

PiSiDo Yardımı

Paketleme Temelleri

Bir pisi paketi “pisi build” komutuyla inşa edilebilir. Pisi inşa komutu bir pspec.xml dosyasını parametre olarak alır ve kurulabilen ikili bir dosya oluşturur. Bir paketi inşa edebilmek için pspec.xml dosyasını parametre olarak verirken diğer başka dosyaların da olması gerekir. İnşa dosyalarının dizilimi şu şekildedir :

```
<paket_adi>
|__ actions.py
|__ pspec.xml
|__ translations.xml
|__ files
|__ comar
|__ packages.py
|__ service.py
```

<paket_adi>, files ve comar dizinlerdir. Bu dosyaları açıklarsak :

<paket_adi> : Bu dizinin ismi önemli değildir. Sadece inşa dosyalarımızı tutar.

actions.py : Bu dosya yapılandırma, derleme ve inşa adımlarını tanımlar. Her adımda ne yapılacağını Actions API kullanarak tanımlar. PiSiDo size, autotools, cmake, kde4, qt4 gibi şablonlar sağlar. Daha fazla bilgi için [Pardus geliştirme sayfalarına](#) başvurun.

pspec.xml : Bu dosya paket bilgisini tanımlar. Paket kaynağını, ayrıntılı bilgisini ve inşa dosyalarının geçmişini tutar. Bu dosya için RNG bir kuralı dosyası vardır. Dosyası [burada](#) bulabilirsiniz. Daha fazla bilgi için [Pardus geliştirme sayfalarına](#) başvurun.

translations.xml : Bu dosya seçimliktir. Paket özetinin ve ayrıntılı açıklamasının farklı dillerde tanımlanmasını sağlar.

files : Bu dizin seçimliktir. Kaynak arşivi ile gelmeyen ek dosyalarınızı buraya ekleyebilirsiniz. Ayrıca yama dosyalarınızı da ekleyebilirsiniz. Buradaki her dosyanın işlenmesi için pspec.xml dosyasında tanımlanmalıdır. PiSiDo bu işlemin en kolay şekilde yapılabilmesi için grafik araçlar sağlar.

