NAME

apt-ftparchive − Utility to generate index files

SYNOPSIS

apt−ftparchive [−**dsq**] [−−**md5**] [−−**delink**] [−−**readonly**] [−−**contents**] [−−**arch** *architecture*]
[−**o=***config_string*] [−**c=***config_file*] {packages *path...* [*override−file* [*pathprefix*]] |
sources *path...* [*override−file* [*pathprefix*]] | contents *path* | release *path* |
generate *config_file section...* | clean *config_file* | {−v | −−version} | {−h | −−help}}

DESCRIPTION

apt−ftparchive is the command line tool that generates the index files that APT uses to access a
distribution source. The index files should be generated on the origin site based on the content of that site.

apt−ftparchive is a superset of the **dpkg-scanpackages**(1) program, incorporating its entire functionality
via the packages command. It also contains a contents file generator, contents, and an elaborate means to
'script' the generation process for a complete archive.

Internally **apt−ftparchive** can make use of binary databases to cache the contents of a .deb file and it does
not rely on any external programs aside from **gzip**(1). When doing a full generate it automatically performs
file−change checks and builds the desired compressed output files.

Unless the −**h**, or −−**help** option is given, one of the commands below must be present.

packages

The packages command generates a package file from a directory tree. It takes the given directory and
recursively searches it for .deb files, emitting a package record to stdout for each. This command is
approximately equivalent to **dpkg-scanpackages**(1).

The option −−**db** can be used to specify a binary caching DB.

sources

The sources command generates a source index file from a directory tree. It takes the given directory
and recursively searches it for .dsc files, emitting a source record to stdout for each. This command is
approximately equivalent to **dpkg-scansources**(1).

If an override file is specified then a source override file will be looked for with an extension of .src.
The −−source−override option can be used to change the source override file that will be used.

contents

The contents command generates a contents file from a directory tree. It takes the given directory and
recursively searches it for .deb files, and reads the file list from each file. It then sorts and writes to
stdout the list of files matched to packages. Directories are not written to the output. If multiple
packages own the same file then each package is separated by a comma in the output.

The option −−**db** can be used to specify a binary caching DB.

release

The release command generates a Release file from a directory tree. It recursively searches the given
directory for uncompressed and compressed Packages, Sources, Contents, Components and icons files
as well as Release, Index and md5sum.txt files by default
(APT::FTPArchive::Release::Default−Patterns). Additional filename patterns can be added by listing
them in APT::FTPArchive::Release::Patterns. It then writes to stdout a Release file containing (by
default) an MD5, SHA1, SHA256 and SHA512 digest for each file.

Values for the additional metadata fields in the Release file are taken from the corresponding variables
under APT::FTPArchive::Release, e.g. APT::FTPArchive::Release::Origin. The supported fields are
Origin, Label, Suite, Version, Codename, Date, NotAutomatic, ButAutomaticUpgrades,
Acquire−By−Hash, Valid−Until, Signed−By, Architectures, Components and Description.

generate

The generate command is designed to be runnable from a cron script and builds indexes according to the given config file. The config language provides a flexible means of specifying which index files are built from which directories, as well as providing a simple means of maintaining the required settings.

clean

The clean command tidies the databases used by the given configuration file by removing any records that are no longer necessary.

THE GENERATE CONFIGURATION

The generate command uses a configuration file to describe the archives that are going to be generated. It follows the typical ISC configuration format as seen in ISC tools like bind 8 and dhcpd. **apt.conf**(5) contains a description of the syntax. Note that the generate configuration is parsed in sectional manner, but **apt.conf**(5) is parsed in a tree manner. This only effects how the scope tag is handled.

The generate configuration has four separate sections, each described below.

Dir Section

The Dir section defines the standard directories needed to locate the files required during the generation process. These directories are prepended certain relative paths defined in later sections to produce a complete an absolute path.

ArchiveDir

Specifies the root of the FTP archive, in a standard Debian configuration this is the directory that contains the ls−LR and dist nodes.

OverrideDir

Specifies the location of the override files.

CacheDir

Specifies the location of the cache files.

FileListDir

Specifies the location of the file list files, if the FileList setting is used below.

Default Section

The Default section specifies default values, and settings that control the operation of the generator. Other sections may override these defaults with a per−section setting.

Packages::Compress

Sets the default compression schemes to use for the package index files. It is a string that contains a space separated list of at least one of the compressors configured via the **APT::Compressor** configuration scope. The default for all compression schemes is '. gzip'.

Packages::Extensions

Sets the default list of file extensions that are package files. This defaults to '.deb'.

Sources::Compress

This is similar to Packages::Compress except that it controls the compression for the Sources files.

Sources::Extensions

Sets the default list of file extensions that are source files. This defaults to '.dsc'.

