Squid 3.0 Configuration Manual

Support and Queries to E-mail squid_support@visolve.com

Home Page: http://www.visolve.com

Disclaimer: This manual is NOT a Squid tutorial. It does not, for example, takes the reader through step-by-step details of Squid installation and configuration. The objective of this manual is to explain, in as much detail as possible, every configuration parameter available in Squid 3.0. As such, the reader is required to have prior knowledge of basic Squid installation and configuration. The details presented in this manual are in the nature of reference material.

For a complete tutorial on Squid, please visit http://www.squid-cache.org

NOTE: 1. Squid 3.0 is NOT a stable version. 2. new says newly added directives to squid 2.4 Stable x

Table of Contents

4. maximum_object_size
5. minimum_object_size
6. maximum_object_size_in_memory
7. ipcache_size
8. ipcache_low
9. ipcache_high
10. fqdncache_size
11. cache_replacement_policy
12. memory_replacement_policy

4. Logfile pathnames and cache directory

1. cache_dir
2. logformat new
3. access_log
4. cache_log
5. cache_store_log
6. cache_swap_log
7. emulate_httpd_log
8. log_ip_on_direct
9. mime_table
10. log_mime_hdrs
11. useragent_log
12. referer_log
13. pid_filename
14. debug_options
15. log_fqdn
16. client_netmask

5. Options for external support programs

1. ftp_user
2. ftp_list_width
3. ftp_passive
4. ftp_sanitycheck
5. check_hostnames new
6. cache_dns_program
7. dns_children
8. dns_retransmit_interval
9. dns_timeout
10. dns_defnames
11. dns_nameservers
12. hosts_file new
13. diskd_program
14. unlinkd_program
15. pinger_program
16. redirect_program
17. redirect_children
18. redirect_concurrency new
8. memory_pools
9. memory_pools_limit
10. via
11. forwarded_for
12. log_icp_queries
13. icp_hit_stale
14. minimum_direct_hops
15. minimum_direct_rtt
16. cachemgr_passwd
17. store_avg_object_size
18. store_objects_per_bucket
19. client_db
20. netdb_low
21. netdb_high
22. netdb_ping_period
23. query_icmp
24. test_reachability
25. buffered_logs
26. reload_into_ims
27. always_direct
28. never_direct
29. header_access
30. header_replace
31. icon_directory
32. error_directory
33. maximum_single_addr_tries
34. snmp_port
35. snmp_access
36. snmp_incoming_address
37. snmp_outgoing_address
38. as_whois_server
39. wccp_router
40. wccp_version
41. wccp_incoming_address
42. wccp_outgoing_address

12. Delay pool parameters

1. delay_pools
2. delay_class
3. delay_access
4. delay_parameters
5. delay_initial_bucket_level
6. incoming_icp_average
7. incoming_http_average
8. incoming_dns_average
9. min_icp_poll_cnt
10. min_dns_poll_cnt
11. min_http_poll_cnt

19. redirect_rewrites_host_header
20. redirector_access
21. auth_param new
22. authenticate_cache_garbage_interval new
23. authenticate_ttl
24. authenticate_ip_ttl
25. external_acl_type new

6. Options for tuning the cache

1. wais_relay_host
2. wais_relay_port
3. request_header_max_size
4. request_body_max_size
5. refresh_pattern
6. quick_abort_min
7. quick_abort_max
8. quick_abort_pct
9. read_ahead_gap new
10. negative_ttl
11. positive_dns_ttl
12. negative_dns_ttl
13. range_offset_limit

