ftp.debian.org.br** olduğu ve bu yansı için elde edilen skor değerinin 365 olduğudur. (Bu değer herkes için farklı olacaktır!).

Şimdi **netselect** tarafından en hızlı olarak belirtilen yansıyı **/etc/apt/sources.list** dosyanıza (bkz. [/etc/apt/sources.list dosyası](#) (sayfa: 4)) ekleyebilir ve ardından [Paketlerin yönetimi](#) (sayfa: 7) bölümündeki yönergeleri takip edebilirsiniz.



### Bilgi

Debian tam yansı listesine [http://www.debian.org/mirror/mirrors\\_full](http://www.debian.org/mirror/mirrors_full) adresinden erişebilirsiniz.

0.3.ds1 sürümünden itibaren **netselect** paketi **netselect-apt** betiğini de içermektedir. Bu betik yukarıdaki işlemleri otomatik olarak yapmaktadır. Dağıtım olarak ne kullanmak istediğinizi belirtmeniz durumunda (öntanımlı olarak kararlı) yansılar arasından sizin için en iyileri seçilecek ve bu karar doğrultusunda en iyi **main** ve **non-US** yansılarını **sources.list** dosyanız oluşturulacaktır. Aşağıdaki örnek kullanım ile kararlı dağıtım için bir **sources.list** dosyası üretilmektedir:

```
# ls sources.list
ls: sources.list: Böyle bir dosya ya da dizin yok
# netselect-apt stable
(...) Bir miktar bekliyoruz
# ls -l sources.list
sources.list
#
```



### Uyarı

**sources.list** dosyası bulunduğunuz dizin altında oluşturulacaktır. Bu dosyayı kullanmak istiyorsanız **/etc/apt** dizini altına taşımalısınız.

Ardından [Paketlerin yönetimi](#) (sayfa: 7) bölümündeki yönergeleri izleyebilirsiniz.

## 2.4. sources.list dosyasına CD-ROM ekleme

Eğer paketlerin kurulumu veya güncellemesi için halihazırda elinizde bulunan bir CD-ROM'u kullanmak istiyorsanız bunu **sources.list** dosyanıza eklemelisiniz. Bunun için CD-ROM'u takıp **apt-cdrom** programını aşağıdaki gibi çalıştırın:

```
# apt-cdrom add
```

CD-ROM bağlanacak ve eğer geçerli bir Debian CD'si ise üzerindeki paketlere ait bilgiler okunacaktır. Eğer CD-ROM ayarlamalarınız biraz standart dışı ise aşağıdaki seçenekleri de kullanabilirsiniz:

```
-h          - bu yardım iletisini gösterir
-d DİZİN    - DİZİN ile CD-ROM'un bağlanacağı yer belirtilir
-r          - Tanınmış bir CD-ROM'u yeniden adlandırır
-m          - Bağlama işlemini gerçekleştirir
-f          - Hızlı kip, paket dosyaları kontrol edilmez
-a          - Tarama kipi ile
```

```
# apt-cdrom -d /home/kov/mycdrom add
```

Ayrıca `sources.list` dosyanıza eklemeyen CD-ROM'u tanıma işlemini yapabilirsiniz:

```
# apt-cdrom ident
```



### Bilgi

**apt-cdrom** CD-ROM aygıtınıza ait ayarların `/etc/fstab` dosyasında düzgün olarak yapılmış olduğu durumda çalışacaktır.

## 3. Paketlerin yönetimi

### 3.1. Paket listesini güncelleme

Paket yönetim sistemi kurulu paketler, kurulu olmayan paketler, kurulabilir paketler vb. hakkında bilgileri kendi özel veritabanında tutmaktadır. **apt-get** programı bu veritabanını kullanarak istenilen paketin nasıl kurulacağını, başka hangi paketlerin daha kurulması gerektiğini öğrenir.

Bu listeyi güncellemek için **apt-get update** komutunu kullanmalısınız. Bu komut `/etc/apt/sources.list` dosyanıza bakar ve belirtilen arşivlerden güncel paket listesini indirir; ayrıntılı bilgi için bkz. [/etc/apt/sources.list dosyası](#) (sayfa: 4).

Paket güncellemeleri ve güvenlikle ilgili güncellemelerden sizin ve sisteminizin haberdar olabilmesi için bu komutu düzenli aralıklarla çalıştırmakta fayda vardır.

### 3.2. Paket kurma

Sonunda sabırsızlıkla beklediğiniz bölüme geldik! Artık `sources.list` dosyanız hazır ve paket listeniz güncel durumda. Tek yapmanız gereken kuracağınız paketi belirtmek.

```
# apt-get install xchat
```

APT hemen kendi veritabanını tarayarak bu paketin en son sürümünü bulacak ve `sources.list` dosyanızda belirtilen arşivden indirmeye başlayacaktır. Bu paketin başka paketlere de bağımlı olması durumunda – ki örneğimizde de öyle – aynı adımları bağımlılık yaratan paketler için de yapacaktır. Aşağıdaki örneğe bakınız:

```
# apt-get install nautilus
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
```

```

bonobo libmedusa0 libnautilus0
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 0 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 17.2MB will be used.
Do you want to continue? [Y/n]

```

**nautilus** paketi bazı kütüphane paketlerine bağımlı durumdadır. Eğer bağımlılık yaratan paketleri de komutu-muza eklemiş olsaydık APT devam etmek istiyor musunuz? sorusunu sormayacak ve hemen paketleri indirmeye başlayacaktı.

Bunun anlamı APT'nin sadece komut satırında belirtilmeyen ancak gereken paketleri de kurmak istediği zaman onay isteyeceğidir.

Aşağıdaki **apt-get** seçenekleri işinize yarayabilir:

```

-h    Bu yardım iletisini gösterir.
-d    Sadece indirir, kurulum yapılmaz
-f    Bütünlük kontrolü başarısız olsa da devam etmeye çalışır
-s    Hiçbir eylem yapmaz, sadece olayı simüle eder
-y    Tüm sorulara Evet cevabı verildiği varsayılır
-u    İşlem sonucunda güncellenecek olan paketleri listeler

```

Birden fazla paket kurulmak üzere tek satırda seçim yapılabilir. Gerekli paketler indirilir ve `/var/cache/apt/archives` dizini altına ilerideki kurulumlarda da kullanılmak üzere kaydedilir.

Aynı komut satırında kaldırılmasını istediğiniz paketleri de belirtebilirsiniz. Bunun için kaldırmak istediğiniz paketin isminin sonuna bir `-` karakteri ekleyerek yazmanız yeterli:

```

# apt-get install nautilus gnome-panel-
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]

```

Paket kaldırmayla ilgili bilgiler için [Paket kaldırma](#) (sayfa: 8) bölümüne bakınız.