Contents::Compress

This is similar to Packages::Compress except that it controls the compression for the Contents files.

Translation::Compress

This is similar to Packages::Compress except that it controls the compression for the Translation−en master file.

DeLinkLimit

Specifies the number of kilobytes to delink (and replace with hard links) per run. This is used in conjunction with the per−section External−Links setting.

FileMode

Specifies the mode of all created index files. It defaults to 0644. All index files are set to this mode

with no regard to the umask.

LongDescription

Specifies whether long descriptions should be included in the Packages file or split out into a master Translation−en file.

TreeDefault Section

Sets defaults specific to Tree sections. All of these variables are substitution variables and have the strings $(DIST), $(SECTION) and $(ARCH) replaced with their respective values.

MaxContentsChange

Sets the number of kilobytes of contents files that are generated each day. The contents files are round−robined so that over several days they will all be rebuilt.

ContentsAge

Controls the number of days a contents file is allowed to be checked without changing. If this limit is passed the mtime of the contents file is updated. This case can occur if the package file is changed in such a way that does not result in a new contents file [override edit for instance]. A hold off is allowed in hopes that new .debs will be installed, requiring a new file anyhow. The default is 10, the units are in days.

Directory

Sets the top of the .deb directory tree. Defaults to $(DIST)/$(SECTION)/binary−$(ARCH)/

SrcDirectory

Sets the top of the source package directory tree. Defaults to $(DIST)/$(SECTION)/source/

Packages

Sets the output Packages file. Defaults to $(DIST)/$(SECTION)/binary−$(ARCH)/Packages

Sources

Sets the output Sources file. Defaults to $(DIST)/$(SECTION)/source/Sources

Translation

Sets the output Translation−en master file with the long descriptions if they should be not included in the Packages file. Defaults to $(DIST)/$(SECTION)/i18n/Translation−en

InternalPrefix

Sets the path prefix that causes a symlink to be considered an internal link instead of an external link. Defaults to $(DIST)/$(SECTION)/

Contents

Sets the output Contents file. Defaults to $(DIST)/$(SECTION)/Contents−$(ARCH). If this setting causes multiple Packages files to map onto a single Contents file (as is the default) then **apt−ftparchive** will integrate those package files together automatically.

Contents::Header

Sets header file to prepend to the contents output.

BinCacheDB

Sets the binary cache database to use for this section. Multiple sections can share the same database.

FileList

Specifies that instead of walking the directory tree, **apt−ftparchive** should read the list of files from the given file. Relative files names are prefixed with the archive directory.

SourceFileList

Specifies that instead of walking the directory tree, **apt−ftparchive** should read the list of files from the given file. Relative files names are prefixed with the archive directory. This is used when processing source indexes.

Tree Section

The Tree section defines a standard Debian file tree which consists of a base directory, then multiple sections in that base directory and finally multiple Architectures in each section. The exact pathing used is

defined by the Directory substitution variable.

The Tree section takes a scope tag which sets the $(DIST) variable and defines the root of the tree (the path is prefixed by ArchiveDir). Typically this is a setting such as dists/buster.

All of the settings defined in the TreeDefault section can be used in a Tree section as well as three new variables.

When processing a Tree section **apt−ftparchive** performs an operation similar to:

```
for i in Sections do
   for j in Architectures do
      Generate for DIST=scope SECTION=i ARCH=j
```

Sections
> This is a space separated list of sections which appear under the distribution; typically this is something like main contrib non−free

Architectures
> This is a space separated list of all the architectures that appear under search section. The special architecture 'source' is used to indicate that this tree has a source archive. The architecture 'all' signals that architecture specific files like Packages should not include information about architecture all packages in all files as they will be available in a dedicated file.

LongDescription
> Specifies whether long descriptions should be included in the Packages file or split out into a master Translation−en file.

BinOverride
> Sets the binary override file. The override file contains section, priority and maintainer address information.

SrcOverride
> Sets the source override file. The override file contains section information.

ExtraOverride
> Sets the binary extra override file.

SrcExtraOverride
> Sets the source extra override file.

BinDirectory Section
> The bindirectory section defines a binary directory tree with no special structure. The scope tag specifies the location of the binary directory and the settings are similar to the Tree section with no substitution variables or SectionArchitecture settings.

Packages
> Sets the Packages file output.

Sources
> Sets the Sources file output. At least one of Packages or Sources is required.

Contents
> Sets the Contents file output (optional).

BinOverride
> Sets the binary override file.

SrcOverride
> Sets the source override file.

ExtraOverride
> Sets the binary extra override file.

SrcExtraOverride
>Sets the source extra override file.

BinCacheDB
>Sets the cache DB.

