

# Pardus Depo Politikası

6 Ekim 2005

Belge Pardus paket depolarını ve bu depolar üzerinde uyulması gereken kuralları listeler.

Belge yalnızca bir kullanım alanına yönelik hazırlanan Pardus işletim sistemi dağıtımı için kuralları tarif eder. Farklı kullanım alanlarına yönelik, farklı Pardus dağıtımları (Pardus, Pardus Sunucu, vb.) bulunabilir. Belgede anlatılan kurallar her Pardus dağıtımı için ayrı ayrı uygulanır.

## 1 İkili paket depoları

İkili paket depoları PİSİ paketlerinin derlenmiş ve kurluma hazır hallerinin bulunduğu depolardır. Pardus içerisinde o sıradaki dağıtım sürümü tarafından kullanılan iki depo vardır:

1. Kararlı (stable) depo
2. Geliştirme (development) deposu

Bu ilk depo isminden anlaşılacağı üzere yayınlanmış olan kararlı Pardus sürümünün paketlerini barındırır. Bu depo sürüm çıkana kadar oluşturulmaz veya tümü ile boş bırakılır.

Geliştirme deposu ise bir sonraki kararlı sürüm için hazırlanmakta olan PİSİ paketlerini içerisinde barındırır.

Yeni sürümler yayımlandıkça kararlı olarak işaretlenmiş birden fazla depo olabilmesine karşın, bir kullanım alanına/kullanıcı kitlesine hazırlanan Pardus sürümü için yalnızca bir tane geliştirme deposu bulunur. Geliştirme deposu yeterli olgunluğa kavuştuğunda ve sürüm tarihi geldiğinde, geliştirme deposu kopyalanarak sürüm için kararlı paket deposu oluşturulur. Bundan sonra geliştirme deposu bir sonraki sürüm için geliştirmenin yapıldığı alan olarak görevine devam eder.

Kararlı depo üzerinde, yazılımların sürüm güncellemeleri ve depoya yeni bir yazılımın/paketin eklenmesi yalnızca deponun kararlılığını ve işlevini korumak için yapılır. Kararlı sürümde

yapılan güncellemelerin amacı yeni özellikleri kararlı sürüme kazandırmak değil, sürümün kararlılığını korumak içindir. Burada yapılacak güncellemelerde programlar arası ilişkilerin bozulmamasına azami önem göstermek gerekmektedir. API ve ABI’nın korunması esas olarak kabul edilir.

Bir kararlı depo içerisinde her hangi bir paketin yalnızca **tek bir sürümü** bulunabilir. Eğer yazılımların farklı sürümlerinin depoda bulunması gerekiyorsa bu pakete **farklı bir isim** verilerek çözülür. Örneğin; gtk1 ve gtk2 paketleri farklı paket isimleri ile depoda bulunmalıdır.

## 1.1 Depo isimlendirmesi

Her kararlı sürüm, sürüm adı ve sürüm numarasından oluşan bir depo ismi ile anılır. Örneğin; Pardus 1.0 sürümü için depo ismi “pardus-1.0” dır.

Geliştirme deposu ise bir kullanım alanı için hazırlanan Pardus için her zaman aynı ismi alır. Örneğin; Pardus isimli işletim sistemi dağıtımı için geliştirme deposunun ismi her zaman “pardus-devel” olarak isimlendirilecektir. Bu isim, geliştirilmekte olan bir sonraki sürümün numarasından bağımsızdır ve tektir.

## 2 Kaynak paket depoları

Kaynak paket depoları PİSİ kaynak dizinlerinin bulunduğu depolardır. Paketler üzerinde yapılan tüm geliştirme kaynak paket depoları üzerinden yürütülür. PİSİ paketleri bu kaynaklar kullanılarak oluşturulur ve ikili depoya yerleştirilir. Bu yüzden, ikili paket depoları ile ilgili kurallar doğrudan kaynak paket depolarına da uygulanır.

Kaynak paket depolarının isimlendirmesi de ikili depolar ile özdeştir.

Kaynak paket depoları, ikili paket depolarından farklı olarak Subversion sürüm kontrol sistemi üzerinde barındırılır. Kararlı sürüm oluşturulacağı zaman geliştirme deposu etiketlenir (taglanır) ve etiketlenmiş kaynaklardan ikili PİSİ paketleri oluşturularak sürüm için kararlı ikili paket deposu oluşturulur.

## 3 Paket geliştiricileri için kurallar

Paket geliştiricileri Subversion üzerinde bulunan kaynak paket depoları üzerinde çalışırlar. Yazılımların kaynak kodlarından PİSİ paketlerinin oluşturulması için gerekli olan geliştirme, paket geliştiricileri tarafından yürütülür.

Paket geliştiricileri 3 farklı sorumluluk grubu ile listelenebilir.

1. **Depo geliştiricisi:** Depo geliştiricileri tüm depodan sorumlu olan ve tüm depo üzerinde işlem yapmaya yetkili kişilerdir.

2. **Kategori geliştiricisi:** PİSİ kaynakları tasarım belgesinde anlatıldığı gibi bir kategori (Category / PartOf) tanımı yaparlar. Kategori geliştiricisi belirli bir kategoriye uyan tüm paketlerden sorumlu olan ve bu paketler üzerinde işlem yapma yetkisine sahip geliştiricidir. Örnek kategoriler arasında, *system/base*, *system/devel*, *desktop/kde* sayılabilir.
3. **Paket geliştiricisi:** Paket geliştiricisi yalnızca sorumluluğunu almış olduğu paketler üzerinde işlem yapma yetkisine sahiptir.