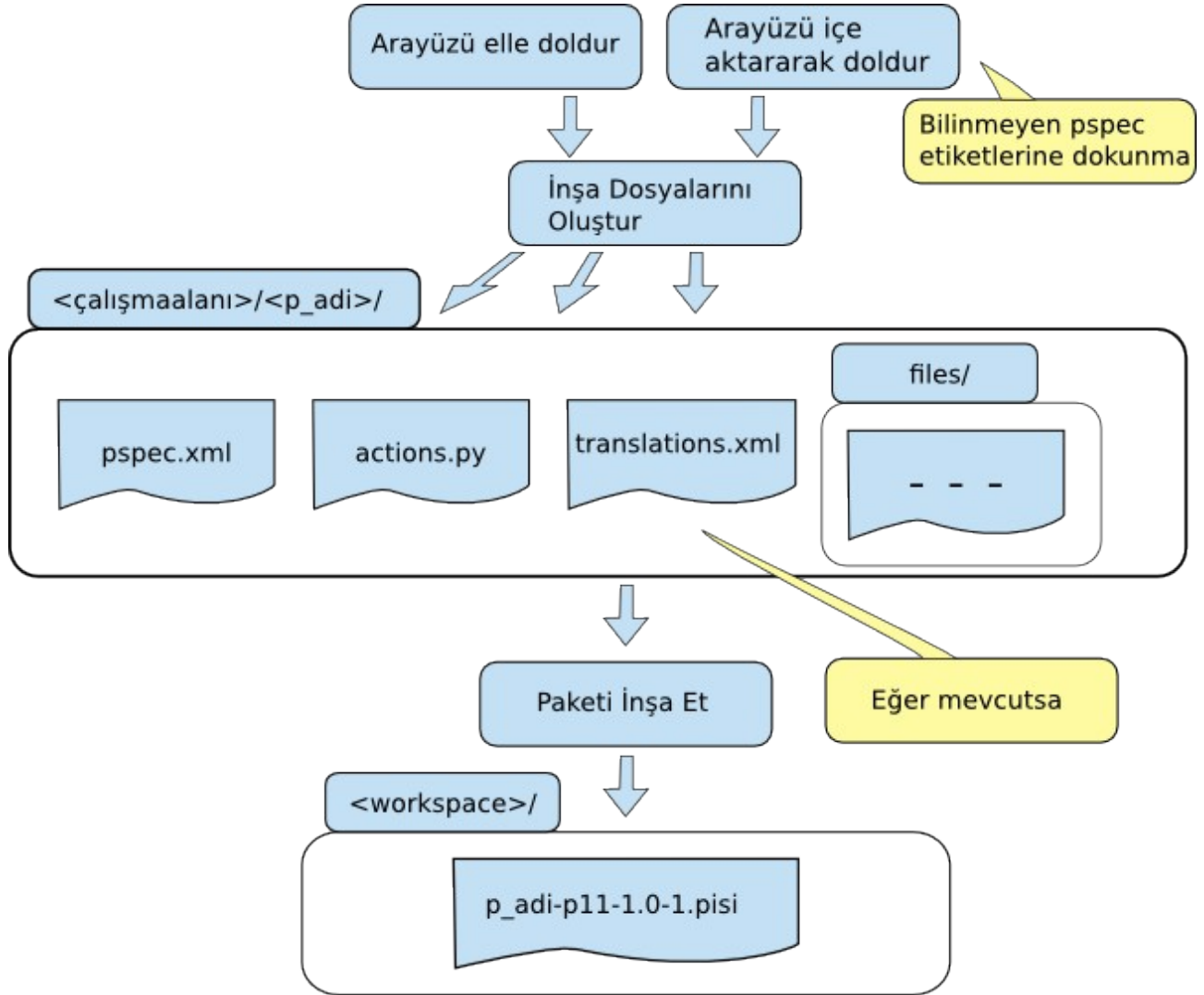
comar : Bu da seçimlik bir dizindir. Bu dizin COMAR (CONfiguration MAnageR) dosyalarını tutar. Bu dosyalar Pardus'a servis işlemlerini tanımlamada yardımcı olurlar. Bu dosyalar PiSiDo'nun ilgi alanı dışında olduğundan dizini ve içeriğindekiyi oluşturmayacaktır. Daha fazla bilgi için [Pardus geliştirme sayfalarına](#) başvurun.

Bir pisi paketi yapabilmek için en azından pspec.xml ve actions.py dosyalarını oluşturmak zorundasınız. Daha sonra şu komutu vererek inşa edebilirsiniz :