7. Timeouts

1. connect_timeout
2. peer_connect_timeout
3. read_timeout
4. request_timeout
5. persistent_request_timeout new
6. client_lifetime
7. half_closed_clients
8. pconn_timeout
9. ident_timeout
10. shutdown_lifetime
12. max_open_disk_fds
13. offline_mode
14. uri_whitespace
15. broken_posts
16. mcast_miss_addr
17. mcast_miss_ttl
18. mcast_miss_port
19. mcast_miss_encode_key
20. nonhierarchical_direct
21. prefer_direct
22. strip_query_terms
23. coredump_dir
24. redirector_bypass
25. ignore_unknown_nameservers
26. digest_generation
27. digest_bits_per_entry
28. digest_rebuild_period
29. digest_rewrite_period
30. digest_swapout_chunk_size
31. digest_rebuild_chunk_percentage
32. chroot
33. client_persistent_connections
34. server_persistent_connections
35. pipeline_prefetch
36. extension_methods
37. request_entities new
38. high_response_time_warning
39. high_page_fault_warning
40. high_memory_warning
41. store_dir_select_algorithm
42. ie_refresh
43. vary_ignore_expire new
44. sleep_after_fork

NETWORK PARAMETERS

Network parameters control network configuration, e.g. communication ports, secure network access and options, SSL options, inter-cache communication, multicast ICP queries etc.

TAG NAME	http_port
Description	Port where Squid will listen for clients http requests
Build Option	Default
Usage	**http_port** port [options] **http_port** hostname:port [options] **http_port** ip_adderss:port [options]
Default	none

Synopsis

This parameter allows the user to define the address on which Squid will listen for client's http requests. This is a required parameter, and there are no defaults.

Without this configuration, Squid will never start.

Arguments

port	Port to which Squid will bind the socket
hostname	hostname to which Squid will bind the socket
ip_address	ip_address to which Squid will bind the socket

When a hostname or IP address is specified (as shown in variations 2 and 3 above), Squid binds the socket to that specific address.

Note: The http_port parameter may be specified multiple times, with different addresses each time. This will cause Squid to listen on multiple ports.

Options are arguments that further control the behavior of the Squid proxy. The supported values are explained in the table below:

Options	Functions
accel	Configure Squid in accelerator mode
transparent	Configure Squid as transparent proxies
vhost	Accelerator using virtual hosts
vport	Accelerator with virtual ip host support
vport=NN	As above, but uses specified port number rather than the http_port number.
defaultsite=xx	Main web site name for accelerators. also implies accel option
protocol=	Protocol to reconstruct accelerated requests with. Defaults to http.

Example(s)

http_port 3128
http_port 172.16.1.53:3300
http_port 172.16.1.53:80 accel defaultsite=visolve.com
http_port 3128 transparent

TAG NAME	https_port

Description	Port where Squid will listen for clients https requests
Build Option	--enable-ssl
Usage	**https_port** [ip:]port cert=certificate.pem [key=key.pem] [options...]
Default	none

Synopsis

This parameter specifies the address where Squid will listen for client's https requests. Its role is significant when Squid is configured in accelerator mode where SSL works to be done.

Arguments

ip	IP Address to which Squid will bind the socket
port	Port to which Squid will bind the socket
cert=certificate.pem	Path and the file name where SSL certificate is located
key=key.pem	Path and the file name where SSL private key for the certificate is located

options controls other additional features and are explained in the table below:

Options	**Functions**
defaultsite=	The name of the https site presented on this port
protocol=	Protocol to reconstruct accelerated requests with. Defaults to https.
cert=	Path to SSL certificate (PEM format)
key=	Path to SSL private key file (PEM format) if not specified, the certificate file is assumed to be a combined certificate and key file
version=	The version of SSL/TLS supported 1 automatic (default) 2 SSLv2 only 3 SSLv3 only 4 TLSv1 only
cipher=	Colon separated list of supported ciphers
options=	Various SSL engine options. The most important being: NO_SSLv2 Disallow the use of SSLv2 NO_SSLv3 Disallow the use of SSLv3 NO_TLSv1 Disallow the use of TLSv1 SINGLE_DH_USE Always create a new key when using temporary/ephemeral DH key exchanges See src/ssl_support.cc or OpenSSL SSL_CTX_set_options documentation for a complete list of options.
clientca=	File containing the list of CAs to use when requesting a client certificate
cafile=	File containing additional CA certificates to use when verifying client certificates. If unset clientca will be used.
capath=	Directory containing additional CA certificates to use when verifying client certificates
dhparams=	File containing DH parameters for temporary/ephemeral DH key exchanges
sslflags=	Various flags modifying the use of SSL: DELAYED_AUTH - Don't request client certificates immediately, but wait until acl processing requires a certificate NO_DEFAULT_CA - Don't use the default CA list built in to OpenSSL.