Eğer herhangi bir şekilde kurulu bir paketin dosyalarına zarar verdiyseniz veya sadece kurulu bir paketin yeniden indirilip tekrar kurulmasını istiyorsanız `--reinstall` seçeneğini örnekteki gibi kullanmalısınız.

```

# apt-get --reinstall install gdm
Reading Package Lists... Done
Building Dependency Tree... Done
0 packages upgraded, 0 newly installed, 1 reinstalled, 0 to remove and 1 not upgraded.
Need to get 0B/182kB of archives. After unpacking 0B will be used.
Do you want to continue? [Y/n]

```

### 3.3. Paket kaldırma



Bir paketi artık kullanmak istemiyorsanız APT ile sisteminizden paketi kaldırabilirsiniz. Bunun için **apt-get remove package** komutunu vermeniz yeterli olacaktır. Örneğin:

```
# apt-get remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets gnome-panel gnome-panel-data gnome-session
0 packages upgraded, 0 newly installed, 4 to remove and 1 not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

Yukarıda da görüldüğü üzere APT kaldırılan pakete bağımlı olan paketler üzerinde de işlem yapmaktadır. Yukarıdaki soruyu onayladığınızda listelenen tüm paketler kaldırılacaktır. APT kullanarak bir paketi, o pakete bağımlı olan diğer paketleri kaldırmadan kaldırmanın :) bir yolu yoktur.

**apt-get**'in yukarıdaki gibi çalıştırılması sonrasında listelenen paketler sistemden kaldırılacak ancak paketlere ait yapılandırma dosyaları –eğer varsa– sistemde kalacaktır. Yapılandırma dosyaları ile birlikte paketleri tamamen kaldırmak için aşağıdaki komutu çalıştırın:

```
# apt-get --purge remove gnome-panel
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
0 packages upgraded, 0 newly installed, 4 to remove and 1 not upgraded.
Need to get 0B of archives. After unpacking 14.6MB will be freed.
Do you want to continue? [Y/n]
```

Burada isminden sonra \* karakteri olan paketlere ait yapılandırma dosyalarının da kaldırılacağını anlıyoruz.

Kurulum **install** yönteminde olduğu gibi kaldırma **remove** işleminde de aynı satırda kurmak istediğiniz paketleri belirtebilirsiniz. Bunun için kurulmasını istediğiniz paketin sonuna + karakteri eklemeniz gereklidir:

```
# apt-get --purge remove gnome-panel nautilus+
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
The following packages will be REMOVED:
  gnome-applets* gnome-panel* gnome-panel-data* gnome-session*
The following NEW packages will be installed:
  bonobo libmedusa0 libnautilus0 nautilus
0 packages upgraded, 4 newly installed, 4 to remove and 1 not upgraded.
Need to get 8329kB of archives. After unpacking 2594kB will be used.
Do you want to continue? [Y/n]
```

**apt-get**, burada kaldırma işleminin yanı sıra kurulmasını istediğimiz paketi ve gerektirdiği diğer paketlerle birlikte listeler.

### 3.4. Paket güncelleme

Paket güncellemeleri APT'nin çok başarılı olduğu bir işlemdir. Tek bir **apt-get upgrade** komutu ile tüm paketlerin güncellenmesi sağlanabilmektedir. Bu komutu kullandığınız dağıtımdaki programların güncel sürümlerine sisteminizi yükseltmek veya tümüyle yeni bir Debian sürümüne yükseltme yapmak amacıyla kullanabilirsiniz. Dağıtım yükseltmeleri için önerilen yol **apt-get dist-upgrade** ile yapılmasıdır; ayrıntılar için [Yeni bir sürüme güncelleme](#) (sayfa: 10) kısmına bakınız.

Bu komutun `-u` seçeneği ile çalıştırılması oldukça yararlıdır. Bu seçeneklerin kullanımıyla APT güncellenecek paketleri listeler. APT paketlerin en son sürümlerini indirecek ve gereken sırada kurulumu gerçekleştirecektir. Böylesi bir güncelleme yapmadan önce **apt-get update** komutu ile paket listenizi de güncelleniz önemlidir. Aşağıdaki örneğe bakalım:

```
# apt-get -u upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages have been kept back
  cpp gcc lilo
The following packages will be upgraded
  adduser ae apt autoconf debhelper dpkg-dev esound esound-common ftp indent
  ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0 libesd0-dev
  libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev liborbit0
  libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit procs psmisc
29 packages upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
Need to get 5055B/5055kB of archives. After unpacking 1161kB will be used.
Do you want to continue? [Y/n]
```

İşlem oldukça basittir. İlk bir kaç satırda **apt-get** bazı paketlerin sistemde tutulmaya devam edileceğini (kept back) söylemektedir. Bunun anlamı, listelenen programların yeni sürümleri olmasına rağmen herhangi bir nedenden ötürü sisteminize kurulamayacak olmasıdır. Büyük olasılıkla ilgili paketler, paket listenizde yer almayan paketlerin veya paket sürümlerinin kurulmasını gerektirmekte, bu işlem yapılamayacağı için de paketler güncellenmemektedir. İkinci olasılık, paketin yeni sürümünün sisteminizde daha önce olmayan yeni bir pakete bağımlı olmasıdır.

Birinci senaryo için kesin bir çözüm yoktur. İkinci durumda problem yaratan paketi **apt-get install** komutu ile kurarak sorunu çözebilirsiniz. Daha temiz bir yöntem ise **dist-upgrade** kullanmaktır, bu sayede sistemde daha önce olmayan yeni paketler de kurulacaktır, bkz. [Yeni bir sürüme güncelleme](#) (sayfa: 10).

### 3.5. Yeni bir sürüme güncelleme

APT'nin bu özelliği sayesinde bir defada tüm sisteminizi yeni bir Debian sürümüne ister Internet üzerinden isterse varolan bir CD üzerinden (bu CD, satın alınmış veya Internet üzerinden indirilmiş bir ISO görüntüsünden elde edilmiş olabilir) yükseltebilirsiniz.

Ayrıca kurulu paketler arasındaki paket bağımlılıklarında değişimler olduğunda kullanılır. **apt-get upgrade** ile yapılan güncellemelerde bu tür paketlere dokunulmaz.

Örnek olarak, kararlı Debian dağıtımının 0 sürümünü kullandığınızı düşünün ve 3. sürümü içeren bir CD edinmiş olun. APT'yi kullanarak sisteminizi bu CD üzerinden yeni sürüme güncelleyebilirsiniz. Bunu yapmak için öncelikle **apt-cdrom** ile (bkz. [sources.list dosyasına CD-ROM ekleme](#) (sayfa: 6)) ile CD'yi `/etc/apt/sources.list` dosyanıza ekleyin ve ardından **apt-get dist-upgrade** komutunu çalıştırın.

Unutmayın, APT her zaman paketin güncel sürümünü arar. Eğer `/etc/apt/sources.list` dosyanızda belirtilen arşivlerde bir paketin CD üzerindeki daha yeni bir sürümü var ise güncelleme sırasında bu sürüm kullanılacaktır.

[Paket güncelleme](#) (sayfa: 9) bölümündeki örneğimizde bazı paketlerin güncelleme sırasında korunduğunu (kept back) görmüştük. Bu problemi şimdi **dist-upgrade** ile çözebiliriz:

```
# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
```

```
The following NEW packages will be installed:
  cpp-2.95 cron exim gcc-2.95 libident libopenldap-runtime libopenldap1
  libpcre2 logrotate mailx
The following packages have been kept back
  lilo
The following packages will be upgraded
  adduser ae apt autoconf cpp debhelper dpkg-dev esound esound-common ftp gcc
  indent ipchains isapnptools libaudiofile-dev libaudiofile0 libesd0
  libesd0-dev libgtk1.2 libgtk1.2-dev liblockfile1 libnewt0 liborbit-dev
  liborbit0 libstdc++2.10-glibc2.2 libtiff3g libtiff3g-dev modconf orbit
  procs psmisc
31 packages upgraded, 10 newly installed, 0 to remove and 1 not upgraded.
Need to get 0B/7098kB of archives. After unpacking 3118kB will be used.
Do you want to continue? [Y/n]
```

Görüldüğü gibi paketler hem güncellenecek hem de yeni bağımlılıklar için yeni paketler de kurulacaktır. Ancak **lilo** paketi halen sistemde korunmaya devam etmektedir (kept back). Muhtemelen bağımlılık dışında daha ciddi bir problemi vardır. Problemin ne olduğunu aşağıdaki komutla görebiliriz:

```
# apt-get -u install lilo
Reading Package Lists... Done
Building Dependency Tree... Done
The following extra packages will be installed:
  cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2
  logrotate mailx
The following packages will be REMOVED:
  debconf-tiny
The following NEW packages will be installed:
  cron debconf exim libident libopenldap-runtime libopenldap1 libpcre2
  logrotate mailx
The following packages will be upgraded
  lilo
1 packages upgraded, 9 newly installed, 1 to remove and 31 not upgraded.
Need to get 225kB/1179kB of archives. After unpacking 2659kB will be used.
Do you want to continue? [Y/n]
```

Yukarıdaki belirtildiği üzere, **lilo** paketi **debconf-tiny** paketi ile yeni oluşan bir çakışmaya sahiptir. Bunun anlamı, **debconf-tiny** paketi kaldırılmadan **lilo** paketinin yüklenemeyeceği ve güncellenemeyeceğidir.

Korunan veya kaldırılan paketlerle ilgili daha ayrıntılı bilgi almak için komutu aşağıdaki gibi kullanabilirsiniz:

```
# apt-get -o Debug::pkgProblemResolver=yes dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Starting
Starting 2
Investigating python1.5
Package python1.5 has broken dep on python1.5-base
  Considering python1.5-base 0 as a solution to python1.5 0
  Holding Back python1.5 rather than change python1.5-base
Investigating python1.5-dev
Package python1.5-dev has broken dep on python1.5
  Considering python1.5 0 as a solution to python1.5-dev 0
  Holding Back python1.5-dev rather than change python1.5
  Try to Re-Instate python1.5-dev
Done
Done
The following packages have been kept back
```

```
gs python1.5-dev
0 packages upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
```

Burada açıkça görüldüğü gibi `python1.5-dev` paketi kurulamayacaktır çünkü `python1.5` paketine bağımlıdır. `python1.5` kurulamadığı için `python1.5-dev` paketi de kurulamamaktadır.

### 3.6. Kullanılmayan paket dosyalarını temizleme

#### **`apt-get clean` ve `autoclean`**

Bir paket kuracağınız zaman APT gerekli dosyaları `/etc/apt/sources.list` dosyanızda belirtilen konaklardan alır ve yerel bir dizin (`/var/cache/apt/archives/`) altında saklar. Ardından kurulumu buradaki dosyalar üzerinden gerçekleştirir, bkz. [Paket kurma](#) (sayfa: 7).

Zamanla dosyaların tutulduğu yerel dizinin boyutları artmaya ve gereksiz yer kaplamaya başlar. APT bu dizini temizlemek için komutlar içerir: **`apt-get`**'in **`clean`** ve **`autoclean`** komutları.

**`apt-get clean`** kilit dosyaları haricinde `/var/cache/apt/archives/` ve `/var/cache/apt/archives/partial/` dizinlerindeki tüm dosyaları siler. Sonuç olarak, eğer bir paketi yeniden kurmak isterseniz (`--reinstall` seçeneği ile) APT gerekli dosyalara artık sahip olmadığından, yeniden indirecektir.

**`apt-get autoclean`** ise sadece artık indirilmesi mümkün olmayan dosyaları siler.

Aşağıdaki örnek **`apt-get autoclean`** özelliğinin nasıl çalıştığını gösteriyor:

```
# ls /var/cache/apt/archives/logrotate* /var/cache/apt/archives/gpm*
logrotate_3.5.9-7_i386.deb
logrotate_3.5.9-8_i386.deb
gpm_1.19.6-11_i386.deb
```

`/var/cache/apt/archives` altında `logrotate` paketi için 2 dosya ve `gpm` paketi için de bir 1 dosya bulunmaktadır.

```
# apt-show-versions -p logrotate
logrotate/stable uptodate 3.5.9-8
# apt-show-versions -p gpm
gpm/stable upgradeable from 1.19.6-11 to 1.19.6-12
```

**`apt-show-versions`** çıktısına baktığımızda `logrotate_3.5.9-8_i386.deb` dosyasının `logrotate` paketi için güncel sürüm olduğunu görüyoruz. Bu durumda `logrotate_3.5.9-7_i386.deb` dosyası hiç bir işe yaramamaktadır. Ayrıca `gpm_1.19.6-11_i386.deb` dosyası da, indirilebilecek daha güncel bir sürümü olduğu için işe yaramamaktadır.

```
# apt-get autoclean
Reading Package Lists... Done
Building Dependency Tree... Done
Del gpm 1.19.6-11 [145kB]
Del logrotate 3.5.9-7 [26.5kB]
```

Sonuçta, **`apt-get autoclean`** ile sadece eski ve işe yaramayacak dosyaların kaldırılmasını sağlamış olduk, bkz. [Sadece belirli bir Debian versiyonuna sahip paketleri güncelleme](#) (sayfa: 14).

### 3.7. APT ile `dselect` kullanımı

**`dselect`** kurulum için Debian paketlerini seçebileceğiniz bir programdır. Kullanımı oldukça karışık ve sıkıcıdır, fakat biraz pratikle konsol tabanlı ncurses kütüphanesi üzerinde çalışan arayüzüne alışılabilir.

**dselect**'in özelliklerinden biri, Debian paketlerinin "şiddetle önerilen" ve "tavsiye edilen" paketlere ait bilgileri işleyebilmesidir ("recommending" ve "suggesting"). Programı çalıştırmak için root iken **dselect**'i çalıştırın. Erişim yöntemleri menüsünden **apt** seçimini yapın. Mutlaka gerekli olmamakla birlikte, eğer bir CDROM kullanmıyor ve paket indirmek istiyorsanız erişim yöntemi olarak **apt** kullanmalısınız.

**dselect** kullanımı hakkında ayrıntılı bilgi almak isterseniz **dselect** belgelerinin bulunduğu <http://www.debian.org/doc/> adresini ziyaret edebilirsiniz.

**dselect** ile seçimlerinizi yaptıktan sonra, aşağıdaki komutu çalıştırın:

```
# apt-get -u dselect-upgrade
```

```
# apt-get -u dselect-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
The following packages will be REMOVED:
  lbxproxy
The following NEW packages will be installed:
  bonobo console-tools-libs cpp-3.0 enscript expat fingerd gcc-3.0
  gcc-3.0-base icepref klogd libdigest-md5-perl libfnlib0 libft-perl
  libgc5-dev libgcc300 libhtml-clean-perl libltdl0-dev libsasl-modules
  libstdc++3.0 metamail nethack proftpd-doc psfontmgr python-newt talk tidy
  util-linux-locales vacation xbill xplanet-images
The following packages will be upgraded:
  debian-policy
1 packages upgraded, 30 newly installed, 1 to remove and 0 not upgraded.
Need to get 7140kB of archives. After unpacking 16.3MB will be used.
Do you want to continue? [Y/n]
```

Şimdi bu ekran çıktısını aynı sistemdeki **apt-get dist-upgrade** çıktısı ile karşılaştıralım:

```
# apt-get -u dist-upgrade
Reading Package Lists... Done
Building Dependency Tree... Done
Calculating Upgrade... Done
The following packages will be upgraded:
  debian-policy
1 packages upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 421kB of archives. After unpacking 25.6kB will be freed.
Do you want to continue? [Y/n]
```

Görüldüğü gibi **dselect** ile yapılan güncellemede yukarıdaki listede olmayan paketler de yüklenmek üzere seçilmiştir. Çünkü güncellenecek olan paketlerin "şiddetle önerilen" ve "tavsiye edilen" paketler alanları da incelenmiş, bu paketler listeye dahil edilmiştir. **dselect**, APT ile birlikte kullanıldığında oldukça güçlü bir araçtır.

### 3.8. Karışık bir sistem nasıl kurulur?

Kullanıcılar bazen bir Debian sürümünü temel sistem olarak kullanırken bir veya birkaç paketini diğer sürümlerden kullanmak isterler.

Ana Debian sürümünü belirtmek için aşağıdaki satırı `/etc/apt/apt.conf` (eğer yoksa oluşturun) dosyanıza ekleyin:

```
APT::Default-Release "sürüm";
```

Buradaki *sürüm* kısmı ana sürüm olarak kullanmak istediğiniz sürüm adıdır. Kullanabileceğiniz sürümler *stable*, *testing* ve *unstable*'dir. Diğer bir sürümden paket yüklemek için APT'yi şu şekilde kullanın:

```
# apt-get -t sürüm install paket_adi
```

Bu şekilde çalıştırabilmek için `/etc/apt/sources.list` dosyasında paketi istediğiniz sürümle ilgili en az bir APT kaynak girdisi olmalı ve paket kaynaktan indirilebilir olmalıdır

Paketin belli bir sürümünü kullanmak isterseyebilirsiniz:

```
# apt-get install paket_adi=sürüm
```

Örneğin aşağıdaki örnekte `nautilus` paketinin `2.2.4-1` sürümü kurulmaktadır:

```
# apt-get install nautilus=2.2.4-1
```



### Uyarı

Debian 'unstable' (kararsız) sürümü Debian paketlerinin en yeni sürümlerinin konulduğu sürümdür. Bu sürümde paketlerle ilgili birkaç paketi veya tüm sistemi etkileyebilecek en küçüğünden en tehlikeli değişikliklere kadar değişimler olur. Bu yüzden de dağıtımın bu sürümü tecrübesiz veya sistemde kararlılık bekleyen kullanıcılar tarafından **kullanılmamalıdır**.

'testing' (deneme) sürümü de 'unstable' sürümünden daha iyi sayılmaz, çünkü güvenlik güncellemelerini zamanında kabul etmemektedir. Sunucular ve diğer üretim sistemleri için kararlı sürüm kullanılmalıdır.

## 3.9. Sadece belirli bir Debian versiyonuna sahip paketleri güncelleme

`apt-show-versions` programı, karışık bir sistem kullanan kullanıcıların sistemlerini güncellemelerinde daha güvenli bir ortam sunar. Örneğin aşağıdaki gibi sadece sisteminizdeki `unstable` paketleri güncelleyebilirsiniz:

```
# apt-get install `apt-show-versions -u -b | grep unstable | cut -d ' ' -f 1`
```

## 3.10. Belirli bir sürüme sahip paketlerin kurulu olarak kalmasını sağlama

### ileri düzey

Bazen bir paketteki bazı kısımları değiştirmeniz gerekebilir ve bunun için de zamanınız olmayabilir veya bu değişiklikleri programın yeni sürümüne yansıtmak istemiyor olabilirsiniz. Örneğin sisteminizi Debian 3.0 dağıtımına güncellemiş fakat belirli bir paketin Debian 2.2 dağıtımındaki halini kullanmaya devam etmek isteyebilirsiniz. Bu problemi çözmek için paketlerinizi **pin**leyebilirsiniz.

Bu özelliği kullanmak oldukça kolaydır, tek yapmanız gereken `/etc/apt/preferences` dosyasını düzenlemekten ibaret.

Dosyanın biçemi şu şekildedir:

```
Package: <paket_adi>
Pin: <pin tanımlaması>
Priority: <pin önceliği>
```

Örnek olarak, `sylpheed` paketinin `0.4.99` sürümünü korumak isterseniz aşağıdaki kaydı buraya ekleyebilirsiniz:

```
Package: sylpheed
Pin: version 0.4.99*
```

Buradaki `*` kullanımına dikkat! Bu karakter ile girdiğimiz **pin** değerinin tüm `0.4.99` ile başlayan sürümler için geçerli olduğunu belirtebiliyoruz. Bu sayede program sürümünün yanı sıra paketin Debian sürümünün değişmesi durumunda da koyduğunuz kurallar geçerli olur, örneğin `0.4.99-1` sürümü ile `0.4.99-10` sürümü.

Pin önceliği "Packages:" ve "Pin:" satırlarına karşılık gelen paketlerin yüklenip yüklenmeyeceğine karar verilmede kullanılır. Yüksek öncelik değeri paketin yüklenmesini sağlar. `apt_preferences (7)` kılavuzunu detaylar için okuyabilirsiniz. Bir kaç örnek değeri yukarıdaki örnek için açıklayalım:

1001

Sylpheed 0.4.99 sürümü apt kullanımı ile başka sürümlerle değiştirilemeyecektir. Daha yüksek sürümleri edinilebilir olsa da apt kullanımında sadece 0.4.99 sürümü kurulacaktır. Bir paketin eski sürümleri ile değiştirilebilir olmasını `pin` değerleri de gözönüne alınarak sağlamak için ("downgradable") öncelik değeri 1000'den büyük hale getirilmelidir.

1000

Etkisi 1001 önceliği ile aynıdır. Farkı ise 0.4.99 sürümünden daha düşük bir sürüme indirmeye (downgrade) izin verilmeyecek olmasıdır.

990

0.4.99 sürümü sadece "APT::Default-Release" değişkeninde belirtilen tercih edilen sürümde daha yeni bir sürümde paket olması durumunda var olanla değiştirilecektir. bkz. [Karışık bir sistem nasıl kurulur?](#) (sayfa: 13)

500

Herhangi bir sürümdeki sylpheed 0.4.99 sürümünden daha büyük sürüm numarasına sahip olan paket tercih edilen paket olacaktır. Düşük sürüm numaraları içinse 0.4.99 geçerliliğini koruyacaktır

100

Herhangi bir sürümdeki sylpheed 0.4.99 sürümünden daha büyük sürüm numarasına sahip olan paket tercih edilen paket olacaktır. 0.4.99 sürümü eğer başka herhangi bir sürümü kurulmadıysa yüklenecektir. Bu öncelik değeri yüklü paketler için kullanılır.

-1

Negatif öncelik değerleri de kullanılabilir. Bu kullanım ile 0.4.99 sürümünün yüklenmesi tamamen önlenmiş olur.

Bir `pin` değeri paketin "Debian dağıtımı"na, "sürümü"ne ve "adı"na göre verilebilir.

Yukarıda gördüğümüz gibi pin değeri vermede sürüm numaraları ve özel karakterler kullanılır.

Sürüme pin değeri verme işlemi APT'nin kullandığı arşivlerdeki veya CD'deki `Release` dosyası ile ilişkilidir. Eğer kullandığınız arşivler `Release` dosyasını sağlamıyorsa bu özellik kullanılamaz. `Release` dosyanızın içeriğini `/var/lib/apt/lists/` dizini altından görebilirsiniz. Sürüm için kullanılacak seçenekler `a` (arşiv), `c` (components – bölüm), `v` (version – sürüm), `o` (origin – kaynak) ve `l` (label – etiket) tir.

Bir örnek:

```
Package: *
Pin: release v=2.2*,a=stable,c=main,o=Debian,l=Debian
Priority: 1001
```

Bu örnekte Debian 2.2\* dağıtımlarını (2.2r2, 2.2r3 vb. – bu seçim ile güvenlik yamaları ve önemli güncellemeler dahil edilmiş oldu), kararlı (`stable`) arşivi, `main` bölümü (`contrib` veya `non-free` de olabilirdi), kaynak ve etiket olarak Debian'ı seçtik. Kaynak (`o=`) `Release` dosyasını kimin ürettiğini belirtir, etiket (`l=`) ise dağıtımın ismidir (Debian dağıtımı için etiket Debian iken Progeny için Progeny'dir). Örnek bir `Release` dosyasının içeriği aşağıdadır:

```
$ cat /var/lib/apt/lists/ftp.debian.org.br_debian_dists_potato_main_binary-i386_Release
Archive: stable
Version: 2.2r3
```



```
Component: main
Origin: Debian
Label: Debian
Architecture: i386
```

## 4. Yardımcı araçlar

### 4.1. Kendi derlediğim paketleri nasıl kuracağım?

#### *equivs*

Bazen bir programın belirli bir sürümünü kullanmak zorunda kalabiliriz. Bu sürüme ait bir Debian paketi mevcut değil ise, programı kaynak kodundan derleyerek kendimiz kurarız. Fakat paket yönetim sistemi bu durumda şaşırabilir. Mesela kullandığınız eposta sunucu yazılımının yeni bir sürümünü derlediğinizi düşünün. Burada problem yok, ancak pek çok Debian paketi sistemde bir MTA (Mail Transport Agent) yazılımı olmasını gerektirir. Eposta sunucunuzu kendiniz derleyerek kurduğunuz için paket yönetim sistemi bundan haberdar değildir ve bağımlılık hataları oluşur.

Burada **equivs** paketi devreye girmektedir. Paketi kurmak için aynı isimli paketi sisteminize kurmalısınız. Bu program ile boş bir paket yaratıp, bağımlılık problemlerini çözebilir ve paket yönetim sisteminin içini rahatlatırsınız.

Başlamadan önce hatırlatmak isteriz ki, Debian için zaten paketi hazırlanmış bir programı farklı derleme seçenekleriyle derlemenin daha güvenli ve uygun yolları vardır. Eğer ne yaptığınızı tam olarak bilmiyorsanız **equivs** kullanmayınız. Ayrıntılı bilgi için bkz. [Kaynak paketlerle çalışma](#) (sayfa: 22).

MTA örneğimize kaldığımız yerden devam edelim. Yeni derlemiş olduğunuz **postfix** paketini sisteminize kurdunuz ve ardından **mutt** paketini kurmak istediniz. Fakat **mutt** paketinin başka bir MTA kurulmasını gerektirdiğini gördünüz ama zaten sisteminizde bir MTA var!

Herhangi bir dizine geçip (örneğin `/tmp`) aşağıdaki komutu çalıştırın:

```
$ equivs-control isim
```

*isim* değerini oluşturmak istediğiniz **control** dosyasının adı ile değiştiriniz. Ardından dosya aşağıdaki şekilde oluşturulacaktır:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: <enter package name; defaults to equivs-dummy>
Version: <enter version here; defaults to 1.0>
Maintainer: <your name and email address; defaults to username>
Pre-Depends: <packages>
Depends: <packages>
Recommends: <packages>
Suggests: <package>
Provides: <(virtual)package>
Architecture: all
Copyright: <copyright file; defaults to GPL2>
Changelog: <changelog file; defaults to a generic changelog>
Readme: <README.Debian file; defaults to a generic one>
Extra-Files: <additional files for the doc directory, commaseperated>
Description: <short description; defaults to some wise words>
```



```
long description and info
.
second paragraph
```

Bu dosyayı kendi isteğimize göre değiştirebiliriz. Tüm alanları açıklamaya şimdilik burada gerek yok, hemen amacımızı gerçekleştirelim:

```
Section: misc
Priority: optional
Standards-Version: 3.0.1

Package: mta-local
Provides: mail-transport-agent
```

Evet, hepsi bundan ibaret. `mutt` paketi sistemde `mail-transport-agent` paketi olmasını gerektirir. Bu sanal bir pakettir ve tüm MTA'lar tarafından sağlanır. Aynı davranış için paket ismi olarak doğrudan `mail-transport-agent` seçebilirdik, ancak sanal paket kavramını ve `Provides` alanını kullanarak bunu gerçekleştirdik.

Şimdi yapmanız gereken paketi oluşturmaktır:

```
# equivs-build paket_ismi
dh_testdir
touch build-stamp
dh_testdir
dh_testroot
dh_clean -k
# Add here commands to install the package into debian/tmp.
touch install-stamp
dh_testdir
dh_testroot
dh_installdocs
dh_installchangelogs
dh_compress
dh_fixperms
dh_installdeb
dh_gencontrol
dh_md5sums
dh_builddeb
dpkg-deb: building package 'paket_ismi' in '../paket_ismi_1.0_all.deb'.

The package has been created.
Attention, the package has been created in the current directory,
```

Ve sonra üretilen `.deb` paketini sisteminize kurmalısınız.

Gördüğünüz gibi `equivs` programının kullanılabileceği pek çok durum vardır. Örneğin `favorilerim` adlı bir paket oluşturabilir ve bu paketin sıklıkla kullandığınız paketlere bağımlı olmasını sağlayabilirsiniz (`Depends` alanı ile). Ardından bu paketi sisteminize kurup, bağımlılık verdiğiniz diğer tüm paketlerin de kurulmasını sağlamış olursunuz.

Örnek `control` dosyaları için `/usr/share/doc/equivs/examples` dizini altına bakabilirsiniz.

## 4.2. Kullanılmayan yerelleştirme dosyalarını kaldırma

### *localepurge*

Çoğu Debian kullanıcısı sadece bir tip yerel dosyası kullanır. Örneğin bir Türk genellikle `tr_TR` yerelini kullanır ve hemen hiç bir zaman `es` kullanma ihtiyacı hissetmez.

`localepurge` paketi bu kullanıcılar için oldukça faydalıdır. Kullanmadığınız yerellerin sisteminizden kaldırılmasını ve yenilerinin de kurulmamasını sağlayarak disk üzerinde size yer kazandırır. Yapmanız gerekeni biliyorsunuz: `apt-get install localepurge`.

Paketin ayarlamaları oldukça kolaydır, `debconf` tarafından ilgili sorular size yöneltilecektir. Ancak ilk soruya cevap verirken dikkatli olun, yanlış bir seçim sonucu kullandığınız yerel dosyalarının silinmesine neden olabilirsiniz. Bu durumda tek çözüm, ilgili yerel dosyalarını içeren paketleri sisteminize yeniden kurmak olacaktır.

### 4.3. Güncellenebilir paketleri nasıl öğrenebilirim?

`apt-show-versions` programı sisteminizde güncellenmeye müsait olan paketler hakkında faydalı bilgiler sunar. `-u` seçeneği ile güncellenebilir paketlerin listesini alabilirsiniz:

```
$ apt-show-versions -u
libeel0/unstable upgradeable from 1.0.2-5 to 1.0.2-7
libeel-data/unstable upgradeable from 1.0.2-5 to 1.0.2-7
```

## 5. Paketler hakkında bilgi toplama

Paketler hakkında bilgi toplamak için APT sisteminin üzerinde çalışan programlar bulunmaktadır. Bu bilgiler sisteme kurulabilen paket listesi, yüklenmiş paketlerin listesi, paketin hangi bölüme ait olduğu, önceliği ve açıklaması vb. olabilir.

