

Pisi - howto



bir İngilizce versiyonu için güçlü bir talep almanca PiSi HowTovar,ben hızlı ve kirli küçük bir tane oluşturmak için çalışacağız. Ben sadece ambalaj için en önemli şeylerden içerecektir anlamına gelir. Ben PiSi-kitap, Pardus wiki ve kendi küçük deneyim bilgi karışacaktır.oriijinal almanca PiSi-NASIL-Kitap ve aşağıdaki listede bulabilirsiniz tüm katılımcılar için Gürkan Zengin'in (aka LinuxFanatic)

Tamamen yanlış veya eksik bir şey varsasayesinde.,Bizi suçlama → KATKI!

Sürüm	Tarih	Kişi	Yorum
0.1	14.02.2011,	GB	:-)Çok ilk versiyonu
0.2	16.02.2011	atolboo	typo,Bazı düzeltmeler.....
0.3	28.02.2011	GB	PiSi-Yap eklendi ve Workdir (thx LinuxFanatic)hakkında bilgi
0.4	10.03.2011	Phix	,gramerortografik, yazım hataları, hataları düzeltme
0.5	2011/11/15	GB	PiSiDo, yama, mimari algılama ve tutmak * Sime hakkında bilgiler eklendi. pyc dosyaları. (Serdar & Arda & a.palius thx)

biz pisi-howto belgesi başlamadan önce, orada . kullanılabilir pardus wiki sadece bazı biraz ingilizce bilgisi olan var olan bir geliştirici kılavuzu olmadığı için, orada da yararlı bilgiler bulabilirsiniz: <http://developer.pardus.org.tr/guides/>

biz bir tavsiye Başlamadan önce: öğrenmek için . dosyaları oluşturmak, mevcut eğitim almak her zaman iyi resmi pardus repo İnşa dosyaları burada bulunabilir: <http://svn.pardus.org.tr/pardus/> Onlar profesyoneller tarafından oluşturulan ve yeni başlayanlar için anlamak kolay bazen değillerdir . Böylece belki dearasında inşa dosyaları içine bir göz alarak Pardususer.de bu yapı dosyalar profesyonel olmayan (sizin ve benim gibi) ve anlamak için belki daha iyi yapılrdeğer. Onlar ne,% 100 doğru ne de çok şık olabilir ama - hey - de işe yarıyor! :-)



PiSi "başarıyla amaçlandığı gibi yüklü Paketler" Pardus ve stand paketi sistemidir.
Paket yöneticisi PiSi için önyüz ve PiSi sahip olduğu tüm fonksiyonlara sahip değil.

Pardus yeni bir paketleme sistemi oluşturulmuş nedeni, paketleri oluşturmak için çok kolay bir yol var.
Ayrıca paket özellikleri yükleme talimatları ayrı çekilmelidir.

Paket Sistem PiSi Python.The oluşturulan yazılım paketi ile yazılmış heavenly sıkıştırılmışkullanır
ve sufix ". pisi". PiSi formatındaki tüm paketleri ya terminali ile kurulabilir
veya>"Paket Yöneticisi ile Aç" sağ tıklama- sağlanan paket sağ pardus sürümü uyanzaman:.

komut ile yeni bir paket oluşturmak Sudo pisi build pspec.xml

PiSibir dizin / iş ve / install / **var / pisi / package-name**yaratacaktır.

"olarak**çalışmak**"dizini Tüm derleme yapılır ve bundan sonra, tüm kurulum dosyaları olmaKuracaktır. /

Bundan sonra, bütün"install" dizinin dosyaları dahil ve pisi paket içine sıkıştırılmış olacaklistesi:.

PiSi paketlerinin nasıl oluşturulacağı hakkında kısa bir

1	indir sıkıştırılmış kaynak dosya (pspec için download linki kaydedin) +1 +
2	ilekurmak sha1sum (pspec için) komut: sha1sum name_of_sourcefile.tar.gz +2 +
3	unzip SourceFile

4	çalışma INSTALL ve README
5	onay bağımlılıklarıINSTALL ve READMEgöre:
6	programı (örneğin) rpm hakkında bilgi ile ilgili diğer bilgiler sitelerida kontrol http://rpm.pbone.net / veya http://www.rpmseek.com/ ya http://www.rpmfind.net/ daArch Linux için pckbuild dosya iyi bir referansolabilir
7	bağımlılıkları Parduskullanılabilir durumdaysaçek
8	çek varsapaket adı birinden farklıdır. / rutinbekleyebiliriz yapılandırmak
9	kaynak dosya (manuel yükleme çalışıyorsa o zaman bakınız)kullanarak paketini yüklemek
10	info.txt tüm toplanan bilgiler toplamak (daha iyi anlaşılması ve sonraki değişiklikler içinoluşturmak) ++3
11	PİSİ / pspec.xml actions.py ve diğer dosyaları
12	çalışan tarafından bir paket oluşturmak: sudo pisi build pspec.xml

- +1 + → Açık kaynak projeleriBir çok SourceForge tarafından barındırılan. SourceForge, birçok aynalar ve kullanıyor
- Gerçek hızlı bir indirme için alınır. Bu avantajı kullanmak için yerine ayna adresi kullanabilirsiniz "gerçek" indirme adresi. Sonra arşivi / etc / pisi / mirrors.conf belirtilen aynasından indirilen olacak gerçek
Adres: **aynalar :/ / sourceforge / freefilesync /FreeFileSync_Sourcecode_v4.1.zip**
- **+2 +** → "pisi-geliştirici araçları" paketi (Pardususer.de repo mevcut) kurduktan sonra bunu yapabilirsiniz yunus içinde eylemler yoluyla modunu (sağ klik)
- **+3 +** → Bir sonraki sayfada bulacaksınız info.txtA excampe: D

