

Actions API

A. Murat Eren (meren@uludag.org.tr)

7 Ekim 2005

Özet

Bu belgede Actions API tanıtılmış ve içerisindeki fonksiyonların işlevleri açıklanmış ve örneklendirilmiştir. PİSİ mimarisi ya da PİSİ paketi hazırlama ile ilgili bilgilere yer verilmemiştir.

İçindekiler

1	Actions	5
2	Actions API	5
3	Pisitools	7
3.1	dobin	7
3.2	dodir	7
3.3	dodoc	7
3.4	doexe	7
3.5	dohard	8
3.6	dohtml	8
3.7	doinfo	8
3.8	dojar	8
3.9	dolib	8
3.10	dolib_a	9
3.11	dolib_so	9
3.12	doman	9
3.13	domo	9
3.14	domove	10
3.15	dosed	10
3.16	dosbin	10
3.17	dosym	11
3.18	insinto	11
3.19	newdoc	11
3.20	newman	12
3.21	remove	12
3.22	removeDir	12
4	Autotools	13
4.1	configure	13
4.2	rawConfigure	13
4.3	compile	13
4.4	make	13
4.5	install	13
4.6	rawInstall	13
4.7	aclocal	13
4.8	autoconf	14
4.9	automake	14
5	Libtools	15
5.1	prelib	15
5.2	prelib_so	15
5.3	gnuconfig_update	15
5.4	libtoolize	15
5.5	gen_usr_ldscript	15

6	Shelltools	16
6.1	can_access_file	16
6.2	can_access_directory	16
6.3	makedirs	16
6.4	chmod	16
6.5	unlink	17
6.6	unlinkDir	17
6.7	move	17
6.8	copy	17
6.9	copytree	17
6.10	touch	18
6.11	cd	18
6.12	ls	18
6.13	export	18
6.14	system	19
6.15	isLink	19
6.16	realPath	19
6.17	baseName	19
6.18	dirName	19
6.19	sym	19
7	Get	20
7.1	pkgDIR	20
7.2	workDIR	20
7.3	installDIR	20
7.4	srcNAME	20
7.5	srcVERSION	20
7.6	srcRELEASE	20
7.7	srcTAG	21
7.8	srcDIR	21
7.9	HOST	21
7.10	CFLAGS	21
7.11	CXXFLAGS	21
7.12	LDFLAGS	21
7.13	docDIR	21
7.14	sbinDIR	21
7.15	infoDIR	21
7.16	manDIR	21
7.17	dataDIR	22
7.18	confDIR	22
7.19	localstateDIR	22
7.20	defaultprefixDIR	22
7.21	exists_binary	22
7.22	getBinutilsInfo	22
7.23	AR	22
7.24	AS	22
7.25	CC	22
7.26	CXX	22
7.27	LD	22
7.28	NM	23

7.29	RANLIB	23
7.30	F77	23
7.31	GCJ	23
8	Kde	24
8.1	configure	24
8.2	make	24
8.3	install	24

1 Actions

Bir uygulamanın ikili bir PİSİ paketi haline getirilmesi sürecinde PİSİ'nin build sistemi tarafından kullanılan minimum bileşenler, paketlenen uygulamanın kaynak kodu, *pspec.xml* dosyası ve *actions.py* dosyasıdır. Ayrıca varsa yamalar ve ek dosyalar da uygulamanın kaynak kodunun parçası olarak düşünülebilir. **pspec.xml** dosyası uygulamanın kaynak kodunu taşıyan arşiv dosyasının nereden çekileceği, bu arşiv dosyası açılırken hangi yamaların uygulanacağı, bu uygulamayı kimin paketliyor olduğu, uygulamanın adı, sürüm numarası gibi meta sayılabilecek bilgilerin saklandığı dosya iken, bu uygulamanın derlenmesi ve build edilmesi aşamasında yapılması gereken işlerin tarifi de, bu belgede bahsedilecek olan **Actions API**'dan faydalanan **actions.py** dosyasıdır. Örnek bir *actions.py* dosyası aşağıdaki gibidir:

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
from pisi.actionsapi import autotools
from pisi.actionsapi import pisitools
from pisi.actionsapi import get
def setup():
    autotools.configure("--enable-nls --bindir=/bin --with-rmt=/usr/sbin/rmt")
def build():
    autotools.make()
def install():
    autotools.rawInstall("DESTDIR=%s" % get.installDIR())
    pisitools.dodoc("ChangeLog", "NEWS", "README", "INSTALL")
    pisitools.remove("/usr/share/man/man1/mt.1")
    pisitools.removeDir("/usr/libexec")
```

actions.py dosyası ile ilgili daha fazla bilgi PİSİ Mimari Belgesi'nden alınabilir.

Bu belgede Actions API üzerinde durulacaktır.

2 Actions API

Actions API, PİSİ paketlerinin oluşturulması esnasında paketi oluşturacak kaynağın hangi işlemlere tabi tutulacağına dair bilgilerin Python programlama dili ile yazıldığı *actions.py* isimli dosya içerisinden paket yapıcılarının kullanabileceği fonksiyonları barındıran kütüphanedir. Paket yapılırken bu kütüphane dışında herhangi bir Python modülü kullanılmaması şiddetle tavsiye edilir, fakat paketleyici bu konuda özgürdür.

actions.py isimli dosya, kaynak koddan bir ikili ve kurulmaya hazır PİSİ paketi oluşturma sürecini tarif eden bir dosyadır. Bu dosya içerisinde yapılacak tarifin doğru şekilde yapılabilmesi için paket yapıcının, yapmakta olduğu paketi yeterince tanıması gereklidir. Bu kaynağın daha önce paketlendiği dağıtımların paket sistemlerinin spec

dosyalarına göz atarak hızlı bir şekilde paketin nasıl oluşturulması gerektiğine dair fikir sahibi olmak mümkündür, fakat bunu PİSİ'nin diğer paket yöneticileri ile arasında organik bir bağ olduğu şeklinde yorumlanması yanlış olur.

Actions API içerisinde paket yapıcılara sunulan fonksiyonlar amaçlarına göre birbirinden ayrılmış, ayrı modüller içerisinde toplanmışlardır. Her bir modülün *actions.py* içerisinde dahil edildiğinde hangi fonksiyonları sunduğu ve bu fonksiyonların özellikleri sonraki başlıklarda incelenmiştir. Bununla beraber modüllerin basitçe amaçları şu şekilde tanımlanabilir:

Autotools: Uygulama derlemek ve kurmak için gerekli olan standart işlevleri sunar.

Pisitoools: *Pisitoools*, içerisinde paket yapıcılarının en sık kullanacağı fonksiyonları sunan modüldür. Özel durumlar dışında tüm işlemler bu modüldeki fonksiyonlar yardımı ile gerçekleştirilmelidir. Bu modül ile beraber gelen fonksiyonların tümü *relative path*'ler ile çalışır ve çok büyük bir kısmı da *work* dizininden *install* dizinine dosya taşıma işlerinde kullanılırlar. Bununla beraber symlink oluşturma, dosyalar üzerinde değişiklik yapma (sed ile), dosya ya da dizin silme gibi ihtiyaçlara da yanıt verir.

Shelltools: Bu modül ile beraber özel durumlarda kullanılabilecek fonksiyonlar sunulur. Kimi işlevler *pisitoools* içindeki işlevlerle benzerlik gösterir, fakat *shelltools* içerisindekilere absolut path verebilir, böylece sıkça karşılaşılmayacak *work* dizininden *work* dizinine, *install* dizininden *install* dizinine dosya taşıma işlemleri, dosya izinlerini değiştirme, kabuk'ta komut çalıştırma, çalışma dizini değiştirme gibi ihtiyaçlar karşılanabilir. Fakat mümkün olan her durumda *Pisitoools* içerisindeki fonksiyonların tercih edilmesi şarttır. Shelltools içerisindeki fonksiyonlar paketleyiciyi tamamen özgür bıraktığı için dikkatli bir şekilde kullanılmalılardır.