Fakat burada amacımız APT ile aynı bilgilere nasıl erişebileceğinizi sizlere anlatmaktır. Mesela, kurmak istediğiniz programın paket adını nasıl öğrenebiliriz?

Bu amaç için birkaç yardımcı araca sahibiz. Önce `apt-cache` ile başlayalım. Bu program APT sistemi tarafından kendi özel veritabanını yönetmede kullanılır. Şimdi pratik kullanımda uygulanmasına geçelim.

### 5.1. Paket isimlerini keşfetme

Örneğin, eski iyi günleri hatırladınız, Atari 2600 günlerinizi... Ve bir Atari emülator programı kurmak için APT'yi kullanmak istiyorsunuz ancak kurmanız gereken paketin ismini bilmiyorsunuz. Yapmanız gereken:

```
# apt-cache search atari
atari-fdisk-cross - Partition editor for Atari (running on non-Atari)
circuslinux - The clowns are trying to pop balloons to score points!
madbomber - A Kaboom! clone
tcs - Character set translator.
atari800 - Atari emulator for svgalib/X/curses
stella - Atari 2600 Emulator for X windows
xmss-x - X binaries for Multi-Emulator Super System
```

Gördüğünüz gibi arama sonucumuzda birkaç paket kısa açıklamalarıyla birlikte listelendi. Herhangi biri hakkında daha ayrıntılı bilgi almak istersek kullanacağımız komut:

```
# apt-cache show stella
Package: stella
Priority: extra
Section: non-free/otherosfs
Installed-Size: 830
Maintainer: Tom Lear <tom@trap.mtview.ca.us>
Architecture: i386
```

```

Version: 1.1-2
Depends: libc6 (>= 2.1), libstdc++2.10, xlib6g (>= 3.3.5-1)
Filename: dists/potato/non-free/binary-i386/otherosfs/stella_1.1-2.deb
Size: 483430
MD5sum: 11b3e86a41a60falc4b334dd96c1d4b5
Description: Atari 2600 Emulator for X windows
  Stella is a portable emulator of the old Atari 2600 video-game console
  written in C++. You can play most Atari 2600 games with it. The latest
  news, code and binaries for Stella can be found at:
  http://www4.ncsu.edu/~bwmott/2600

```

Bu defaki ekran çıktısında paket hakkında daha ayrıntılı bilgi var ve bu bilgiler doğrultusunda paketi kurmak isteyip istemediğimize karar verebiliriz. Eğer paket sisteminizde kurulu fakat yeni bir sürümü mevcut ise, her iki sürüm hakkındaki bilgiler listelenecektir.

```

# apt-cache show lilo
Package: lilo
Priority: important
Section: base
Installed-Size: 271
Maintainer: Russell Coker <russell@coker.com.au>
Architecture: i386
Version: 1:21.7-3
Depends: libc6 (>= 2.2.1-2), debconf (>=0.2.26), logrotate
Suggests: lilo-doc
Conflicts: manpages (<<1.29-3)
Filename: pool/main/l/lilo/lilo_21.7-3_i386.deb
Size: 143052
MD5sum: 63fe29b5317fe34ed8ec3ae955f8270e
Description: LInux LOader - The Classic OS loader can load Linux and others
  This Package contains lilo (the installer) and boot-record-images to
  install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
.
  You can use Lilo to manage your Master Boot Record (with a simple text screen)
  or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

Package: lilo
Status: install ok installed
Priority: important
Section: base
Installed-Size: 190
Maintainer: Vincent Renardias <vincent@debian.org>
Version: 1:21.4.3-2
Depends: libc6 (>= 2.1.2)
Recommends: mbr
Suggests: lilo-doc
Description: LInux LOader - The Classic OS loader can load Linux and others
  This Package contains lilo (the installer) and boot-record-images to
  install Linux, OS/2, DOS and generic Boot Sectors of other OSes.
.
  You can use Lilo to manage your Master Boot Record (with a simple text screen)
  or call Lilo from other Boot-Loaders to jump-start the Linux kernel.

```

Burada ilk sırada yeni sürüme ait bilgiler, ikinci sırada ise zaten sisteminizde kurulu olan sürüme ait bilgiler listelenir. Bir paket hakkında daha genel bilgi edinmek için aşağıdaki komutu kullanabilirsiniz:

```
# apt-cache showpkg penguin-command
```

```

Package: penguin-command
Versions:
1.4.5-1 (/var/lib/apt/lists/download.sourceforge.net_debian_dists_unstable_main_binary-i

Reverse Depends:
Dependencies:
1.4.5-1 - libc6 (2 2.2.1-2) libpng2 (0 (null)) libSDL-mixer1.1 (2 1.1.0) libSDL1.1 (0 (
Provides:
1.4.5-1 -
Reverse Provides:

```

Bir paketin bağımlı olduğu paketlerin listesini almak için:

```

# apt-cache depends penguin-command
penguin-command
  Depends: libc6
  Depends: libpng2
  Depends: libSDL-mixer1.1
  Depends: libSDL1.1
  Depends: zlib1g

```

Özetle, aradığımız paket adını bulmak için bir kaç silaha sahibiz.

## 5.2. Paket adlarını bulmak için dpkg kullanma

Paket adını bulmada diğer bir yöntem de o paket tarafından kullanıldığını bildiğiniz bir dosyadan hareket etmektir. Örneğin belirli bir ".h" dosyasını içerdiğini bildiğiniz paketin ismini aşağıdaki gibi bulabilirsiniz:

```

# dpkg -S stdio.h
libc6-dev: /usr/include/stdio.h
libc6-dev: /usr/include/bits/stdio.h
perl: /usr/lib/perl/5.6.0/CORE/nostdio.h

```

veya:

```

# dpkg -S /usr/include/stdio.h
libc6-dev: /usr/include/stdio.h

```

Eğer sisteminize kurulu bir paketin tam ismini öğrenmek isterseniz aşağıdaki komut işinizi görecektir:

```

# dpkg -l | grep mozilla
ii  mozilla-browser 0.9.6-7          Mozilla Web Browser

```

Bu komut paket isimlerini belirli bir uzunluktan sonra kesmektedir. Yukarıdaki örnekte paketin tam ismi `mozilla-browser`'dir. Bu problemi gidermek için `COLUMNS` çevresel değişkenini aşağıdaki gibi ayarlamalısınız:

```

[kov]@[couve] $ COLUMNS=132 dpkg -l | grep mozilla
ii  mozilla-browser 0.9.6-7          Mozilla Web Browser
-  core and browser

```

veya açıklamasından yola çıkarak paket adını elde edebilirsiniz:

```

# apt-cache search "Mozilla Web Browser"
mozilla-browser - Mozilla Web Browser

```

## 5.3. Programları anında kurma

Bir program derliyorsunuz ve aniden bir hata oluştu! Çünkü program sisteminizde olmayan bir `.h` dosyasına ihtiyaç duyuyor. **auto-apt** programı sizi bu senaryolardan kurtaracaktır. Gerektiğinde ilgili paketi kurmak isteyip istemediğinizi soracak, derleme işlemini durduracak ve programı çekip kurduktan sonra derleme işleminin devam etmesini sağlayacaktır.

Peki nasıl yapacaksınız:

```
# auto-apt run komut
```

Burada *komut* yerine derleme için kullanacağınız komutu yazmalısınız.

```
# auto-apt run ./configure
```

Ardından gerekli paketleri kurmak isteyip istemediğinizi soracak, **apt-get**'i çağırarak kurulumları otomatik olarak gerçekleştirecektir. Eğer bir X sunucusu üzerinde çalışıyorsanız, varsayılan metin arayüzü grafik bir arayüzle otomatik olarak değiştirilecektir.