Example(s)

https_port 443 cert=/usr/local/ssl/cert.pem key=/usr/local/ssl/key.pem defaultsite=visolve.com

TAG NAME	ssl_unclean_shutdown

Description	Used to handle bugs in browsers which does not fully support SSL
Build Option	--enable-ssl
Usage	**ssl_unclean_shutdown on|off**
Default	ssl_unclean_shutdown off

Synopsis
Some browsers like MSIE will indicate bugs during SSL shutdown. During such conditions, making this tag "on" will handle those bugs.

Arguments

on/off	Enable or disable ssl_unclean_shutdown

TAG NAME	ssl_engine
Description	Defines Hardware SSL acceleration which is to be used
Build Option	--enable-ssl
Usage	**ssl_engine** engine
Default	none

Synopsis
The openssl engine to use. For Example(s), you will need to set this if you would like to use hardware SSL acceleration.

Arguments

engine	Hardware SSL accelerator to be used

TAG NAME	sslproxy_client_certificate
Description	Used to define clients SSL certificate for proxying https:// URLs
Build Option	--enable-ssl
Usage	**sslproxy_client_certificate** path/certificatefile
Default	none

Synopsis
When proxying https:// URLs requests, this tag defines the clients SSL certificate path and the certificate file to be used for verification.

Arguments

path/ certificatefile	Path and the file that holds the clients SSL certificate

Example(s)
sslproxy_client_certificate /usr/local/ssl/cert.pem

TAG NAME	sslproxy_client_key
Description	Defines clients SSL certificate key for proxying https:// URLs
Build Option	--enable-ssl
Usage	**sslproxy_client_key** path/key.pem
Default	none

Synopsis
When Squid is used as a proxy server for https:// URLs requests, this tag defines the clients SSL certificate key's path and the file that holds the key.

Arguments

path/key. pem	Path and the file that contains the clients certificate key

Example(s)
sslproxy_client_key /usr/local/ssl/certkey.pem

TAG NAME	sslproxy_version
Description	Defines the SSL version level to be used when proxying https:// URLs
Build Option	--enable-ssl
Usage	**sslproxy_version** version
Default	sslproxy_version 1

Synopsis
When SSL certificate is used for proxying https:// URLs, this tag can be used to define the SSL version level that will be used for handling encrypted
connections.

Arguments

version	SSL version level

Example(s)
sslproxy_version 3

TAG NAME	sslproxy_options
Description	This defines the SSL engine options to be used when proxying https:// URLs
Build Option	--enable-ssl
Usage	**options** option
Default	none

Synopsis
When proxying https:// URLs, this tag is used to specify various SSL options.

Arguments

option	SSL options

Example(s)
sslproxy_options NO_SSLv2

TAG NAME	sslproxy_cipher
Description	SSL cipher list to be used when proxying https:// URLs
Build Option	--enable-ssl
Usage	**sslproxy_cipher** cipher
Default	none

Synopsis
This tag sets the ciphers on which SSL will decide during the negotiation phase of the SSL connection when proxying https:// URLs

Arguments

cipher	SSL proxy cipher to be used

TAG NAME	sslproxy_cafile

Description	Defines the file that contains CA certificate
Build Option	--enable-ssl
Usage	**sslproxy_cafile** filename
Default	none

Synopsis
This tag defines the file that contains CA certificate to be used for verifying server certificates when Squid is used as a proxy server for https://URLs.

Arguments
filename	File that contains CA certificate

Example(s)
sslproxy_cafile /usr/local/ca1.pem

TAG NAME	sslproxy_capath
Description	Defines the directory for the file containing CA certificate
Build Option	--enable-ssl
Usage	**sslproxy_capath** path
Default	none

Synopsis
While proxying https:// URLs, this tag defines the path where the CA certificate file to be used when verifying server certificates is located.

