

İSİM

rsync – rcp yerine daha hızlı ve esnek bir araç

KULLANIM

rsync [*seçenek*] ... *kaynak* [*kaynak*] ... [*kullanıcı@*] *konak:hedef*

rsync [*seçenek*] ... [*kullanıcı@*] *konak:kaynak hedef*

rsync [*seçenek*] ... *kaynak* [*kaynak*] ... *hedef*

rsync [*seçenek*] ... [*kullanıcı@*] *konak::kaynak* [*hedef*]

rsync [*seçenek*] ... *kaynak* [*kaynak*] ... [*kullanıcı@*] *konak::hedef*

rsync [*seçenek*] ... **rsync**:// [*kullanıcı@*] *konak[:port]/kaynak* [*hedef*]

AÇIKLAMA

rsync çoğunlukla **rcp** gibi davranan bir uygulamadır, ancak daha fazla seçenek içerir ve özellikle hedefteki dosyanın mevcut olması halinde dosya aktarımını çok daha hızlı gerçekleştiren **rsync** uzak güncelleme protokolünü kullanır.

rsync uzak güncelleme protokolü **rsync**'in ağ bağlantısı üzerinden sadece iki dosya grubu arasındaki farkları aktarmasını mümkün kılar. Bunu bu pakete eşlik eden teknik raporda açıklanan verimli bir toplama sağlaması arama algoritmasını kullanarak yapar.

rsync'e özgü bazı ek özellikler:

- bağların, aygıtların, sahip, grup ve izinlerin kopyalanmasını destekler
- GNU **tar**'a benzer exclude ve exclude-from seçenekleri
- CVS'nin yok sayacağı dosyaların yoksayılmasını sağlayan CVS dışlama (exclude) kipi
- **rsh**, **ssh** gibi uzak şeffaf kabukları kullanabilir
- root kullanıcısının yetkilerine ihtiyaç duymaz
- görünmeyen maliyetleri asgari düzeyde tutmak için dosya aktarımlarında borulama yapar
- hem anonim hem de kimlik doğrulamalı **rsync** sunucu desteği (yansılama için ideal)

GENEL

rsync sekiz farklı yöntemle kullanılabilir:

- yerel dosyaları kopyalamak için. Bu çağrıda ne kaynak ne de hedef dosya yolu bir : ayracı içermez.
- yerel makineden uzaktaki bir makineye bir uzak kabuk uygulaması (**ssh** veya **rsh** gibi) kullanarak dosyaları kopyalamak için. Bu çağrıda sadece hedef dosya yolu tek bir : ayracı içerir.
- uzaktaki bir makineden yerel makineye bir uzak kabuk uygulaması (**ssh** veya **rsh** gibi) kullanarak dosyaları kopyalamak için. Bu çağrıda sadece kaynak dosya yolu bir : ayracı içerir.
- uzaktaki bir **rsync** sunucusundan yerel makineye kopyalama yapmak için. Bu çağrıda sadece kaynak dosya yolu bir :: ayracı veya bir **rsync**:// URL'si içerir.

- yerel makineden uzaktaki bir **rsync** sunucusuna kopyalama yapmak için. Bu çağrıda sadece hedef dosya yolu bir `::` ayracı veya bir `rsync::` URL'si içerir.
- uzaktaki makinede bulunan **rsync** sunucusunu ve bir uzak kabuk uygulamasını kullanarak, uzaktaki makineden yerel makineye kopyalama yapmak için. Bu çağrıda kaynak dosya yolunun bir `::` ayracı içermesi yanında `--rsh=komut` (ya da `-e komut`) seçeneği de kullanılır.
- uzaktaki makinede bulunan **rsync** sunucusunu ve bir uzak kabuk uygulamasını kullanarak, uzaktaki makineye yerel makineden kopyalama yapmak için. Bu çağrıda hedef dosya yolunun bir `::` ayracı içermesi yanında `--rsh=komut` (ya da `-e komut`) seçeneği de kullanılır.
- uzaktaki makinede bulunan dosyaların listesini almak için. Bu işlem uzaktaki makineden kopyalama işlemindeki gibi ancak yerel hedefi belirtmeden yapılır.

Tüm durumlarda (burada listelenenler dışında kalanlar dahil) kaynak ve hedef belirtimlerinden biri daima yerel olmalıdır (yani **rsync** ile iki uzak makine arasında işlem yapamazsınız).

AYARLAR

Kurulum işlemleri için **README** dosyasına bakınız.

Kurduktan sonra, bir uzak kabuk üzerinden eriştiğiniz (veya **rsync** artalan kipi protokolü kullanarak eriştiğiniz) her makinede **rsync** kullanabilirsiniz. Uzak dosya aktarımları için, günümüzde **rsync** iletişimde **ssh** kullanmasına rağmen öntanımlı olarak kullanacağı uzak kabuk **rsh** veya **remsh** gibi bir kabuk olarak da ayarlanabilir.

Bundan başka `-e` komut satırı seçeneğinin argümanı olarak veya **RSYNC_RSH** ortam değişkenine atayarak da istediğiniz bir uzak kabuk uygulamasını belirtebilirsiniz.

Yüksek derecede güvenlik istendiğinde ortak kanaat **ssh** kullanmaktır.

Yalnız, **rsync**'in hem yerel hem de uzak makine üzerinde kurulu olması gerektiğini unutmayın.

KULLANIMI

rsync'i **rcp**'yi nasıl kullanıyorsanız öyle kullanabilirsiniz. Biri uzak makine olmak üzere bir kaynak ve bir hedef belirtmelisiniz.

Şüphesiz sözdizimini açıklamanın en kolay yolu bir örnek vermektir:

```
rsync -t *.c foo:src/
```

Bu komut çalışma dizinindeki `*.c` kalıbı ile eşleşen tüm dosyaları `foo` makinesinin `src` dizinine kopyalayacaktır. Bu dosyalardan uzak makinede mevcut olanları için **rsync** uzak güncelleme protokolü kullanılarak sadece farklar gönderilir. Bu konuda daha ayrıntılı bilgi için teknik rapora bakınız.

```
rsync -avz foo:src/bar /data/tmp
```

Bu komut, `foo` makinesindeki `src/bar` dizinindeki tüm dosyaları varsa alt dizinlerdeki de içererek yerel makinenin `/data/tmp` dizinine kopyalar. Dosyaların, sembolik bağların, aygıtların, özniteliklerinin, izinlerinin, sahiplerinin, v.s.sinin korunmasını sağlayan "arşiv" kipinde aktarım yapılır. Buna ek olarak, aktarım işleminin verileri içeren kısımlarının boyutunu küçültmek için sıkıştırma uygulanır.

```
rsync -avz foo:src/bar/ /data/tmp
```

Kaynak yolunun sonuna eklenen / işareti hedefte ek bir dizin oluşturmaktan kaçınılması davranışını değiştirir. Sona eklenen bu / işaretinin "dizini ismiyle kopyala" davranışının zıddı olarak "bu dizinin içeriğini kopyala" anlamına geldiğini düşünebilirsiniz, fakat her iki durumda da dizinin öznelikleri hedefteki dizine kopyalanacaktır. Başka bir deyişle, aşağıdaki komutların her biri ile dosyalar aynı şekilde, `/dest/foo` dizinin öznelikleri ayarlanarak kopyalanır:

```
rsync -av /src/foo /dest
rsync -av /src/foo/ /dest/foo
```

Ayrıca, **rsync**'i sadece yerel kipte de kullanabilirsiniz. Bu durumda hedef ve kaynak dosya yollarının ikisi de isimlerinde ':' içermeyecektir. Kopyalama işlemi güçlendirilmiş bir kopyalama komutu verilmiş gibi yürüyecektir.

```
rsync somehost.mydomain.com::
```

Bu komut, `somehost.mydomain.com` makinesindeki kullanılabilecek tüm anonim **rsync** modüllerini listeleyecektir. (Daha fazla bilgi için aşağıdaki bölümlere bakınız.)

İLERİ DÜZEY KULLANIM

Bir uzak makineden çok sayıda dosya için istek yaparken aralarında boşluk bırakılmış kaynak dosya yollarını tırnak içine alarak belirtebilirsiniz. Bazı örnekler:

```
rsync makine::'modülismi/dizin1/dosya1 modülismi/dizin2/dosya2' /hedef
```

Bu komut, bir **rsync** sunucusundan `/hedef` dizinine `dosya1` ve `dosya2` dosyalarını kopyalayacaktır. Her argüman aynı "modülismi/" önekini içermeli ve arada sadece bir boşluk karakteri bulunmalıdır. Diğer tüm boşlukların dosya isimlerine ekleneceği varsayılır.

```
rsync -av makine:'dizin1/dosya1 dizin2/dosya2' /hedef
```

Bu komut `dosya1` ve `dosya2` dosyalarını `/hedef` dizinine kopyalayacaktır. Sözcük ayırma işlemi uzak kabuk tarafından yapılır, yani eğer sözcük ayırma gerçekleşmemişse, bunun sebebi uzak kabuğun bu işlem için yapılandırılmamış olmasıdır (nedendir bilinmez, çok yaygındır). İsmi boşluk içeren bir dosyayı kopyalamak isterseniz, ya bunu uzak kabuğun anlayabileceği bir yöntemle yapmalı ya da boşlukların yerine dosya ismi kalıpları kullanmalısınız. İki örnek:

```
rsync -av makine:'isminde\ boşluk\ olan\ dosya' /hedef
rsync -av makine:isminde?boşluk?olan?dosya /hedef
```

İkinci örnekte, eşleşmeyen kalıpları kabuğunuzun anlayacağı varsayılmıştır. Bu takdirde bunları tersbölü öncelemeli hale getirecektir.

BİR RSYNC SUNUCUSUNA BAĞLANMAK

Dosya aktarımı için **rsync**'i bir uzak kabuk olmaksızın da kullanmak mümkündür. Bu durumda, TCP 873. port üzerinde çalışan bir uzak **rsync** sunucusuna bağlanacaksınız.

RSYNC_PROXY ortam değişkenine bir makine:port çifti belirterek bir vekil sunucu üzerinden de bağlantı sağlayabilirsiniz. Burada, vekil sunucunuzun yapılandırmasının 873. port üzerinden bağlantıya izin verdiğinden emin olmalısınız.

rsync'i bu yöntemle kullanmanın bir uzak kabuk üzerinden kullanmaya farkları şunlardır:

- ya makine ismini dosya yolundan ayırmak için tek bir : yerine :: kullanırsınız ya da bir `rsync://` URL biçimini kullanırsınız.

- uzak sunucu bağlandığınızda günün iletisini basabilir.
- uzak sunucu için bir dosya yolu belirtmezseniz, sunucu üzerinde erişilebilir olan dosya yollarının listesini alırsınız.
- bir yerel hedef belirtmezseniz, uzak sunucu istediğiniz dosyaların bir listesini yollar.

Uzak sunucu üzerindeki bazı dosya depoları kimlik doğrulaması gerektirebilir. Böyle bir durumda bağlanmak istediğinizde sizden parola istenecektir. Parola istemi almamak için parolanızı `RSYNC_PASSWORD` ortam değişkenine kaydedebileceğiniz gibi, `--password-file` seçeneğini de kullanabilirsiniz. Bu kullanım **rsync**'i bir betik ile kullandığınız durumlarda yararlıdır.



Uyarı

Bazı sistemlerde ortam değişkenlerini tüm kullanıcılar görebilmektedir. Bu tür sistemlerde `--password-file` seçeneğinin tercih edilmesi önerilir.

BİR RSYNC SUNUCUSUNA BİR UZAK KABUKTAN BAĞLANMAK

Bağlantı için **ssh** veya **rsh** kullanarak ve dosya aktarımını **rsync** sunucusunun yeteneklerinden yararlanarak yapmanın mümkün olması bazan kullanışlı olabilmektedir. Özellikle, bir uzak makineye **ssh** üzerinden bağlanmak istediğinizde (şifreleme için ya da güvenlik duvarından geçebilmek için) ama bunun yanında **rsync** sunucusunun yeteneklerinden de yararlanmak istediğiniz durumlarda bu yararlıdır (*BİR UZAK KABUKTAN BİR RSYNC SUNUCUSU ÇALIŞTIRMAK* (sayfa: 4) bölümüne bakınız).

Kullanıcı açısından, **rsync**'in bu yöntemle kullanılması, komut satırından uzak kabuk uygulamasının `--rsh=komut` ile belirtilmesi dışında bir **rsync** sunucusuna bağlanmaktan farksızdır. (Burada, `RSYNC_RSH` ortam değişkenine bir atama yaparak uzak kabuk uygulaması belirtmek, beklenen işlevselliği sağlamayacaktır.)

Uzak kabuk kullanıcıasını, **rsync** sunucu kullanıcıısından ayırmak için uzak kabuk komutunda `'-l kullanıcı'` seçeneğini kullanabilirsiniz:

```
rsync -av --rsh="ssh -l ssh-kullanıcısı" \
    rsync-kullanıcısı@makine::modül[/yol] yerel-yol
```

"ssh-kullanıcısı" ssh seviyesinde kullanılmalıdır; "rsync-kullanıcısı" ise uzak makinenin `rsyncd.conf` dosyasından sınanacaktır.

BİR RSYNC SUNUCUSU ÇALIŞTIRMAK

Bir **rsync** sunucusu bir yapılandırma dosyası kullanılarak yapılandırılır. Daha ayrıntılı bilgi için **rsyncd.conf (5)** kılavuz sayfasına bakınız. **rsync**'i bir uzak kabuk uygulaması üzerinden ve root olarak çalıştırmadıkça, öntanımlı olarak `/etc/rsyncd.conf` dosyası kullanılır. Aksi takdirde, uzak makinedeki çalışma dizininde (genellikle `$HOME`) bulunan `rsyncd.conf` öntanımlıdır.

BİR UZAK KABUKTAN BİR RSYNC SUNUCUSU ÇALIŞTIRMAK

rsync sunucusu yapılandırma dosyası ile ilgili tüm bilgiyi **rsyncd.conf (5)** kılavuz sayfasında bulabilirsiniz.

Uzak kullanıcı root (**chroot**, **setuid/setgid**, v.s.) olmadıkça bazı yapılandırma seçenekleri kullanılamaz olacaktır. Eğer **rsync** sunucusunu bir uzak kabuk uygulaması üzerinden kullanıyorsanız, `services` dosyasında **rsync** sunucu portunun bulunması ya da **inetd**'nin yapılandırılması gerekli değildir.

Bir tek kullanımlık ssh anahtarı dışında bir **rsync** sunucusu çalıştırmak için **rsyncd.conf (5)** kılavuz sayfasındaki bu bölüme bakınız.

ÖRNEKLER

Burada **rsync** kullanımıyla ilgili bazı örnekler bulacaksınız.

Büyük MS Word dosyaları ve eposta dizinlerinden oluşan eşimin ev dizinini yedeklemek için şöyle bir **cron** işi çalıştırıyorum:

```
sync -Cavz . arvidsjaur:backup
```

Her gece bir PPP bağlantısı üzerinden bu dizini ayrıca, kendi makinemin **arvidsjaur** dizinine kopyalıyorum.

Samba kaynak ağacını eşzamanlı tutmak için şu Makefile hedeflerini kullanıyorum:

```
get:
    rsync -avuzb --exclude '*' samba:samba/ .

put:
    rsync -Cavuzb . samba:samba/

sync: get put
```

Bu sayede bağlantının öbür ucundaki bir CVS diziniyle benimkini güncellememi sağlıyor. Uzak makinede bir takım CVS işlemleri yaptıktan sonra uzak CVS protokolü ile bunları kendi makinemde güncellemek gecikmelerden dolayı pek verimli olmuyor.

"eski" ve "yeni" ftp siteleri arasında bir dizini şu komutla yansılıyorum:

```
rsync -az -e ssh --delete ~ftp/pub/samba nimbus:"~ftp/pub/tridge"
```

Bu komut her saat başı **cron**'dan çalıştırılıyor.