**auto-apt** programı verimli çalışabilmek için kendi özel veritabanını tutar. Bu veritabanının güncel tutulması programın beklenen davranışı gösterebilmesi açısından önemlidir. Veritabanını güncellemek için **auto-apt update**, **auto-apt updatedb** ve **auto-apt update-local** komutlarını çalıştırmalısınız.

## 5.4. Bir dosyanın hangi pakete ait olduğunu bulma

Bir paketi kurdunuz ancak **apt-cache** ile nasıl çağıracağınızı bilmiyorsunuz. Fakat programın dosya adını biliyorsunuz veya paketten çıkan başka herhangi bir dosyanın adını bilmektesiniz. Bu durumda **apt-file** programı ile paket adını bulabilirsiniz:

```
$ apt-file search dosya_adı
```

Bu komut aynı **dpkg -S** gibi çalışır ancak ayrıca verdiğiniz dosyayı içermesine rağmen sisteminizden sonradan kaldırılmış olan paketleri de listeler. Bu komut ayrıca bir derleme sırasında ihtiyaç duyulan dosyaları içeren paketleri bulmak için de kullanılabilir. Fakat bu tür sorunlar için **auto-apt** kullanmak daha iyi bir yöntemdir. Detaylar için bkz. [Programları anında kurma](#) (sayfa: 20).

Bir paketin içeriğini aşağıdaki komutla listeleyebilirsiniz:

```
$ apt-file list paket_adı
```

**apt-file** tüm paketler hakkında içerdikleri dosyalara ilişkin kayıtları tutar. Aynı **auto-apt**'de olduğu gibi bu veritabanının da zamanla güncellenmesi gereklidir:

```
# apt-file update
```

Öntanımlı olarak **apt-file** **auto-apt** ile aynı kaynağı kullanır, bkz. [Programları anında kurma](#) (sayfa: 20).

## 5.5. Paketlerdeki değişikliklerden haberdar olma

Sisteminize kurulan her paket kendi belge dizini altına (`/usr/share/doc/paket_adı`) `changelog.Debian.gz` adlı bir dosya atar. Bu dosyada ilk sürümden son sürüme kadar paket üzerinde yapılan değişiklikler belirtilir. Bu dosyayı **zless** ile okuyabilirsiniz, ancak tüm paketleri bu şekilde takip etmek epey güç bir uğraştır.

Bu işi otomatik hale getirmek için yardımcı bir araç mevcuttur: **apt-listchanges**. Programı kullanmak için önce **apt-listchanges** paketini sisteminize kurmalısınız. Kurulum esnasında **Debconf** tarafından sorulan soruları istediğiniz gibi yanıtlayabilirsiniz.

İlk soru `apt-listchanges`'in paketlerdeki değişiklikleri nasıl göstereceğine karar vermeniz gereken kısımdır. Değişikliklerin kendinize elektronik posta olarak iletilmesini seçebilirsiniz, otomatik güncellemelerde bu faydalı olabilir veya bir `less` gibi bir sayfalama aracı ile güncellemelere devam etmeden önce değişikliklere bir göz atmak isteyebilirsiniz. Eğer `apt-listchanges`'in güncellemeler sırasında otomatik olarak çalışmasını istemiyorsanız `none` seçeneğini seçin.

`apt-listchanges` yüklendikten sonra paketler apt ile indirildikten sonra (veya bir CD veya bağlanmış aygıttan alındıktan sonra), paketleri yüklemekten önce bu paketlerde yapılmış değişiklikleri listeleyecektir.

## 6. Kaynak paketlerle çalışma

### 6.1. Kaynak paketleri indirme

Özgür yazılım dünyasında kaynak kodlar üzerinde çalışma, hata bulma ve düzeltme oldukça yaygındır. Bunu yapabilmek için programın kaynak kodlarını indirmeniz gereklidir. APT sistemi, dağıtım içerisinde yer alan paketlerin kaynak kodlarını, `.deb` paketi oluşturulabilmesi için yapılan değişiklikleri de içerecek şekilde indirmenizi sağlayacak komutlara sahiptir.

Debian kaynak paketlerinin kullanılmasının gerektiği diğer bir durum ise, belirli bir paketin güncel sürümünü, `unstable` sürümünü örneğin, `stable` sürümü için yeniden hazırlamak istediğiniz durumdur. Bir paketi `stable` dağıtım için oluşturduğunuzda bağımlılıklar da `stable` dağıtıma göre düzenlenir.

Bunun için `/etc/apt/sources.list` dosyanızda `unstable` sürümle ilgili `deb-src` satırlarının bulunması gereklidir. Ayrıntı için bkz. [/etc/apt/sources.list dosyası](#) (sayfa: 4).

Bir kaynak paketi indirmek için aşağıdaki komutu kullanınız:

```
# apt-get source paket_adı
```

Bu komut üç dosyanın indirilmesini sağlayacaktır: bir `.orig.tar.gz`, bir `.dsc` ve bir de `.diff.gz`. Eğer paket Debian'a özgü ise, son iki dosya indirilmez ve dosya adı büyük olasılıkla `orig` önekini içermez.

`.dsc` dosyası `dpkg-source` tarafından kaynak paketi `paket_adı-sürüm` dizini altına açmak için kullanılır. İndirilen her bir debian kaynak paketi açıldıktan sonra, `debian` adında bir dizin içerir. Bu dizinde `.deb` paketini oluşturmak için yapılan tüm değişiklikler bulunmaktadır.

Kaynak paket indirildikten hemen sonra `.deb` paketinin oluşturulmasını istiyorsanız `auto-build` komutu işinize yarayacaktır. Bunun için komutu `-b` seçeneği ile aşağıdaki gibi kullanmalısınız:

```
# apt-get -b source paket_adı
```

Eğer indirdikten hemen sonra yerine daha ileriki zamanlarda `.deb` paketini oluşturmak isterseniz aşağıdaki komut bunu yapabilirsiniz:

```
# dpkg-buildpackage -rfakeroot -uc -b
```

Bu komutu kaynak paketin açıldığı dizin içerisindeyken çalıştırmalısınız. İşlem sonunda üretilen paketi aşağıdaki gibi sisteminize kurabilirsiniz:

```
# dpkg -i paket.deb
```

`apt-get`'in `source` yöntemi ile diğerleri arasında birtakım farklar bulunmaktadır. Burada paketin indirilebilmesi için root kullanıcısı olunmasına gerek yoktur, indirilen paketler o an bulunulan dizin içerisine açılırlar.

### 6.2. Kaynak paketleri derlemek için gerekli paketler

Doğal olarak derleyeceğiniz programın ihtiyaç duyduğu kütüphanelerin sisteminizde bulunması zorunludur. Tüm kaynak paketler `debian/control` dosyalarında `Build-Depends` : adlı bir alana sahiptir, bu alanda paketin kaynak kodundan derlenebilmesi için ihtiyaç duyduğu paketler belirtilir.

APT bu paketleri kolayca indirmek ve kurmak için gerekli komuta sahiptir. `apt-get build-dep paket_adı` komutu ile derlemek istediğiniz paketin `Build-Depends` alanında belirtilen paketlerin sisteminize kurulması sağlanır.

```
# apt-get build-dep gmc
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  comerr-dev e2fslibs-dev gdk-imlib-dev imlib-progs libgnome-dev libgnorba-dev
  libgpmgl-dev
0 packages upgraded, 7 newly installed, 0 to remove and 1 not upgraded.
Need to get 1069kB of archives. After unpacking 3514kB will be used.
Do you want to continue? [Y/n]
```

Burada `gmc` paketinin kaynak halinden derlenebilmesi için gerekli paketler sisteme kurulmaktadır. Bu komutun kaynak paketi indirmediğini unutmayın, kaynak paketi `apt-get source` ile ayrıca indirmelisiniz.