Arguments
path	Path where CA certificate file is located

Example(s)
sslproxy_capath /usr/local/

TAG NAME	sslproxy_flags
Description	Specifies the way how SSL should act while proxying https:// URLs
Build Option	--enable-ssl
Usage	**sslproxy_flags** flags
Default	none

Synopsis
When Squid is used as a proxy server for https://URLs, this tag is used to defines the nature of SSL's behaviour.

Arguments
Flags	Meaning
DONT_VERIFY_PEER	Accept certificates even if they fail to verify
NO_DEFAULT_CA	Don't use the default CA list built in to OpenSSL

Example(s)
sslproxy_flags NO_DEFAULT_CA

TAG NAME	icp_port

Description	Port number through which Squid sends and receives ICP queries
Build Option	Default
Usage	**icp_port** portnumber
Default	icp_port 0

Synopsis
Defines the port for ICP packets to be sent and received from neighbour caches.

Arguments
portnumber	Port to which Squid will bind the socket

Example(s)
icp_port 3030

TAG NAME	htcp_port
Description	Port number through which Squid sends and receives HTCP queries
Build Option	Default
Usage	**htcp_port** portnumber
Default	htcp_port 4827

Synopsis
This tag defines the port address through which HTCP packets will be sent and received from neighbour caches.

Arguments
portnumber	Port to which Squid will bind the socket

Example(s)
htcp_port 2134

TAG NAME	mcast_groups
Description	Defines list of multicast groups which your server should join to receive multicasted ICP queries
Build Option	Default
Usage	**mcast_groups** ip_address
Default	none

Synopsis
Multicast is essentially the ability to send one IP packet to multiple receivers. Your server will join to the multicat groups defined by the IP Addresses.

This option is to be set only if you want to RECEIVE multicast queries.

ICP replies are always sent via unicast, so this option does not affect whether or not you will receive replies from multicast group members.

Arguments
ip_address	ip_address of the multicast groups to join

Example(s)
mcast_groups 239.128.16.128 224.0.1.20

TAG NAME	udp_incoming_address, udp_outgoing_address

Description	Defines the address for sending and receiving ICP packets
Build Option	Default
Usage	**udp_incoming_address** ip_address **udp_outgoing_address** ip_address
Default	udp_incoming_address 0.0.0.0 udp_outgoing_address 255.255.255.255

Synopsis

These tags defines the interface through which ICP packets are sent and received. The default behavior is to not bind to any specific address.

A *udp_incoming_address* value of 0.0.0.0 indicates that Squid should listen for UDP messages on all available interfaces.

If *udp_outgoing_address* is set to 255.255.255.255 (the default) then it will use the same socket as *udp_incoming_address*. Only change this if you want to have ICP queries sent using another address than where this Squid listens for ICP queries from other caches.

Arguments

ip_address	ip_address to which Squid binds the ICP socket

Note: u*dp_incoming_address* and *udp_outgoing_address* cannot have the same value since they both use port 3130.

Example(s)

udp_incoming_address 172.16.1.35
udp_outgoing_address 192.168.150.6

NEIGHBOUR SELECTION ALGORITHM

Configurations needed for communication of Squid with the neighbor caches are done under this category.

TAG NAME	cache_peer
Description	This specifies other caches in cache hierarchy
Build Option	Default
Usage	**cache_peer** hostname type http_port icp_port [options]
Default	none

Synopsis

This defines how to treat the neighbour peer's in cache hierarchy. This is used during inter cache communication.

Arguments

hostname	The cache peer to which communication is to be established
type	The way how the cache peer be treated (either as '*parent*', '*sibling*' or '*multicast*').
proxy_port	Port number where the cache listens for other peers requests.
icp_port	Used for querying neighbor caches about objects. To have a non-ICP neighbor specify '7' for the ICP port and make sure the neighbour machine has the UDP echo port - enabled in its /etc/inetd.conf file.