Benim kişisel lezzet paketleme öncesi yazılım hakkında gerekli tüm bilgileri toplamak için vardır. Bu info.txt var sonra ben PiSiDo kullanarak packging başlatmak veya varolan benzer build dosyaları kullanarak. Burada bir bilgi textfile hakkında excampe şudur:

info.txt	Açıklama
Anasayfa: http://sourceforge.net/projects/wake-on-lan/ Download: http://mesh.dl.sourceforge.net/project/wake-on-lan/wol/0.7.1/wol-0.7.1.tar.gz sha1sum:7edd1a8c2d4baffc3b6baf8fcd12e786f9fc1fbd	pspec.xml için bazı genel bilgiler
bağımlılıkları isim özel bash	psec.xml için bağımlılıklar - paketlerin isimleri sometimes farklı
INSTALL olabilir: 1. Paketin kaynak kodu ve türü içeren dizini 'cd ' ' / 'Sizin sisteminiz için paket yapılandırmak için yapılandırın. 2. Tip 'paketi derlemek için 'make. 3. Tip 'programları ve herhangi bir veri dosyalarını yüklemek için 'make install ve dokümantasyon oluşturun:.	actions.py Bilgi <pre> pisi.actionsapi ithalat autotools pisi.actionsapi ithalat pisitools gelen pisi.actionsapi ithalattanelde WorkDir = "wol-" + get.srcVERSION () def setup (): autotools.configure ("-- prefix = / usr") def build (): autotools.make () def </pre>
Bilgi: (http://rpm.pbone.net) Özet: Lan Ystemcide Wake Açıklama: Bu wol olduğunu Lan istemci Wake On. sihirli paket uyumlu uyanır. Böyle wake-on-lan-ethernet kartları ile kutuları gibimakineleri Bazı iş istasyonları bir şifre ile wake-on-lan uzanır SecureON sağlar. Bu özellik aynı . zamanda wol tarafından sağlanmaktadır:	Özeti ve Açıklama pspec.xml için ve translations.xml için de tercüme
RPM İçerik	ind psec.xml: <pre> <Files> <Path fileType="executable"> / usr / bin / </ Path> <Path fileType="doc"> / usr / share / doc / </ Path> <Yolu fileType = "yerel"> / usr / share / locale / </ Path> <Path fileType="info"> / usr / share / info / </ Path> </ Files> </pre>

Bazı terminali PiSi-Komutlar hakkında Genel Bakış:

Yükleme paketi	sudo pisi it PAKETİ
paketi çıkarın	sudo pisi rm PAKETİ

Güncelleme depoları (paketlerin indirme en son endeks)	sudo pisi ur
Güncelleme tüm sistem	kadarsudo pisi
paket yöneticisilistelenen Listesi depoları	pisi lr
paket yöneticisiADRESdeposunu ekleyin	sudo pisi ar ADI
paket deposu Kaldırığın-yönetici	sudo pisi rr ADI ADRES
aktif depolarıiçinde paket ara	pisi sr PAKETİ
paketiyeiden yükleyin	sudo pisi it PAKETİ-yeniden
paketi yaratıcısıhakkında bilgiler	pisi suçu PAKETİ
build dosyaları paketi oluşturma	pspec.xmlsudo pisi build
tüm paketlerkontrol	sudopisionay
check-komutuhakkındaDosyanızı>	pisi & pisi-check.txt && kontrol grep-i'yolsuzluk' pisi-check.txt
paket veritabanındatemizkilitlelerini kaldırın	sudo pisi
yüklemesi bir hata (enerji kaybı veya bağlı durursabir şey) sonra belki bekleyen paketleritekraryapılandırmak	sudo pisi cp
pisipaketiSil önbellek (geçici dosyaları)	sudopisi dc
kurmadanindir pisi	pisiPAKETİ getir
birpgraph.dot dosyası saklanırgerçek klasöründegerekir.Tüm"Kgraphviewer"ile görüntüleyebilirsiniz	pisi graph PAKETİ
idam pisi eylemleringöster geçmişi (history-manager gibi)	pisi geçmişi
deposu indeks dosyalarıOluştur	pisi index
göster paket bilgileri	pisi bilgi PAKET
Listebelirli bir depodaki bütün mevcut paketleri	pisi la REPONAME
Listesibileşenler (paket kategorisi [component.xml parçası])	pisi lc
Listesi belirli bir kategorinin üyesi olan ve text.txtolarak kaydedebilirsiniz paketleri yüklü	pisi li- bileşeni applications.games> text.txt
Listesi Yeni paketleri Aktif depolarda	pisi ln
mevcut yükseltmeleriListesi	pisi lu
PiSi veritabanıRebuild	pisi rbd

komut"**pisihelp**"sunuyor.PİSİ temel işlevlerini genel bir bakış hepsini Bahsedildiği gibi, paket yönetici GUI fonksiyonların en değil, :-)

paketleri oluşturmak için sisteminhazırlanması,

Öncelikle standart bir Pardus kurulum dahil edilmez gerekli tüm paketleri yüklemek zorunda.kullanıyorum:Aşağıdaki komutu ile yapabilirsiniz

