

**NAME**

AptPkg::Cache - APT package cache interface

**SYNOPSIS**

```
use AptPkg::Cache;
```

**DESCRIPTION**

The AptPkg::Cache module provides an interface to **APT**'s package cache.

**AptPkg::Cache**

The AptPkg::Cache package implements the **APT** pkgCacheFile class as a hash reference (inherits from AptPkg::hash). The hash keys are the names of packages in the cache, and the values are AptPkg::Cache::Package objects (which in turn appear as hash references, see below).

*Constructor*

```
new([LOCK])
```

Instantiation of the object uses configuration from the `$AptPkg::Config::_config` and `$AptPkg::System::_system` objects (automatically initialised if not done explicitly).

The cache initialisation can be quite verbose — controlled by the value of `$_config->{quiet}`, which is set to “2” (quiet) if the `$_config` object is auto-initialised.

The cache directory is locked if `LOCK` is true.

It is important to note that the structure of the returned object contains self-referential elements, so some care must be taken if attempting to traverse it recursively.

*Methods*

`files`

Return a list of AptPkg::Cache::PkgFile objects describing the package files.

`packages`

Return an AptPkg::PkgRecords object which may be used to retrieve additional information about packages.

`get`, `exists`, `keys`

These methods are used to implement the hashref abstraction: `$obj->get($pack)` and `$obj->{$pack}` are equivalent.

`is_multi_arch`

Cache is multi-arch enabled.

`native_arch`

Native architecture.

**AptPkg::Cache::Package**

Implements the **APT** pkgCache::PkgIterator class as a hash reference.

*Keys*

`Name`

`Section`

`Arch`

Package name, section and architecture.

`FullName`

Fully qualified name, including architecture.

`ShortName`

The shortest unambiguous package name: the same as `Name` for native packages, and `FullName` for non-native.

SelectedState

InstState

CurrentState

Package state from the status file.

SelectedState may be `Unknown`, `Install`, `Hold`, `DeInstall` or `Purge`.

InstState may be `Ok`, `ReInstReq`, `HoldInst` or `HoldReInstReq`.

CurrentState may be `NotInstalled`, `UnPacked`, `HalfConfigured`, `HalfInstalled`, `ConfigFiles` or `Installed`.

In a numeric context, SelectedState, InstState and CurrentState evaluate to an `AptPkg::State::` constant.

VersionList

A reference to an array of `AptPkg::Cache::Version` objects describing available versions of the package.

CurrentVer

An `AptPkg::Cache::Version` object describing the currently installed version (if any) of the package.

RevDependsList

A reference to an array of `AptPkg::Cache::Depends` objects describing packages which depend upon the current package.

ProvidesList

For virtual packages, this is a reference to an array of `AptPkg::Cache::Provides` objects describing packages which provide the current package.

Flags

A comma separated list of flags: `Auto`, `Essential` or `Important`.

In a numeric context, evaluates to a combination of `AptPkg::Flag::` constants.

[Note: the only one of these you need worry about is `Essential`, which is set based on the package's header of the same name. `Auto` seems to be used internally by `APT`, and `Important` seems to only be set on the apt package.]

Index

Internal `APT` unique reference for the package record.

### **AptPkg::Cache::Version**

Implements the `APT` `pkgCache::VerIterator` class as a hash reference.

*Keys*

VerStr

Section

Arch

The package version, section and architecture.

MultiArch

Multi-arch state: `None`, `All`, `Foreign`, `Same`, `Allowed`, `AllForeign` or `AllAllowed`.

In a numeric context, evaluates to an `AptPkg::Version::` constant.

ParentPkg

An `AptPkg::Cache::Package` object describing the package providing this version.

DescriptionList

A list of `AptCache::Cache::Description` objects describing the files which describe a package version. The list includes both `Package` files and `Translation` files, which contain translated `Description` fields.

TranslatedDescription

An `AptCache::Cache::Description` object for the current locale, which will generally be a `Translation` file.

**DependsList**

A reference to an array of AptPkg::Cache::Depends objects describing packages which the current package depends upon.

**ProvidesList**

A reference to an array of AptPkg::Cache::Provides objects describing virtual packages provided by this version.

**FileList**

A reference to an array of AptPkg::Cache::VerFile objects describing the packages files which include the current version.

**Size**

The *.deb* file size, in bytes.

**InstalledSize**

The disk space used when installed, in bytes.

**Index**

Internal **APT** unique reference for the version record.

**Priority**

Package priority: `important`, `required`, `standard`, `optional` or `extra`.

In a numeric context, evaluates to an AptPkg::VerPriority:: constant.

**AptPkg::Cache::Depends**

Implements the **APT** pkgCache::DepIterator class as a hash reference.

*Keys***DepType**

Type of dependency: `Depends`, `PreDepends`, `Suggests`, `Recommends`, `Conflicts`, `Replaces` or `Obsoletes`.

In a numeric context, evaluates to an AptPkg::Dep:: constant.

**ParentPkg****ParentVer**

AptPkg::Cache::Package and AptPkg::Cache::Version objects describing the package declaring the dependency.

**TargetPkg**

AptPkg::Cache::Package object describing the depended package.

**TargetVer**

For versioned dependencies, TargetVer is a string giving the version of the target package required.

**CompType****CompTypeDeb**

CompType gives a normalised comparison operator (`>`, `>=`, etc) describing the relationship to TargetVer ("" if none).

CompTypeDeb returns Debian-style operators (`>>` rather than `>`).

In a numeric context, both CompType and CompTypeDeb evaluate to an AptPkg::Dep:: constant.

Alternative dependencies (`Depends: a | b`) are identified by all but the last having the AptPkg::Dep::Or bit set in the numeric representation of CompType (and CompTypeDeb).

**Index**

Internal **APT** unique reference for the dependency record.

**AptPkg::Cache::Provides**

Implements the **APT** pkgCache::PrvIterator class as a hash reference.

*Keys*

**Name**

Name of virtual package.

**OwnerPkg****OwnerVer**

AptPkg::Cache::Package and AptPkg::Cache::Version objects describing the providing package.

**ProvideVersion**

Version of the virtual package. [Not currently supported by dpkg]

**Index**

Internal **APT** unique reference for the provides record.

**AptPkg::Cache::VerFile**

Implements the **APT** pkgCache::VerFileIterator class as a hash reference.

*Keys***File**

An AptPkg::Cache::PkgFile object describing the packages file.

**Offset****Size**

The byte offset and length of the entry in the file.

**Index**

Internal **APT** unique reference for the version file record.

**AptPkg::Cache::PkgFile**

Implements the **APT** pkgCache::PkgFileIterator class as a hash reference.

*Keys***FileName**

Packages file path.

**IndexType**

File type: Debian Package Index, Debian dpkg status file.

**Archive****Component****Version****Origin****Label****Site**

Fields from the Release file.

**IsOk**

True if the cache is in sync with this file.

**Index**

Internal **APT** unique reference for the package file record.

**AptPkg::Cache::DescFile**

Implements the **APT** pkgCache::DescFileIterator class as a hash reference.

*Keys***File**

An AptPkg::Cache::PkgFile object describing the packages file.

**SEE ALSO**

*AptPkg::Config* (3pm), *AptPkg::System* (3pm), *AptPkg* (3pm), *AptPkg::hash* (3pm),  
*AptPkg::PkgRecords* (3pm), *AptPkg::Policy* (3pm).

**AUTHOR**

Brendan O'Dea <bod@debian.org>