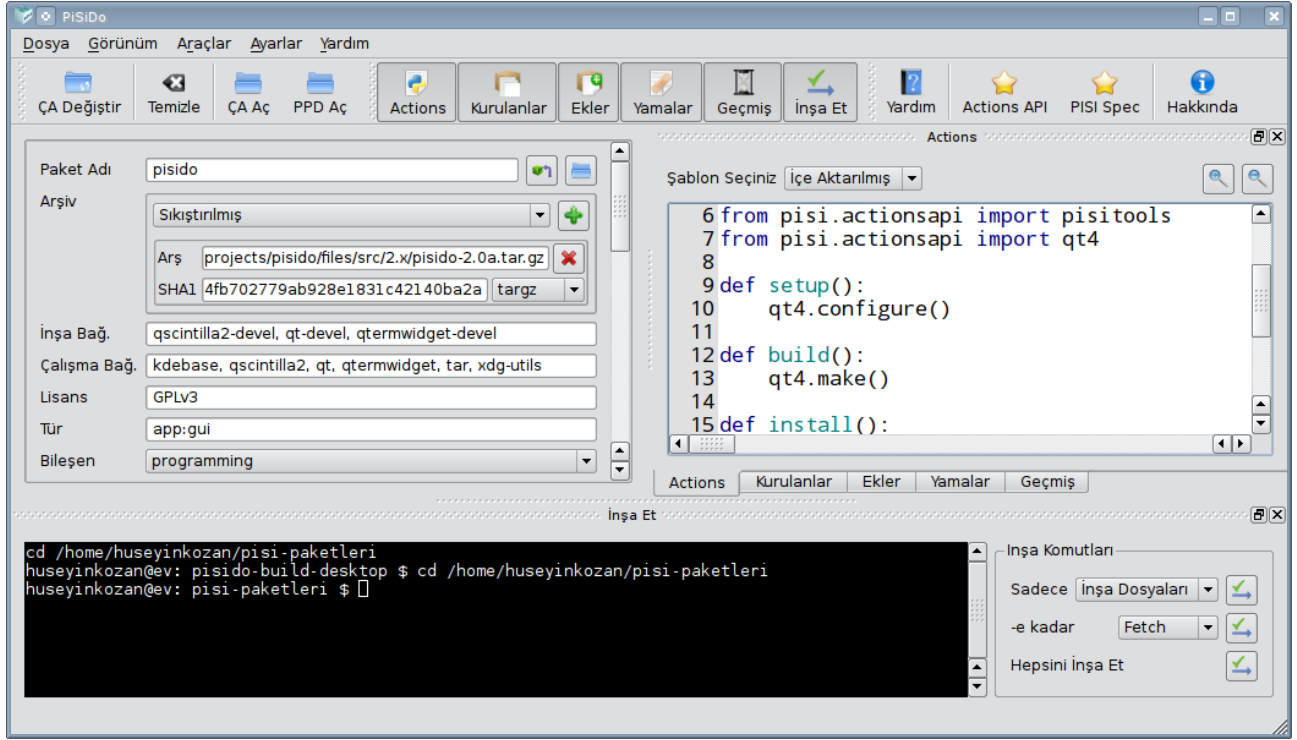
```
$ sudo pisi build pspec.xml
```

Resmi olmayan [Pardus Viki](#)'si üzerinde bu konuda işe yarar bilgiler bulabilirsiniz. Resmi belgeler [Pardus geliştirme sayfalarındadır](#).

PiSiDo Paketi Nasıl Yapar



PiSiDo Grafik Arabirimi



Merkezi Alanlar

Paket Adı:

Paketin adıdır. Bu alandaki ad kullanılarak çalışma alanında bir dizin oluşturulur. Tuşladığınız her karakterde uygulama var olan dizinleri tarar ve bulduğu inşa dizinini otomatik olarak size listeler. Var olan paket dizinini yazıp Enter'e basarak mevcut paketi içe aktarabilirsiniz.

Arşiv:

Burada paketin kaynağını tanımlayabilirsiniz. Seçim kutusunun iki değeri vardır:

1. Sıkıştırılmış: Yerel dosyalar içindir. Ekledikten sonra uygulama sha1 değerini hesaplayacaktır.
2. Url: Yerel ve uzak dosyalar içindir. Kaynak yolunu ve sha1 değerini sizin girmeniz gerekmektedir.

İpucu : Hatalı bir sha1 değeri girip inşa etmeyi deneyin. Pisi dosyanızı arşiv dizinine indirecektir. Daha sonra konsolu kullanarak sha1 değerini alabilirsiniz. Sha1 değerine 1111.....11 girin ve inşa edin. Hata aldıktan sonra şunu deneyin:

```
$ sha1sum /var/cache/pisi/archives/pisido-2.0a.tar.gz
```

İnşa Bağ.:

Burada inşa bağımlılıklarını tanımlayabilirsiniz. Pisi inşa sistemi paketi inşa etmeden önce bunları kuracaktır. Virgül ile ayırarak birden fazla paket adı yazabilirsiniz. Ayrıca sürümü kısıtlamak için şu özel parametreleri de kullanabilirsiniz :