ÖZETLE SEÇENEKLER

Burada **rsync**'de kullanılabilen seçeneklerin bir özetine yer verilmiştir (**rsync --help** komutu bu listeyi çıktılar). Bu seçenekler bir sonraki bölümde ayrıntılı olarak açıklanmıştır.

-v, --verbose	çıktı daha ayrıntılı olur
-q, --quiet	hatalar dışında bilgi verilmez
-c, --checksum	sadece toplam sağlamaya bakılır, zaman ve boyut değişikliklerine bakılmaz
-a, --archive	arşiv kipi; -rlptgoD (-H 'sız) ile aynı
-r, --recursive	dizinlerin alt dizinlerine inilir
-R, --relative	görelî dosya yolları kullanılır
--no-relative	--relative kapatılır
--no-implied-dirs	-R uygulanmış dizinler gönderilmez
-b, --backup	yedekleme yapılır (--suffix ve --backup-dir seçeneklerine bakın)
--backup-dir=DİZİN	sadece belirtilen dizinin içeriği yedeklenir
--suffix=SONEK	yedekleme soneki
	(--backup-dir yoksa ~ öntanımlıdır)
-u, --update	hedeftekiler daha yeniyse bu dosyalar atlanır

<code>--inplace</code>	hedef dosyaları yerinde günceller
<code>-d, --dirs</code>	altdizinlere inmeden dizinleri aktarır
<code>-l, --links</code>	sembolik bağları sembolik bağ olarak kopyalar
<code>-L, --copy-links</code>	sembolik bağ değil bağın son hedefi kopyalanır
<code>--copy-unsafe-links</code>	sadece güvensiz sembolik bağlar dönüştürülür
<code>--safe-links</code>	ağacın dışını hedefleyen sembolik bağlar yoksayılır
<code>-H, --hard-links</code>	sabit bağlar korunur
<code>-K, --keep-dirlinks</code>	sembolik bağlı dizinler alıcıda dizin yapılır
<code>-p, --perms</code>	izinler korunur
<code>-o, --owner</code>	sahip korunur (sadece root)
<code>-g, --group</code>	grup korunur
<code>-D, --devices</code>	aygıtlar korunur (sadece root)
<code>-t, --times</code>	zamanlar korunur
<code>-O, --omit-dir-times</code>	zamanlar korunurken dizinler bunun dışında tutulur
<code>-S, --sparse</code>	seyrek dosyaları en verimli şekilde elde eder
<code>-n, --dry-run</code>	işlemi yapıyormuş gibi gösterir ama yapmaz
<code>-W, --whole-file</code>	dosyaları tamamen kopyalar (rsync algoritması ile farkları değil)
<code>--no-whole-file</code>	daima arttırımlı rsync algoritması kullanılır
<code>-x, --one-file-system</code>	dosya sistemi sınırlarının dışına çıkılmaz
<code>-B, --block-size=BOYUT</code>	sabit bir blok boyunda sağlama toplamına zorlar
<code>-e, --rsh=KOMUT</code>	kullanılacak uzak kabuk uygulaması belirtilir
<code>--rsync-path=UYGULAMA</code>	uzak makinede rsync'in çalıştıracağı uygulama
<code>--existing</code>	sadece hedefte mevcut olan dosyalar güncellenir
<code>--ignore-existing</code>	hedefte mevcut olan dosyalar yoksayılır
<code>--remove-sent-files</code>	alınan dosyalar/sembolik bağlar göndericide silinir
<code>--del</code>	<code>--delete-during</code> ile aynı
<code>--delete</code>	göndericide olmayan dosyalar alıcıda silinir
<code>--delete-before</code>	alıcıda aktarım öncesi silme yapılır (öntanımlı)
<code>--delete-during</code>	alıcıda aktarım sırasında silme yapılır
<code>--delete-after</code>	alıcıda aktarım sonrası silme yapılır
<code>--delete-excluded</code>	alıcıda ayrıca dışlanmış dosyalar da silinir
<code>--ignore-errors</code>	G/Ç hataları olsa da siler

--force	dizinler boş olmasa bile silinir
--max-delete=SAYI	SAYIdan fazla dosya silinmez
--max-size=BOYUT	BOYUTTan daha büyük dosyalar aktarılmaz
--partial	kısmen aktarılmış dosyalar tutulur
--partial-dir=DİZİN	kısmen aktarılmış dosyalar DİZİNde tutulur
--delay-updates	güncellenecek tüm dosyalar sona bırakılır
--numeric-ids	kullanıcı/grup kimliklerini isimlerle eşleştirmez
--timeout=SÜRE	G/Ç zaman aşımını SÜRE saniyeye ayarlar
-I, --ignore-times	zaman ve boyut bakımından eşleşen dosyalar atlanmaz
--size-only	boyutları aynı olan dosyalar atlanır
--modify-window=SAYI	doğruluğu arttırmak için değişiklik zamanları karşılaştırılır
-T, --temp-dir=DİZİN	geçici dosyaları DİZİNde oluşturur
-y, --fuzzy	hedefte dosya yoksa benzerini bulur
--compare-dest=DİZİN	alınan dosyaları ayrıca DİZİNdekilerle karşılaştırır
--link-dest=DİZİN	değişmediyse DİZİNdeki dosyalara sabit bağ
-z, --compress	aktarım sırasında dosya verisi sıkıştırılır
-C, --cvs-exclude	dosyalar CVS'nin yaptığı gibi yoksayılır
-f, --filter=KURAL	bir dosya süzme kuralı ekler
-F	--filter='dir-merge /.rsync-filter' ve yinelemeli: --filter='- .rsync-filter' ile aynı
--exclude=KALIP	KALIP ile eşleşen dosyalar dışlanır
--exclude-from=DOSYA	dışlama kalıpları DOSYAdan okunur
--include=KALIP	KALIP ile eşleşen dosyalar dışlanmaz
--include-from=DOSYA	dışlanmayan kalıplar DOSYAdan okunur
--files-from=DOSYA	aktarılacak dosyaların isimlerinin listesi DOSYAdan okunur
-0, --from0	*from seçeneklerindeki dosya listeleri boş karakter sonlandırmalı kabul edilir
--version	sürüm numarasını basar
--port=PORT	başka bir port belirtmek için
--blocking-io	uzak kabukta engellemeli G/Ç kullanılır
--no-blocking-io	engellemeli G/Ç öntanımlı olduğunda bunu iptal eder
--stats	dosya aktarımı sırasında durum bilgileri verir
--progress	aktarım sırasında gelişimi gösterir
-P	--partial --progress ile aynı
-i, --itemize-changes	tüm güncellemeler için bir değişiklik özeti çıktılar
--log-format=BIÇİM	aktarımları belirtilen biçimde günlükler
--password-file=DOSYA	parola DOSYAdan okunur
--list-only	dosyaları kopyalamaz, sadece listeler
--bwlimit=KBPS	saniyede kBayt cinsinden G/Ç band genişliği
--write-batch=DOSYA	bir betikli güncellemeyi DOSYAya yazar

```

--read-batch=DOSYA    bir betikli güncellemeyi DOSYAdan okur
--checksum-seed=SAYI  blok/dosya sağlama toplamı tohumunu ayarlar
-4, --ipv4            IPv4 tercih edilir
-6, --ipv6            IPv6 tercih edilir
-h, --help            bu yardım iletisini gösterir

```

rsync ayrıca bir artalan süreci olarak da çalıştırılabilir. Bu durumda şu seçenekleri kabul eder:

```

--daemon              bir rsync artalan süreci olarak çalışır
--address=ADRES       belirtilen adres kullanılır
--bwlimit=KBPS        saniyede kBayt cinsinden G/Ç band genişliği
--config=DOSYA        rsyncd.conf yerine kullanılacak DOSYA
--no-detach           üst süreçten ayrık çalışmaz
--port=PORT           belirtilen PORTu dinler
-v, --verbose         ayrıntı düzeyini arttırır
-4, --ipv4            IPv4 tercih edilir
-6, --ipv6            IPv6 tercih edilir
-h, --help            bu yardım iletisini gösterir

```

AÇIKLAMALARIYLA SEÇENEKLER

rsync GNU uzun seçenekler paketini kullanır. Çoğu komut satırı seçeneği hem kısa hem de uzun seçenek içerir. Bazı seçenekler ise sadece uzun seçeneklerden oluşur. '=' işareti içeren seçeneklerdeki parametreler isteğe bağlıdır; '=' yerine boşluk kullanılabilir.

-h

--help

rsync tarafından kabul edilen seçenekleri açıklayan kısa bir yardım iletisi basar.

--version

rsync sürüm numarasını basar ve çıkar.

-v

--verbose

Bu seçenek aktarım sırasında verilen bilgi miktarını arttırır. **rsync** öntanımlı olarak sessiz çalışır. Tek **-v** belirtildiğinde, hangi dosyaların aktarıldığını gösterip sonunda bir özet basar. İki **-v** belirtildiğinde atlanan dosyaları gösterir ve sonunda daha fazla bilgi verir. İki'den fazla **-v** kullanımı sadece **rsync** hata ayıklaması yapıyorsanız anlamlıdır.

Aktarılan dosya isimlerinin çıktılanmasında **--log-format** seçeneğinin öntanımlı olarak "%n%L" ile kullanıldığı varsayılarak sadece dosyaların isimleri ve varsa sembolik bağlar hedefleri ile belirtilir. Tek **-v** seviyesinden bir ayrıntıda bir dosyanın öznitelikleri değişmişse bu gösterilmez. Değiştirilen özniteliklerin öğelere göre listelenmesi isterseniz (**--itemize-changes** seçeneğiyle ya da **--log-format** seçeneğine "%i" ekleyerek), istemcideki çıktıda değiştirilen tüm öğelerin ayrıntı düzeyi arttırılır. Daha fazla bilgi için **--log-format** seçeneğinin açıklamasına bakınız.

-q

--quiet

Bu seçenek aktarım sırasında verilen bilgi miktarını azaltır. Özellikle uzak sunucudaki bilgi iletileri baskılanır. Bu seçenek **rsync**'i **cron**'dan çalıştırırken yararlıdır.

-I

--ignore-times

Normalde **rsync** hedefte de aynı boyutta ve aynı değişiklik zamanına sahip dosyalar varsa bunları atlayacaktır. Bu seçenek bu "hızlı sinama" davranışını kapatır.

--size-only

Normalde **rsync** hedefte de aynı boyutta ve aynı değişiklik zamanına sahip dosyalar varsa bunları atlayacaktır. Bu seçenekle, dosyaların zamanlarına bakılmaksızın hedefte de aynı boyutta iseler aktarılmazlar. Zaman değerlerinin doğru tutulmadığı başka bir yansılama sistemini kullandıktan sonra **rsync**'i kullanmaya başlayacaksanız bu seçenek yararlıdır.

--modify-window=sayı

İki zaman değerini karşılaştırırken, fark *sayı*'ya eşitse **rsync** bu değerleri eşit kabul eder. Bu fark normalde 0 dır (tam eşitlik), ancak bu farkı bazı durumlarda daha büyük bir değere ayarlamak daha yararlıdır. Örneğin, MS Windows FAT dosya sistemi (zaman değerleri 2 saniyelik çözünürlüğe sahiptir) taraflardan biri olduğunda **--modify-window=1** yararlıdır (1 saniyelik farklara izin vermek).

-c

--checksum

Bu seçenek göndericiyi aktarımdan önce tüm dosyaları 128 bitlik MD4 sağlama toplamını kullanarak sinamaya zorlar. Sağlama toplamı doğrudan alıcıda sinanır ve mevcut olanlar arasında aynı sağlama toplamı, aynı boyutlu ve aynı isimli olanlar aktarılmaz. Bu seçenek aktarım işlemini oldukça yavaşlatabilir.

-a

--archive

Bu seçenek **-rlptgoD** seçeneklerine eşdeğerdir. Alt dizinlere inerek işlem yapılmasını ve hemen hemen herşeyin korunmasını istemenin en kestirme yoludur; bir şey dışında: **--files-from** belirtilmişse **-r** uygulanmayacaktır.

Çok sayıda bağlı dosyanın varlığı pahalıya mal olacağından **-a** seçeneği sabit bağları korumaz. Bunu istiyorsanız **-H** seçeneğini ayrıca belirtmelisiniz.

-r

--recursive

Bu seçenek alt dizinlerinde ardışık olarak kopyalanmasını sağlar. Ayrıca **--dirs (-d)** seçeneğine de bakınız.

-R

--relative

Görelî dosya yolları kullanılır. Yani, komut satırının gönderici kısmında belirtilen dosya yolunun son parçası değil, dosya yolunun tamamı kopyalanır. Bu seçenek aynı anda çok sayıda dizini göndermek için yararlıdır. Örneğin şöyle bir komut,

```
rsync /foo/bar/foo.c remote:/tmp/
```

uzak makinede /tmp/foo.c dosyasını oluşturur. Bunun yerine şu komutu kullansaydınız,

```
rsync -R /foo/bar/foo.c remote:/tmp/
```

uzak makinede /tmp/foo/bar/foo.c dosyası oluşturulacaktı (dosya yolunun tamamı kopyalanır). Karşıya yollanacak yol uzunluğunu sınırlamak için şöyle yapabilirsiniz:

```
cd /foo
rsync -R bar/foo.c remote:/tmp/
```

Bu komut uzak makinede /tmp/bar/foo.c dosyasını oluşturur.

--no-relative

--relative seçeneğini iptal eder. Bu özellikle **--files-from** seçeneğinin gerektiği durumlarda **--relative** seçeneği uygulanmaksızın dosyaları işleme sokmakta yararlıdır.

--no-implied-dirs

--relative seçeneği ile birlikte kullanıldığında her dosya yolundaki dolaylı dizinler, aktarımın parçası olarak yinelenmez. Bu dosya aktarımını daha elverişli yaptığı gibi ayrıca dosya yolunun dolaylı parçalarında iki tarafta eşleşmeyen sembolik bağların varlığını mümkün kılar. Örneğin, `"/path/foo/file"` dosyasını **-R** ile aktarırsanız, hedefte `"/path"` ve `"/path/foo"` ile kaynağın dizin ve sembolik bağları tamamen eşleştirilecektir. **--no-implied-dirs** seçeneğinin kullanımı bu dolaylı dizinlerin her ikisinin de atlanmasına yol açar. Yani, eğer `"/path"` bir makine üzerinde gerçek bir dizin, ancak diğer makinede bir sembolik bağ ise, **rsync** bunu değiştirmeye çalışmayacaktır.

-b

--backup

Bu seçenekle, evvelce mevcut hedef dosyaların her aktarımında ya da silinmesinde bu dosyaların isimleri değiştirilir. Yedek dosyaların nereye gideceğini ve nasıl bir sonek alacaklarını **--backup-dir** ve **--suffix** seçenekleri ile belirtebilirsiniz. Eğer **--backup-dir** seçeneğini belirtmezseniz, **--omit-dir-times** seçeneği etkin kılınacaktır.

--backup-dir=dizin

--backup seçeneği ile birlikte kullanıldığında bu seçenekle yedek dosyaların hangi dizinde saklanacağı belirtilebilir. Bu, arttırımlı yedekleme yaparken çok yararlıdır. Buna ek olarak **--suffix** seçeneğini kullanarak bir yedek soneki de belirtebilirsiniz (aksi takdirde dosyalar belirtilen dizine kendi isimleri ile yedeklenir).

--suffix=sonek

Bu seçenek **--backup** (**-b**) ile kullanıldığında, belirtilen *sonek* öntanımlı yedekleme sonekinin yerine kullanılır. **--backup-dir** belirtilmemişse, öntanımlı sonek `~` işaretidir, aksi takdirde boş bir dizgedir.

