List of HTTP header fields

HTTP header fields are components of the message header of requests and responses in the Hypertext Transfer Protocol (HTTP). They define the operating parameters of an HTTP transaction.

The header fields are transmitted after the request or response line, the first line of a message. Header fields are colon-separated name-value pairs in clear-text string format, terminated by a carriage return (CR) and line feed (LF) character sequence. The end of the header fields is indicated by an empty field, resulting in the transmission of two consecutive CR-LF pairs. Long lines can be folded into multiple lines; continuation lines are indicated by presence of space (SP) or horizontal tab (HT) as first character on next line.[1] Few fields can also contain comments (i.e. in. User-Agent, Server, Via fields), which can be ignored by software.[2]

A core set of fields is standardized by the Internet Engineering Task Force (IETF) in RFC 2616 and other updates and extension documents (e.g., RFC 4229), and must be implemented by all HTTP-compliant protocol implementations. Additional field names and permissible values may be defined by each application.

The permanent registry of headers [3] and repository of provisional registrations [4] are maintained by the IANA.

Many field values may contain a quality (*q*) key-value pair, specifying a weight to use in content negotiation.[5]

There are no limits to the size of each header field name or value, or the number of headers, in the standard itself. However, most servers, clients and proxy software impose some limits for practical and security reasons. For example, the Apache 2.3 server by default limits each header size to 8190 bytes, and there can be at most 100 headers in single request.[6]

Requests

Field name	Description	Example
Accept	Content-Types that are acceptable	`Accept: text/plain`
Accept-Charset	Character sets that are acceptable	`Accept-Charset: utf-8`
Accept-Encoding	Acceptable encodings. See HTTP compression.	`Accept-Encoding: gzip, deflate`
Accept-Language	Acceptable human languages for response	`Accept-Language: en-US`
Accept-Datetime	Acceptable version in time	`Accept-Datetime: Thu, 31 May 2007 20:35:00 GMT`
Authorization	Authentication credentials for HTTP authentication	`Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==`
Cache-Control	Used to specify directives that MUST be obeyed by all caching mechanisms along the request/response chain	`Cache-Control: no-cache`
Connection	What type of connection the user-agent would prefer	`Connection: keep-alive`
Cookie	an HTTP cookie previously sent by the server with `Set-Cookie` (below)	`Cookie: $Version=1; Skin=new;`
Content-Length	The length of the request body in octets (8-bit bytes)	`Content-Length: 348`
Content-MD5	A Base64-encoded binary MD5 sum of the content of the request body	`Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ==`
Content-Type	The MIME type of the body of the request (used with POST and PUT requests)	`Content-Type: application/x-www-form-urlencoded`
Date	The date and time that the message was sent	`Date: Tue, 15 Nov 1994 08:12:31 GMT`
Expect	Indicates that particular server behaviors are required by the client	`Expect: 100-continue`