Options	Functions
proxy-only	to specify that objects fetched from this cache should not be saved locally.
weight=n	to specify a weighted parent. The weight must be an integer. The default weight is 1, larger weights are favored more.
basetime=n	to specify a base amount to be subtracted from round trip times of parents. It is subtracted before division by weight in calculating which parent to fetch from. If the rtt is less than the base time then the rtt is set to a minimal value.
ttl=n	to specify a IP multicast TTL to use when sending an ICP queries to this address. Only useful when sending to a multicast group. Because we don't accept ICP replies from random hosts, you must configure other group members as peers with the multicast-responder' option below.
no-query	NOT to send ICP queries to this neighbor.

background-ping	only send ICP queries to this neighbor infrequently. This is used to keep the neighbor round trip time updated and is usually used in conjunction with weighted-round-robin.
default	if this is a parent cache which can be used as a "last-resort." You should probably only use 'default' in situations where you cannot use ICP with your parent cache(s).
round-robin	to define a set of parents which should be used in a round-robin fashion in the absence of any ICP queries.
weighted-round-robin	to define a set of parents which should be used in a round-robin fashion with the frequency of each parent being based on the round trip time. Closer parents are used more often.
carp	to define a set of parents which should be used as a CARP array. The requests will then be distributed among the parents based on the CARP load balancing hash function based on their weight.
multicast-responder	indicates that the named peer is a member of a multicast group. ICP queries willnot be sent directly to the peer, but ICP replies will be accepted from it.
closest-only	indicates that, for ICP_OP_MISS replies, we'll only forward CLOSEST_PARENT_MISSes and never FIRST_PARENT_MISSes.
no-digest	NOT to request cache digests from this neighbor.
no-netdb-exchange	disables requesting ICMP RTT database (NetDB) from the neighbor.
no-delay	to prevent access to this neighbor from influencing the delay pools.
login=user:password	if this is a personal/workgroup proxy and your parent requires proxy authentication. The string can include URL escapes (i.e. %20 for spaces). This also means that % must be written as %%.
login=PASS	if users must authenticate against the upstream proxy. This will pass the users credentials as they are to the peer proxy. This only works for the Basic HTTP authentication scheme. To combine this with proxy_auth both proxies must share the same user database as HTTP only allows for one proxy login. Also be warned that this will expose your users proxy password to the peer. USE WITH CAUTION
login=*:password	to pass the username to the upstream cache, but with a fixed password. This is meant to be used when the peer is in another administrative domain, but it is still needed to identify each user. The star can optionally be followed by some extra information which is added to the username. This can be used to identify this proxy to the peer, similar to the login=username:password option above.
connect-timeout=nn	to specify a peer specific connect timeout (also see the peer_connect_timeout directive)
digest-url=url	to tell Squid to fetch the cache digest (if digests are enabled) for this host from the specified URL rather than the Squid default location.
allow-miss	to disable Squid's use of only-if-cached when forwarding requests to siblings. This is primarily useful when icp_hit_stale is used by the sibling. To extensive use of this option may result in forwardingloops, and you should avoid having two-way peerings with this option. (for Example(s) to deny peer usage on requests from peer by denying cache_peer_access if the source is a peer)
max-conn	to limit the amount of connections Squid may open to this peer.
htcp	to send HTCP, instead of ICP, queries to the neighbor. You probably also want to set the "icp port" to 4827 instead of 3130.
originserver	causes this parent peer to be contacted as a origin server. Meant to be used in accelerator setups.
name=xxx	if you have multiple peers on the same host but different ports. This name can then be used to differentiate the peers in cache_peer_access and similar directives.
forceddomain=name	to forcibly set the Host header of requests forwarded to this peer. Useful in accelerator setups where the server (peer) expects a certain domain name and using redirectors to feed this domainname is not feasible.
ssl	to indicate that connections to this peer should bs SSL/TLS encrypted.
sslcert= /path/to/ssl/certificate	to specify a client SSL certificate to use when connecting to this peer.
sslkey= /path/to/ssl/key	to specify the private SSL key corresponding to sslcert above. If 'sslkey' is not specified then 'sslcert' is assumed to reference a combined file containing both the certificate and the key.
sslversion=1|2|3|4	to specify the SSL version to use when connecting to this peer 1 = automatic (default) 2 = SSL v2 only 3 = SSL v3 only 4 = TLS v1 only
sslcipher=...	to specify the list of valid SSL chipers to use when connecting to this peer