Libtools: Kütüphanelerin doğru şekilde yapılandırılması, uygulamanın doğru şekilde derlenebilmesi için ihtiyaç duyulan derleme öncesi ve sonrası işlevlerini sunar.

Get: Derleme öncesi yapılandırma, derleme ve kurulum esnasında ihtiyaç duyulabilecek çeşitli ortam değişkenlerinin ve paket ile ilgili bilgilerin elde edilmesi için işlevler sunar.

Kde: KDE uygulamaları için derleme öncesi yapılandırma, derleme ve kurulum işlevlerini sunar.

Takip eden bölümlerde bu modüller içerisindeki fonksiyonlar tek tek açıklanmış ve örneklenmiş, en son bölümde de örnek bir *actions.py* dosyası verilmiştir. Fonksiyonlar modüllerin altbaşlıkları olarak gösterilmiştir, fonksiyon ismini takip eden satırda fonksiyonun aldığı parametreler belirtilmiş (bir fonksiyon hiç parametre almadığı *None* kelimesi ile gösterilmiştir), takip eden paragrafta nasıl bir işlev sunduğu açıklanmış, bu açıklamanın ardından da örnek bir kullanımı verilmiştir.

3 Pisitools

3.1 dobin

(sourceFile, destinationDirectory = '/usr/bin')

work dizini içerisindeki yolu *sourceFile* ile verilen bir çalıştırılabilir dosyayı öntanımlı olarak */bin*, *destinationDirectory* parametresi ile tam yolu verilmiş dizin içerisine koyar.

- `pisitools.dobin("sed/sed", "/bin")`
- `pisitools.dobin("zipsplit")`

3.2 dodir

(destinationDirectory)

Install dizini içerisine verilen isimde bir dizini oluşturur.

- `pisitools.dodir("/usr/include/awk")`
- `pisitools.dodir("/usr/" + get.HOST() + "/include")`
- `pisitools.dodir("/usr/share/doc/%s/examples" % get.srcTAG())`

3.3 dodoc

*(*sourceFiles)*

work dizini içerisindeki yolu verilen dosya ya da dosya listesini *install* dizini altındaki */usr/share/doc/PACKAGE* altına kopyalar. Metakarakter kullanılabilir.

- `pisitools.dodoc("README")`
- `pisitools.dodoc("*.html")`
- `pisitools.dodoc("FAQ", "README", "ChangeLog.*", "algorithm.txt")`

3.4 doexe

(sourceFile, destinationDirectory)

work dizinindeki yolu verilen bir dosyayı *destinationDirectory* ile belirlenen *install* dizini içerisindeki hedefe çalıştırılabilir dosya olarak kopyalar. *sourceFile* için Metakarakter kullanılabilir.

- `pisitools.doexe("extras/scsi-devfs.sh", "/etc/udev/scripts/")`
- `pisitools.doexe("etc/hotplug/*.rc", "/etc/hotplug/")`

3.5 dohard

(sourceFile, destinationFile)

sourceFile ve *destinationFile* arasında bir hardlink oluşturur (FIXME).

3.6 dohtml

*(*sourceFiles)*

work dizini içerisinde yolu verilen dosya ya da dosya listesini *install* dizni altındaki */usr/share/doc/PACKAGE/html* altına kopyalar. Metakarakter kullanılabilir.

- `pisitools.dohtml("index.html")`
- `pisitools.dohtml("doc/*")`

3.7 doinfo

*(*sourceFiles)*

work dizini içerisinde yolu verilen dosya ya da dosya listesini *install* dizni altındaki */usr/share/info* altına kopyalar. Metakarakter kullanılabilir.

- `pisitools.doinfo("*.info")`

3.8 dojar

(None)

jar dosyalarını */usr/share/PACKAGE/lib* altına kopyalar ve */usr/share/PACKAGE/classpath.env* içerisine ekler (FIXME).

3.9 dolib

(sourceFile, destinationDirectory = '/usr/lib')

work dizini içerisindeki bir kütüphaneyi öntanımlı olarak *install* dizini altında */usr/lib* içerisine kopyalar.

- `pisitools.dolib("libz.a")`
- `pisitools.dolib("lib/libpci.a")`
- `pisitools.dolib("libbz2.so.1.0.2", "/lib")`

3.10 dolib_a