From	The email address of the user making the request	`From: user@example.com`
Host	The domain name of the server (for virtual hosting), and the TCP port number on which the server is listening. The port number may be omitted if the port is the standard port for the service requested.[7] Mandatory since HTTP/1.1. Although domain name are specified as case-insensitive,[8][9] it is not specified whether the contents of the Host field should be interpreted in a case-insensitive manner[10] and in practice some implementations of virtual hosting interpret the contents of the Host field in a case-sensitive manner.	`Host: en.wikipedia.org:80` `Host: en.wikipedia.org`
If-Match	Only perform the action if the client supplied entity matches the same entity on the server. This is mainly for methods like PUT to only update a resource if it has not been modified since the user last updated it.	`If-Match: "737060cd8c284d8af7ad3082f209582d"`
If-Modified-Since	Allows a *304 Not Modified* to be returned if content is unchanged	`If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT`
If-None-Match	Allows a *304 Not Modified* to be returned if content is unchanged, see HTTP ETag	`If-None-Match: "737060cd8c284d8af7ad3082f209582d"`
If-Range	If the entity is unchanged, send me the part(s) that I am missing; otherwise, send me the entire new entity	`If-Range: "737060cd8c284d8af7ad3082f209582d"`
If-Unmodified-Since	Only send the response if the entity has not been modified since a specific time.	`If-Unmodified-Since: Sat, 29 Oct 1994 19:43:31 GMT`
Max-Forwards	Limit the number of times the message can be forwarded through proxies or gateways.	`Max-Forwards: 10`
Pragma	Implementation-specific headers that may have various effects anywhere along the request-response chain.	`Pragma: no-cache`
Proxy-Authorization	Authorization credentials for connecting to a proxy.	`Proxy-Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==`
Range	Request only part of an entity. Bytes are numbered from 0.	`Range: bytes=500-999`
Referer[sic]	This is the address of the previous web page from which a link to the currently requested page was followed. (The word “referrer” is misspelled in the RFC as well as in most implementations.)	`Referer: http://en.wikipedia.org/wiki/Main_Page`
TE	The transfer encodings the user agent is willing to accept: the same values as for the response header Transfer-Encoding can be used, plus the "trailers" value (related to the "chunked" transfer method) to notify the server it expects to receive additional headers (the trailers) after the last, zero-sized, chunk.	`TE: trailers, deflate`
Upgrade	Ask the server to upgrade to another protocol.	`Upgrade: HTTP/2.0, SHTTP/1.3, IRC/6.9, RTA/x11`
User-Agent	The user agent string of the user agent	`User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:12.0) Gecko/20100101 Firefox/12.0`
Via	Informs the server of proxies through which the request was sent.	`Via: 1.0 fred, 1.1 example.com (Apache/1.1)`
Warning	A general warning about possible problems with the entity body.	`Warning: 199 Miscellaneous warning`

Common non-standard request headers

Non-standard header fields were conventionally marked by prefixing the field name with `X-` .[11] However, this convention became deprecated in June 2012 due to the inconveniences it caused when non-standard headers then became standard.[12] For example, `X-Gzip` and `Gzip` are now both supported headers for compressed HTTP requests and responses.

Field name	Description	Example
X-Requested-With	mainly used to identify Ajax requests. Most JavaScript frameworks send this header with value of `XMLHttpRequest`	`X-Requested-With: XMLHttpRequest`
DNT[13]	Requests a web application to disable their tracking of a user. This is Mozilla's version of the X-Do-Not-Track header (since Firefox 4.0 Beta 11). Safari and IE9 also have support for this header.[14] On March 7, 2011, a draft proposal was submitted to IETF.[15] The W3C Tracking Protection Working Group is producing a specification.[16]	`DNT: 1 (Do Not Track Enabled)` `DNT: 0 (Do Not Track Disabled)`
X-Forwarded-For[17]	a *de facto* standard for identifying the originating IP address of a client connecting to a web server through an HTTP proxy or load balancer	`X-Forwarded-For: client1, proxy1, proxy2` `X-Forwarded-For: 129.78.138.66, 129.78.64.103`
X-Forwarded-Proto[18]	a *de facto* standard for identifying the originating protocol of an HTTP request, since a reverse proxy (load balancer) may communicate with a web server using HTTP even if the request to the reverse proxy is HTTPS	`X-Forwarded-Proto: https`
Front-End-Https[19]	Non-standard header used by Microsoft applications and load-balancers	`Front-End-Https: on`
X-ATT-DeviceId[20]	Allows easier parsing of the MakeModel/Firmware that is usually found in the User-Agent String of AT&T Devices	`x-att-deviceid: MakeModel/Firmware`
X-Wap-Profile[21]	Links to an XML file on the Internet with a full description and details about the device currently connecting. In the example to the right is an XML file for an AT&T Samsung Galaxy S2.	`x-wap-profile:` [22]
Proxy-Connection[23]	Implemented as a misunderstanding of the HTTP specifications. Common because of mistakes in implementations of early HTTP versions. Has exactly the same functionality as standard Connection header.	`Proxy-Connection: keep-alive`

Responses

Field name	Description	Example
Access-Control-Allow-Origin	Specifying which web sites can participate in cross-origin resource sharing	`Access-Control-Allow-Origin: *`
Accept-Ranges	What partial content range types this server supports	`Accept-Ranges: bytes`
Age	The age the object has been in a proxy cache in seconds	`Age: 12`
Allow	Valid actions for a specified resource. To be used for a *405 Method not allowed*	`Allow: GET, HEAD`
Cache-Control	Tells all caching mechanisms from server to client whether they may cache this object. It is measured in seconds	`Cache-Control: max-age=3600`
Connection	Options that are desired for the connection[24]	`Connection: close`