PathPrefix
>Appends a path to all the output paths.

FileList, **SourceFileList**
>Specifies the file list file.

THE BINARY OVERRIDE FILE

The binary override file is fully compatible with **dpkg-scanpackages**(1). It contains four fields separated by spaces. The first field is the package name, the second is the priority to force that package to, the third is the section to force that package to and the final field is the maintainer permutation field.

The general form of the maintainer field is:

old [// oldn]* => new

or simply,

new

The first form allows a double−slash separated list of old email addresses to be specified. If any of those are found then new is substituted for the maintainer field. The second form unconditionally substitutes the maintainer field.

THE SOURCE OVERRIDE FILE

The source override file is fully compatible with **dpkg-scansources**(1). It contains two fields separated by spaces. The first field is the source package name, the second is the section to assign it.

THE EXTRA OVERRIDE FILE

The extra override file allows any arbitrary tag to be added or replaced in the output. It has three columns, the first is the package, the second is the tag and the remainder of the line is the new value.

OPTIONS

All command line options may be set using the configuration file, the descriptions indicate the configuration option to set. For boolean options you can override the config file by using something like **−f−**,**−−no−f**, **−f=no** or several other variations.

−−md5, **−−sha1**, **−−sha256**, **−−sha512**
>Generate the given checksum. These options default to on, when turned off the generated index files will not have the checksum fields where possible. Configuration Items: APT::FTPArchive::*Checksum* and APT::FTPArchive::*Index*::*Checksum* where *Index* can be Packages, Sources or Release and *Checksum* can be MD5, SHA1, SHA256 or SHA512.

−d, **−−db**
>Use a binary caching DB. This has no effect on the generate command. Configuration Item: APT::FTPArchive::DB.

−q, **−−quiet**
>Quiet; produces output suitable for logging, omitting progress indicators. More q's will produce more quiet up to a maximum of 2. You can also use **−q=#** to set the quiet level, overriding the configuration file. Configuration Item: quiet.

−−delink
>Perform Delinking. If the External−Links setting is used then this option actually enables delinking of the files. It defaults to on and can be turned off with **−−no−delink**. Configuration Item: APT::FTPArchive::DeLinkAct.

−−contents

Perform contents generation. When this option is set and package indexes are being generated with a cache DB then the file listing will also be extracted and stored in the DB for later use. When using the generate command this option also allows the creation of any Contents files. The default is on. Configuration Item: APT::FTPArchive::Contents.

−s, **−−source−override**

Select the source override file to use with the sources command. Configuration Item: APT::FTPArchive::SourceOverride.

−−readonly

Make the caching databases read only. Configuration Item: APT::FTPArchive::ReadOnlyDB.

−a, **−−arch**

Accept in the packages and contents commands only package files matching *_arch.deb or *_all.deb instead of all package files in the given path. Configuration Item: APT::FTPArchive::Architecture.

APT::FTPArchive::AlwaysStat

apt-ftparchive(1) caches as much as possible of metadata in a cachedb. If packages are recompiled and/or republished with the same version again, this will lead to problems as the now outdated cached metadata like size and checksums will be used. With this option enabled this will no longer happen as it will be checked if the file was changed. Note that this option is set to "false" by default as it is not recommend to upload multiple versions/builds of a package with the same version number, so in theory nobody will have these problems and therefore all these extra checks are useless.

APT::FTPArchive::LongDescription

This configuration option defaults to "true" and should only be set to "false" if the Archive generated with **apt-ftparchive**(1) also provides Translation files. Note that the Translation−en master file can only be created in the generate command.

−h, **−−help**

Show a short usage summary.

−v, **−−version**

Show the program version.

−c, **−−config−file**

Configuration File; Specify a configuration file to use. The program will read the default configuration file and then this configuration file. If configuration settings need to be set before the default configuration files are parsed specify a file with the **APT_CONFIG** environment variable. See **apt.conf**(5) for syntax information.

−o, **−−option**

Set a Configuration Option; This will set an arbitrary configuration option. The syntax is **−o Foo::Bar=bar**. **−o** and **−−option** can be used multiple times to set different options.

EXAMPLES

To create a compressed Packages file for a directory containing binary packages (.deb):

apt−ftparchive packages *directory* | **gzip** > Packages.gz

SEE ALSO

apt.conf(5)

DIAGNOSTICS

apt−ftparchive returns zero on normal operation, decimal 100 on error.

BUGS

[APT bug page][1]. If you wish to report a bug in APT, please see /usr/share/doc/debian/bug−reporting.txt or the **reportbug**(1) command.

AUTHORS

      Jason Gunthorpe

      APT team

NOTES

    1.   APT bug page
       http://bugs.debian.org/src:apt