```
(sourceFile, destinationDirectory = '/usr/lib')
```

work dizini içerisindeki bir statik kütüphaneyi öntanımlı olarak *install* dizini altında */usr/lib* içerisine gerekli dosya izinleri ile kopyalar.

- pisitools.dolib_a("lib/libpci.a")
- pisitools.dolib_a("libdb1.a")

3.11 dolib_so

```
(sourceFile, destinationDirectory = '/usr/lib')
```

work dizini içerisindeki bir paylaşımlı kütüphaneyi öntanımlı olarak *install* dizini altında */usr/lib* içerisine gerekli dosya izinleri ile kopyalar.

- pisitools.dolib_so("libdb1.so.2")

insert the static library into */usr/lib* with permission 0755

3.12 doman

```
(*sourceFiles)
```

work dizini içerisindeki yolu verilen dosya ya da dosya listesini *install* dizini altındaki */usr/share/man/* altına kopyalar. Metakarakter kullanılabilir.

- pisitools.doman("logrotate.8")
- pisitools.doman("doc/bash.1", "doc/bashbug.1", "doc/builtins.1", "doc/rbash.1")
- pisitools.doman("*.1-8")

3.13 domo

```
(sourceFile, locale, destinationFile)
```

work dizini içerisindeki yolu *sourceFile* ile verilen bir *po* dosyasını, *locale* ile belirlenen dil için derleyerek bir *mo* dosyası oluşturur ve bu dosyayı *install* altındaki */usr/share/locale/LOCALE/LC_MESSAGES* dizini içerisine *destinationFile* ile verilen isimle kopyalar.

- pisitools.domo("po/tr.po", "tr", "pam_login.mo")

3.14 domove

```
(sourceFile, destination, destinationFile = ")
```

install dizini içerisindeki yolu *sourceFile* ile verilen dosyayı, yine *install* dizini içerisindeki yolu *destination* ile verilen dizin içerisine eğer belirtilmişse *destinationFile* ismi ile taşır.

- `pisitools.domove("/usr/bin/passwd", "/bin/")`
- `pisitools.domove("/usr/bin/yacc", "/usr/bin", "yacc.bison")`
- `pisitools.domove("/usr/docs/", "/usr/share/doc/%s/html/" % get.srcTAG())`

3.15 dosed

```
(sourceFile, findPattern, replacePattern = ")
```

work dizini içerisindeki yolu verilen *sourceFile* içerisinde istenen değişiklikleri yapar (FIXME: dosed dökümantasyonunun biraz daha genişletilmesi gerekli).

- `pisitools.dosed("gcc/version.c", "<URL:http://gcc.gnu.org/bugs.html>", "<URL:http://bugs.uludag.org.tr>")`
- `pisitools.dosed("sshd_config", "(?m)^(^#UsePAM).*", r"UsePAM yes")`
- `pisitools.dosed("unix/Makefile", "-O3", get.CFLAGS())`
- `pisitools.dosed("Make.Rules", "^HAVE_NDBM_H=yes", "HAVE_NDBM_H=no")`
- `pisitools.dosed("Makefile.def", "CC=cc", "CC=%s" % get.CC())`
- `pisitools.dosed("automake.texi", "(?m)^(@setfilename.*)automake", r"\1automake1.7")`

3.16 dosbin