-u

--update

Bu seçenek, hedefte mevcut olan bir dosyanın değişiklik zamanının kaynak dosyanınkinden daha yeni olduğu durumda atlanmasını sağlar. (Eğer hedef dosyanın değişiklik zamanı kaynağınkıyla aynıysa, ama boyutları farklıysa güncelleme yapılacaktır.)

`--update` seçeneğinin şu anki gerçekleşiminde gönderici ve alıcı arasında dosya biçimlerinde bir fark varsa, bu, güncelleme için tarihlerden daha önemli olarak ele alınır. Başka bir deyişle, hedef bir dosyaya sahipken, kaynak bir dizin veya bir sembolik bağa sahipse, aktarım dosyalarını zaman değerlerine bakılmaksızın yapılır. Bu davranış ileride değişebilir (bu konuda bir fikriniz varsa, eposta listesinde bunu açıklamaktan çekinmeyin).

--inplace

Bu seçenek **rsync**'in dosyanın yeni bir kopyasının oluşturulmamasına ve yerine taşınmamasına sebep olur. Bunun yerine **rsync** mevcut dosyanın üzerine yazacaktır, yani **rsync** algoritması ağ yükünü tamamen azaltmayı başaramaz (henüz veriyi uyumlu sıralamayı denemediğinden). Buna bir istisna, **rsync** yedek dosyaları aktarımda temel dosya olarak kullanma yeterliğinde olduğundan, seçeneğin **--backup** ile birlikte kullanılmasıdır.

Bu seçenek blok temelli değişiklikler olan ya da sonuna ekleme yapılan çok büyük dosyalarla, disk ve ağ bakımından sınırlı sistemlerde yararlıdır.

Bu seçenek **--partial** (kesilmeye uğramış bir aktarımda dosya silinmez) seçeneğini uygular, fakat **--partial-dir** ve **--delay-updates** seçenekleriyle çelişir. **rsync** 2.6.4 öncesinde **--inplace** ayrıca **--compare-dest** ve **--link-dest** seçenekleriyle de çalışıyordu.



Uyarı

Dosya verisi aktarım sırasında kararsız bir durumda olacaktır (ve büyük ihtimalle aktarım kesintiye uğradıktan sonra da), dolayısıyla, kullanılmakta olan dosyaları bu seçeneği kullanarak güncellememelisiniz. Ayrıca, alıcı kullanıcının yazma izni olmayan dosyaların **rsync** tarafından yerinde güncellenemeyeceğini de unutmayın.

-d

--dirs

Gönderen tarafın saptanan her dizini içereceğini söyler. **--recursive** seçeneğinin tersine, bir dizinin içeriği, dizin komut satırında "." ya da bir bölü çüzgisi ile biten bir isim ("**foo/**" gibi) ile belirtilmedikçe kopyalanmaz. Bu seçenek ya da **--recursive** seçeneği olmaksızın, **rsync** saptadığı tüm dizinleri atlayacaktır (ve herbiri için bir ileti çıktılacaktır).

-l

--links

Saptanan her sembolik bağ hedefte oluşturulur.

-L

--copy-links

Sembolik bağ saptandığında sembolik bağ değil, hedefindeki dosya kopyalanır. **rsync**'in eski sürümlerinde, bu seçenek ayrıca alıcı tarafa, dizin sembolik bağı gibi sembolik bağları izlemesini söylemek gibi bir yan etkiye sahipti. Günümüzde ise, **rsync**, böyle bir durumda, bu ek davranışı sağlamak için sizin **--keep-dirlinks** (**-K**) seçeneğini belirtmenizi gerektirir. Buna tek istisna, **-K** seçeneğini anlayacak çok eski bir **rsync** ile dosyaların gönderilmesidir — bu durumda **-L** seçeneği daha eski bir **rsync** ile hala **-K** seçeneğinin yan etkisine sahip olacaktır.

--copy-unsafe-links

Bu seçenek **rsync**'e, kopyalanan ağacın dışındaki bir hedefi gösteren sembolik bağları kopyalamasını söyler. Mutlak sembolik bağlar ayrıca sıradan dosyalar olarak ele alınır, böylece bunlar **--relative** kullanıldığında kaynak dosya yolunun kendindeki sembolik bağlar olurlar.

safe-links

Bu seçenek **rsync**'e, kopyalanan ağacın dışındaki bir hedefi gösteren sembolik bağları yoksaymasını söyler. Bu seçeneğin **--relative** seçeneğiyle birlikte kullanılması beklenmedik sonuçlara yol açabilir.

-H

--hard-links

Bu seçenek **rsync**'e, sabit bağları aynı kaynak sistemdeki gibi yeniden oluşturmasını söyler. Bu seçenek olmaksızın sabit bağlar sıradan dosyalar olarak ele alınırlar.

rsync'in, eğer bağı her iki parçası da dosya listesi ile gönderilmişse, sabit bağları saptayabileceğini unutmayın.

Bu seçenek işlemi yavaşlatabilir, gerçekten bu işlevselliğe ihtiyacınız varsa kullanın.

-K

--keep-dirlinks

Alıcı tarafta, bir sembolik bağ bir dizini gösteriyorsa, bunun gönderici tarafta bir dizinle eşleştiği varsayılacaktır.

-W

--whole-file

Bu seçenekle arttırımlı **rsync** algortiması kullanılmaz ve bunun yerine dosya olduğu gibi gönderilir. Eğer kaynak ile hedef arasındaki band genişliği diskin band genişliğinden (özellikle disk, ağ dosya sisteminin bir parçası ise) daha yüksekse bu seçenek daha hızlı olabilir. Kaynak ve hedef, yerel dosya yolları olarak belirtilmişse bu seçenek öntanımlıdır.

--no-whole-file

--whole-file öntanımlı olduğunda bu seçenek onu kapatır.

-p

--perms

Bu seçenek **rsync**'e, izinleri hedefte aynı kaynak sistemdeki gibi belirlemesini söyler.

Bu seçenek olmaksızın, her yeni dosya izinlerini kaynak dosyanın izinlerine ve alıcı taraftaki **umask**'a göre belirlenir, diğer tüm dosyalar (güncellenenler dahil) kendi mevcut izinleriyle kalır (bu, **cp** gibi diğer dosya kopyalama araçlarında da böyledir).

-o

--owner

Bu seçenek **rsync**'e, hedef dosyanın sahibini aynı kaynak dosyadaki gibi belirlemesini söyler. Çoğu sistemde sadece ayrıcalıklı kullanıcı dosya aidiyetini belirleyebilir. Öntanımlı olarak, koruma isim tarafında yapılır, fakat bazı durumlarda son çare olarak kullanıcı kimliği kullanılır. Bu konu **--numeric-ids** seçeneğinin açıklamasında daha ayrıntılı incelenmiştir.

-g

--group

Bu seçenek **rsync**'e, hedef dosyanın grubunu aynı kaynak dosyadaki gibi belirlemesini söyler. Alıcı tarafta **rsync**'i ayrıcalıklı kullanıcı çalıştırmıyorsa, sadece alıcının üyesi olduğu gruplar için bu yapılabilir. Öntanımlı olarak işlem grup ismine göre yapılır, ancak bazı durumlarda son çare olarak grup kimliği de kullanılabilir. Bu konu **--numeric-ids** seçeneğinin açıklamasında daha ayrıntılı incelenmiştir.

-D

--devices

Aktarılan karakter ve blok aygıtları hedefte yeniden oluşturulur. Bu seçeneği sadece ayrıcalıklı kullanıcı kullanabilir.

-t

--times

Bu seçenek **rsync**'e, dosyaları değişiklik zamanlarına göre aktarmasını ve bunları uzak sistemde güncellemesini söyler. Bu seçenek kullanılmazsa, eniyileme yüzünden değiştirilmemiş dosyaların dışlanması verimli olamaz; başka bir deyişle, **-t** veya **-a** seçeneğinin yokluğu sonraki aktarımda **-I** kullanılmış gibi tüm dosyaların güncellenmesine yol açar (**rsync** algoritması güncellemeyi oldukça verimli yapacağından, eğer dosyalar gerçekten değişmemişse, **-t** kullanmanız çok daha iyi olacaktır).

-O

--omit-dir-times

Eğer **rsnc** değişiklik zamanlarını güncelliyorsa (**--times** ile), bu seçenek belirtildiğinde izinleri bunun dışında tutar. Eğer alıcı tarafta NFS izinleri paylaşıyorlarsa, **-O** seçeneğini kullanmak daha iyidir. Bu seçenek **--backup** seçeneği **--backup-dir** olmaksızın kullanılmış gibi sonuç verir.

-n

--dry-run

Bu seçenek **rsync**'e, herhangi bir dosya aktarımı yapmamasını, sadece yapıyormuş gibi işlemleri raporlamasını söyler.

-S

--sparse

Bu seçenekle **rsync**, seyrek dosyaların hedefte daha az yer kaplayacak şekilde elde edilmesini sağlamaya çalışır.



Bilgi

Bu seçeneği hedefteki dosya sistemi Solaris "tmpfs" ise kullanmayın. Boş bölgelerin düzgünce atlanmasını sağlayamaz ve dosya sonunda bozulmalar oluşur.

-x

--one-file-system

Bu seçenek **rsync**'e, aktarım sırasında dosya sistemi sınırlarının dışına çıkmamasını söyler. Bu seçenek, tek bir dosya sisteminin içeriğini aktarmak için yararlıdır.

--existing

Bu seçenek **rsync**'e, hedefte yeni dosya oluşturulmasını gerektirecek aktarımları yapmamasını sadece güncelleme yapmasını söyler.

--ignore-existing

Bu seçenek **rsync**'e, hedefte mevcut olan dosyaları güncellememesini söyler.

--remove-sent-files

Bu seçenek **rsync**'e, hedefte yeni oluşturulan dosyalarla içeriği güncellenenleri, gönderen taraftan silmesini söyler. Dizinler ve aygıtlar ile sadece öznitelikleri değiştirilmiş dosyalar silinmez.

--delete

Bu seçenek **rsync**'e, hedefte olup kaynakta olmayan dosyaları hedeften silmesini söyler, fakat sadece dosyalar izinleri ile belirtilmişse bu yapılır. Yani, **rsync**'den tüm dizini ("**dir**" veya "**dir/**" gibi) göndermesini istemeli, izin içeriğini belirten dosya kalıpları ("**dir/***" gibi) belirtmemelisiniz. Dosya kalıpları belirtildiğinde kalıp kabuk tarafından yorumlanır ve kabuk **rsync**'den dosyaları kapsayan dizini değil, tek tek dosyaları aktarmasını ister. Aktarımdan dışlanan dosyalar dan biriyse, **--delete-excluded** kullanılmamışsa ya da kurallar sadece gönderen tarafla eşleşenleri imliyorsa hedefte silinecek olanlar da dışlanır (*SÜZME KURALLARI* (sayfa: 24) bölümündeki içirme/dışlama belirteçlerine bakınız).

Alt izinlere inilmiyorsa bu seçenek etkisizdir.

Bu seçenek doğru şekilde kullanılmazsa tehlikeli olabilir. Önce **--dry-run** (**-n**) seçeneğiyle silinecek dosyaları görüp, önemli dosyaların silinmeyeceğinden emin olduktan sonra asıl işlemi başlatmak çok daha iyidir.

Gönderen tarafta bir G/Ç hatası saptandığında hedefteki dosya silme işlemi iptal edilir. Bu, gönderen tarafın alıcı taraftaki dosyaları tümünden silmesi gibi geçici dosya sistemi hatalarına karşı (NFS hataları gibi) bir önlemdir. Bu davranışı **--ignore-errors** seçeneğiyle iptal edebilirsiniz.

--delete seçeneği silme zamanını belirten **--delete-*** seçenekleriyle, **--delete-excluded** dahil, herhangi bir sorun olmaksızın birleştirilebilir. Ancak, silme zamanını belirten **--delete-*** seçeneklerinden biri belirtilmemişse, **rsync** ötanımlı olarak **--delete-before** algoritmasını seçer. İleri sürümlerde bu seçim **--delete-during** algoritmasıyla değiştirilebilir. Ayrıca, **--delete-after** seçeneğine de bakınız.

--delete-before

Alıcı taraftaki dosya silme işlemlerinin aktarımın başlangıcında yapılmasını sağlar. Bu seçenek **--delete** veya **--delete-excluded** seçeneği ile birlikte silme zamanını belirten **--delete-*** seçeneklerinden biri belirtilmemişse ötanımlıdır. Dosya silinmesi ile ilgili daha ayrıntılı bilgi için **--delete** seçeneğinin açıklamasına bakınız.

Aktarımın başında silme yapılması, eğer dosya sistemi doluluk bakımından sınırdan ise ve ancak fazlalıkların kaldırılmasıyla dosya aktarımı mümkün olabileceyse yararlıdır. Ancak, bu seçenek aktarımın başlamasını geciktirdiğinden aktarımın zamanaşımına uğramasına (**--timeout** belirtilmişse) sebep olabilir.

--delete-during

--del

Alıcı taraftaki dosya silme işlemlerinin aktarımı sırasında yapılmasını sağlar. Bu yöntem silme işleminin aktarımdan önce ve sonra yapıldığı algoritmalarından daha hızlıdır. Fakat ilk olarak **rsync**'in 2.6.4 sürümüyle kullanılabilir olmuştur. Dosya silinmesi ile ilgili daha ayrıntılı bilgi için **--delete** seçeneğinin açıklamasına bakınız.

--delete-after

Alıcı taraftaki dosya silme işlemlerinin aktarım tamamlandıktan sonra yapılmasını sağlar. Eğer aktarımın bir parçası olarak dizin bazında yeni dosya birleşimleri yapıyorsanız ve bunların ihracının aktarımın silme fazında etkin olmasını istiyorsanız bu seçenek yararlı olacaktır. Dosya silinmesi ile ilgili daha ayrıntılı bilgi için **--delete** seçeneğinin açıklamasına bakınız.

--delete-excluded

Alıcı tarafta olup, gönderici tarafta olmayan dosyaların alıcı tarafta silinmesine ek olarak, alıcı tarafta dışlanan dosyalarında silinmesini sağlar (**--exclude** seçeneğine bakınız). Alıcı tarafta tek tek dosyaların dışlanması bu yöntemle silinmesi ve silinmesi istenmeyen dosyaların bu seçenekten korunması ile ilgili bilgiler için [SÜZME KURALLARI](#) (sayfa: 24) bölümüne bakınız. Dosya silinmesi ile ilgili daha ayrıntılı bilgi için ise **--delete** seçeneğinin açıklamasına bakınız.

--ignore-errors

G/Ç hataları olsa bile **--delete** işleminin sürdürülmesini sağlar.

--force

Gönderici tarafta olmayan ancak alıcı tarafta olan dizinlerin içleri dolu bile olsa silinmesini sağlar. Bu sadece **--delete** seçeneği olmaksızın böyledir, çünkü silme işlemleri artık içiçe dizinler içeren bir tepe dizinin silinmesi şeklinde yapılmaktadır. Etkin olması için **--recursive** seçeneği (**-a** tarafından uygulanır) gerekir.

--max-delete=sayı

sayı'dan fazla dosya ya da dizin silinmez. Yansılama yapılırken çok büyük ağaçların yıkıma yol açmaması için yararlıdır.

--max-size=boyut

boyut bayttan daha büyük dosyalar aktarılmaz. *boyut* belirtilirken, sonrak olarak K, M, G gibi bayt çarpanları belirtilebileceği gibi kesirli değerler de belirtilebilir (**--max-size=1.5m** gibi).

-B**--block-size=***blokboyu*

rsync algoritmasında sabit bir blok boyunun kullanımına zorlar. Normalde, bu değer güncellenen her dosya için ayrı ayrı seçilir. Daha ayrıntılı bilgi için teknik rapora bakınız.