Ben aşağıdaki dizin yapısını

```
benim PiSi-Paketler
|
|__ paket-ismi
|  |
|  |__ kaynak      (yazılım kaynağı ileINSTALL / README, ....) gibi talimatları okumak
|  |
|  |__ yapı        (Pardus paket için inşa dosyaları ile)
|  |  |  |  |
|  |  |  |  |__ pspec.xml
|  |  |  |  |__ actions.py
|  |  |  |  |__ translations.xml
|  |  |  |  |__ dosyaları
|  |  |  |
|  |  |  |__ INFO (kişisel notlar örneğin paket hakkında tüm toplanan bilgiler ile textfile)
|  |  |
|  |  |__ paket-ismi
|  |  |  |
|  |  |  |__ kaynak
|  |  |  |__ .....
|  |  |  |__ .....
```

pspec.xml paketi hakkındabilgiler. Yeni bir paket sürüm oluşturmak istediğinizde, sadece wih pspec.xml çalışmak zorundagenellikle. tüm seçenekleri de bakabilirsiniz: [link1](#) ve [link2](#) boş bir pspec.xmlbulunabilir BURAYA

actions.py derleme içingerekli komutları- actions.py oluşturulduktan sonra, başka bir değişikliğe ihtiyaç vardır.

translations.xml paket yöneticisiiiçindegösterilen paketi hakkında İngilizce dışındaki dillerdeKısa bilgi ve açıklama

comar comar için özel bir komut dosyası (bilgi ve çok çok nadiren gerekli ötesinde)

dosyalar klasör paketi (yamalar, simgeler, K-Menü masaüstü dosyaları gibi ...) için deek opsiyonel

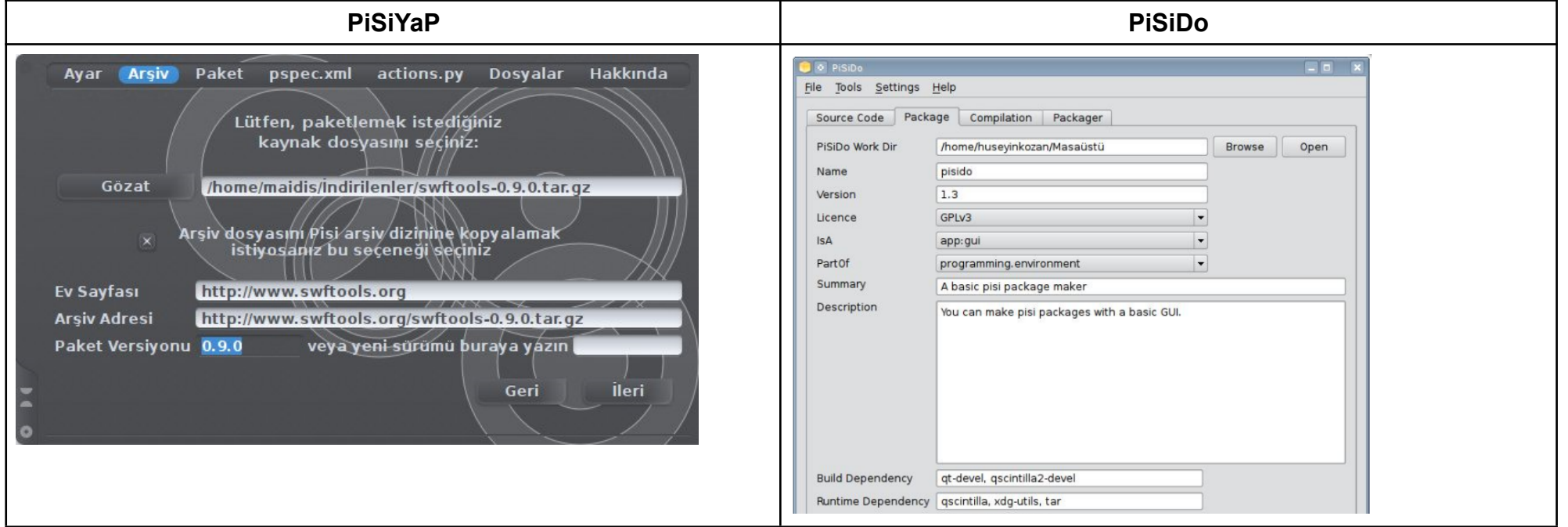
dosyalarıbilgisayarınızda bir paket oluşturmak için, çukur pspec.xml da içeren yapı-klasörü gidin. Bir terminalden aşağıdaki komutu çalıştırın:

sudo pisi build pspec.xml
sudo pisi bi <http://www.webaddress.org/folder/pspec.xml>

PiSi paketlerinin oluşturulması için Grafik önyüzleridir

Orada kullanılabılır dosyaları oluşturmak çalışan iki yardımcı araçlardır.

PiSiDo çok kullanışlı ve yararlı bir araçtır. Eğer P2011 ya da Pardus user.de repo mevcuttur



Bilginizi ve PiSi know-how yeterince iyi, bir katılımcı olmak ve üzerinde korunabilir: resmi Pardus deposu paketleri yeni-katılımcı-rehber

- de de yapabilirsiniz bugtracking içinde Yeni İşler
- Kendi paket deposu oluşturabilir veya
- varolan bir ambalaj takımına katkıda bulunabilir.

pspec.xml (paket özellikleri)

Kaynak

pspec.xml başlığında sonrabölümünde,kaynak bulacaksınız. . Bu bölüm paketin kaynak ve paketi derlemek için gerekli bağımlılıkları hakkında temel bilgi verir Ayrıca vardır: derleme süreci başlamadan önce lisans, kategori, paketin yolları ve hangi yamaları (değişiklikler) uygulanması gerekirbölümü.

Paketi

Bu bölümde, ambalaj sonra oluşturulacak dosyalar hakkında bilgi verilmiştir.