ssloptions=...	to specify various SSL engine options NO_SSLv2 Disallow the use of SSLv2 NO_SSLv3 Disallow the use of SSLv3 NO_TLSv1 Disallow the use of TLSv1
cafile=...	to specify a file containing additional CA certificates to use when verifying the peer certificate
capath=...	to specify a directory containing additional CA certificates to use when verifying the peer certificate
sslflags=...	to specify various flags modifying the SSL implementation DONT_VERIFY_PEER - Accept certificates even if they fail to verify. NO_DEFAULT_CA - Don't use the default CA list built in to OpenSSL. DONT_VERIFY_DOMAIN - Don't verify that the peer certificate matches the server name
sslname=	to specify the peer name as advertised in it's certificate. Used for verifying the correctness of the received peer certificate. If not specified the peer hostname will be used.
front-end-https	to enable the "Front-End-Https: On" header needed when using Squid as a SSL frontend infront of Microsoft OWA. See MS KB document Q307347 for details on this header. If set to auto then the header will only be added if the request is forwarded as a https://URL.

Example(s)

cache_peer proxy.visolve.com parent 3128 3130 default
cache_peer 172.16.1.57 parent 3128 3130 proxy-only
cache_peer 172.16.1.123 sibling 3129 5500 weight=2

TAG NAME	cache_peer_domain
Description	Used to limit the domains for which a neighbour cache will be queried
Build Option	Default
Usage	**cache_peer_domain** cache-host domain [domain ...]
Default	none

Synopsis

In case if there are more number of cache peers, then using this tag we can direct the query to that cache peer for particular domains alone. Prefixing the domain with "!" will be queried for objects NOT in that domain.

Arguments

cache-host	The cache peer to be queried for the specified domain
domain	The domain for which the cache peer to be queried

Example(s)

cache_peer_domain 172.16.1.57 .co.in

TAG NAME	neighbor_type_domain
Description	Using this tag, we can modify the define nerighbour type for particular domains
Build Option	Default
Usage	**neighbor_type_domain** neighbour parent|sibling domain domain ...
Default	none

Synopsis

There may be situations where an already defined neighbour to be treated differently for particular domains alone. This can be achieved using this directive.

Arguments

neighbour	The neighbour which to be treated diffrently
parent|sibling	How the neighbour to be treated (parent/sibling)
domain	The domain for which the cache peer to be treated differently

Example(s)

cache_peer parent 172.16.1.57 3128 3130
neighbor_type_domain 172.16.1.57 sibling.com

TAG NAME	icp_query_timeout
Description	Used to define the inter-cache query timeout
Build Option	Default
Usage	**icp_query_timeout** time(msec)
Default	icp_query_timeout 0

Synopsis

Based on the round trip time of recent ICP queries, Squid normally determines an optimal ICP query timeout. If you want to override this value, you can specify the timeouts using this tag.

The value specified is in Milliseconds.

Arguments

time	Fixed time period for ICP queries

Example(s)

icp_query_timeout 2000

TAG NAME	maximum_icp_query_timeout
Description	Defines ICP query timeout value to a maximum limit
Build Option	Default
Usage	**maximum_icp_query_timeout** time(msec)
Default	maximum_icp_query_timeout 2000

Synopsis

Normally the ICP query timeout is determined dynamically. But sometimes it can lead to very large values (say 5 seconds). Use this option to put an upper limit on the dynamic timeout value.

The value specified is in Milliseconds.

Note: Do NOT use this option to always use a fixed (instead of a dynamic) timeout value. To set a fixed timeout see the icp_query_timeout directive.

Arguments

time	Maximum upper time limit

Example(s)

maximum_icp_query_timeout 4000

TAG NAME	minimum_icp_query_timeout
Description	Defines ICP query timeout value to a minimum limit
Build Option	Default
Usage	**minimum_icp_query_timeout** time(msec)
Default	minimum_icp_query_timeout 5

Synopsis

As in the previous tag, ICP query timeouts to very small value, even lower than the normal latency variance on your link due to traffic. Use this option to put an lower limit on the dynamic timeout value.