-e**--rsh=***komut*

Bu seçenek **rsync**'in yerel ve uzak kopyaların arasında iletişim için bir uzak kabuk uygulaması belirleyebilmeyi sağlar. Genelde, **rsync** öntanımlı olarak **ssh** kullanacak şekilde yapılandırılır, fakat bir yerel ağda **rsh** kullanmayı da tercih edebilirsiniz.

Bu seçenek [*kullanıcı@*]*makine::modül/yol* ile kullanılırsa, uzak makinedeki **rsync** sunucusunu çalıştırmak için belirtilen uzak kabuk *komut*'u kullanılacak ve aktarım için uzak makinede çalışan **rsync** sunucusunun soket bağlantısı değil, uzak kabuk bağlantısı kullanılacaktır. Ayrıca, yukarıdaki *BİR RSYNC SUNUCUSUNA BİR UZAK KABUKTAN BAĞLANMAK* (sayfa: 4) bölümüne de bakınız.

komut içinde **rsync** sadece tek bir argümana izin verir. Örnek:

```
-e "ssh -p 2234"
```



Bilgi

ssh kullanıcıları makineye özel bağlantı seçeneklerini kendi *.ssh/config* dosyası ile kişiselleştirebilirler.

Uzak kabuk uygulamasını seçmek için *RSYNC_RSH* ortam değişkenini, **-e** seçeneğinde belirtilebilen bir değeri atayarak kullanabilirsiniz.

Ayrıca, bu seçenekten etkilenen **--blocking-io** seçeneğinin açıklamasına da bakınız.

--rsync-path=*uygulama*

Uzak makinede başlangıçta hangi uygulamanın çalıştırılacağını belirtmek için kullanılır. Çoğunlukla, **rsync**'in uzak kabuğun öntanımlı yolları dışında bulunduğu durumda kullanılır (**--rsync-path=/usr/local/bin/rsync** gibi). *uygulama* kabuk tarafından çalıştırılacağından herhangi bir uygulama, betik ya da bir dizi komut olabilir. Bir dizi komut belirtiyorsanız **rsync**'in iletişimde kullanacağı standart girdi ve standart çıktının devredışı kalmamasına dikkat etmelisiniz.

Aşağıdaki örnekte uzak makinede **--relative** seçeneği ile kullanmak için farklı bir öntanımlı yol belirtilmektedir:

```
rsync -avR --rsync-path="cd /a/b && rsync" makine:c/d /e/
```

-C**--cvs-exclude**

Sistemler arasında aktarımı istenmeyen geniş bir dosya kümesinin dışlanması için bir kısayol olarak kullanışlıdır. CVS'nin yoksayılacak dosyaları saptamakta kullandığı algoritma kullanılır.

Dışlanan dosyaların listesi şöyle ilklendirilmiş olsun:

```
RCS SCCS CVS CVS.adm RCSLOG cvslog.* tags TAGS
.make.state .nse_depinfo *~ #* .* , * _$* *$ *.old *.bak
*.BAK *.orig *.rej .del-* *.a *.olb *.o *.obj *.so *.exe
*.Z *.elc *.ln core .svn/
```

\$HOME/.cvsignore içindeki dosyalarla *CVSIGNORE* ortam değişkeninde (tüm isimler bir boşlukla ayrılarak belirtilir) listelenen her dosya bu listeye eklenecektir.

Son olarak, `.cvsignore` dosyası ile aynı dizindeki dosyalardan `.cvsignore` dosyasında listelenmiş kalıplarla eşleşen dosyalar yok sayılır. **rsync**'in süzme ve dışlama uyguladığı dosyaların aksine bu kalıplar boşluklarla ayrılır. Daha fazla bilgi için **cv**s (1) kılavuz sayfasına bakınız.

Bu seçeneği kendi **--filter** kurallarınızla birlikte belirtirseniz, **-C** seçeneğinin komut satırının neresinde olduğuna bakılmaksızın, CVS dışlananlarının sizin kurallarınızın sonuna ekleneceğini unutmayın. Bu, onlara sizin belirttiğiniz kurallara göre daha düşük öncelik verir. CVS dışlananlarının süzme kurallarınız arasındaki yerini kendiniz belirlemek isterseniz, komut satırı seçeneği olarak **-C** belirtmek yerine **--filter=:C** ve **--filter=-C** seçeneklerini belirtmelisiniz (bu şekilde komut satırında belirtebileceğiniz gibi bir süzme dosyasına `:"C` ve `-"C` kurallarını yerleştirerek de belirtebilirsiniz). İlk kural her dizinde `.cvsignore` dosyalarına bakılmasını, ikinci kural ise yukarıda bahsedilen CVS dışlananlarını bir seferliğine dahil edilmesini sağlar.

-f

--filter=kural

Bu seçenek aktarılacak dosyalar listesinden belli dosyaların seçilerek dışlanması için kurallar eklemenizi mümkün kılar. Bu seçenek alt dizinlerin ardışık aktarıldığı aktarımlarda kullanıldığında çok yararlıdır.

Dışlanacak dosyaların listesini oluşturmak için komut satırında çok sayıda **--filter** seçeneği belirtebilirsiniz.

Bu seçenek hakkında daha ayrıntılı bilgi için [SÜZME KURALLARI](#) (sayfa: 24) bölümüne bakınız.

-F

Bu seçenek iki defa belirtildiğinde komuta iki **--filter** kuralı eklemeye eşdeğerdir. İlk seçenek şu kuralı ekler:

```
--filter=': /.rsync-filter'
```

Bu kural, **rsync**'in dizin hiyerarşisi dahilinde serpiştirilmiş `.rsync-filter` dosyalarına bakmasını ve aktarımda bu dosyalardaki süzme kurallarını kullanmasını sağlar. **-F** seçeneği ikinci kez kullanılmışsa şu kuralı uygular:

```
--filter='- .rsync-filter'
```

Bu kural `.rsync-filter` dosyalarının aktarılmamasını sağlar.

Bu seçenek hakkında daha ayrıntılı bilgi için [SÜZME KURALLARI](#) (sayfa: 24) bölümüne bakınız.

--exclude=kalıp

Bu seçenek bir dışlama kuralı tanımlanmasını sağlayan **--filter** seçeneğinin basitleştirilmiş bir biçimidir. Seçenek, normal süzme kurallarının kural çözümleme sözdizimine izin vermez.

Bu seçenek hakkında daha ayrıntılı bilgi için [SÜZME KURALLARI](#) (sayfa: 24) bölümüne bakınız.

--exclude-from=dosya

Bu seçenek, dışlama kurallarının belirtilen dosyadan alınmasını sağlaması dışında **--exclude** seçeneğinin benzeridir. *dosya* içindeki boş satırlar ile `';` veya `'#' ile başlayan satırlar yoksayılar. dosya olarak - belirtilmişse girdiler standart girdiden okunur.`

--include=kalıp

Bu seçenek bir içerme kuralı tanımlanmasını sağlayan **--filter** seçeneğinin basitleştirilmiş bir biçimidir. Seçenek, normal süzme kurallarının kural çözümleme sözdizimine izin vermez.

Bu seçenek hakkında daha ayrıntılı bilgi için [SÜZME KURALLARI](#) (sayfa: 24) bölümüne bakınız.

--include-from=dosya

Bu seçenek, içirme kurallarının belirtilen dosyadan alınmasını sağlar. *dosya* olarak **-** belirtilmişse girdiler standart girdiden okunur.

--files-from=*dosya*

Bu seçeneği kullanarak aktarılacak dosyaların tam listesini belirtebilirsiniz. Liste *dosya*'dan ya da **-** belirtilmişse standart girdiden okunur. Ayrıca, bu seçenekle birlikte öntanımlı uygulanan seçeneklere bağlı olarak **rsync**'in öntanımlı davranışını kolayca değiştirebilirsiniz:

- **--relative** (**-R**) seçeneği uygulanarak *dosya* içindeki her öğede belirtilen yol bilgisi korunmuştur (bu davranışı kapatmak için **--no-relative** seçeneğini kullanın).
- **--dirs** (**-d**) seçeneği uygulanarak listede belirtilen dizinlerin atlanmayıp, hedefte oluşturulması sağlanmıştır.
- **--archive** (**-a**) seçeneği uygulanmasına rağmen bu seçeneğin uygulamalarından biri olan **--recursive** (**-r**) seçeneği uygulanmadığından, bu davranışın uygulanmasını istiyorsanız **--recursive** (**-r**) seçeneğini ayrıca belirtmelisiniz.

dosya'dan okunan dosya isimlerinin hepsi kaynak dizine göre değerlendirilir — öğelerin başındaki **"/**"lar ile daha üst dizinlere çıkılmasını sağlayan **"/**"lar kaldırılır. Örnek:

```
rsync -a --files-from=/tmp/foo /usr uzak:/yedek
```

/tmp/foo içinde **"bin"** (hatta **"/bin"**) diye bir girdi varsa, */usr/bin* dizini hedefte */yedek/bin* dizini olarak oluşturulur (ancak, *dosya* içinde dosyalar açıkça listelenmemişse ya da komut satırında **-r** seçeneği yoksa */usr/bin* dizinindeki dosyalar aktarılmaz). Ayrıca, **--relative** seçeneğinin etkisini (öntanımlı olarak etkindir) sadece dosyadan okunan yol bilgisini yinelemek için saklar ancak, kaynak için belirtilmiş yolun yinelenmesine zorlamaz (örnekte */usr*).

Ek olarak, eğer *dosya*'yı **"uzak:"** öneki ile belirtirseniz, bu dosya yerel makineden değil, uzak makineden okunur (**uzak:** ile belirtilen makine, aktarımın yapıldığı makine olmalıdır). Bir kısaltma olarak, "aktarımın yapıldığı uzak makine" anlamında sadece **:"** önekini de kullanabilirsiniz. Örnek:

```
rsync -a --files-from=:/yol/dosya-listesi uzak:/ /tmp/kopya
```

Bu komut **"uzak"** makinesindeki */yol/dosya-listesi* dosyasında listelenmiş dosyaları kopyalayacaktır.

-O

--from0

Bu seçenek **rsync**'e, bir dosyadan okunan dosya isimlerinin sonlandırılmasının satırsonu karakteri (LF) veya satırbaşı karakteri (CR) ya da her bu ikisinin birlikte kullanılmasıyla (CR+LF) değil sadece boş karakterle (**'\0'**) yapıldığını belirtir. Bu seçenek **--exclude-from**, **--include-from**, **--files-from** ve bir **--filter** kuralı ile belirtilen dosyaları etkiler. Dosya isimlerinin boşluk ile ayrıldığı **.cvsignore** dosyalarını kullanan **--cvs-exclude** seçeneğini etkilemez.

-T

--temp-dir=*dizin*

Bu seçenekle, dosyalar aktarılırken alıcı tarafta geçici olarak oluşturulan dosyaların nereye konulacağını belirtmek için kullanılır. Öntanımlı davranış, geçici dosyaların alıcı dizinde oluşturulmasıdır.

-y

--fuzzy

Bu seçenekle, hedefte bulunmayan bir dosya için benzer bir dosya aranacağını belirtilir. Şimdiki algoritma, hedef dosya için aynı dizinde, aynı boyda ve aynı değişiklik zamanlı ya da aynı isimli bir dosya arar. Böyle bir dosya varsa, bu dosya bulanık bir hedef dosya olarak aktarımın hızlandırılması amacıyla kullanılır.



Bilgi

--delete seçeneği bu amaçla kullanılabilecek bir dosyanın işlem öncesinde silinmesine yol açabileceğinden, bunun yerine ya **--delete-after** seçeneğini kullanın ya da dosyanın silme işlemine konu olmayacak şekilde olmasına dikkat edin.

--compare-dest=dizin

Bu seçenekle, alıcı makinede hedef dizinde olmayan dosyalar için karşılaştırmanın *dizin* dizinine göre yapılacağı belirtilir. Eğer *dizin* içinde göndericinin dosyasının aynısı varsa bu dosya aktarılmaz. Bu, sadece bir önceki yedeklemeden beri değişmiş dosyaların yedeklenmesi için kullanışlıdır.

2.6.4 sürümünden başlayarak, çok sayıda **--compare-dest** dizini belirtilebilmektedir. Eğer hedefte bir eşleşme bulunamazsa, dosya bu dizinlerde aranır.

Eğer *dizin* görel bir dizin olarak belirtilmişse bu dizinin hedef dizinde olduğu kabul edilir. Ayrıca **--link-dest** seçeneğine de bakınız.

--link-dest=dizin

Bu seçenek, *dizin* içindeki değişmemiş dosyaların hedef dizine sabit bağlı yapılması dışında **--compare-dest** seçeneği gibi davranır. Sabit bağlı yapılacak dosyalar öznitelikleri (izinler, sahipler, v.s) bakımından kaynaktakilerle tamamen aynı olmalıdır. Örnek:

```
rsync -av --link-dest=$PWD/öncü-dizin \
    makine:kaynak-dzn/ yeni-dzn/
```

2.6.4 sürümünden başlayarak, çok sayıda **--link-dest** dizini belirtilebilmektedir. Eğer hedefte bir eşleşme bulunamazsa, dosya bu dizinlerde aranır.

Eğer *dizin* görel bir dizin olarak belirtilmişse bu dizinin hedef dizinde olduğu kabul edilir. Ayrıca **--compare-dest** seçeneğine de bakınız.



Bilgi

2.6.1 sürümünden önceki sürümlerde, **-o** seçeneği (**-a** seçeneği de bu seçeneği uygular) belirtilmişse, **--link-dest** seçeneği root olmayan bir kullanıcı için istendiği gibi çalışmayacaktır (yazılım hatası). Bu hatadan korunmak için böyle bir **rsync** ile **-o** seçeneğini kullanmamalısınız.

-z

--compress

Bu seçenekle hedef makineye gönderilen dosyaların verilerini **rsync**, aktarım miktarını azaltmak için sıkıştıracaktır. Özellikle, yavaş bağlantılar için çok yararlıdır.



Bilgi

Hedef makinede sıkıştırma için aynı araçlar kullanılıyorsa, sıkıştırılmış veri sıkıştırma ile ilgili dolaylı bilgileri içermeyeceğinden aktarımda daha iyi sıkıştırma yapılabilmektedir.

--numeric-ids

Bu seçenekle **rsync**, her iki uçta kullanıcı ve grup isimlerini değil, kullanıcı ve grup kimliklerini eşleştirmeye çalışacaktır.

Öntanımlı olarak **rsync**, dosyaların sahibini belirlemekte kullanıcı ve grup isimlerini kullanır. Özel kullanıcı ve grup kimliği olan 0 kimliği, bu seçenek belirtilmemiş olsa bile, kullanıcı ve grup isimleriyle asla eşleştirilmez.

Eğer kullanıcı ve grup isimleri her iki uçta eşleşmiyorsa, bunların yerine kaynak sistemdeki kullanıcı ve grup kimlikleri kullanılır. **chroot** ayarlarının **rsync**'in kullanıcı ve grup isimlerini arama yeteneklerini nasıl etkilediğini ve bunları nasıl değiştirebileceğinizi öğrenmek için [rsyncd.conf \(5\)](#) kılavuz sayfasındaki **use chroot** modül seçeneğinin açıklamasına bakınız.

--timeout=süre

Bu seçenikle G/Ç zamanasını saniye cinsinden belirtebilirsiniz. Belirtilen süre içinde bir aktarım gerçekleşmezse, **rsync** işlemi sonlandıracaktır. Öntanımlı değer, bir zamanasımı olmadığını belirten 0 değeridir.

--port=port

Öntanımlı 873. port yerine kullanılacak portu belirtmek için kullanılır. URL sözdizimi port belirtmek için zaten bir yöntem içerdiğinden, bu sadece bir **rsync** sunucusuna ":" kullanarak bağlanıyorsanız gerekir. Bu seçeneğin [SUNUCU SEÇENEKLERİ](#) (sayfa: 23) bölümündeki açıklamasına da bakınız.

blocking-io

Uzak kabuk aktarımı başlatılırken engellenebilen G/Ç kullanılmasını belirtir. Uzak kabuk **rsh** veya **remsh** ise, **rsync** öntanımlı olarak engellenebilen G/Ç, aksi takdirde engellenemeyen G/Ç kullanır (**ssh** ile engellenemeyen G/Ç tercih edilir).

no-blocking-io

Öntanımlı olarak engellenebilen G/Ç kullanılıyorsa, bunu kapatır.