Bu ana sistem yazılımı çalıştırmak için gerekli olan tüm "bağımlılıklar" ile başlar. Sonra, "Dosyalar" oluşturulan paketindeki tüm yolları tanımlar. Son olarak, "Ek Dosyalar" kaynak paketinin bir parçası değildir dosyalarını tanımlar. BuDaha"dosyalar" adlı yapı-klasörde saklayın ve kullanabilirsiniz

hakkında.önce, kaynak dosyaları, (gerekiyorsa) yamalı derlenmiş ve daha sonra / var / pisi / paket_adı / install aktarılır. Belirtilen

Geçmiş bölümünde

güncelleştirmeleri Bilgiburada listelenir. Bir hata ya da bir donanım veya sürüm değişikliği düzeltmek için bir ihtiyaç olduğunda, sürüm sayısı artırılmalıdır. Bu yeni paket PiSi ve Paket Yöneticisi bir güncelleme olarak kullanıcıya sunuluyor emin olmaktır. Bu sürüm numarası, böylece karışıklığı önlemek, paket adının sonra eklenecektir. Sürüm numarası her farklı sürüm ile bir artırılmalıdırörnek:.

pspec.xml 'deki bir

```
<? Xml version = "1.0">  
<DOCTYPE PİSİ SİSTEMİ
```



```
(zararlı görünüyor) nazardan bilgisayarınızı korumak için uygulama </ Summary>  
<Description> Knazar şey yapmak için tasarlanmış bir uygulamadır. </ Description>  
<Arşiv sha1sum = "f7c7f7e16e7c4804020885b84e6a5300455ecd40"
```

```
versionFrom = "4.4.3"> kdelibs-devel</ Dependency>
</ BuildDependencies>
</ Source>
```

belirtilen Sürüm olmadan→ de mümkündür: <Dependency> kdbase </ Dependency>

```
<Package>
<Adı> Knazar </ Name>
<RuntimeDependencies>
  <Dependency versionFrom="4.4.3"> kdbase </ Dependency>
</ RuntimeDependencies>
<Files>
  <Path fileType="executable"> / usr / bin </ Path>
  <Path fileType="doc"> / usr / share / doc </ Path>
  <Path fileType="localedata"> / usr / share / locale </ Path>
  <Path fileType="data"> / usr / share / icons </ Path>
  <Path fileType="data"> / usr / share / applications </ Path>
</ Files>
</ Package>
```

→ de mümkün belirtilen Sürüm olmadan: <Dependency> kdbase </ Dependency>

```
<History>
<update Release="10">
  <Date> 2010-05-13 </ Date>
  <version> 1.1 </ Version>
  <Comment>, kdepimlibs yeni sürümü qt sıkı bağımlılık yaz ..... </ Comment>
  <Adı> Homer Simpson </ Name>
  <EMAIL> xxxxx@pardus.org.tr </ Email>
</ Update>
|
|
<update release="1">
  <Tarih > 2005-09-28 </ Date>
  <version> 0.1 </ Version>
  <Comment> First release. </ Comment>
  <Adı> Homer Simpson </ Name>
  <EMAIL> xxxxx@pardus.org.tr </ Email >
</ Update>
</ Tarih>
</ PISI>
```

adı Kaynak ve paketinin/ (ayrıca translations.xml olarak) aynı olmalıdır gereken

Anasayfa istprojenin anabağlantı

Paketleyici-Adı ve e-posta adıdır ve, elbette, İlk paketleyicisine geçerli e-posta

Lisans adresi,yazılım lisansı olduğunu. Bir yazılım birden fazla lisans var olabilir. Olası lisansburada

Simge (. Png) bitirmeden simge adıdır bulabilirsiniz. Simge ismi çok sık pakettir. adı

PartOf Bu bu paket ait olduğu bileşenin tanımlayan component.xml içeriği olan Mümkün Bileşen bilgi kategorilerini Pardus sunucu üzerinde bulunabilir:
<https://svn.pardus.org.tr/pardus/> (component.xml arayın)

IsA olduğunu

& Açıklama Paket Yöneticisi GUI gösterilen paket hakkında bilgiler. Özet kısa ve net olmalı, Açıklama Daha fazla bilgi ile daha uzun olabilir.

Arşiv yazılım Kaynak (sağlama, türü ve yeri) 'dir. Olası türleri: tar.bz2 / zip / tar.gz / tar / gz /bz2 /

Yol fileType dosyalar yüklendikten sonra nerede belirtir. halledeceğimiz tüm fileType özelliklerini bulabilirsiniz Buraya veya buraya

çeşitli düzeyi arasında 4 boşluk olduğunu unutmayın.
Tab izin verilmez ve hatalara neden olabilir!

Sen PiSiDo kullanıyorsanız bu umurumda değil.
PiSiDo hakkında o.

Translation.xml

Burada bir çeviri Özeti ve Açıklama pspec.xml içinde verebilir. Dil kodu (**tr / fr // de nl**).

Örnek:

```
<? Xml version = "1.0">  
<PISI>
```

```
<source>
  <Adı> knazar </ Name>
  <summary xml:lang="tr"> Elem tere FIS, kem gözlere şiş </ Summary>
  <Özeti lang ="de">xml:!Schutz vor dem boesen Blick </ Summary>
  <summary xml:lang="fr"> Éloignez le mauvais oeil </ Summary>
  <Description xml:lang="tr"> Knazar hiçbirşey yapmamak Click özellikle tasarlanmış Bir programdır. </ Description>
  <Description xml:lang="de"> Knazar schuetzt vor dem boesen Blick und şapka sonst keine Funktion. </ Description>
  <Description xml:lang="fr"> Knazar est un petit Program nasip à éloigner le mauvais oeil. </ Description>
</ Kaynak
</ PISI>
```

Actions.py

Bu paketi muhtemelen inşaat en karmaşık parçasıdır.
Genellikle / çok sık Linux Yazılım üç komut ile

"configure" yüklenir:./yapılandırmak
bölümünde bölümünde kurulumunda
bölümünde build:
bölümünde make install sudo
bölümünde yüklemek



Burada PİSİ nasıl yazılım oluşturmak için talimat vermek.