- qt-devel[>4.7] : qt-devel paketinin sürümü 4.7'den büyük olmalıdır
- qt-devel[=4.7] : qt-devel paketinin sürümü 4.7'ye eşit olmalıdır
- qt-devel[<4.7] : qt-devel paketinin sürümü 4.7'den küçük olmalıdır
- qt-devel[>>4.7] : qt-devel paketinin yayımı 4.7'den büyük olmalıdır
- qt-devel[==4.7] : qt-devel paketinin yayımı 4.7'ye eşit olmalıdır
- qt-devel[<<4.7] : qt-devel paketinin yayımı 4.7'den küçük olmalıdır

Örneğin : qt-devel[>4.7], gtk2-devel[2.20]

Çalışma Bağı:

Burada çalışma zamanı bağımlılıklarını tanımlayabilirsiniz. Pisi paketi kurmadan önce bağımlılıkları kuracaktır. Virgül ile ayırarak birden fazla paket adı yazabilirsiniz. Ayrıca sürümü kısıtlamak için şu özel parametreleri de kullanabilirsiniz :

- qt[>4.7] : qt paketinin sürümü 4.7'den büyük olmalıdır
- qt[=4.7] : qt paketinin sürümü 4.7'ye eşit olmalıdır
- qt[<4.7] : qt paketinin sürümü 4.7'den küçük olmalıdır
- qt[>>4.7] : qt paketinin yayımı 4.7'den büyük olmalıdır
- qt[==4.7] : qt paketinin yayımı 4.7'ye eşit olmalıdır
- qt[<<4.7] : qt paketinin yayımı 4.7'den küçük olmalıdır

Örneğin : qt[>4.7], gtk2[2.20]

Lisans:

Burada paketin lisans türünü seçebilirsiniz.

ISA:

Burada paket türünü seçebilirsiniz. Uygulama (application), servis (service), yerelleştirme (localization) gibi.

PartOf:

Burada paketin hangi bileşende olduğunu tanımlayabilirsiniz. Masaüstü (desktop), oyun (game), donanım (hardware) gibi.

Anasayfa:

Burada paketin web sayfasını tanımlayabilirsiniz.

Özet:

Burada kısa bir açıklama tanımlayabilirsiniz. Bu, paketleri kullanıcıya listelerken kullanılacaktır.

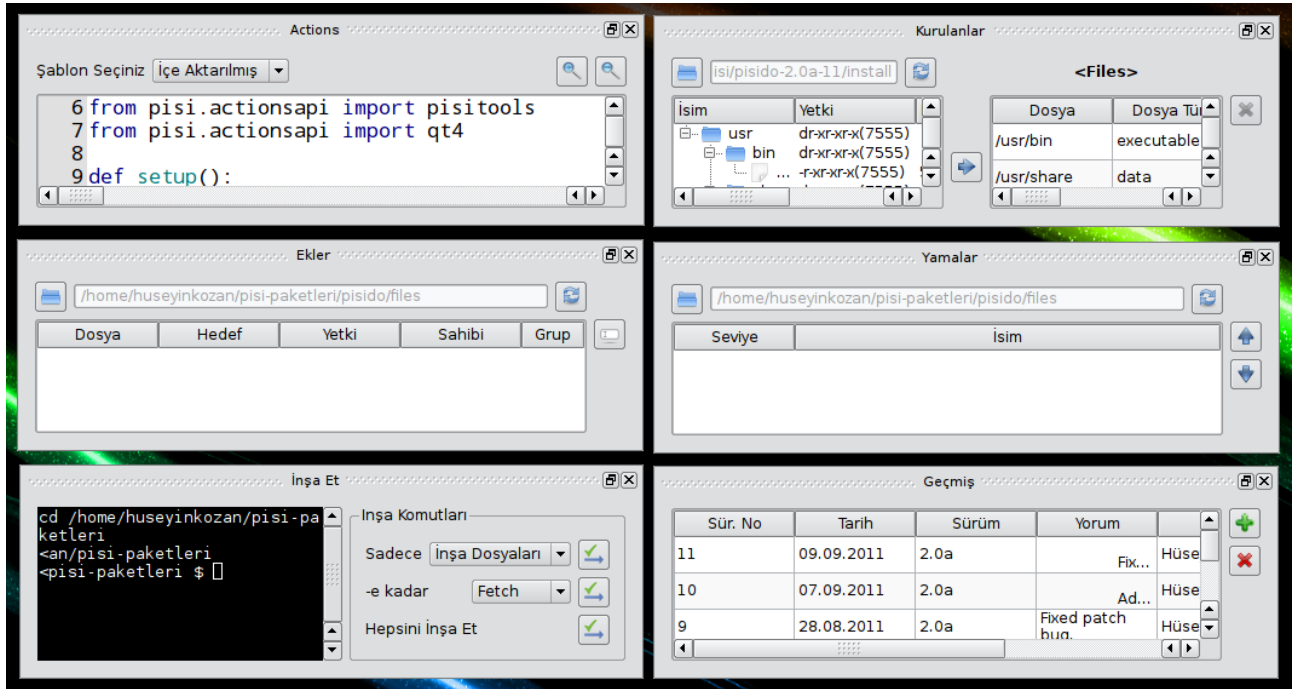
Açıklama:

Burada ayrıntılı açıklama tanımlayabilirsiniz.

Lisan:

Burada Özet ve Açıklama alanlarında girdiğiniz bilgileri farklı diller için tanımlayabilirsiniz. İçerik aktarırken Özet-Açıklama ikilisi birlikte tanımlanmayan ve diğer etiketlerde hata bulunan girdiler yüklenmez.

Yapışık Pencereler



Actions Dosyası

Actions dosyası pisi inşa sistemi tarafından inşa işlemini yönetmek için kullanılır. Seçim kutusu yardımıyla şablonları seçebilirsiniz. Actions dosyası python ve Actions API kullanır. Daha fazla bilgi için [Pardus geliştirme sayfalarına](#) başvurun.

Actions dosyası şablonları kullanıcıya yardımcı olmak için bazı yorum satırlarına sahiptir. Ayrıca PiSiDo'ya özel bazı değişkenler de vardır. Bu değişkenler inşa dosyaları oluşturulurken uygun değerlerle değiştirilirler. Özel değişkenler şunlardır :

- `__package_name__` : Paket Adı alanıyla değiştirilir
- `__version__` : [Geçmiş Penceresindeki](#) en son güncellemenin sürümüyle değiştirilir
- `__summary__` : [Özet](#) alanıyla değiştirilir

Uyarı ! : Yaptığınız değişiklikler inşa dosyalarını oluşturuncaya kadar kaydedilmeyecektir. İnşa dosyalarının oluşturulması [İnşa Et Penceresi](#)'nde bulunan herhangi bir komut tarafından otomatik olarak yapılmaktadır.

Kurulan Dosyalar

Burada kurulabilecek her dosyanın nereye kurulacağını tanımlayabilirsiniz. *Kurulanlar* listesi başarılı bir inşadan sonra doldurulacaktır. Uygulama [pisi inşa dizinini](#) değişikliklere karşı izler. Eğer inşa başlarda başarısız olunursa liste temizlenecektir.

Gerekli bir alan olduğundan paketteki kurulan bütün dosyalar için kök dizini (/) gösteren önatımlı bir etiket tanımlıdır. Dosyalar veya dizinler için uygun etiketleri tanımladıktan sonra önatımlı etiketi silmelisiniz.

Ek Dosyalar

Bu alan seçimlidir. Uygulama <çalışmaalanı>/<paket_adi>/files/ dizinini değişikliklere karşı izler. El yordamıyla bu dizine yeni dosyalar ekleyerek kurulum esnasında nereye konacaklarını buradan tanımlayabilirsiniz.