-i

--itemize-changes

Her dosyanın öznitelik değişikliklerini de içererek yapılmış değişikliklerin dosyalara göre tasnif edilmiş bir listesini ister. Bu, **--log-format='%i %n%L'** belirtmekle aynıdır.

"%i" belirtimi 9 karakter uzunlukta şifresel bir çıktıya sahiptir. Genel biçimi *UXcstpoga* dizgesine benzer. Burada **U** karakteri yapılan güncellenmenin çeşidini, **X** karakteri dosya türünü diğer karakterler ise güncellenmişlerse öznitelikleri belirten bir karakterle değiştirilir.

U karakterinin yerini alan güncelleme türleri şunlardır:

- **<** karakteri yerel makineye aktarılan (alınan) bir dosyayı belirtir.
- **>** karakteri yerel makineye aktarılan (gönderilen) bir dosyayı belirtir.
- **c** karakteri yerel makinede değiştirilen/oluşturulan bir dosyayı belirtir (bir dizin ya da sembolik bağın oluşturulması gibi).
- **h** karakteri başka bir öğeye sabit bağ olan bir öğeyi belirtir (**--hard-links** gerektirir).
- **.** (nokta) karakteri sadece öznitelikleri değişmiş bir dosya belirtir.
- **=** karakteri öğenin iki uçta da aynı olduğunu belirtir (bu sadece ve sadece yüksek bir ayrıntı seviyesinde çıktılanır).

X karakterinin yerini alan dosya türleri şunlardır: bir dosya için **f**, bir dizin için **d**, bir sembolik bağ için **L** ve bir aygıt için **D**.

UXcstpoga dizgesinin diğer harfleri, gerçek karakterler olup, öğenin ilgili öznitelikleri güncellenmişse çıktılanırlar, aksi takdirde "." ile değiştirilirler. Bunun üç istisnası vardır:

1. yeni oluşturulmuş bir öğe için her harf bir "+" ile değiştirilir.

2. İki uçtada aynı olan öğelerde her harf bir boşlukla değiştirilir.
3. Bilinmeyen her öznitelik bir "?" ile değiştirilir (bu, daha eski bir **rsync** ile konuşurken ortaya çıkar).

Harflerin belirttiği öznitelikler şunlardır:

- **c** dosyanın sağlama toplamalarının farklı olduğunu ve dosyanın aktarımla güncelleneceğini belirtir (**--checksum** gerektirir).
- **s** dosyanın boyutlarının farklı olduğunu ve dosyanın aktarımla güncelleneceğini belirtir
- **t** değişiklik zamanlarının farklı olduğunu ve sunucudaki değer ile güncelleneceğini belirtir (**--times** gerektirir). Bazan bu karakterin yerine **T** kullanılır ve zamanın aktarım zamanına ayarlanacağını belirtir. **T** harfi, bir dosya ya da dizin **--times** seçeneği belirtilmeksizin aktarılırken ya da bir sembolik bağ aktarılırken çıkarılır.
- **p** izinlerin farklı olduğunu ve sunucudaki değer ile güncelleneceğini belirtir (**--perms** gerektirir).
- **o** sahiplerin farklı olduğunu ve sunucudaki değer ile güncelleneceğini belirtir (**--owner** ve **root** yetkileri gerektirir).
- **g** grupların farklı olduğunu ve sunucudaki değer ile güncelleneceğini belirtir (**--group** ve **grup** yetkileri gerektirir).
- **a** harfi gelecekte ACL'ler gibi genişletilmiş dosya özniteliklerini desteklemek üzere yedek olarak ayrılmıştır.

Olası diğer bir çıktı daha vardır: dosyalar silinirken silinecek her öğe için "%i" belirtimi "deleting" dizgesini çıktılar (yeterince son sürüm bir **rsync** ile konuşulduğu ve bunun silme işlemlerini bir ayrıntı olarak çıkılamayıp günlük kaydı tuttuğu varsayımıyla).

--log-format=biçim

İşlenen her dosya için kullanıcıya **rsync** istemcisinin nasıl bir çıktı göstereceğini belirlemenizi sağlar. *biçim* her biri % karakteri ile öncelenmiş tek karakterli belirtilimlerden oluşan bir dizgedir. Olası belirtilimlerin listesi için [rsyncd.conf\(5\)](#) kılavuz sayfasındaki **log format** seçeneğinin açıklamasına bakınız. (**--log-format** seçeneği bir sunucunun günlük kaydında ne çıkılayacağını belirlemez.)

biçim dizgesinde **--itemized-changes** belirtimi (%i) kullanılmamışsa, belli bir yolla (bir dosyanın aktarılması, bir sembolik bağ veya aygıtın yeniden oluşturulması, bir dizine dokunulması) güncellenen her dosya için çıktı üretilecektir, aksi takdirde, çıkıtlama herhangi bir yolla güncellenen her öğeye genişletilir (alıcı tarafın sürümü en azından 2.6.4 ise). "%i" belirtiminin çıktısı ile ilgili ayrıntılar için **--itemized-changes** seçeneğinin açıklamasına bakınız.

--verbose seçeneği "%n%L" biçiminin uygulanmasını sağlar. **--log-format** seçeneğini isterseniz, **-v** (**--verbose**) seçeneğini belirtmeksizin kullanabileceğiniz gibi bu seçeneği de kullanarak her dosya için bu biçimdeki bir çıktının üretmesini de sağlayabilirsiniz.

Günlüklemenin aktarımın sonunda yapılmasını sağlayan aktarım istatistiği belirtilimlerinden biri belirtilmedikçe **rsync**, günlük biçimi dizgesini bir dosyanın aktarımından önce çıkıtlayacaktır. Bu geç günlükleme etkin olduğunda ve ayrıca **--progress** belirtilmişse **rsync** ayrıca aktarılan dosyanın ismini işlem bilgisinden önce (şüphesiz, **--log-format** çıktısından sonra) çıkıtlayacaktır.

--stats

Bu seçenek **rsync**'e, **rsync** algoritmasının verileriniz için ne kadar verimli olduğunu görmeyi sağlayacak dosya aktarım istatistiklerini ayrıntılı olarak basmasını söyler.

--partial

Aktarım kesintiye uğramışsa, öntanımlı olarak **rsync**, aktarımı yarım kalmış dosyayı silecektir. Bazı durumlarda kısmen aktarılmış dosyaları silinmemesi istenebilir. Aktarımın bir sonraki bağlantıda tamamlanmasını hızlandırmak üzere kısmen aktarılmış dosyaların silinmemesi için bu seçeneği kullanabilirsiniz.

--partial-dir=dizin

Aktarımı yarım kalmış dosyaların silinmesini önlemenin daha iyi bir yolu bu tür dosyalar için ayrı bir *dizin* belirtmektir. Sonraki aktarımda, **rsync** bu dosyanın aktarımını tamamlamak için belirtilen *dizin* içindeki dosyayı kullanacak ve dosyanın aktarımı tamamlandıktan sonra bu dosyayı silecektir. Yalnız, eğer **--whole-file** belirtilmişse (ya da dolaylı uygulanıyorsa), **rsync** artırımlı algoritmayı kullanmaksızın dosyaları göndereceğinden, *dizin* içinde tamamlanmayı bekleyen bir dosya bulunursa basitçe silinecektir.

Eğer *dizin* mevcut değilse, **rsync** *dizin*'i oluşturacaktır (sadece son *dizin*, yolun tamamı değil). Bu davranışın sonucu olarak, hedef dosyanın bir *dizin* oluşturmayı gerektirdiği durumlarda görece yol belirtmek (**--partial-dir=.rsync-partial** gibi) daha iyidir, bu takdirde eksik dosya işlem sonunda silindiğinde bu *dizin* de silinir.

Eğer *dizin* ile mutlak bir yol belirtilmemişse, mevcut dışlamalarınızın sonunda **rsync** ayrıca **--exclude** değeri olarak bir *dizin* ekleyecektir. Böylece *dizin* içindeki dosyaların aktarımı ve ayrıca alıcı tarafta bu dosyaların zamanından önce silinmesi engellenecektir. Bir örnek: yukarıdaki **--partial-dir** seçeneği diğer süzme kurallarının sonuna bir **--exclude=.rsync-partial/** kuralı ekleyecektir. Eğer kendiniz bir takım süzme kuralları kullanıyorsanız, bu kuralın daha yüksek öncelik alması için kendi kurallarınız içinde uygun bir yerde belirtebilirsiniz (kurallarınızın sonunda bir **--exclude='*'** kuralı varsa, özdevimli eklenen bir kural hiçbir zaman erişilebilir olmayacaktır).

**Önemli**

--partial-dir ile belirtilen dizine diğer kullanıcılar yazamamalıdır, aksi takdirde bu bir güvenlik açığı oluşturur. Yani **"/tmp"** gibi bir *dizin* belirtmekten KAÇININ.

Ayrıca *dizin* değerini **RSYNC_PARTIAL_DIR** ortam değişkeninde belirtebilirsiniz. Bu değişkene bir değerin atanmış olması **--partial** seçeneğinin etkin olmasını sağlamaz ama komut satırında **--partial** seçeneğini belirtmekle **--partial-dir** seçeneğine *dizin* belirtilmiş gibi işlem yapılması sağlanır. Örneğin, **--partial-dir=.rsync-tmp** ile **--progress** seçeneğini birlikte kullanmak yerine, kabukta **RSYNC_PARTIAL_DIR=.rsync-tmp** şeklinde bir atama yaptığınızı ve **.rsync-tmp** *dizin*ini eksik kalan aktarımı tamamlamak için komut satırında sadece **-P** seçeneğini belirterek kullandığınızı varsayalım. **--partial** seçeneğinin bu ortam değişkenine bakmadığı durumlar, (1) **--inplace** seçeneğinin belirtilmiş olması (**--inplace** seçeneği **--partial-dir** ile çeliştiğinden), (2) **--delay-updates** seçeneğinin belirtilmiş olmasıdır (aşağıya bakınız).

Sunucu yapılandırmasının **refuse options** ayarı, amacı gereği **--partial-dir** seçeneğinin **--partial** seçeneğini uygulamamasına sebep olur. Bu, yani, **--partial** seçeneğinin reddedilmesi, **--partial-dir** tarafından sağlanan daha güvenilir üsluba hala izin veriyor olmakla, aktarımı yarım kalmış hedef dosyaların tekrar yazılmasına izin vermemekte kullanılabilir.

--delay-updates

Bu seçenek, güncellenen her dosyayı aktarımın sonunda isimleri değiştirilmek suretiyle yerine taşınmak üzere bir dizin içinde geçici bir dosyaya koyar. Bu dosyaların güncellenmesini daha atomik yapmayı amaçlar. Öntanımlı olarak bu dosyalar her dosyanın hedef dizini içindeki ".~tmp~" isimli bir dizine konur, fakat bu davranışı **--partial-dir** seçeneğini belirterek değiştirebilirsiniz. Bu seçenek **--inplace** seçeneği ile çelişir.

**Bilgi**

Bu seçeneğe özel olarak, **--partial-dir** yerine `RSYNC_PARTIAL_DIR` ile **--partial** belirtmek `RSYNC_PARTIAL_DIR` değerinin kullanılmasını sağlamaz. Ayrıca, sunucu yapılandırmasının **refuse options** ayarı, amacı gereği **--partial-dir** seçeneğine özel uygulamalar da yapılmaz.

Bu seçenek alıcı tarafta biraz daha fazla bellek kullanımına (aktarılan dosya başına bir bit) ve güncellenen tüm dosyaları tutacak kadar boş disk alanına gereksinim duyar. Ayrıca, yol mutlak olduğunda güncellenen dosyaların hepsi tek bir dizine konulacağından, aktarımda dosyaların herhangi birinin aynı isimde olma olasılığının olmaması durumu dışında, **--partial-dir** ile mutlak dosya yolu belirtmemelisiniz.

Ayrıca, çok daha atomik bir güncelleme algortiması içeren, **rsync** paketinin "support" alt dizinindeki "atomic-rsync" perl betiğine de bakın (**--link-dest** seçeneğini ve dosyalar için paralel bir hiyerarşi kullanır).

--progress

Aktarımın gelişimini gösteren çıktı üretilmesini sağlar. Bu, bezgin kullanıcıya bazı şeyleri kollayabilme imkanı verir. Açıkça belirtilmemişse, **--verbose** seçeneğini de uygulanır.

Bir dosya aktarılırken, üretilen çıktı şöyle görünür:

```
782448 63% 110.64kB/s 0:00:04
```

Bu çıktı kullanıcıya o anki dosya boyunu, aktarılmış verinin yüzdesini (hattaki veri ile aktarılmış verinin toplamı olarak), aktarım hızını ve aktarımın tamamlanması için gerekli kalan süreyi gösterir.

Bir dosya aktarıldıktan sonra, üretilen çıktı şöyle görünür:

```
1238099 100% 146.38kB/s 0:00:08 (5, 57.1% of 396)
```

Bu çıktı kullanıcıya aktarımı biten dosyanın son boyutunu, aktarımın %100 tamamlandığını, dosyanın son aktarım hızını, dosyanın aktarımı için kullanılan süreyi ve ek olarak toplam aktarımın boyut ve adet olarak ne kadarının tamamlandığını parantez içinde gösterir.

-P

--partial --progress seçenekleri için kısayoldur. Amacı, kesintiye uğraması muhtemel uzun bir aktarım için bu iki seçeneğin birlikte belirtilmesini kolaylaştırmaktır.

--password-file=dosya

Uzak **rsync** sunucusuna erişim için parola içeren bir dosya belirtebilmeyi sağlar. Bu seçenek, aktarım için bir uzak kabuk kullanıldığında değil, sadece yerleşik aktarım kullanarak bir **rsync** sunucusuna erişirken yararlıdır. Dosya herkesçe okunabilir olmamalı ve içinde parolayı içerek tek bir satır bulunmalıdır.

--list-only

Bu seçenek dosyaların aktarılmasına değil, kaynak dosyaların listelenmesine sebep olur. Bu seçenek, eğer bir hedef belirtilmezse çalışır, dolayısıyla doğrudan kullanmaya ihtiyaç duymazsınız. Ancak, alt dizinlere inilmeyen bir liste üretmek için tecrübeli bir kullanıcının **-r --exclude='/*/*'** seçeneklerini kullanmaktan kaçınabilmesi için bir kolaylık sağlar.

--bwlimit=kbps

Azami aktarım hızının kB/s cinsinden belirtilebilmesini sağlar. **rsync**'i çok büyük dosyaları (MB'larca) aktarmak için kullanırken oldukça verimlidir. **rsync** aktarımlarının doğası gereği, veri bloklar halinde gönderilir, eğer **rsync** aktarımın oldukça hızlı olduğunu saptarsa, sonraki bloku göndermeden önce bekler. Sonuçta, aktarım hızının ortalaması belirtilen hıza eşit olur. Sıfır değeri bir sınırlama olmadığı anlamına gelir.

--write-batch=dosya

Aynı hedefe yapılacak eşdeğer bir çağrı için **--read-batch** seçeneği ile kullanılmak üzere bir dosya oluşturur. Ayrıntılar için *BETİK KİPİ* (sayfa: 31) bölümüne bakınız.

--read-batch=dosya

Eveelce **--write-batch** ile oluşturulan *dosya* içindekileri uygular. Eğer *dosya* olarak "-" belirtilirse betik verisi standart girdiden okunur. Ayrıntılar için *BETİK KİPİ* (sayfa: 31) bölümüne bakınız.

