## NAME

apt-transport-http − APT transport for downloading via the Hypertext Transfer Protocol (HTTP)

## DESCRIPTION

This APT transport allows the use of repositories accessed via the Hypertext Transfer Protocol (HTTP). It is available by default and probably the most used of all transports. Note that a transport is never called directly by a user but used by APT tools based on user configuration.

HTTP is an unencrypted transport protocol meaning that the whole communication with the remote server (or proxy) can be observed by a sufficiently capable attacker commonly referred to as a "man in the middle" (MITM). However, such an attacker can *not* modify the communication to compromise the security of your system, as APT's data security model is independent of the chosen transport method. This is explained in detail in **apt-secure**(8). An overview of available transport methods is given in **sources.list**(5).

## OPTIONS

Various options can be set in an **apt.conf**(5) file to modify its behavior, ranging from proxy configuration to workarounds for specific server limitations.

### Proxy Configuration

The environment variable **http_proxy** is supported for system wide configuration. Proxies specific to APT can be configured via the option Acquire::http::Proxy. Proxies which should be used only for certain hosts can be specified via Acquire::http::Proxy::*host*. Even more fine−grained control can be achieved via proxy autodetection, detailed further below. All these options use the URI format *scheme*://[[*user*][:*pass*]@]*host*[:*port*]/. Supported URI schemes are socks5h (SOCKS5 with remote DNS resolution), http and https. Authentication details can be supplied via **apt_auth.conf**(5) instead of including it in the URI directly.

The various APT configuration options support the special value DIRECT meaning that no proxy should be used. The environment variable **no_proxy** is also supported for the same purpose.

Furthermore, there are three settings provided for cache control with HTTP/1.1 compliant proxy caches: Acquire::http::No−Cache tells the proxy not to use its cached response under any circumstances. Acquire::http::Max−Age sets the allowed maximum age (in seconds) of an index file in the cache of the proxy. Acquire::http::No−Store specifies that the proxy should not store the requested archive files in its cache, which can be used to prevent the proxy from polluting its cache with (big) .deb files.

### Automatic Proxy Configuration

Acquire::http::Proxy−Auto−Detect can be used to specify an external command to discover the HTTP proxy to use. The first and only parameter is a URI denoting the host to be contacted, to allow for host−specific configuration. APT expects the command to output the proxy on stdout as a single line in the previously specified URI format or the word DIRECT if no proxy should be used. No output indicates that the generic proxy settings should be used.

Note that auto−detection will not be used for a host if a host−specific proxy configuration is already set via Acquire::http::Proxy::*host*.

See the **squid-deb-proxy-client**(1) and **auto-apt-proxy**(1) packages for example implementations.

This option takes precedence over the legacy option name Acquire::http::ProxyAutoDetect.

### Connection Configuration

The option Acquire::http::Timeout sets the timeout timer used by the method; this value applies to the connection as well as the data timeout.

The used bandwidth can be limited with Acquire::http::Dl−Limit which accepts integer values in kilobytes per second. The default value is 0 which deactivates the limit and tries to use all available bandwidth. Note that this option implicitly disables downloading from multiple servers at the same time.

The setting Acquire::http::Pipeline−Depth can be used to enable HTTP pipelining (RFC 2616 section 8.1.2.2) which can be beneficial e.g. on high−latency connections. It specifies how many requests are sent in a pipeline. APT tries to detect and work around misbehaving webservers and proxies at runtime, but if you know that yours does not conform to the HTTP/1.1 specification, pipelining can be disabled by setting

the value to 0. It is enabled by default with the value 10.

Acquire::http::AllowRedirect controls whether APT will follow redirects, which is enabled by default.

Acquire::http::User−Agent can be used to set a different User−Agent for the http download method as some proxies allow access for clients only if the client uses a known identifier.

Acquire::http::SendAccept is enabled by default and sends an Accept: text/* header field to the server for requests without file extensions to prevent the server from attempting content negotiation.

## EXAMPLES

```
Acquire::http {
        Proxy::example.org "DIRECT";
        Proxy "socks5h://apt:pass@127.0.0.1:9050";
        Proxy−Auto−Detect "/usr/local/bin/apt−http−proxy−auto−detect";
        No−Cache "true";
        Max−Age "3600";
        No−Store "true";
        Timeout "10";
        Dl−Limit "42";
        Pipeline−Depth "0";
        AllowRedirect "false";
        User−Agent "My APT−HTTP";
        SendAccept "false";
};
```

## SEE ALSO

**apt.conf**(5) **apt_auth.conf**(5) **sources.list**(5)

## 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