Senbulacaksınız

Siz bulacak"make"

Siz install" bulacaksınız"make

actions.py Örnek

```
# / usr / bin / pythoncoding:
# - * - utf-8 - * -
#
Copyright 2005 TUBITAK / UEKAE
# GNU Genel Kamu Lisansı, sürüm 2 Lisanslıdır.
#dosyayıhttp://www.gnu.org/licenses/old-licenses/gpl-2.0.txt bak

eldepisi.actionsapi ithalat
```

```
pisi.actionsapi ithalat autotools
pisi.actionsapi ithalat pisitoolsgelen

WorkDir= "softwarename-" + get.srcVersion ()

def setup ():
    autotools.configure ()

def build ():
    autotools.make ()

def install ():
    autotools.rawInstall ("DESTDIR = % s "% get.installDIR ())
    pisitools.dodoc ('KOPYALANMASI', 'YAZARLAR', 'Değişiklikler',
'README')
```

işlevler ve olanaklar daha eksiksiz bir genel bakış için Actions_API bir wiki içine bir göz atın: http://en.pardus-wiki.org/Actions_API

.Actions API ambalajlayıcı için paketleme işlemini kolaylaştıran modüler bir yapıya sahiptir

Bu işlevleri kullanmak için, onlar actions.py tanımlanmış olmalıdır:

```
pisi.actionsapi ithalat cmaketools gelen
pisi.actionsapi ithalattanelde
pisi.actionsapi ithalat pisitools.
```

Autotools:Actions API büyük modülleri bulunmaktadır

- uygulamaları oluşturmak ve yüklemek için Standart işlevler
- **Cmaketools:** bina cmake ile yapılandırılan uygulamaları için Fonksiyonlar
- **Pisitools:.** Temelde taşımak için kullanılan En işlemleri dosya veya dizin silme, sed üzerinden, sembolik bağlantılarını dosya işleme yani için dizin kurmak için çalışma dizinden dosyaları, pisitools dahil fonksiyonları uygundur
- **Shelltools:.** özel operasyonlar için işlevler. Dışında pisitools gelen, shelltools mutlak yollar yerine görelî yolları faaliyet yeteneğine sahiptirler. Paketleyicisine dosya sisteminin en karanlık köşelerinde kullanma yeteneği verilmesi, shelltools sorumlu bir şekilde kullanılması gerekmektedir
- **Libtools:.** Ön inşa ve yapılandırma kütüphaneler için işlemleri sonradan inşa

- **alın:.** İhtiyaç duyulan ortam değişkenleri veya paketler hakkında bilgi almak için Fonksiyonlar inşaat ve montaj aşamalarında
- **Kde:..**, yapılandırma bina ve KDE uygulamalarını yüklemek için Fonksiyonlar
- **Perlmodules:** Perl modüllerinin kurulumu için işlevler
- **Pythonmodules:.** Python modülleri için işlevler
- **Scons:.** bina araçları yeni nesil autotools, scons Karşılıkları.

Ayrıca bakınız: <http://developer.pardus.org.tr/guides/packaging/actionsapi/index.html>

WorkDir = "softwarename-" + get.srcVersion ()

WorkDir ile paketlenmiş dosyaya PiSi klasörün adını verir kaynağı. Pisi / var / pisi içinde / bina ve inşa için aynı WorkDir-adını kullanır. WorkDir sourcefile içinde seçilen isim bağlıdır.

Bu yazılımın adını kullanmak çok yaygındır. Ve ile "+ srcVersion ()" PiSi pspec.xml verilen Kaynak sürümünü algılar.

(<Arşiv> Ve tarih)./ var/pisi/softwarename-0.1-1 /değil:yeşil exsample aşağıdaki WorkDir sonuçlanacaktır

Bazı testler yapmak ve birkaç kez bir paket derlemekvarsa,/ var / pisi / XXX klasör silmek için kötü bir fikir

Ayrıcaolası seçenekleri: ". "WorkDir =

WorkDir = "paketadı"

WorkDir hakkındaEk bilgiler:

Bazen kaynak arşivlerinde (tar.gz, tar.bz2, vb) bir dizin (= ayrıca "üst düzey dizin" olarak bilinen) olan . kaynak içerik mağazalar bunu ayıklamak, bunu göreceksiniz.

Örnek:

mysoftware-2.4.tar.gz

|
|- **mysoftware-2.4 /**

|
|- dosya ve klasörleri (benioku, kopyalama, yükleme kaynakları, başlık dosyaları), make, vb scriptler yapılandırmak ...

Çünkü bu ^ ^ ^ kaynak arşiv, aradığınız WorkDir değişken dizin adını tanımlamak zorunda **"mysoftware-" + get.srcVERSION ()**

Yani dizin yapısı PiSi "iş" dizini olarak aşağıda görünüyor:

```
/ var/pisi/mysoftware-2.4-1/work /
|
|- mysoftware-2 .4 /
|
|- dosya ve klasörleri (benioku, kopyalama, yükleme kaynakları, header dosyaları, yapılandırma ), vb komut yapmak ...
```

Ve bazen kaynak arşivleri vardır - Yukarıdaki örnekte aykırı herhangi bir "üst düzey dizin"sahip

Örnekolmadığı.

```
mysoftware-2.4.tar.gz
|
- dosya ve klasörleri (benioku, | kaynakları, başlık dosyaları), make, vb scriptler yapılandırmak ...durumda,yükleme,
```