Eğer amacınız belirli bir kaynak paketi derleyebilmek için gereken paketlerin listesini almak ise bunun için `apt-cache show` türevi bir komut kullanılır, bkz. [Paketler hakkında bilgi toplama](#) (sayfa: 18). Bu araç `Build-Depends` satırında yazan bilgiler yanında başka bilgiler de göstermektedir.

```
# apt-cache showsrc paket_adı
```

## 7. Hatalarla başa çıkma

### 7.1. Genel hatalar

Hatalı durumlar her zaman olabilir, hataların çoğu basit bir şekilde düzeltilebilecek türdendir. Aşağıda sıklıkla karşılaşılan hatalar ve çözüm yöntemlerinin bir listesi verilmiştir:

- Eğer `apt-get install paket_adı` komutunun ardından aşağıdaki gibi bir çıktı aldıysanız:

```
Reading Package Lists... Done
Building Dependency Tree... Done
W: Couldn't stat source package list 'http://people.debian.org unstable/ Packages'
W: You may want to run apt-get update to correct these missing files
E: Couldn't find package penguineyes
```

`/etc/apt/sources.list` dosyasında yaptığınız son değişiklikten sonra `apt-get update` komutunu çalıştırmayı unutmuşsunuz demektir.

- Eğer hata aşağıdakine benzer ise:

```
E: Could not open lock file /var/lib/dpkg/lock - open (13 Permission denied)
E: Unable to lock the administration directory (/var/lib/dpkg/), are you root?
```

`apt-get`'in `source` dışında bir yöntemini normal kullanıcı haklarıyla kullanmaya çalışıyorsunuzdur.

Benzer bir hatayı, aynı anda birden fazla `apt-get` programını çalıştırmamız durumunda ya da `dpkg` programı işlem yaparken APT kullanmaya çalışmamız durumunda alabilirsiniz. Aynı anda kullanılabilecek tek yöntem `source`'tır.

- Eğer kurulum işleminiz herhangi bir nedenle yarıda kesilmiş ve ardından hiç bir paketi kurma ve kaldırma işlemi yapamaz duruma gelmişseniz, aşağıdaki komutlarla APT'nin gerekli düzeltmeleri yapmasını sağlayabilirsiniz:

```
# apt-get -f install
# dpkg --configure -a
```

Ve tekrar deneyin. Bazen yukarıdaki komutlardan ikincisini birden fazla defa tekrarlamanız gerekebilir. `unstable` dağıtımı kullanmaya başlayacak olan kullanıcıların öğrenmesi gereken ilk ders budur.

- Eğer **apt-get update** kullanımı sırasında "E: Dynamic MMap ran out of room" şeklinde bir hata ile karşılaşılırsanız `/etc/apt/apt.conf` dosyasına şu satırı ekleyin:

```
APT::Cache-Limit 10000000;
```

## 7.2. Nereden yardım bulabilirim?

Debian paket yönetim sistemi hakkındaki ayrıntılı belgeleri inceleyebilirsiniz. Programların `--help` ve kılavuz (man) sayfaları da oldukça yararlı bilgiler içerir. Ayrıca `/usr/share/doc` dizini altındaki her programa ait belgeleri inceleyebilirsiniz, APT için bu dizin `/usr/share/doc/apt`'dir.

Eğer aradığınız sorunun cevabını bu belgelerde de bulamazsanız cevabı Debian eposta listelerinde aramalısınız. Debian listeleri hakkında ayrıntılı bilgilere <http://www.debian.org/> adresinden ulaşabilirsiniz.

Belirttiğimiz listeler Debian kullanıcıları içindir, diğer dağıtımları kullananlar, dağıtımlarının eposta listelerinde kendilerine uygun cevapları bulabilirler.

## 8. Hangi Linux dağıtımları APT destekliyor?

Aşağıda APT'yi destekleyen dağıtımların bir listesi yer almaktadır:

- Debian GNU/Linux (<http://www.debian.org/>) – APT'nin geliştiriminin yapıldığı ve devam ettiği dağıtımdır.
- Conectiva (<http://www.conectiva.com.br/>) – APT sisteminin RPM ile kullanılabilmesi için çalışan ilk dağıtımdır.
- Libranet (<http://www.libranet.com/>)
- Mandrake (<http://www.mandrake.com/>)
- PLD (<http://www.pld.org.pl/>)
- Vine (<http://www.vinelinux.org/>)
- APT4RPM (<http://apt4rpm.sf.net/>)
- Alt Linux (<http://www.altlinux.ru/>)
- Red Hat (<http://www.redhat.com/>)
- Sun Solaris (<http://www.sun.com/>)
- SuSE (<http://www.suse.de/>)
- Yellow Dog Linux (<http://www.yellowdoglinux.com/>)

## 9. Teşekkürler

Debian-BR ve Debian projesindeki arkadaşlarıma, desteklerini esirgemedikleri için teşekkürlerimi sunuyorum. Ayrıca projemize ve benzer özgür projelere destek veren CIPSGA'ya teşekkür ediyorum.



Ve özel teşekkürler:

Yooseong Yang <yooseong (at) debian.org>

Michael Bramer <grisu (at) debian.org>

Bryan Stillwell <bryan (at) bokeoa.com>

Pawel Tecza <pawel.tecza (at) poczta.fm>

Hugo Mora <h.mora (at) melix.com.mx>

Luca Monducci <luca.mo (at) tiscali.it>

Tomohiro KUBOTA <kubota (at) debian.org>

Pablo Lorenzoni <spectra (at) debian.org>

Steve Langasek <vorlon (at) netexpress.net>

Arnaldo Carvalho de Melo <acme (at) conectiva.com.br>

Erik Rossen <rossen (at) freesurf.ch>

Ross Boylan <RossBoylan (at) stanfordalumni.org>

Matt Kraai <kraai (at) debian.org>

Aaron M. Ucko <ucko (at) debian.org>

Jon Åslund <d98-jas (at) nada.kth.se>

## 10. Bu belgenin yeni sürümleri

Bu belge [Debian-BR<sup>\(B36\)</sup>](#) projesi kapsamında üretilmiş olup, herkesin Debian kullanımına yardımcı olmak amacıyla oluşturulmuştur.

Belgenin yeni sürümlerine her zaman Debian Belgelendirme Projesi sayfalarından erişebilirsiniz: [http://www.debian.org/doc/ddp<sup>\(B37\)</sup>](http://www.debian.org/doc/ddp<sup>(B37)</sup>).

Her türlü yorum ve görüşlerinizi <kov (at) debian.org> adresinden bana gönderebilirsiniz.

## Notlar

Belge içinde dipnotlar ve dış bağlantılar varsa, bunlarla ilgili bilgiler bulundukları sayfanın sonunda dipnot olarak verilmeyip, hepsi toplu olarak burada listelenmiş olacaktır.

<sup>(B2)</sup> [../howto/gpl.pdf](#)

<sup>(B36)</sup> <http://www.debian-br.org/>

<sup>(B37)</sup> <http://www.debian.org/doc/ddp>

Bu dosya (apt-howto.pdf), belgenin XML biçiminin T<sub>E</sub>XLive ve belgeler-xsl paketlerindeki araçlar kullanılarak PDF biçimine dönüştürülmesiyle elde edilmiştir.

22 Ocak 2007