Content-Encoding	The type of encoding used on the data. See HTTP compression.	`Content-Encoding: gzip`
Content-Language	The language the content is in	`Content-Language: da`
Content-Length	The length of the response body in octets (8-bit bytes)	`Content-Length: 348`
Content-Location	An alternate location for the returned data	`Content-Location: /index.htm`
Content-MD5	A Base64-encoded binary MD5 sum of the content of the response	`Content-MD5: Q2hlY2sgSW50ZWdyaXR5IQ==`
Content-Disposition[25][26][27]	An opportunity to raise a "File Download" dialogue box for a known MIME type with binary format or suggest a filename for dynamic content. Quotes are necessary with special characters.	`Content-Disposition: attachment; filename="fname.ext"`
Content-Range	Where in a full body message this partial message belongs	`Content-Range: bytes 21010-47021/47022`
Content-Type	The MIME type of this content	`Content-Type: text/html; charset=utf-8`
Date	The date and time that the message was sent	`Date: Tue, 15 Nov 1994 08:12:31 GMT`
ETag	An identifier for a specific version of a resource, often a message digest	`ETag: "737060cd8c284d8af7ad3082f209582d"`
Expires	Gives the date/time after which the response is considered stale	`Expires: Thu, 01 Dec 1994 16:00:00 GMT`
Last-Modified	The last modified date for the requested object, in RFC 2822 format	`Last-Modified: Tue, 15 Nov 1994 12:45:26 GMT`
Link	Used to express a typed relationship with another resource, where the relation type is defined by RFC 5988	`Link: </feed>; rel="alternate"`[28]
Location	Used in redirection, or when a new resource has been created.	`Location: http://www.w3.org/pub/WWW/People.html`
P3P	This header is supposed to set P3P policy, in the form of `P3P:CP="your_compact_policy"`. However, P3P did not take off,[29] most browsers have never fully implemented it, a lot of websites set this header with fake policy text, that was enough to fool browsers the existence of P3P policy and grant permissions for third party cookies.	`P3P: CP="This is not a P3P policy! See http://www.google.com/support/accounts/bin/answer.py?hl=en&answer=151657 for more info."`
Pragma	Implementation-specific headers that may have various effects anywhere along the request-response chain.	`Pragma: no-cache`
Proxy-Authenticate	Request authentication to access the proxy.	`Proxy-Authenticate: Basic`

Refresh	Used in redirection, or when a new resource has been created. This refresh redirects after 5 seconds. This is a proprietary, non-standard header extension introduced by Netscape and supported by most web browsers.	`Refresh: 5; url=http://www.w3.org/pub/WWW/People.html`
Retry-After	If an entity is temporarily unavailable, this instructs the client to try again after a specified period of time (seconds).	`Retry-After: 120`
Server	A name for the server	`Server: Apache/2.4.1 (Unix)`
Set-Cookie	An HTTP cookie	`Set-Cookie: UserID=JohnDoe; Max-Age=3600; Version=1`
Status	The HTTP status of the response	`Status: 200 OK`
Strict-Transport-Security	A HSTS Policy informing the HTTP client how long to cache the HTTPS only policy and whether this applies to subdomains.	`Strict-Transport-Security: max-age=16070400; includeSubDomains`
Trailer	The Trailer general field value indicates that the given set of header fields is present in the trailer of a message encoded with chunked transfer-coding.	`Trailer: Max-Forwards`
Transfer-Encoding	The form of encoding used to safely transfer the entity to the user. Currently defined methods [30] are: chunked, compress, deflate, gzip, identity.	`Transfer-Encoding: chunked`
Vary	Tells downstream proxies how to match future request headers to decide whether the cached response can be used rather than requesting a fresh one from the origin server.	`Vary: *`
Via	Informs the client of proxies through which the response was sent.	`Via: 1.0 fred, 1.1 example.com (Apache/1.1)`
Warning	A general warning about possible problems with the entity body.	`Warning: 199 Miscellaneous warning`
WWW-Authenticate	Indicates the authentication scheme that should be used to access the requested entity.	`WWW-Authenticate: Basic`

Common non-standard response headers

Non-standard header fields are conventionally marked by prefixing the field name with `X-` .