```
(sourceFile, destinationDirectory = '/usr/sbin')
```

work dizininde içerisindeki yolu *sourceFile* ile verilen bir çalıştırılabilir dosyayı ön-tanımlı olarak */sbin* verilmiş ise *destinationDirectory* ile tam yolu verilmiş dizin içerisine kopyalar.

- `pisitools.dosbin("traceroute6")`
- `pisitools.dosbin("extras/scsi_id/scsi_id", "/sbin")`

3.17 dosym

(sourceFile, destinationFile)

install dizini içerisinde göstereceği yol *sourceFile* ile verilmiş olan ve *install* dizin içerisindeki yolu *destinationFile* ile verilmiş bir *symlink* oluşturur.

- pisitools.dosym("gzip", "/bin/gunzip")
- pisitools.dosym("libdb1.so.2", "/usr/lib/libdb.so.2")
- pisitools.dosym("../bin/lsmmod", "/sbin/lsmmod")
- pisitools.dosym("/usr/X11R6/include/X11", "/usr/include/X11")

3.18 insinto

(destinationDirectory, sourceFile, destinationFile = "")

work dizini içerisindeki yolu *sourceFile* ile verilmiş olan bir dosyayı *install* dizini içerisindeki yolu *destinationDirectory* ile verilmiş olan dizin içerisine, eğer verilmişse *destinationFile* ismi ile, eğer verilmemişse kendi adı ile kopyalar. Dosyanın *work* dizini içerisindeki *uid/guid* ve izinleri korunur. *destinationFile* parametresi kullanılmamışsa, *sourceFile* için metakarakter kullanılabilir.

- pisitools.insinto("/opt/rar/bin", "rar")
- pisitools.insinto("/etc/", "doc/nanorc.sample", "nanorc")
- pisitools.insinto("/etc/hotplug", "etc/hotplug/*map")

3.19 newdoc

(sourceFile, destinationFile)

work dizini içerisindeki yolu *sourceFile* ile verilmiş olan dosyayı */usr/share/doc/PACKAGE/* dizini içerisine *destinationFile* adı ile kopyalar.

- pisitools.newdoc("extras/volume_id/README", "README_volume_id")
- pisitools.newdoc("gprof/ChangeLog.linux", "gprof/ChangeLog.linux")
- pisitools.newdoc("bfd/PORTING", "bfd/PORTING")

3.20 newman

(sourceFile, destinationFile)

work dizini içerisindeki yolu *sourceFile* ile verilmiş olan dosyayı */usr/share/man/manPREFIX/* dizini içerisine *destinationFile* adı ile kopyalar.

- `pisitools.newman("less.nro", "less.1")`

3.21 remove

(sourceFile)

install dizini içerisindeki yolu *sourceFile* ile verilmiş olan dosyayı siler.

- `pisitools.remove("/usr/lib/libdb_cxx.so")`

3.22 removeDir

(destinationDirectory)

install dizini içerisindeki yolu *destinationDirectory* ile verilmiş olan dizini ve alt dizinlerini siler.

- `pisitools.removeDir("/usr/lib")`

4 Autotools

4.1 configure

```
(parameters = " )
```

configure source with given parameters= "--with-nls --with-libusb --with-something-usefull"

4.2 rawConfigure

```
(parameters = " , prefix=" )
```

configure source with given parameters= "--prefix=/usr --libdir=/usr/lib --with-nls"

4.3 compile

```
(parameters = " )
```

4.4 make

```
(parameters = " )
```

make source with given parameters= "all" || "doc" etc.

4.5 install

```
(parameters = " )
```

install source into install directory with given parameters

4.6 rawInstall

```
(parameters = " )
```

install source into install directory with given parameters= PREFIX=%s % get.installDIR()

4.7 aclocal

```
(parameters = " )
```

generates an aclocal.m4 based on the contents of configure.in.

4.8 autoconf

```
(parameters = " )
```

generates a configure script

4.9 automake

```
(parameters = " )
```

generates a makefile

5 Libtools

5.1 preplib

(sourceDirectory = '/usr/lib')

5.2 preplib_so

(sourceDirectory)

5.3 gnuconfig_update

(None)

En yeni config.* dosyalarını kaynak üzerine kopyalar.

5.4 libtoolize

(parameters = "")

5.5 gen_usr_ldscript

(dynamicLib)

6 Shelltools

6.1 can_access_file

(*sourceFile*)

Tam yolu *sourceFile* ile verilmiş olan dosyanın erişilebilir olup olmadığını bilgisini döndürür.

- `shelltools.can_access_file("/usr/share/terminfo/%s" % termfile)`