-4**--ipv4****--6****--ipv6**

rsync'e socketleri oluştururken IPv4/IPv6 tercih etmesini söyler. Bu sadece **rsync**'in doğrudan kendi denetimi altındaki socketleri etkiler. Ayrıca, bu seçeneklerin *SUNUCU SEÇENEKLERİ* (sayfa: 23) bölümündeki açıklamasına da bakınız.

--checksum-seed=tamsayı

MD4 sağlama toplamı tohumunun *tamsayı* olarak belirtilebilmesini sağlar. Bu 4 baytlık tohum her blok ve dosyanın MD4 sağlama hesaplamasında kullanılır. Öntanımlı olarak bu tohum o anki zaman değeri olarak sunucu tarafından üretilir. Bu seçenek, yinelenebilir blok ve dosya sağlamaları isteyen uygulamalar için ya da kullanıcının daha bir rasgele tohum kullanmak istediği yerlerde kullanışlıdır. *tamsayı* olarak sıfır değeri verilirse öntanımlı tohum kullanılacaktır.

SUNUCU SEÇENEKLERİ

Bir **rsync** sunucu başlatılırken kullanılan seçenekler:

--daemon

Bu seçenek **rsync**'in artalandaki çalışacağını belirtir. Başlattığınız sunucuya bir **rsync** istemcisi `makine::modül` veya `rsync://makine/modül/` sözdizimini kullanarak erişebilir.

Eğer standart girdi bir socket ise, **rsync**, **inetd** üzerinden çalıştığını varsayacaktır, aksi takdirde, uçbirimini bırakıp bir artalan süreci haline gelecektir. **rsync** istemcileri tarafından yapılan her bağlantı isteğinde süreç, yapılandırma dosyasını (`rsyncd.conf`) okuduktan sonra isteğe yanıt verecektir. Daha fazla bilgi için *rsyncd.conf* (5) kılavuz sayfasına bakınız.

--address=adres

Öntanımlı olarak **rsync**, **--daemon** seçeneği ile başlatıldığında ya da bir **rsync** sunucuya bağlanırken bir şablon adres kullanır. Bu seçenekle kullanılacak adres olarak belli bir IP adresi (veya konakismi) belirtebilirsiniz. Bu **--config** seçeneği ile birlikte kullanılarak sanal adresleme yapabilmeyi mümkün kılar. Ayrıca, *rsyncd.conf* (5) kılavuz sayfasındaki **address** seçeneğinin açıklamasına da bakınız.

--bwlimit=kbps

Sunucu sürecinin gönderdiği veri için kilobayt/s cinsinden azami aktarım hızını belirtmek için kullanılır. İstemci bu durumda bile bir **--bwlimit** değeri belirtebilir, ancak bu istemci taraftaki band genişliğini daraltmak için yapılabilir. Eğer sunucunun belirttiği hız bu yolla aşılacak istenirse değer aşağıya, sunucunu hızına yuvarlanır. Bazı ek ayrıntılar için bu seçeneğin [istemci seçeneklerindeki açıklamasına](#) (sayfa: 23) da bakınız.

--config=dosya

Öntanımlı yapılandırma dosyasında farklı bir yapılandırma dosyası belirtmek için kullanılır. Sunucu süreci root dışında bir uzak kullanıcı tarafından, bir uzak kabuk uygulaması üzerinden çalıştırılıyorsa öntanımlı yapılandırma dosyası çalışma dizinindeki (genellikle `$HOME`) `rsyncd.conf` dosyası, aksi takdirde `/etc/rsyncd.conf`'dur.

--no-detach

Bir sunucu olarak çalışmada **rsync**'in uçbirimini bırakıp bir artalan süreci haline gelmesini engeller. Bu seçenek, **rsync**, Cygwin üzerinde bir hizmet olarak çalıştırılırken gereklidir. Ayrıca, daemon-tools ya da AIX'in Sistem Özkaynak Denetçisi gibi bir uygulama tarafından yönetildiğinde yararlıdır. Seçeneğin ayrıca, **rsync** bir hata ayıklayıcı altında çalıştırılırken de kullanılması tavsiye edilir. Bu seçenek, **rsync**, **inetd** veya **sshd** üzerinden çalışıyorsa etkisizdir.

--port=port

Sunucu tarafından öntanımlı olarak kullanılan 873. port yerine başka bir port belirtmek için kullanılır. Ayrıca `rsyncd.conf (5)` kılavuz sayfasındaki **port** seçeneğinin açıklamasına da bakınız.

-v

--verbose

Sunucu sürecinin başlatılması sırasında günlük bilgilerinin miktarını arttırmak için kullanılır. Sunucunun, istemci bağlandıktan sonraki ayrıntı seviyesi istemci tarafından kullanılan seçeneklerle ve modülün yapılandırma bölümündeki **max verbosity** değeriyle denetlenir.

-4

--ipv4

--6

--ipv6

rsync sunucusunun bağlantıları dinleyeceği soketleri oluştururken IPv4/IPv6 tercih etmesini sağlarlar. Çekirdekteki bir yazılım hatası nedeniyle Linux'un eski sürümlerinde bu seçeneklerden biri gerekli olabilir (port kullanılacağı zaman bir "adres zaten kullanımda" hatası alıyorsanız, **rsync** sunucusunu başlatırken **--ipv6** veya **--ipv4** belirtmeyi deneyin).

-h

--help

--daemon seçeneğinden sonra belirtilirse, **rsync** sunucusu başlatılırken kullanılabilecek seçenekleri kısaca açıklayan bir yardım sayfası basar.

SÜZME KURALLARI

Süzme kuralları hangi dosyaların aktarılacağını (içerileceğini – include), hangi dosyaların atlanacağını (dışlanacağını – exclude) seçmek için esnek bir yöntem sağlar. Kurallar ya doğrudan include/exclude seçenekleri ile ya da daha hassas içerme/dışlama kalıpları ile (bir dosyadan okuyarak) belirtilebilir.

rsync, aktarılacak dosya/dizinlerin listesindeki her isim için ayrı ayrı içerme/dışlama kurallarını tarar ve ilk eşleşen kalıbı uygular: bir dışlama kalıbıyla eşleşen dosya atlanır; bir içerme kalıbı ile eşleşen dosya atlanmaz; eşleşen bir kalıp yoksa dosya atlanmaz.

rsync, komut satırında belirtilen include/exclude seçeneklerinden sıralı bir süzme kuralları listesi oluşturur. Süzme kurallarının sözdizimi şöyledir:

```
kural [kalıp_veya_dosyaismi]
kural, belirteç [kalıp_veya_dosyaismi]
```

kural ismi olarak, aşağıda açıklanan kısa ya da uzun kural isimlerinden birini kullanabilirsiniz. Eğer kısa isimli bir kural kullanırsanız, isteğe bağlı olan *belirteç*'ten *kuralı* bir ' ' ile ayırın. *kalıp* veya *dosyaismi*'ni (varsa) bir boşluk veya altçizgiden sonra yazmalısınız. Kullanılabilecek kural örnekleri:

uzun	kısa	açıklama
exclude	-	dışlama kalıbı belirtir
include	+	içerme kalıbı belirtir.
merge	.	ek kuralları okumak için katıştırılacak dosya,
dir-merge	:	bir dizin içi katıştırma dosyası belirtir.
hide	H	aktarımdan gizlenecek dosyalar için kalıp.
show	S	aktarımdan gizlenmeyecek dosyalar için kalıp.
protect	P	silinmekten koruncak dosyalar için kalıp.
risk	R	silinmekten korunmayacak dosyalar için kalıp.
clear	!	o anki include/exclude listesini temizler (argümentsiz)

Kurallar bir dosyadan okunduğunda, "#" ile başlayan açıklama satırları ile boş satırlar yoksayıılır.

--include ve **--exclude** komut satırı seçenekleri yukarıdaki kural sözdizimini değil, sadece kalıplarla ek olarak listeyi temizleyen ünlem (!) imini (ve kalıplar bir dosyadan okunuyorsa boş ve açıklama satırlarını) kabul ederler. Bir include seçeneği için, bir kalıp "- " (tire ve boşluk) veya "+ " (artı ve boşluk) ile başlamıyorsa, kural bir "+ " olarak değerlendirilir. Bir **--exclude** seçeneği için bir kalıp bir "- " ile başlamalıdır.

Bir **--filter** seçeneği ise kuralın başlangıcında daima bir uzun ya da kısa kural ismi içermelidir.

Ayrıca, **--filter**, **--include** ve **--exclude** seçenekleri tek bir kalıp/kural alabilir. Çok sayıda kalıp/kural eklemek için komut satırında bu seçenekleri tekrarlayabilir veya **--filter**, **--include-from** ve **--exclude-from** seçeneklerinin dosya katıştırma sözdizimini kullanabilirsiniz.

İçerme/Dışlama Kalıpları

Yukarıda [SÜZME KURALLARI](#) (sayfa: 24) bölümünde bahsedildiği gibi dosyaları "+", "-" gibi kalıpları kullanarak aktarabilir ya da atlayabilirsiniz. İçerme ve dışlama kurallarının herbirinde aktarılacak dosyaların isimleriyle eşleşen kalıplar belirtilir. Bu kalıplar çeşitli biçimler alabilir:

- Eğer kalıp bir / ile başlıyorsa, dosya hiyerarşisi içinde belli bir noktaya demirlenir, aksi takdirde yol isminin sonuna eşlenir. Bu düzenli ifadelerde kalıbı ^ ile başlatmaya benzer. Örneğin, **"/foo**" kalıbı ya "aktarımın kökü"ndeki (genel bir kural için) ya da katıştırılan dosyanın dizinindeki (**dir-merge** kuralı için) **foo** isimli bir dosya ile eşleşirdi. Nitelenmemiş bir **foo** ise, ağacın herhangi bir yerindeki **foo** isimli bir dosya veya dizin ismi ile eşleşirdi, çünkü algoritma kökten uca doğru alt dizinlere inilerek uygulanır; yani, her yol elemanı sonunda dosya ismi olacak bir sırada ele alınır. **sub/foo** ise hiyerarşinin herhangi bir noktasındaki **sub** isimli bir dizin içindeki bir **foo** ile eşleşirdi. Aktarımın kökü ile eşleşen bir kalıbın nasıl belirtileceğinin ayrıntılı açıklaması için [İçerme/Dışlama Kalıplarının Kök Dizinle İlişlendirilmesi](#) (sayfa: 29) bölümüne bakınız..
- Eğer kalıp bir / ile bitiyorsa, sadece bir dizinle eşleşecektir, bir dosya, bağ ya da aygıtla değil.

- Eğer kalıp ***?[]** dosyaismi kalıp karakterlerinden birini içeriyorsa, ifade kabuğun dosyaismi eşleştirme kuralları kullanılarak eşleştirilir. Aksi takdirde basit bir dizge eşleme kullanılır.
- Tek yıldızlı bir kalıp ("*****") bölü çizgilerinde duracakken, çift yıldızlı kalıp ("******") çok sayıda bölü çizgisi ile eşleşecektir.
- Eğer bir kalıp bir **/** (sonda değil) ya da bir "*****" içeriyorsa baştan sona tüm dizinleri dahil tam dosya yolu ile eşleştirilir. Eğer kalıp bir **/** ya da "******" içermiyorsa, sadece dosyaisminin son bileşeni ile eşleştirilir (algoritmanın altdizinlere inilerek uygulandığını hatırlayın, yani "tam dosyaismi" başlangıç dizininden başlayarak son dizine kadar yolun herhangi bir parçası ile eşleşebilir).

Şuna dikkat edin, **--recursive (-r)** seçeneğini (**-a** ile uygulanan) kullandığınızda, her yolun her alt bileşeni yukarıdan aşağıya ziyaret edilir, yani içirme/dışlama kalıpları altdizinlere inilerek her alt bileşenin tam ismine uygulanır (örn, `/foo/bar/baz` içerilirken `/foo` ve `/foo/bar` dışlanamaz). **rsync** göndereceği dosyaları ararken, dışlama kalıpları aslında dizin basamaklama aşamasını kısıtladığı eder. Eğer bir kalıp belli bir üst dizini dışlıyorsa, bir daha derin içirme kalıbını sonuçsuz bırakabilir, çünkü **rsync**, hiyerarşinin dışlanan bir bölümünden daha derine inmez. Bu özellikle "*****" ile biten bir kural kullanırken önem kazanır. Örneğin, bu çalışmayacaktır:

```
+ /bir/yol/bu-dosya-bulunmaz
+ /bu-dosya-bulunur
- *
```

Bu kural listesi, "bir" üst dizini '*****' kuralından dolayı dışlayacağından başarısız olur, yani **rsync**, "bir" ve "bir/yol" dizinleri içindeki hiçbir dosyayı ziyaret etmez. Hiyerarşideki tüm dizinleri ziyaret ettirmenin bir yolu tek bir kuralın listeye dahil edilmesidir: `"+ */"` ("**- ***" kuralından önce herhangi bir yere yerleştirebilirsiniz). Diğer bir çözüm ise ziyaret edilmesi gereken üst dizinlerin hepsi için belirli bir içirme kuralı eklemektir. Örneğin, bu kural kümesi doğru çalışır:

```
+ /bir/
+ /bir/yol/
+ /bir/yol/bu-dosya-bulunur
+ /bu-dosya-da-bulunur
- *
```

Dışlama/içirme eşleştirme örnekleri:

- `"- *.o"` kalıbı `*.o` ile eşleşen tüm dosya isimlerini dışlar.
- `"- /foo"` kalıbı aktarımın kök dizinindeki `"foo"` isimli bir dosyayı dışlar.
- `"- foo/"` kalıbı `"foo"` isimli herhangi bir dizini dışlar.
- `"- /foo/*/bar"` kalıbı aktarımın kök dizinindeki `"foo"` dizininden iki seviye aşağıdaki `"bar"` isimli dosyayı dışlar.
- `"- /foo/**/bar"` kalıbı aktarımın kök dizinindeki `"foo"` dizininden iki veya daha alt seviyedeki `"bar"` isimli dosyayı dışlar.
- `"+ */"`, `"+ *.c"` ve `"- *` kalıplarının birleşimi tüm dizinler ve C kaynak dosyaları dışında herşeyi dışlar.
- `"+ foo/"`, `"+ foo/bar.c"` ve `"- *` kalıplarının birleşimi sadece `"foo"` dizinini ve `"foo/bar.c"` dosyasını içerecektir (`"foo"` dizini açıkça içerilmelidir yoksa "*****" tarafından dışlanırdı).

Katıştırma Dosyası Süzme Kuralları

Yukarıda [SÜZME KURALLARI](#) (sayfa: 24) bölümünde bahsedildiği gibi dosyaları bir **merge** (.) ya da bir **dir-merge** (:) süzme kuralı belirterek süzme kurallarınıza tüm dosyaları katıştırabilirsiniz.

İki çeşit katıştırma dosyası vardır: tek dosyalık (':') ve diziniçi dosyalar ('::'). Tek dosyalık bir katıştırma dosyası bir defa okunur ve "." kuralındaki süzme listesi ile birlikte işleme sokulur. Diziniçi katıştırma dosyası türünde ise, **rsync** adım adım bütün altdizinleri tarayarak ismi belirtilen dosyayı buldukça dosya içeriğini o anki kural listesine ekler. Bu diziniçi kural dosyaları aktarım alt dizinlerinde aranması gerektinden gönderen tarafta oluşturulmalıdır. Ayrıca, bu kural dosyalarının silinmeyecek dosyaları etkilemesi isteniyorsa, alıcı tarafa aktarılmalı gerekecektir (aşağıda, [Diziniçi Katıştırma Kuralları ve Silme](#) (sayfa: 30) bölümüne bakınız).