Field name	Description	Example
X-Frame-Options[31]	Clickjacking protection: "deny" - no rendering within a frame, "sameorigin" - no rendering if origin mismatch	`X-Frame-Options: deny`
X-XSS-Protection[32]	Cross-site scripting (XSS) filter	`X-XSS-Protection: 1; mode=block`
X-Content-Security-Policy, X-WebKit-CSP[33]	Content Security Policy definition.	`X-WebKit-CSP: default-src 'self'`
X-Content-Type-Options[34]	The only defined value, "nosniff", prevents Internet Explorer from MIME-sniffing a response away from the declared content-type. This also applies to Google Chrome, when downloading extensions.[35]	`X-Content-Type-Options: nosniff`
X-Powered-By[36]	specifies the technology (e.g. ASP.NET, PHP, JBoss) supporting the web application (version details are often in `X-Runtime`, `X-Version`, or `X-AspNet-Version`)	`X-Powered-By: PHP/5.4.0`

X-UA-Compatible[37]	Recommends the preferred rendering engine (often a backward-compatibility mode) to use to display the content. Also used to activate Chrome Frame in Internet Explorer.	`X-UA-Compatible: IE=EmulateIE7`
		`X-UA-Compatible: IE=edge`
		`X-UA-Compatible: Chrome=1`

Effects of selected HTTP header fields

Avoiding caching

If a web server responds with `Cache-Control: no-cache` then a web browser or other caching system(intermediate proxies) must not use the response to satisfy subsequent responses without first checking with the originating server (this process is call validation). This header field is part of HTTP version 1.1, and is ignored by some caches and browsers. It may be simulated by setting the `Expires` HTTP version 1.0 header field value to a time earlier than the response time. Notice that no-cache is not instructing the browser or proxies about whether or not to cache the content. it just tells the browser and proxies to validate the cache content with the server before using it (this is be done by using if-Modified-Since,If-Unmodified-Since, If-Match ,If-None-Match attributes mentioned above) . What no-cache actually means is browser or proxies are not going to use the cache contents just based on "freshness criteria" of the cache content. Another common way how most of the website avoid having shown old content to the user is to set the `Cache-Control: max-age=0`. This again instructs the user agent that the content is stale and to validate it before using.

The header field `Cache-Control: no-store` is intended to instruct a browser application to make a best effort not to write it to disk (i.e not to cache it).

The request that a resource should not be cached is no guarantee that it will not be written to disk. In particular, the HTTP/1.1 definition draws a distinction between history stores and caches. If the user navigates back to a previous page a browser may still show you a page that has been stored on disk in the history store. This is correct behavior according to the specification. Many user agents show different behavior in loading pages from the history store or cache depending on whether the protocol is HTTP or HTTPS.

The `Pragma: no-cache` and `Cache-Control: no-cache` header field is an HTTP/1.0 header intended for use in requests header also. It is a means for the browser to tell the server and any intermediate caches that it wants a fresh version of the resource, not for the server to tell the browser not to cache the resource. `Pragma: no-cache` is implementation specific, and some user agents do pay attention to this header in responses, but the HTTP/1.1 RFC specifically warns against relying on this behavior.

References

[1] "HTTP/1.1: Notational Conventions and Generic Grammar" (http://www.w3.org/Protocols/rfc2616/rfc2616-sec2.html#sec2.2). W3.org. . Retrieved 2012-03-13.

[2] "HTTP/1.1: Header Field Definitions" (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.43). W3.org. . Retrieved 2012-03-13.

[3] http://www.iana.org/assignments/message-headers/perm-headers.html

[4] http://www.iana.org/assignments/message-headers/prov-headers.html

[5] "RFC 2616 §3.9" (http://www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.9). W3.org. . Retrieved 2012-03-13.

[6] "core - Apache HTTP Server" (http://httpd.apache.org/docs/2.3/mod/core.html#limitrequestfieldsize). Httpd.apache.org. . Retrieved 2012-03-13.

[7] "Hypertext Transfer Protocol -- HTTP/1.1" (https://tools.ietf.org/html/rfc2616#section-14.23). IETF. June 1999. . Retrieved 2012-08-16.