```
if shelltools.can_access_file("/usr/share/terminfo/%s" % termfile):  
    do_something_here
```

6.2 can_access_directory

(*destinationDirectory*)

Tam yolu *destinationDirectory* ile verilmiş olan dizinin erişilebilir olup olmadığını bilgisini döndürür.

6.3 makedirs

(*destinationDirectory*)

Tam yolu *destinationDirectory* ile verilmiş olan dizini oluşturur.

- `shelltools.makedirs("%s/build" % get.workDIR())`
- `shelltools.makedirs("%s/build-default-i686-pc-linux-gnu-nptl" % get.workDIR())`

6.4 chmod

(*sourceFile*, *mode* = 0755)

Tam yolu *sourceFile* ile verilen dosyanın izinlerini değiştirir. Metakarakter kullanılabilir.

- `shelltools.chmod("config/config.sub")`
- `shelltools.chmod(get.installDIR() + "/lib/libz.so.*")`
- `helltools.chmod("%s/usr/lib/misc/pt_chown" % get.installDIR(), 4711)`
- `shelltools.chmod(get.installDIR() + "/etc/ssh/sshd_config", 0600)`

6.5 unlink

(*sourceFile*)

Tam yolu *sourceFile* ile verilmiş olan dosyayı siler.

- shelltools.unlink(get.workDIR() + '/' + get.srcDIR() + "/missing")

6.6 unlinkDir

(*sourceDirectory*)

Tam yolu *sourceDirectory* ile verilmiş olan dizini ve alt dizinlerini siler.

- shelltools.unlinkDir(get.workDIR() + "/tmpbuild")

6.7 move

(*sourceFile*, *destinationFile*)

Tam yolu *sourceFile* ile verilmiş dosya ya da dizini tam yolu *destinationFile* ile verilmiş dosya ya da dizine taşır. Metakarakter kullanılabilir.

- shelltools.move("ac-wrapper.sh", "%s/usr/lib/misc/" % get.installDIR())
- shelltools.move("proc/*.h", "%s/usr/include/proc/" % get.installDIR())

6.8 copy

(*sourceFile*, *destinationFile*)

Tam yolu *sourceFile* ile verilmiş olan bir dosyayı tam yolu *destinationFile* ile verilmiş bir dosya olarak kopyalar.

- shelltools.copy("Makefile.pre.in", "%s/usr/lib/python2.3/config/" % get.installDIR())

6.9 copytree

(*source*, *destination*, *sym=False*)

Tam yolu *source* ile verilmiş olan dizini, tam yolu *destination* ile verilmiş hedefe kopyalar.

- shelltools.copytree("include/linux/", "%s/usr/include/linux/" % get.installDIR())
- shelltools.copytree("include/asm-generic/", "%s/usr/include/asm-generic/" % get.installDIR())

6.10 touch

(*sourceFile*)

Tam yolu *sourceFile* ile verilen dosyanın son erişim tarihini erişim anı yapar, eğer dosya yoksa onu yaratır.

- shelltools.touch(get.workDIR() + "aclocal.m4")
- shelltools.touch("gcc/c-gperf.h")

6.11 cd

(*directoryName* = " ")

Çalışma dizinini tam yolu *directoryName* ile verilen dizin içerisine değiştirir.

- shelltools.cd("build_unix")
- shelltools.cd("%s/build-default-i686-pc-linux-gnu-nptl" % get.workDIR())

6.12 ls

(*source*)

Tam yolu *source* ile verilen dizin ya da dosyaların bir listesini döndürür, bu liste içerisindeki her bir dosya üzerinde bir işlem yapmak gerektiği durumlarda kullanılabilir, metakarakterlerden yararlanılabilir.

- shelltools.ls(get.installDIR() + "/usr/lib/*w.*")