Bu kopyalama,herhangi bir dizinin adını tanımlamak gerekmez. Sadece bir nokta "." Yazarak/:kaynak dosya içeriğini PiSi "iş" dizininde doğrudan olması demektir

```
/ var/pisi/mysoftware-2.4-1/work
||
Dosya ve klasörler (benioku, kopyalama, yükleme kaynakları, başlık dosyaları - ), yapmak, komut dosyalarını yapılandırma vb ...
```

actions.pybazı olası komutlar genel bakış ve örnekler

AutoTools / kurulumderlemek içinstandart komutları

autotools.configure ()

autotools.configure ("prefix = / usr /")

autotools.rawConfigure **önek%s/ usr / - ile-doxy "%** get.installDIR()

autotools.make düşünüyorum))

(autotools.install

autotools.install("**libdir=%s/ usr / lib /"%** get.installDIR())

autotools.rawInstall("**DESTDIR=%s"%** get.installDIR())

aynıdır. / configure → datadır, ev sahibi gibi tüm önemli Seçenekleri , kurulum dizini,
PiSi doğru değerleriile dolacaktır

aynıdır / configure-prefix = / usr /("-.

Ben yapılandırmak için farklı olanı emin değilim, ama senidaha fazla seçenek verebilir

rawConfigure(ile

yapmak aynıdır

make install aynıdır

Verilen seçeneklerle make install edilir

yükleme olarak, ama ekstra verilen Yükleme klasörü ile mümkündür

→ çok sık Yazılım / usr / local / share / fakat Pardus kullanımı / usr / share içine yüklemekçalışır ve bu sorunu gidermek için size rawinstalliçinde seçenek verebilir

CMAKETOOLS autotools benzer . cmake derleme yeni bir yoludur.

```
cmaketoools.configure
cmaketoools.configure(installPrefix = "%s"% (get.kdeDIR()))
cmaketoools.make ()
cmaketoools.install ()
cmaketoools.install("libdir=% s/ usr / lib / "% get.installDIR())
cmaketoools.install("libdir=%s/ usr / lib / "% get.installDIR())
cmaketoools.rawInstall("DESTDIR=%s"% get.installDIR())
```

/ configure, ama cmakeiçin doğru ayarları ilegibi aynı. olduğu burada bir gördüğünüzÖrnek ve aynıdır.. / configure-installPrefix = / usr / kde cmakeiçin yapmak aynıdır cmakeiçin make install aynıdır verilen seçenekleriyle cmake için make install edilir cmake için make install edilir Verilen seçeneklerle cmake için yüklemek mümkün, ama ekstra verilen Installationfolder ile → çok sık Yazılım içine / usr / local / share / yüklemeye çalışır ama Pardus kullanıyor / usr / share verawinstall içinde seçenek verebilir Bunu düzeltmek

GET

```
get.curDIR ()
get.curKERNEL ()
get.curPERL ()
için)get.curPYTHON
get.pkgDIR
get.workDIR ()
())get.srcNAME
get.srcVERSION ()
get.srcRELEASE ()
get.docDIR ()
get.sbinDIR ()
get.manDIR ()
get.kdeDIR ()
üzerine ...öyleve:
```

gerçek yoludöndürür
çekirdek sürümdöndürür
(Perl sürümünü döndürür
döndürürPython sürümü
/ var / pisi /'dayapı-klasörün yolunu döndürür
/ var / pisi /(içindeçalışma klasörün yolunu döndürür
verilen Kaynak Adı döner pspeckaynağı bölümünde
kaynakile sürüm numarasıdöndürür:
sürüm numarası (pspec tarih-sayı) döndürür
doc dosyaları (Pardus ot yolunu döndürür / usr / share / doc /)
sistem başlangıç (Pardus sistem dosyalarının yolunu döndürür: / sbin /)
döndürür adam dosyalarının yolunu (Pardus: / usr / share /adam /)
/ usr / kde /) kde klasöründe (Pardus yolunu döndürür

actions.py(GenelBakış ve bazı komutlar örnekleri

KDE kde bazı özel komutlar

```
kde.configure )
kde.make ()
kde.install ()
```

kdeiçin önemli olan tüm seçenekleri autotools.configure aynıdır
kdeiçin önemli olan tüm seçenekleri autotools.make aynıdır
kde için önemli olan tüm seçenekleri autotools.install aynıdır

PERL bazı komutlar perl spesifik

perl.configure ()
perl.make ()
perl.install ()

perl'in önemli olan tüm seçenekleri autotools.configure aynıdır
perl'in önemli olan tüm seçenekleri autotools.make aynıdır
aynıdır perl'in önemli olan tüm seçenekleri ile autotools.install

PYTHON bazı python özel komutları

python.configure ()
python.make ()
python.install ()

python'un önemli olan tüm seçenekleri ile autotools.configure aynıdır
python için önemli olan tüm seçenekleri ile autotools.make aynıdır
python'un önemli olan tüm seçenekleri ile autotools.install aynıdır..