Bazı örnekler:

```
merge /etc/rsync/default.rules
. /etc/rsync/default.rules
dir-merge .per-dir-filter
dir-merge,n- .non-inherited-per-dir-excludes
:n- .non-inherited-per-dir-excludes
```

Bir **merge** veya **dir-merge** kuralı şu belirteçleri kabul eder:

- Bir **-** belirteci, dosyaiçi açıklamalar hariç başka bir kural çözümleme olmaksızın, sadece dışlama kalıplarından oluşan bir dosya belirtir.
- Bir **+** belirteci, dosyaiçi açıklamalar hariç başka bir kural çözümleme olmaksızın, sadece içirme kalıplarından oluşan bir dosya belirtir.
- Bir **C** belirteci, bir dosyanın CVS uyumluluğu anlamında okunması gerektiğini belirtir. Bu **'n'**, **'w'**, ve **'-'** belirteçlerini etkin kılar, fakat ayrıca, belirtilecek liste temizleme imine de (!) izin verir. Bir dosyaismi belirtilmemişse dosya isminin **".cvsignore"** olduğu varsayılır.
- Bir **e** belirteci, katıştırma dosyasını aktarımdan dışlayacaktır; örneğin, **"dir-merge,e .rules"** kuralı **"dir-merge .rules"** ve **"- .rules"** kuralları gibidir.
- Bir **n** belirteci, kuralların altdizinlerde miras alınmayacağını belirtir.
- Bir **w** belirteci, kurallar için satırlara göre değil boşluklara göre sözcük ayırma uygulanacağını belirtir. Bu ayrıca açıklamalara da izin vermeyecektir.



Bilgi

Bir kuralın öneki olan boşluklar özel olarak ele alınır, yani **"- foo + bar"** dizgesi iki kurala ayrılacaktır (ek olarak önek çözümlemesinin iptal edilmediği varsayımıyla).

- Ayrıca, **+** veya **-** kuralı için aşağıdaki belirteçlerden herhangi birini belirtebilirsiniz. Örneğin, **"dir-merge,s .filt"** ve **":sC"** kurallarının herbiri sadece sunucu tarafında uygulanacak diziniçi katıştırma kurallarını oluştururken, **"merge,-/ .excl"** kuralıyla **.excl** dosyasının içeriği mutlakyol dışlamaları olarak ele alınır.

Bir **+** veya **-** kuralından sonra kullanılabilecek belirteçler:

- Bir **/** belirteci kuralın dosya sisteminin köküne göre bir mutlak yol olarak ele alınacağını belirtir. Örneğin, **"-/ /etc/passwd"** kuralı aktarım sırasında **/etc** dizininden gönderilen dosyalardan **passwd** dosyasının dışlanacağını belirtir.

- Bir "**!**" belirteci kuralın, kalıbın eşleşmediğinde etkili olacağını belirtir. Örneğin, "**-! */**" kuralı dizin olmayan herşeyi dışlayacaktır.
- Bir "**C**" belirteci, tüm genel CVS dışlama kurallarının yerinde "**-C**" dışlamaları olarak yerleştirileceğini belirtmek için kullanılır. Bir argüman almaz.
- Bir "**s**" belirteci, kuralın sadece gönderen tarafta uygulanacağını belirtir. Bir kural gönderen tarafta etkin olduğunda, dosyaların aktarılmasını engeller. **--delete-excluded** belirtilmedikçe, bir kuralın her iki tarafı da etkilemesi öntanımlıdır, aksi takdirde kuralın sadece gönderen tarafı etkilemesi öntanımlıdır. Sunucu tarafında etkili içirme/dışlama kuralları belirtmenin diğer yolları için **hide (H)** ve **show (S)** kurallarına da bakınız.
- Bir "**C**" belirteci, kuralın alıcı tarafa uygulanacağını belirtir. Bir kural alıcı tarafta etkili olduğunda, dosyaların silinmesini engeller. Daha fazla bilgi için üstteki "**s**" belirtecine bakınız. Ayrıca, alıcı tarafta etkili içirme/dışlama kuralları belirtmenin diğer yolları için **protect (P)** ve **risk (R)** kurallarına da bakınız.

dir-merge kuralları **n** belirteci kullanılmadıkça katıştırma dosyasının bulunduğu her alt dizinde miras alınır. Her alt dizinin kuralları, miras alınan kurallara göre daha yüksek öncelik almak üzere üst dizindeki **dir-merge** kurallarının başına eklenir. Birleşik kural kümesi katıştırma dosyasının bulunduğu noktalara göre gruplanır, böylece genel kurallar listesinde evvelce belirtilmiş bir kuralın **dir-merge** kuralları ile değiştirilmesi mümkün olur. Bir diziniçi katıştırma dosyasında bir liste temizleme kuralına rastlandığında, sadece o katıştırma dosyasının bulunduğu noktaya kadar miras alınan kurallar temizlenir.

Bir tek dosyalık kuralın bir **dir-merge** dosyasından miras alınmasını engellemenin diğer bir yolu, dosyanın başına bir **/** getirmektir. Bir diziniçi katıştırma dosyasındaki bu tür kurallar katıştırma dosyasının bulunduğu dizine göre ele alınır, böylece bir **"/foo"** kalıbı **dir-merge** süzme kuralı dosyasının bulunduğu dizindeki **"foo"** dosyasıyla eşleşir.

--filter=". dosya" üzerinden belirteceğiniz örnek süzme dosyası:

```
merge /home/user/.global-filter
- *.gz
dir-merge .rules
+ *. [ch]
- *.o
```

Bu, bu listenin başına **/home/user/.global-filter** dosyasının içeriğini ve ayrıca, bunu da alt dizinde **".rules"** dosyasının sonuna ekleyecektir. Her dizinin başında önce tüm kurallar okunup üst dizinden alınan genel kurallar buna eklenir.

Eğer bir diziniçi katıştırma dosyası, ilk aktarım dizininin üst dizini olan bir dosya yolu ile belirtilmişse, **rsync** belirtilen dosya için, aktarım dizininin başlangıcından itibaren tüm üst dizinleri tarayacaktır. Örneğin, böyle bir süzgeç (**-F** seçeneğine bakınız),

```
--filter=': /.rsync-filter'
```

rsync'e **.rsync-filter** dosyasını kökten aşağı doğru, aktarımın bir parçası olarak gönderilen dizinlerde dosya normal taramaya başlamadan önce aktarımın üst dizininden itibaren tüm dizinlerde aramasını söyler (Bilgi: bir **rsync** sunucusu için kök dizin daima modülün dizinidir).

Diziniçi katıştırma dosyaları için bu tür öntarama örnekleri:

```
rsync -avF /src/path/ /dest/dir
rsync -av --filter=': ../../.rsync-filter' /src/path/ /dest/dir
rsync -av --filter=': .rsync-filter' /src/path/ /dest/dir
```

İlk iki komut ".rsync-filter" dosyasını "/src/path" içinde aramaya başlamadan önce "/" ve "/src" dizinlerinde arar. Son komut ise, üst dizin taramasından kaçınır ve ".rsync-filter" dosyasını sadece aktarımın parçası olan dizinlerde arar.

Bir ".cvsignore" dosyasının içeriğini kalıplarınıza dahil etmek isterseniz, ":C" kuralını kullanmalısınız. Bu kural, .cvsignore dosyalarını diziniçi katıştırma dosyaları haline getirir, fakat dosya CVS uyumluluğu bağlamında ayrıştırılır. **--cvs-exclude** (-C) seçeneğinin bulunduğu yerde etkili olmasını istiyorsanız kurallarınızın arasına bu noktada ":C" kuralını yerleştirebilirsiniz. Bu olmaksızın, **rsync** .cvsignore dosyası için **dir-merge** kuralını diğer kurallarınızın sonuna ekleyecektir (komut satırı seçeneklerinizden daha düşük bir öncelik vererek). Örnek:

```
cat <<EOT | rsync -avC --filter='.' -' a/ b
+ foo.o
:C
- *.old
EOT
rsync -avC --include=foo.o -f :C --exclude='*.old' a/ b
```

Yukarıdaki **rsync** komutlarının ikisi de aynı işi yapar. Her ikisi de diziniçi .cvsignore kurallarını listenin sonuna değil, ortasına katıştıracaktır. Bu, onların dizine özel kurallarının sizin kurallarınızın tümünden daha düşük önemde olması yerine :C'ye kadar olan kuralların yerine geçmesini mümkün kılar. Diğer CVS dışlama kurallarını etkilemek için (örn, öntanımlı dışlananlar listesi, \$HOME/.cvsignore dosyasının içeriği ve \$CVSIGNORE değişkeninin değeri), **-C** komut satırı seçeneğini belirtmemeli, onun yerine kurallarınıza bir **"-C"** kuralı yerleştirmelisiniz; örn, **"--filter=-C"**.

Liste Temizleyen Süzme Kuralı

Yukarıda **SÜZME KURALLARI** (sayfa: 24) bölümünde bahsedildiği gibi mevcut içirme/dışlama listesini "!" süzme kuralını kullanarak temizleyebilirsiniz. "Mevcut" liste ya genel kural listesi (kural, süzme seçenekleri çözümlenirken saptanmışsa) ya da diziniçi katıştırma dosyası kurallarından (bunlar kendi alt listelerinde miras alınırlar, dolayısıyla üst dizinin kurallarını bu kuralla temizleyebilirsiniz) oluşur.

İçirme/Dışlama Kalıplarının Kök Dizine İlişkilendirilmesi

Evvelce de bahsedildiği gibi, genel içirme/dışlama kalıpları aktarımın kök dizini ile ilişkilendirilebilir (katıştırma dosyasının dizini ile ilişkilendirilen diziniçi kalıplarının tersine). Aktarımı, göndericiden alıcıya bir isim altağacının gönderilmesi olarak düşünürseniz, aktarımın kök dizini hedef dizinde yinelenen olan ağaç başlangıcıdır. Bu kök, bir / ile başlayan kalıplarla ilişkilendirilir.

Eşleşme aktarımın köküne göre yapıldığından, kaynak yolun sonundaki bölü çizgisini değiştirmek ya da **--relative** seçeneğinin kullanımında değişiklik yapmak eşleştirmenizde kullanılması gereken yolu etkiler. Aşağıdaki örneklerde bu davranış modellenmiştir.

Biri "/home/ben/foo/bar", diğeri "/home/sen/bar/baz" olan iki dosyamız olsun. Aşağıda bu iki dosyanın çeşitli komut kullanımlarında nasıl farklı aktarıldıkları gösterilmiştir.

```
Örnek komut: rsync -a /home/ben /home/sen /hedef
+/- kalıbı: /ben/foo/bar
+/- kalıbı: /sen/bar/baz
Hedef dosya: /hedef/ben/foo/bar
```

```
Hedef dosya: /hedef/sen/bar/baz
```

```
Örnek komut: rsync -a /home/ben/ /home/sen/ /hedef
+/- kalıbı: /foo/bar          (dikkat: "ben" yok)
+/- kalıbı: /bar/baz          (dikkat: "sen" yok)
Hedef dosya: /hedef/foo/bar
Hedef dosya: /hedef/bar/baz
```

```
Örnek komut: rsync -a --relative /home/ben/ /home/sen /hedef
+/- kalıbı: /home/ben/foo/bar    (dikkat: tam dosyayolu)
+/- kalıbı: /home/sen/bar/baz    (keza bu da)
Hedef dosya: /hedef/home/ben/foo/bar
Hedef dosya: /hedef/home/sen/bar/baz
```

```
Örnek komut: cd /home; rsync -a --relative ben/foo sen/ /hedef
+/- kalıbı: /ben/foo/bar        (belirtilen yolla başlar)
+/- kalıbı: /sen/bar/baz        (keza bu da)
Hedef dosya: /hedef/ben/foo/bar
Hedef dosya: /hedef/sen/bar/baz
```

Hangi ismi süzdüğünüzü görmenin en kolay yolu, **--verbose** kullanarak çıktıya bakmak ve ismin önüne bir / koymaktır (herhangi bir dosya kopyalamaya hazır değilseniz, **--dry-run** seçeneğini kullanın).

Diziniçi Katıştırma Kuralları ve Silme

Bir silme seçeneği olmaksızın, diziniçi katıştırma dosyası kuralları sadece gönderen tarafta konuyla ilgilidir, dolayısıyla aktarımı etkilemeksizin rahatça katıştırma dosyalarını dışlayabilirsiniz. Bunu kolayca yapmanın yolu, örnekte görüldüğü gibi bu dışlamaya 'e' belirtecini eklemektir:

```
rsync -av --filter=': .haric' --exclude=.haric mkn:kaynak/dizin /hedef
rsync -av --filter=':e .haric' mkn:kaynak/dizin /hedef
```

Buna rağmen, alıcı tarafta bir silme yapmak VE bazı dosyaların da silinerek dışlanmasını isterseniz, alıcı tarafın dışlanacak dosyaları bilmesini sağlamanız gerekecektir. En kolayı, aktarıma diziniçi katıştırma dosyalarını dahil etmek ve **--delete-after** kullanmaktır. Bu yolla, birşeyleri silmeden önce alıcı tarafta gönderen taraftaki dışlama kuralları alınmış olacaktır:

```
rsync -avF --delete-after mkn:kaynak/dizin /hedef
```

Yine de, katıştırma dosyaları aktarımın bir parçası değilse, ya bazı genel dışlama kuralları belirtmeli (komut satırında belirtilebilir) ya da alıcı tarafta diziniçi katıştırma dosyalarının bulunmasını kendiniz sağlamalısınız. İlkine bir örnek (uzak .kural dosyalarının kendilerini dışladıkları varsayılıyor):

```
rsync -av --filter=': .kural' --filter=': /benim/ek.kuralım'
--delete mkn:kaynak/dizin /hedef
```

Bu örnekte ek.kuralım dosyası aktarımın her iki tarafını da etkileyebilir, fakat (gönderen tarafta) kurallar diziniçi katıştırma kuralından sonra belirtildiğinden .kural dosyasından katıştırılan kurallardan daha düşük öncelikli olurlar

Son örnekte, uzak taraf .rsync-filter dosyalarını aktarımdan dışlamaktadır, ancak biz alıcı tarafta neyin silineceğini denetim altına almak için kendi .rsync-filter dosyalarımızı kullanmak istiyoruz. Bunu yapmak için diziniçi katıştırma dosyalarını özellikle dışlamalı (böylece silinmezler) ve neyin silinip silinmeyeceğini denetim altına almak için kuralları yerel dosyalara koymalıyız. Bu komutlardan biri gibi:

```
rsync -av --filter=':e /.rsync-filter' --delete \
    mkn:kaynak/dizin /hedef
rsync -avFF --delete mkn:kaynak/dizin /hedef
```

BETİK KİPİ

Betik kipi bir çok sistem için aynı olan güncellemeleri uygularken kullanılabilir. Bir dizin ağacının çok sayıda makine üzerinde kopyalarının bulunduğunu varsayalım. Şimdi, kaynak ağacında bazı değişiklikler olduğunu ve bunların diğer makinelere de aktarılması gerektiğini varsayalım. Bu işlemi betik kipini kullanarak yapmak için, **rsync**, kaynak ağacındaki değişiklikleri hedef ağaçlardan birinde uygulayan **--write-batch** seçeneği ile çalıştırılır. **--write-batch** seçeneği, yapılan işlemlerin aynı dizin ağacını içeren hedeflerde uygulanmak üzere bir "betik dosyası" içinde saklanmasını sağlar.

Kaydedilen değişiklikleri diğer hedef ağaçlara uygulamak için **rsync**, bu dosyanın ismi seçeneğe argüman olarak belirtilerek **--read-batch** seçeneği ile çalıştırılır. **rsync**, betik dosyasında saklanmış bilgileri kullanarak hedef ağacı günceller.