[8] "DOMAIN NAMES - CONCEPTS AND FACILITIES" (https://tools.ietf.org/html/rfc1034). IETF. November 1987. . Retrieved 2012-05-06.

[9] "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION" (https://tools.ietf.org/html/rfc1035). IETF. November 1987. . Retrieved 2012-05-06.

[10] "Hypertext Transfer Protocol -- HTTP/1.1" (https://tools.ietf.org/html/rfc2616). IETF. June 1999. . Retrieved 2012-05-06.

[11] Simtec Limited. "2. HTTP Headers" (http://www.httpwatch.com/httpgallery/headers/). . Retrieved 2010-09-10.

[12] Internet Engineering Task Force (2012-06-01). "RFC 6648" (http://tools.ietf.org/html/rfc6648). . Retrieved 2012-11-12.

[13] "Try out the "Do Not Track" HTTP header" (http://blog.sidstamm.com/2011/01/try-out-do-not-track-http-header.html). . Retrieved 2011-01-31.

[14] "Web Tracking Protection: Minimum Standards and Opportunities to Innovate" (http://blogs.msdn.com/b/ie/archive/2011/03/14/web-tracking-protection-minimum-standards-and-opportunities-to-innovate.aspx). . Retrieved 2011-03-24.

[15] IETF Do Not Track: A Universal Third-Party Web Tracking Opt Out (http://tools.ietf.org/html/draft-mayer-do-not-track-00) March 7, 2011

[16] W3C Tracking Preference Expression (DNT) (http://www.w3.org/2011/tracking-protection/drafts/tracking-dnt.html), January 26, 2012

[17] Amos Jeffries (2010-07-02). "SquidFaq/ConfiguringSquid - Squid Web Proxy Wiki" (http://wiki.squid-cache.org/SquidFaq/ConfiguringSquid#head-3518b69c63e221cc3cd7885415e365ffaf3dd27f). . Retrieved 2009-09-10.

[18] Dave Steinberg (2007-04-10). "How do I adjust my SSL site to work with GeekISP's loadbalancer?" (http://www.geekisp.com/faq/6_65_en.html). . Retrieved 2010-09-30.

[19] "Helping to Secure Communication: Client to Front-End Server" (http://technet.microsoft.com/en-us/library/aa997519(v=exchg.65).aspx). 2006-07-27. . Retrieved 2012-04-23.

[20] "ATT Device ID" (http://developer.att.com/developer/forward.jsp?passedItemId=5300270). . Retrieved 2012-01-14.

[21] "WAP Profile" (http://www.developershome.com/wap/detection/detection.asp?page=profileHeader). . Retrieved 2012-01-14.

[22] http://wap.samsungmobile.com/uaprof/SGH-I777.xml

[23] "HTTP/1.1 Proxy-Connection header field" (http://www.http-stats.com/header/Proxy-Connection). . Retrieved 2012-07-13.

[24] "RFC 2616 §14.10" (http://tools.ietf.org/html/rfc2616#section-14.10). Tools.ietf.org. . Retrieved 2012-03-13.

[25] "RFC 2183" (http://www.ietf.org/rfc/rfc2183.txt). . Retrieved 2012-05-17.

[26] "RFC 2616 §19.5.1" (http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1). . Retrieved 2012-05-17.

[27] "RFC 6266 UFT8" (http://tools.ietf.org/html/rfc6266#page-5). . Retrieved 2012-05-17.

[28] Indicate the canonical version of a URL by responding with the Link rel="canonical" HTTP header (http://support.google.com/webmasters/bin/answer.py?hl=en&answer=139394) Retrieved: 2012-02-09

[29] W3C P3P Work Suspended (http://www.w3.org/P3P)

[30] http://www.iana.org/assignments/http-parameters

[31] Eric Lawrence (2009-01-27). "IE8 Security Part VII: ClickJacking Defenses" (http://blogs.msdn.com/ie/archive/2009/01/27/ie8-security-part-vii-clickjacking-defenses.aspx). . Retrieved 2009-06-10.