Scons python dayanmaktadır ve üç klasik komutları benzer kendi araçlarını kullanarak / ***configure && &&make make install***

pythonmodules.configure ()
pythonmodules.make ()
pythonmodules.install ()

python'un önemli olan tüm seçenekleri autotools.configure aynıdır
python'un önemli olan tüm seçenekleri autotools.make aynıdır
python'un önemli olan tüm seçenekleri autotools.install aynıdır

bakışve actions.py bazı komutlarınörnekleri

PISITOOLS bazı sadece / usr / var / pisi / package-name / iş "klasörü etkileyen pisi / pisitools özel işlevleri kolay kullanımlı

pisitools.dobin("dosya", "iç"destinationfolder")

olarak çalıştırılabilir dosyayı işaretler ve hedef klasörkopyalar:
hiçbir hedef verilirse, o dosya / usr / bin /kopyalanacak
Örneğin pisitools.dobin("Pavi"+get.srcVERSION ()+". sh")

pisitools.dodir("klasör")

bir klasör oluşturur Örneğin: pisitools.dodir("/usr / share / doc / Pavi")

pisitools.dodoc("dosya")

yazılım doc klasörü içine dosyaları
örneğikopyalar:

docdoc, ancak burada sadece html dosyaları/ usr / share / doc / paket_adı / html /kopyalanır,

pisitools.doinfo("dosya")

docdoc benzer ancak burada sadece bilgi dosyaları / usr / share / info /kopyalanır

pisitools. doexe("dosya","destinationfolder")

bilgi dosyası hedef klasöre kopyalar ve yürütülebilir olarak
örnekişaretler:

Herhangi bir dizin, dosyalar/ usr / lib /kopyalanır

pisitools.dolib("libz.a","/ lib /")da, pisitools.dolib("file:örneğin("yeni."))

pisitools.domo","destinationfolder","destinationfile")

..lokalizasyonu için mo dosyaları içine yerelle po dosyaları derler ve / usr / share / locale / XX /haline dosyayı
örnekkopyalar:

Bir hedef klasör içine dosya ve bunuaadlandırmak

pisitools.dosed("dosya","searchpattern","replace_with_that")

verilen model için bir dosya içinde arar ve değiştirir
örnek.:

pisitools.dosym("sourcefile","destinationfile")

"dobin" olarak aynı, ancak varsayılan hedef klasörü değil / usr / bin / ama / usr / sbin /edilir
sembolik ("ln-s" komutuna benzer) oluşturur
Örneğin:

Belirli bir dizine mevcut tüm özelliklere sahip bir dosya, yeni bir dosya adı isteğe

örnek:pisitools.insinto("/opt / rar / bin/", "rar")

pisitools.rename("old_filename","new_filename")

bir yeniden adlandırır dosya; dizinin beyanı gerekli
örnek:pisitools.rename("/usr / bin /bash", "bash.old")

pisitools.removeDir("dizin")

siler verilen dizinin; dizinin bildirimi olduğunu

pisitools.remove(zorunlu"dosya")

bir dosya ("rm" benzer) kaldırır; dizinin beyanızorunluolan

actions.py Bakış ve bazı komutlar örnekleri

SHELLTOOLS shelltools ile kabuk komutları ile çalışabilirsiniz. Bazen pisitools benzer, ancak pisitools ile DAİMA çalışmak
/var / pisi ve / çalışma klasöründeshelltools Eğer çalışma klasörü dışında faaliyet ve belki sisteme zarar verir oysa. Onun
mümkünse Yanipisitools yerine shelltools tercih etmelisiniz. Sen shelltools kullanıyorsanız bubelirtilmişgereklidir **Workdir**
olmasıdoğrueder:

shelltools.can_access_file("dosya")

bir dosya varsa(if komutları ile birlikte kullanılır)kontrol
Örneğin **ise:** shelltools.can_access_file("/usr / bin /firefox") # bunu **else:** # bunu

shelltools.can_access_directory("yol/klasör")

Bir dizin varsa(kullanılıyorsa komutları ile)alır:
örnek **if:** shelltools.can_access_directory ("/ usr / share / applications /") # bunu **else:** # Buyapın:

shelltools.makedirs("klasör")

klasörünü oluşturur
örnek shelltools.makedirs(get.workDIR() + "/build")

shelltools.chmod("dosya", "attributes")

bir dosya özniteliklerini değiştirir
Örneğin: shelltools.chmod("/usr / bin /Pavi", 0755)

shelltools.unlink("dosya")

siler verilen dosyası
örneği: shelltools.unlink(get.workDIR() + "/" + get.srcDIR () "/ eksik"+)
→ örnekte verilen dosya üzerinde: / var / pisi / paket_adı / iş / eksik

shelltools.unlinkDir("klasör")

siler belirli bir klasöre
örneği: shelltools.unlinkDir(get.workDIR() + "/ tmpbuild")
Yukarıdaki örnekte → verilen klasör: / var / pisi / paket_adı / tmpbuild

shelltools.move("dosya", "yol")

bir veya daha fazla dosya hareket
shelltools.moveshare: örnek ("*png.", "/usr / / pixmaps'i /")

shelltools.copy("dosya", "yolörneği")

"verilen yol kopya bir veya daha fazla
dosya: shelltools.copy("lib / *", "/usr / lib /")

shelltools.copypath("SourcePath", "destinationpath")

yeni bir yola tüm klasörler ile bir dizini kopyalar

shelltools.touch("dosya")

boş bir metin dosyası oluşturur:
örneği shelltools.touch(get.workDIR() + "pavirc")

shelltools.cd("yol")

değişiklikleri dizine ("cd" e benzer)
Örneğin: shelltools.cd("/usr / share/")

shelltools.ls("yol")

verilen dizinin tüm dosyaların listesini
örnekte okur: için dosya shelltools.ls içinde (get.installDIR () + "/usr / lib / * w* ."): *daha fazla işlem için # komutları (tüm dosyaları bu isimde bir "w" var)*

shelltools.sym("kaynak", "Hedef")

ln-s ") benzer bir sembolik bağ (yaratır"

shelltools.systemSHELL-KOMUT")

kabuk tam erişim sağlar. Buradasınız: mevcut tüm kabuk komutları kullanabilirsiniz