Sorumluluk grupları sorunların hızlı bir şekilde giderilebilmesi ve farklı boyutlarda kararların hızlı bir şekilde alınabilmesi için oluşturulmuştur. Herhangi bir geliştirici, sorumluluğunu aşan bir müdahale yapmak istediği zaman izin isteyerek çalışabilir. Üzerinde anlaşılmış konularda yapılan düzeltmeler/güncellemeler genellikle izin istemeye gerek duyulmadan kabul edilir :).

### 3.1 Genel kurallar

- Kaynak paket deposu üzerinde çalışan geliştiricilerin paket deposu kurallarına ve yayınlanmış bir sürüm zaman planı varsa bu zaman planına uyması gerekir.
- Paket geliştiricileri hazırladıkları PİSİ kaynak dizinlerini hazırlarlar, fakat PİSİ kaynaklarının .pisi ikili paketlerine dönüştürülmesi doğrudan paket geliştiricileri tarafından gerçekleştirilmez. Bununla birlikte, geliştiricinin kaynak paketi sisteminde oluşturduğu, test ettiği ve sorunlarından arındırdığı kabul edilir.
- Geliştiriciler üzerinde çalıştıkları kaynak paket deposunun en güncel versiyonu ile çalışmak zorundadırlar. Bu hem kararlı sürümde, hem de geliştirme sürümünde yapılan çalışma için geçerlidir.
- Geliştiriciler depoya gönderdikleri her paketten sorumludurlar, paketin derlenebilir, sorunsuz ve çalışır olduğundan emin olmak zorundadırlar. Paket tüm testleri geçmelidir.
- Eksik veya tamamlanmamış paketler depoya asla gönderilmemelidir. Her paketin pspec.xml, actions.py dosyaları doğru olarak yazılmış olmalı. Yamaları ve ek dosyaları files/ dizini altında bulunmalı. COMAR betikleri ise comar/ dizini içerisinde bulunmalıdır. Paket derlenebiliyor ve kurulabiliyor olmalıdır.
- Geliştirici paket güncellemelerinde pakete ait olan ChangeLog dosyasını düzgün, anlaşılır şekilde güncellemelidir.
- Geliştirici aynı anda birden fazla paket ile ilgili depoya gönderim yapmamalıdır. Örneğin; Bir geliştirici 8 paket üzerinde birden değişiklik yaptı ise, her paket için ayrı ayrı gönderimde bulunmalıdır (yani toplamda 8 ayrı gönderim). Kısaca gönderimler atomik olmalıdır.

- Bir geliştirici bir başka geliştiricinin paketine acil durumlar dışında (güvenlik açığı, geliştiriciye uzun zamandır ulaşamama gibi) müdahale etmemelidir. Bu kural KATEGORİ sorumlulukları için ise sorumlu olduğu kategori dışındaki paketleri kapsamaktadır. Bunun yerine ilgili paket ile ilgili düzeltmesini paket sorumlusuna veya KATEGORİ sorumlusuna göndermeli ve sorunun ne olduğunu ve nasıl çözüldüğünü ayrıntılı şekilde tarif etmelidir.
- Depo'ya paket için gerekli yamalar, ek dosyalar veya post/pre betikleri, PSPEC dosyası, actions.py ve ÇOMAR betikleri dışında hiçbir şey konmamalıdır. Paketin kaynak kodu, geliştiricinin fotoğrafı ya da uygulamaya ait ekran görüntüsü de bu kurala dahildir :).

### 3.2 Paket isimlendirme kuralları

Paket isimlendirmesi ile ilgili kurallar pisi kaynak dizinindeki package\_versions.tex belgesinde detaylı olarak anlatılmaktadır. Bu bölümde kısaca bir PİSİ paketinin isimlendirmesi anlatılacaktır. Bu kural serisine uymayan isimlendirmeler hatalı kabul edilir.

1. Bir pisi paketinin isimlendirme kuralı şu şekilde formülize edilebilir. PAKET-VERSİYON{\_sonekNUMARA}-REVİZYON.
2. Paket ismi ana geliştiricinin pakete verdiği isimdir, değiştirilemez fakat genişletilebilir (kaynağının paketlenmesi sırasında oluşturulacak bir alt paket gcc-doc olarak isimlendirilebilir)
3. Paket versiyonu ana geliştiricinin pakete verdiği sürüm numarasıdır, değiştirilemez. (util-linux-2.4z, kernel-2.6.9.4, gcc-3.3.6 gibi)
4. \_sonek ile belirtilen kısım “alpha”, “beta”, “pre”, “rc” ve “p [patch level]” den biri olabilir. Bunların kendi arasında sıralaması şöyledir; alpha < beta < pre < rc < Son eksiz paket < p. (1.1\_alpha1 < 1.1\_beta1 < 1.1 < 1.1\_p4 )
5. Revizyon geliştirici tarafından pakete verilen ve devamlı artan bir sayıdır. Pakete yapılan değişikliklerde bu revizyon numarası her zaman arttırılmalıdır.