NAME

apt-transport-mirror − APT transport for more automated mirror selection

DESCRIPTION

This APT transport isn't implementing a protocol to access local or remote repositories on its own, but acquires a mirrorlist and redirects all requests to the mirror(s) picked from this list, accessing them via other transports like **apt-transport-http**(1). The basic functionality has been available since apt 0.7.24, but was undocumented until apt 1.6 which contained a complete rework of the transport and its supported features. Note that a transport is never called directly by a user but used by APT tools based on user configuration.

If the acquisition of a file via a mirror fails, the method ensures that another possible mirror from the list is automatically tried until either the file is retrieved or no mirror is left in the list, transparently handling server downtimes and similar problems.

The security implications of the transport depend on the security considerations associated with the transport used to acquire the mirrorlist and the transports involved in accessing the chosen mirror(s) by the transport.

OPTIONS

This transport has no configuration options at present. The mirror selection is based entirely on the mirrors offered in the mirrorlist and the files APT needs to acquire.

Mirrorlist format

A mirrorlist contains one or more lines each specifying a URI for a mirror. Empty lines and those starting with a hash character (#) are ignored. A URI always starts with a URI scheme which defines the transport used for this mirror. If for example the URI starts with http:, the responsible transport is **apt-transport-http**(1) which might have specific requirements for the format of the remaining part of the URI.

Metadata about a mirror can be given on the same line, separated from the URI by a tab. Multiple items of metadata can themselves be separated by either tabs or spaces. (This is an advanced feature only available with apt >= 1.6. Earlier apt versions will fail to parse mirrorlists using this feature.)

Since apt 1.6 the use of compressed mirrorlists is also supported. Note that the filename of the mirrorlist must specify the compression algorithm used; there is no auto−detection based on file contents.

Mirror selection by metadata

As specified in the format, a mirror can have additional metadata attached to prevent a mirror from being selected for acquiring a file not matching this metadata. This way the mirrorlist can e.g. contain partial mirrors serving only certain architectures and APT will automatically choose a different mirror for files requiring an unlisted architecture. Supported are limits for the architecture (arch), codename of the release (codename), component of the repository the file is in (component), language the file applies to (lang), suite name of the release (suite) and type of the file (type).

Fallback order for mirrors

If no priority is given for a mirror via the metadata key priority, the order in which mirrors are contacted is random. If a certain set of mirrors should be tried first before any of another set is tried, a priority can be explicitly set. The mirrors with the lowest number are tried first. Mirrors which have no explicit priority set default to the highest possible number and are therefore tried last. The choice between mirrors with the same priority is again random.

Allowed transports in a mirrorlist

The availability and choice of transports in a mirrorlist is limited by how the APT client is accessing the mirrorlist. If a local transport like file or copy is used, the mirrorlist can also include local sources, while a mirrorlist accessed via http can not. Additionally, a mirrorlist can not contain a mirrorlist or other wrapping transports (like apt−transport−tor). See the documentation of these transports on how to use them with the mirror method.

Note that apt versions before 1.6 do not support any other transport than http.

EXAMPLES

Basic example

A basic mirrorlist example supported by all apt versions with a mirror method (>= 0.7.24) in which the client will pick any of the three mirrors:

http://ftp.de.debian.org/debian/
http://ftp.us.debian.org/debian/
http://deb.debian.org/debian/

Assuming a file with this content is stored as /etc/apt/mirrorlist.txt on your machine it can be used like this in **sources.list**(5) (since apt 1.6):

deb mirror+file:/etc/apt/mirrorlist.txt buster main

All versions of the mirror method support a mirrorlist accessible via HTTP, so assuming it is available at http://apt.example.org/mirror.lst the sources.list entry from above could instead be written as:

deb mirror://apt.example.org/mirror.lst buster main

Note that since apt 1.6 the use of mirror+http should be preferred over mirror for uniformity. The functionality is the same.

Example with metadata−enhanced mirror selection

As explained in the format definition apt versions before 1.6 do not support this and will fail parsing the mirrorlist. The example mirrorlist is intentionally complicated to show some aspects of the selection. The following setup is assumed: The first mirror is a local mirror accessible via the file method, but potentially incomplete. The second mirror has a great connection, but is a partial mirror insofar as it only contains files related to the architectures amd64 and all. The remaining mirrors are average mirrors which should be contacted only if the earlier ones didn't work.

file:/srv/local/debian/mirror/	priority:1 type:index
http://partial.example.org/mirror/	priority:2 arch:amd64 arch:all type:deb
http://ftp.us.debian.org/debian/	type:deb
http://ftp.de.debian.org/debian/	type:deb
https://deb.debian.org/debian/	

In this setup with this mirrorlist the first mirror will be used to download all index files assuming the mirrorlist itself is accessed via a local transport like file. If it isn't, if the mirror is otherwise inaccessible or if it does not contain the requested file another mirror will be used to acquire the file, chosen depending on the type of the file: An index file will be served by the last mirror in the list, while a package of architecture amd64 is served by the second and those of e.g. architecture i386 by one of the last three.

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.

AUTHOR

APT team

NOTES

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