programları için Standart Linux (Pardus) dosya yapısını

/ Usr / bin /

program için program başlatıcı ya da sembolik (bazen belki programı-bin

dosyaları ve program

K-Menüsü (çoğunlukla png dosya biçimi) gösterilen yazılım
En iyi sonuç simge için olmalıdır

de → program link K-Menü

Eğer seçim var doğru kategori programın linki (oyun, video, sistem,) sahip olmak için
[Http://standards.freedesktop.org/menu-spec/1.0/apa:dogru bir . html](http://standards.freedesktop.org/menu-spec/1.0/apa:dogru-bir-.html)

masaüstü dosyası örneği:

```
[Desktop Entry]
Encoding = UTF-8
Version = 1.0
Adı = Areca Yedekleme
GenericName kolay bir şekilde içinde = Yedekleme dosyaları
GenericName[de] = Datensicherung einfach gemacht
GenericName[nl] = Yedekleme van bestanden maken
Yorum = basit, çok yönlü ve güvenilir bir yedekleme çözümü
Yorum[de] = Eine einfache, vielseitige und verlässliche Backuplösung.
Yorum[nl] = Een eenvoudige, veelzijdige en betrouwbare oplossing voor yedekleme oplossing.
Icon = /usr/share/pixmaps/i/areca.png / (pixmap standart klasör da ise "areka" iyi
```

Eğer örnek dosya görebilirsiniz, ayrıca[de, nl] GenericName ve yorumun bir çevirilebilir
sadece GenericName da K-Menüsü ve Yorum gösterilen edilecek olduğu

a. çalıştırma, komut veya dosya burada yer almalıdır. bin aracılığıyla paket yöneticisi dışında

bir yama dosyası oluşturma

bir kaynak veya bir. desktop dosyasındaki bazı şeyleri düzeltmek için kurulu, bir yama dosyası kullanmak mümkündür. yama olmalıdır dizin / için pspec.xml 'deki yanındaki dosyası dahil.

bir yama oluşturmak için, sadece aynı klasöre orijinal dosyayı kopyalamak ve (xxxx.orig) yeniden adlandırın. Sonra yeni dosyasında değişiklikler. Bundan sonra, terminal başlatın ve aynı dizinde aşağıdaki komutu yazın test.txt.orig):

diff-Naur / metin dosyalarıyla / test.txt.orig / metin dosyalarıyla / test.txt> test.patch

orijinal dosya (/ metin dosyalarıyla /

```
A
B
C
D
```

yeni dosya (/ metin dosyalarıyla / test.txt):

```
A
BB
CCC

D
```

Patch (test.patch):

```
--- / metin dosyalarıyla / test.txt.orig 2011-02-13 23:20:45.501000079 +0100
+++ / metin dosyalarıyla / test . txt    2011-02-13 23:21:15.841000080 +0100
@@ -1,4 +1,5 @@
A-B-C

+ BB
+ CCC
+
D
```

yama kullanmak için sadece oluşturulan yama dosyasını kaydetmek içine dosyaları oluşturmak dosyaların klasör ve pspec.xml içine aşağıdaki kırmızı çizgi ekleyin

```
level = "1"> test.patch </ Patch>
</ Patches>
</ Source>
```

level süre dışarı kesilecek ne kadar gösterir yama.

İlk örneği you would have to use level="1":

```
--- qrencode/configure_org 2010-02-03 00:43:16.000000000 +0100
+++ qrencode/configure 2011-06-03 00:14:07.566744000 +0200
```

In second example You would have to use level="2":

```
--- a/b/configure_org 2010-02-03 00:43:16.000000000 +0100
+++ c/d/configure 2011-06-03 00:14:07.566744000 +0200
```

Patching is not difficult :-)

If during building package PiSi is removing some necessary use a „Keepfile“ function in actions.py!

Here is an example:

```
from pisi.actionsapi import pisitools
from pisi.actionsapi import shelltools

WorkDir = "."

NoStrip = ["/"] # stripping disabled
KeepSpecial=["python"] # do not remove .pyc files

content_of_rpmunpack = "xxxx.rpm"

def setup():
    shelltools.system("rpm2targz xxxx.rpm")
```



files (eg *.pyc-files) then You can

```
shelltools.system("tar -zxvf xxxx.tar.gz")

def install():
    pisitools.insinto("/", "usr")
    pisitools.insinto("/", "etc")
```

More hidden special functions You will find in PiSi build python script:

<http://svn.pardus.org.tr/uludag/trunk/pisi/pisi/operations/build.py>

If You have to detect the architecture of Your system by actions.py (ie.g. if only precompiled software is available), then You can use the „get.ARCH()“ function in actions.py!

Have a look:

```
if get.ARCH() == "x86_64":
    pisitools.dobin("bin/amd64/xxxx")
elif get.ARCH() == "i686":
    pisitools.dobin("bin/i386/xxxx")
```

Another way to do is that, You can hold the directory names in a dictionary.

```
bin_dirs = { "i686" : "i386", "x86_64" : "amd64" }
```

and write a single line of code:

```
pisitools.dobin("bin/%s/makemkvcon" % bin_dirs[get.ARCH()])
```

You can search for examples of get.ARCH() usage in the main repository (say, 2011/devel)

```
grep -rHln get.ARCH . --exclude-dir=.svn
```

I would recommend reading the `/usr/lib/pardus/pisi/actionsapi` source code.

What You use in `actions.py` is right there. Eg `pisitools.dobin` is the `dobin` function in `/usr/lib/pardus/pisi/actionsapi/pisitools.py` and there are a bunch of documentation in the `__doc__` values of the functions.

Also, investigating examples in the repositories will help a lot

G oodby e...



...and have fun with Pardus and the growing Pardus community !