[32] Eric Lawrence (2008-07-02). "IE8 Security Part IV: The XSS Filter" (http://blogs.msdn.com/b/ie/archive/2008/07/02/ie8-security-part-iv-the-xss-filter.aspx). . Retrieved 2010-09-30.

[33] "Content Security Policy" (http://www.w3.org/TR/CSP/). 2012. . Retrieved 2013.

[34] Eric Lawrence (2008-09-03). "IE8 Security Part VI: Beta 2 Update" (http://blogs.msdn.com/b/ie/archive/2008/09/02/ie8-security-part-vi-beta-2-update.aspx). . Retrieved 2010-09-28.

[35] "Hosting - Google Chrome Extensions - Google Code" (http://code.google.com/chrome/extensions/hosting.html). . Retrieved 2012-06-14.

[36] "Why does ASP.NET framework add the 'X-Powered-By:ASP.NET' HTTP Header in responses? - Stack Overflow" (http://stackoverflow.com/questions/1288338/why-does-asp-net-framework-add-the-x-powered-byasp-net-http-header-in-response). . Retrieved 2010-09-30.

[37] "Defining Document Compatibility: Specifying Document Compatibility Modes" (http://msdn.microsoft.com/en-us/library/ie/cc288325(v=vs.85).aspx#SetMode). 2011-04-01. . Retrieved 2012-01-24.

External links

- RFC 4229: HTTP Header Field Registrations. December 2005 (contains a more complete list of HTTP headers)
- RFC 2616: IETF HTTP/1.1 RFC
- RFC 2965: IETF HTTP State Management Mechanism RFC
- HTTP/1.1: Header Field Definitions (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html)
- HTTP/1.1 headers from a web server point of view (http://www.and.org/texts/server-http)
- HTTP Request Header Viewer (http://www.MyHTTP.info/)
- HTTP Response Header Viewer (http://viewdns.info/httpheaders/) - Retrieves the HTTP response headers of any domain.
- HTTP Header Viewer with Google App Engine (http://url-info.appspot.com/)
- Internet Explorer and Custom HTTP Headers - EricLaw's IEInternals - Site Home - MSDN Blogs (http://blogs.msdn.com/b/ieinternals/archive/2009/06/30/internet-explorer-custom-http-headers.aspx)
- crwlr.net - HTTP Header index (http://crwlr.net/)
- HTTP Header with Privacyinfo (http://www.privacyinfo.org/http-headers) - Display your HTTP request and response headers

Article Sources and Contributors

List of HTTP header fields *Source*: http://en.wikipedia.org/w/index.php?oldid=540075659 *Contributors*: 2602:304:CF7B:1770:810D:21A4:8617:BC1C, AWiersch, Aaronmhamilton, Abdull, Adam McMaster, Ammubhave, Apoorv020, ArchiSchmedes, Avaloan, BlaF, Blevintron, Bluezy, Bombshop, Brianary, Cbuckley, Ccutrer, Checat, Cowo, Dan East, Didz93, Dpant, Edsu, EvanCarroll, FrankAndProust, GBuilder, GoingBatty, Gurch, Guy Macon, Gynvael, HenryHayes, Holiverh, Hughesey, Husky, Ianbicking, Impala2009, Ixfd64, JakobVoss, Jarble, Jesdisciple, Jpala, Juergeen, Justinc, Kbrose, Kenticus, Kgfleischmann, KiloByte, Koyama, Kravietz, Lotje, Mabdul, Mac, Mbloore, Mcenedella, Mindmatrix, Mmxx, Mnot, Mpmel, Nevyn, Notanumber, Octahedron80, Oore, OrangeDog, Oxymoron83, Philip Trueman, Pleasancoder, Pmc, Pmsyyz, Poke64, Povlhp, PoweredByLolcats, Psz, RadioFan, Rawlogic, Rich Farmbrough, SD5, Scientus, SeL, Sliwers, Smithkennedy, SmoothPorcupine, Spandrawn, Sstrader, SunnySideOfStreet, Sylvainlasnier, Tellatime-corp, The Thing That Should Not Be, TheJosh, Timmywimmy, Timotheus1, Timwi, Towel401, Tweisbach, Vadmium, Vladkornea, Voltaic, Walling, Widefox, Will Thompson, Wseltzer, Xmm0, 148 anonymous edits

License