Rahat olsun diye, **--write-batch** seçeneği kullanılırken ek bir dosya daha oluşturulur. Bu dosyanın ismi, betik dosyasının isminin sonuna **".sh"** eklenerek oluşturulur. Bu **.sh** dosyası betik dosyasını kullanarak hedefi güncelleyecek komut satırını içerir. Bir Bourne(–benzeri) kabuk kullanılarak çalıştırılabilir. İsteğe bağlı olarak özgün yol yerine hedef ağacın dosya yolu aktarılabilir. Bu, hedef ağacın kaynak ağacının yolundan farklı olduğu durumlarda yararlıdır.

Dosya durumu, sağlama toplamı sınanması ve veri bloklarının üretimi işlemleri betik dosyasına bir kere kaydedildi mi, hedef ağaçlarda defalarca kullanılabilir. Aynı verileri her konağa ayrı ayrı göndermek yerine çok sayıda konağı aynı anda güncelleyecek çok görevli aktarım protolları de kullanılabilir.

Örnekler:

```
$ rsync --write-batch=foo -a mkn:/kaynak/dzn/ /hedef1/dzn/
$ scp foo* uzak:
$ ssh uzak ./foo.sh /hedef2/dzn/

$ rsync --write-batch=foo -a /kaynak/dzn/ /hedef1/dzn/
$ ssh remote rsync --read-batch=- -a /hedef2/dzn/ <foo
```

Bu örneklerde, **rsync** **/kaynak/dzn/**'den **/hedef1/dzn/**'i güncellemekte kullanılmış ve bu işlemi yinelemek için işlemler **"foo"** ve **"foo.sh"** dosyalarına kaydedilmiştir. **"uzak"** konağı betiğe kayıtlı veri **/hedef2/dzn/** dizinine gönderilerek güncellenir. Bu iki örnek arasındaki farklar betik dosyalarının hangi esneklikte kullanılabileceğini göstermek içindir:

- İlk örnek ilk kopyanın yerel olmayacağını gösterir — her iki yönde de aktarım için ister uzak kabuk sözdizimini, ister **rsync** sunucu sözdizimini kullanabilirsiniz.
- İlk örnek uzak makine üzerinde **--read-batch** seçenekli komutu çalıştırırken doğru **rsync** seçeneklerini oluşturulan **"foo.sh"** dosyasından alır.
- İkinci örnek betik verisini standart girdiden okur, böylece betik dosyasının önce uzak makineye kopyalanması gerekmez. Bu örnekte değiştirilmiş bir **--read-batch** seçeneği kullanılmasını gerektirdiğinden **"foo.sh"** betiği kullanılmamıştır, fakat betiği kullanmak için dosyayı kendiniz düzenleyebilirsiniz. (Ancak, başka bir seçeneğin (**--exclude-from=-** gibi) standart girdiyi kullanmadığından emin olmalısınız.)

Yetersizlikler

Betik okuma seçeneği, hedef ağacın, betiğin oluşturulduğu hedefteki ağaçla aynı olacağı kabulüne dayanır. Hedef ağaçta bir fark saptandığında işlem ya bir hata verilmeksizin iptal edilir (dosya güncel görünüyorsa) ya da dosya güncellemesi yapılmaya çalışılır ve bundan sonra eğer dosyanın doğrulanması başarısız olursa işlem bir hata üretilerek iptal edilir. Bu, eğer işlem kesintiye uğratılırsa bir betik okuma işleminin yeniden başlatılmasının güvenli olması gerektiği anlamına gelir. Eğer betikli güncellemenin dosyanın boyutuna ve tarihine bakılmaksızın daima yapılabilmesini isterseniz, **-I** seçeneğini kullanın (betik okunurken). Eğer bir hata oluşursa, hedef ağaç bir ihtimal, kısmen güncellenmiş durumda olacaktır. Bu durumda, **rsync** betikli güncelleme ile değil, normal işlemle kullanılarak hedef ağaç düzeltilebilir.

Tüm hedeflerde kullanılan **rsync** sürümü, betik dosyasını üretmekte kullanılan en azından biraz daha yeni olmalıdır. Eğer, betik dosyasının protokol sürümü onu okuyan **rsync**'in sürümünden daha yeni ise **rsync** bir hata vererek sonlanacaktır.

--dry-run (-n) seçeneği betik kipinde çalışmaz ve bir çalışma anı hatası ile sonlanır.

Bir betik dosyası okunurken, siz belirtmemiş olsanız bile, betik dosyası üretilirken kullanılan ve betik dosyasındaki veriler için gerekli olan bazı seçenekler zorunlu olarak kullanılacaktır. Diğer seçenekler değiştirilebilir, hatta değiştirilmelidir de. Örneğin, **--write-batch** seçeneği **--read-batch** seçeneği olarak değişmeli, **--files-from** kullanılmamalıdır, bunun yanında **--filter/--include/--exclude** seçenekleri ise **--delete** seçeneklerinden biri belirtilmedikçe gerekli olmaz.

betik.sh dosyasını oluşturan kod, kabuk betiğinin sonuna tek bir listeye dönüştürerek **--filter/--include/--exclude** seçeneklerini bir belge niteliğinde ekler. Tecrübeli bir kullanıcı bu bilgiyi **--delete** tarafından silinmesi istenenlerde bir değişiklik varsa, dışlama listesini değiştirmekte kullanabilir. Normal bir kullanıcı ise, bu ayrıntıları yoksayabilir ve betikli veriyi uygun **--read-batch** komutuyla çalıştıran kabuk betiğini kullanabilir.

rsync'deki özgün betik kipi "**rsync+**" üzerine inşa edilmiş olsa da en son sürümde yeni bir gerçeklenim kullanılmaktadır.

SEMBOOLİK BAĞLAR

rsync kaynak dizini içinde bir sembolik bağ saptadığında üç temel davranış söz konusudur:

- Öntanımlı olarak, sembolik bağlar oldukları gibi aktarılmazlar. Bir sembolik bağ varsa, "skipping non-regular file" (normal dosya olmadığından atlanıyor) iletisini alırsınız ve dosya atlanır.
- Eğer **--links** seçeneği belirtilmişse, sembolik bağlar alıcı tarafta aynı hedefle yeniden oluşturulurlar. BİLGİ: **--archive** seçeneği dolaylı olarak **--links** seçeneğini uygular.
- Eğer **--copy-links** seçeneği belirtilmişse, alıcı tarafa sembolik bağın ismiyle hedefindeki dosya aktarılır.

rsync ayrıca, "güvenilir" ve "güvensiz" sembolik bağları ayımsayabilir. Bunun kullanıldığı yere bir örnek olarak, bir sanalyöre yansılama yapılrken, yörenin genel bölümündeki **/etc/passwd** sembolik bağının aktarımda içerilmediğinden emin olunmak istenmesi gösterilebilir. **--copy-unsafe-links** seçeneğinin kullanımı sembolik bağların hedefindeki dosyaların kopyalanmasına sebep olacaktır. **--safe-links** seçeneğinin kullanımı ise güvensiz bağların atlanmasına sebep olacaktır.

Mutlak yol içeren (/ ile başlayan) sembolik bağlarla, bulunulan dizinden daha yukarıdaki dizinleri gösteren ". ." bileşenlerini içeren sembolik bağlar güvensiz olarak kabul edilir.

TANI KOYMA

rsync ara sıra biraz şifreli gibi görünen hata iletileri üretir. Bunlardan biri en kafa karıştııcı olan "protocol version mismatch — is your shell clean?" (protokol sürümü yanlış — kabuğunuz temiz mi?) iletisidir.

Bu ileti genellikle, **rsync**'in aktarım için kullandığı akım üzerinde istenmeyen veriler üreten uzak kabul oluşumları ya da başlatma betiklerinizden kaynaklanır. Bu soruna tanı koymanın yolu, uzak kabuğu şöyle çalıştırmaktır:

```
ssh uzakmakine /bin/true > out.dat
```

Bundan sonra oluşturulan `out.dat` dosyasına bakın. Herşey yolunda gitmişse dosya sıfır uzunlukta olacaktır. Eğer yukarıdaki hatayı alıyorsanız, dosya bazı veriler içerecektir. Dosyanın içeriğine bakarak bunları nelerin üretebileceğini bulmaya çalışın. Çok yapılan bir hata, etkileşimsiz oturum açmak için komutlar içeren yanlış yapılandırılmış kabuk betikleridir (`.cshrc` veya `.profile` gibi).

Süzme kalıplarında hata ayıklamak istiyorsanız, **-vv** seçeneğini belirtmeyi deneyin. **rsync**, bu ayrıntı seviyesinde dosyaların tek tek neden içerildiğini veya dışlandığını gösterecektir.

ÇIKIŞ DEĞERLERİ

- 0 Başarılı
- 1 Sözdizimi ya da kullanım hatası
- 2 Protokol uyumsuzluğu
- 3 girdi/çıkı dosyalarını, izinlerini seçerken oluşan hatalar
- 4 İstenen eylem desteklenmiyor; 64 bitlik dosyaları desteklemeyen bir platformda bu tür dosyalarla çalışılmak isteniyor; istemci tarafından desteklendiği halde sunucu tarafından desteklenmeyen bir seçenek kullanılmış.
- 5 Sunucu/istemci protokolü başlatılırken hata.
- 10 Soket G/Ç hatası
- 11 Dosya G/Ç hatası
- 12 **rsync** protokolü veri akımında hata
- 13 Uygulama tanılarında hata
- 14 Süreçlerarası iletişim (IPC) kodunda hata
- 20 Bir SIGUSR1 veya SIGINT alındı
- 21 waitpid() bazı hatalar döndürdü
- 22

nüve bellek tamponları ayrılırken hata

23

Hatadan dolayı kısmî aktarım

24

Gözden kaçan kaynak dosyalarından dolayı kısmî aktarım

30

Veri aktarımında zamanaşımı

ORTAM DEĞİŞKENLERİ

CVSIGNORE

`.cvsignore` dosyalarındaki gibi yoksayılacak kalıpları belirtmek için kullanılır. Daha fazla bilgi için **--cvs-exclude** seçeneğinin açıklamasına bakınız.

RSYNC_RSH

rsync tarafından kullanılan öntanımlı kabuğun yerine kullanılacak kabuğu belirtmek için kullanılır. Komut isminden sonra komut satırı seçeneği olarak sadece **-e** seçeneği kullanılabilir.

RSYNC_PROXY

rsync sunucusuna bağlanırken **rsync** istemcisi tarafından kullanılacak vekil sunucuyu belirtmek için kullanılır. Değeri bir `konakadı:port` çifti olarak belirtmelisiniz.

RSYNC_PASSWORD

Kimlik kanıtlaması gerektiren bir **rsync** sunucusuna bağlanırken komut satırında parola belirtmek zorunda kalmamak için parola bu değişkene atanabilir. Aktarım **ssh** gibi bir kabuk üzerinden gerçekleştiriliyorsa bu değişkene bakılmayacaktır.

USER veya LOGNAME

USER veya LOGNAME değişkenleri bir **rsync** sunucusuna gönderilecek kullanıcı ismini saptamak için kullanılır. Böyle bir tanım yapılmamışsa, öntanımlı kullanıcı ismi olarak `"nobody"` kullanılır.

HOME

Kullanıcının öntanımlı `.cvsignore` dosyasını bulmakta kullanılır.

İLGİLİ DOSYALAR

`/etc/rsyncd.conf` veya `rsyncd.conf`

İLGİLİ BELGELER

rsyncd.conf(5)

HATALAR

Zamanlar unix `time_t` türünden değerler olarak aktarılır.

Dosyalar FAT dosya sistemine aktarılırken, **rsync** dosyalarda değişiklik yapmayabilir. **--modify-window** seçeneğinin açıklamasına bakınız.

Dosya izinleri, aygıtlar, vs. temel sayısal değerleriyle aktarılır.

Ayrıca, **--delete** seçeneğinin açıklamasına da bakınız.

Lütfen hataları raporlayınız! Hataları raporlamak için <http://rsync.samba.org/> adresine bakınız.

RSYNC HAKKINDA

rsync GNU kamu lisansı altında dağıtılmaktadır. Ayrıntılar için **COPYING** dosyasına bakınız.

Sanalyöremize <http://rsync.samba.org/> adresinden erişebilirsiniz. Bu kılavuz sayfasında yanıtlarını bulamadığınız sorular için sanalyöremizde bir FAQ-O-Matic bulunmaktadır.

rsync'in ana ftp'si: <ftp://rsync.samba.org/pub/rsync>.

Bu uygulamayı istiyorsanız, sizden bunu duymak bizi mennun edecektir.

Bu uygulamada, Jean-loup Gailly ve Mark Adler tarafından yazılmış zlib sıkıştırma kütüphanesi kullanılmaktadır.

TEŞEKKÜR

rsync'in yamanması, sınanması ve önerileri için Richard Brent, Brendan Mackay, Bill Waite, Stephen Rothwell ve David Bell'e teşekkür ederiz. Burada bahsetmeyi unuttuğum kişiler kalmışsa, affola.

Ayrıca, David Dykstra, Jos Backus, Sebastian Krahmer, Martin Pool, Wayne Davison ve J.W. Schultz'a özellikle teşekkür ederiz.

YAZANLAR

rsync ilk olarak Andrew Tridgell ve Paul Mackerras tarafından yazıldı. Daha sonra bir çok kişi tarafından geliştirildi.

Destek ve geliştirme için <http://lists.samba.org> adresindeki eposta listelerine bakınız.

ÇEVİREN

Nilgün Belma Bugüner <[nilgun \(at\) belgeler.gen.tr](mailto:nilgun(at)belgeler.gen.tr)>, Mart 2005

YASAL UYARI

Bu çevirinin telif hakkı yukarıda belirtilen çevirmen(ler)e aittir. Özgün belgenin telif hakkı ve lisans bilgileri varsa ve belge içinde belirtilmemişse belge sonunda belirtilmiş olacaktır. Bu çevirinin lisansı, özgün belge için belirtilmiş bir lisans varsa ve bu lisans çevirinin de aynı lisansa sahip olmasını gerektiriyorsa onunla aynıdır, yoksa GNU GPL lisansı ve her iki durumda da ek olarak aşağıdaki koşullar geçerlidir. GNU GPL lisansı <<http://www.gnu.org/licenses/gpl.html>> adresinden edinilebilir.

BU BELGE ÜCRETSİZ OLARAK RUHSATLANDIĞI İÇİN, BELGENİN İÇERDİĞİ BİLGİLERİN VEYA KODLARIN NİTELİKLERİ İÇİN İLGİLİ KANUNLARIN İZİN VERDİĞİ ÖLÇÜDE HERHANGİ BİR GARANTİ VERİLMEMEKTEDİR. AKSİ YAZILI OLARAK BELİRTİLMEDİĞİ MÜDDETÇE TELİF HAKKI SAHİPLERİ VE/VEYA BAŞKA ŞAHISLAR BELGELERİ "OLDUĞU GİBİ", AŞIKAR VEYA ZİMNEN, SATILABİLİRLİĞİ VEYA HERHANGİ BİR AMACA UYGUNLUĞU DA DAHİL OLMAK ÜZERE HİÇBİR GARANTİ VERMEKSİZİN DAĞITMAKTADIRLAR. BELGELERİN KALİTESİ VEYA PERFORMANSI İLE İLGİLİ TÜM SORUNLAR SİZE AİTTİR. HERHANGİ BİR HATA VEYA EKSİKLİKTEN 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, BELGENİN İÇERDİĞİ BİLGİNİN KULLANIMI VEYA KULLANILAMAMASI (VEYA VERİ KAYBI OLUŞMASI, VERİNİN YANLIŞ HALE GELMESİ, SİZİN VEYA ÜÇÜNCÜ ŞAHISLARIN ZARARA UĞRAMASI VEYA BİLGİNİN 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.

28 Şubat 2005

rsync(1)

Bu dosya (man1-rsync.pdf), belgenin XML biçiminin
T_EXLive ve belgeler-xsl paketlerindeki araçlar kullanılarak
PDF biçimine dönüştürülmesiyle elde edilmiştir.

30 Ocak 2007