```
for file in shelltools.ls(get.installDIR() + "/usr/lib/*w.*"):  
    do_something_with_file_here..
```

- shelltools.ls(get.installDIR() + "/bin/")

6.13 export

(*key*, *value*)

key ile tanımlanmış isimdeki bir ortam değişkenini *value* değeri ile set eder.

- shelltools.export("WANT_AUTOCONF", "2.5")
- shelltools.export("CXX", get.CXX())
- shelltools.export("LDFLAGS", get.LDFLAGS() + "-Wl,-z,now")
- shelltools.export("LC_ALL", "C")

6.14 system

(*command*)

command ile verilen komutu kabuğa işletir, kabuktan dönen sonuç geridönüş değeridir.

- shelltools.system("./update-pciids.sh &> /dev/null")
- shelltools.system("python setup.py build")

6.15 isLink

(*sourceFile*)

Tam you *sourceFile* ile verilen dosyanın link olup olmadığı bilgisini döndürür.

- shelltools.isLink(get.installDIR() + '/maybe/link')

6.16 realPath

(*sourceFile*)

Tam you *sourceFile* ile verilen linkin işaret ettiği gerçek dizinin yolunu döndürür.

- shelltools.realPath(get.installDIR() + link)

6.17 baseName

(*sourceFile*)

sourceFile ile verilen dosya yolunun gösterdiği dosyanın adını döndürür (os.path.basename()). Aşağıdaki örnekteki gibi durumlarda kullanılabilir.

- shelltools.baseName(shelltools.realPath(link))

6.18 dirName

(*sourceFile*)

Tam yolu *sourceFile* ile verilen bir dosya yolunun son dizinlerden ibaret olan kısmını döndürür (bu kimi zaman verilen *sourceFile*'ın tamamıdır).

6.19 sym

(*sourceFile*, *destinationFile*)

Göstereceği yol *sourceFile* ile verilmiş olan ve tam yolu *destinationFile* ile verilmiş bir *symlink* oluşturur.

7 Get

7.1 pkgDIR

(*None*)

Paketin açıldığı ve içerisinde *work* ve *install* dizinlerinin olduğu dizinin tam yolunu döndürür.

7.2 workDIR

(*None*)

work dizininin tam yolunu döndürür.

7.3 installDIR

(*None*)

install dizininin tam yolunu döndürür.

7.4 srcNAME

(*None*)

Kaynak paketinin adını döndürür (automake-wrapper-1.7-3 için, automake-wrapper).

7.5 srcVERSION

(*None*)

Paketin kaynak versiyonunu döndürür (automake-wrapper-1.7-3 için, 1.7).

7.6 srcRELEASE

(*None*)

Paket kaynağının dağıtım tarafından çıkarılmış kaçınıcı sürümü olduğunu döndürür (automake-wrapper-1.7-3 için, 3)

7.7 srcTAG

(None)

Paket adı + '-' + mainstream sürüm numarası + '-' + dağıtım sürüm numarası'nda oluşan ismi döndürür (automake-wrapper-1.7-3 için, automake-wrapper-1.7-3).

7.8 srcDIR

(None)

7.9 HOST

(None)

7.10 CFLAGS

(None)

7.11 CXXFLAGS

(None)

7.12 LDFLAGS

(None)

7.13 docDIR

(None)

7.14 sbinDIR

(None)

7.15 infoDIR

(None)

7.16 manDIR

(None)

7.17 dataDIR

(None)

7.18 confDIR

(None)

7.19 localstateDIR

(None)

7.20 defaultprefixDIR

(None)

7.21 exists_binary

(bin)

7.22 getBinutilsInfo

(util)

7.23 AR

(None)

7.24 AS

(None)

7.25 CC

(None)

7.26 CXX

(None)

7.27 LD

(None)

7.28 NM

(None)

7.29 RANLIB

(None)

7.30 F77

(None)

7.31 GCJ

(None)

8 Kde

8.1 configure

(parameters = ")

8.2 make

(None)

8.3 install

(None)