Yamalar

Bu seçimlik bir alandır. Uygulama <çalışmaalanı>/<paket_adi>/files/ dizinini değişikliklere karşı izler. El yordamıyla bu dizine yeni yamalar (*.patch) ekleyerek yama sırasını bu pencereden tanımlayabilirsiniz.

Geçmiş

Bu alan paket inşa dosyalarının geçmiş bilgisini gösterir. Paketçi tarafından pspec.xml dosyasında tanımlanmıştır. Yeni güncelleme bilgisi ekleyebilirsiniz. Ayrıca, paketçi bilgilerini [Uygulamayı Yapılandır](#)'dan tanımlayabileceğinizi belirtelim.

İnşa Et

Bu pencerede paketinizi inşa etmek için bazı komutlar verebilirsiniz. İşte desteklenen komutlar bunlar :

- Sadece
 - İnşa Dosyaları : İnşa dosyalarını oluştur ve dur.
 - Paket : İnşa dosyalarını oluştur ve başarılı bir inşayı paketlemeye çalış. Bu actions.py'yi değiştirip paketi yeniden derlemek zorunda kalmadan inşa etmek için kullanışlıdır. -e kadar'ın en son adımı olarak düşünülebilir.
- -e kadar
 - Fetch : İnşa dosyalarını oluştur ve pisi build pspec.xml -fetch komutunu ver. Kaynak arşivini indirip duracaktır.
 - Unpack : Fetch adımını uygula, arşivin sha1 değerini hesapla, arşivi aç, yamaları uygula ve dur.
 - Setup : Unpack adımını uygula, ayarla (yapılandır) ve dur.
 - Build : Setup adımını uygula, inşa et (derle) ve dur.
 - Install : Build adımını uygula, kur ve dur.
 - Check : Install adımını uygula, paketi denetle ve dur.

Daha fazla bilgi için [Pardus geliştirme sayfalarına](#) başvurun.

PiSiDo Menüleri

Dosya

Çalışma Alanını Değiştir

Çalışma alanını değiştirir. Çalışma alanı uygulamanın inşa dosyalarını ve inşa edilen pisi

dosyalarını kaydettiği ana dizindir. Çalışma alanı penceresinin başlangıçta açılıp açılmayacağını kendi üstünden ayarlayabilirsiniz.

Alanları Temizle

Grafik arabirim alanlarının hepsini temizler.

Çıkış

Programdan çıkar. Uygulama kaydedilmemiş değişiklikler için sizi uyarmayacaktır.

Görünüm

Bu menüden her yapışık pencereyi ve araç çubuğunu gizleyip gösterebilirsiniz .

Araçlar

Çalışma Alanını Aç

Dosya yöneticisi ile uygulamanın açılışında belirlediğiniz çalışma alanını açar.

PISI Paketleme Dizinini Aç

Dosya yöneticisi ile `/var/pisi` altında bulunan pisi çalışma dizinini açar. Bunu [Ayarlar](#)'dan değiştirebilirsiniz.

Ayarlar

Uygulama Lisanı...

Buradan önatımlı uygulama lisanını değiştirebilirsiniz. Değişiklik uygulamayı yeniden başlattığınızda etkili olacaktır.

Uygulamayı Yapılandır...

Uygulamaya özel ayarları buradan değiştirebilirsiniz. Paketçi bilgisini tanımlayarak geçmişe güncelleme eklerken yardımcı olmasını sağlayabilirsiniz. Pisi paketleme dizinini, [Actions API](#) sayfasını ve [PISI Spec](#) dosyasının yolunu değiştirebilirsiniz.

Yardım

Yardım...

Seçilen lisan için uygun olan yardım dosyasını açar.

PISI Spec

PISI Spec RNG dosyasını açar. Bu değer [Uygulamayı Yapılandır](#)'dan değiştirilebilir.

Actions API

Actions API sayfasını açar. Bu değer [Uygulamayı Yapılandır](#)'dan değiştirilebilir.

Hakkında...

Uygulama bilgisini gösterir.

Qt Hakkında...

Qt araç seti bilgisini gösterir.