The value specified is in Milliseconds.

Note: Do NOT use this option to always use a fixed (instead of a dynamic) timeout value. To set a fixed timeout see the icp_query_timeout directive.

Arguments

time	Minimum lower time limit

Example(s)
minimum_icp_query_timeout 4000

TAG NAME	mcast_icp_query_timeout
Description	In case of multicast peer's, the value specified in this tag determines how long should Squid wait to count all replies from its peers
Build Option	Default
Usage	**mcast_icp_query_timeout** time(msec)
Default	mcast_icp_query_timeout 2000

Synopsis
For Multicast peers, Squid regularly sends out ICP "probes" to count how many other peers are listening on the given multicast address. This tag determines the time how long Squid should wait to count all replies from its peers.

The value specified is in Milliseconds.

Arguments

time	Time period to wait

Example(s)
mcast_icp_query_timeout 3000

TAG NAME	dead_peer_timeout
Description	Defines the time period after which Squid will declare the corresponding peer as dead
Build Option	Default
Usage	**dead_peer_timeout** time(sec)
Default	dead_peer_timeout 10 seconds

Synopsis
This allows Squid to define the time period for declaring a peer cache as "dead." If there are no ICP replies received with in the specified amount of time, Squid will declare that peer as dead and will not expect to receive any further ICP replies. However, it continues to send ICP queries, and will mark the peer as alive upon receipt of the first subsequent ICP reply.

Note: This timeout also affects when Squid expects to receive ICP replies from peers. If more than dead_peer seconds have passed since the last ICP reply was received, Squid will not expect to receive an ICP reply on the next query. Thus, if your time between requests is greater than this timeout, you will see a lot of requests sent DIRECT to origin servers instead of to your parents.

Arguments

time	Time period to decide the cache peer as dead

Example(s)
dead_peer_timeout 50 seconds

TAG NAME	hierarchy_stoplist
Description	Use this tag not to query neighbour caches for certain objects
Build Option	Default
Usage	**hierarchy_stoplist** words
Default	none

Synopsis
Certain words defined in this tag when matched in the URLs, directs Squid not to query neighbour caches.

Arguments

words	Words to be matched for direct access

Example(s)
hierarchy_stoplist cgi-bin ?

TAG NAME	no_cache
Description	Use this to force certain objects to never be cached
Build Option	Default
Usage	**no_cache allow|deny** acl ...
Default	none

Synopsis

A list of ACL elements which, if matched, cause the request not to be satisfied from the cache and the reply not to be cached. In other words, use this to force certain objects to never be cached.

You must use the word 'DENY' to indicate the ACL names which should NOT be cached.

Arguments

allow/deny	Allow or deny caching of objects on matching the acl
acl	The condition/rule to be matched for which caching of those objects can be allowed or denied

Example(s)

```
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
```

TAG NAME	background_ping_rate
Description	Defines the rate of ICP pings
Build Option	Default
Usage	**background_ping_rate** time
Default	background_ping_rate 10 seconds

Synopsis

Squid normally sends ICP pings to the siblings. This directive defines the ICP ping rate.

Arguments

time	Background pinging rate

Example(s)

```
background_ping_rate 10 seconds
```

OPTIONS WHICH AFFECT THE CACHE SIZE

Tags under this section deals with cache memory configurations like cache memory size, swap size, maximum and minimum object size, cache and memory replacement policies.

TAG NAME	cache_mem
Description	cache_mem defines the ideal amount of memory to be used for In-Transit objects, Hot Objects, Negative-Cached objects
Build Option	Default
Usage	**cache_mem** size
Default	cache_mem 8 MB

Synopsis

Data for these objects are stored in 4 KB blocks. This parameter specifies the ideal upper limit on the total size of 4 KB blocks allocated.

In-transit objects have priority over the others. When additional space is needed for incoming data, *Negative-cached* and *Hot* objects will be released. In other words, the negative-cached and hot objects will fill up any unused space not needed for *In-transit* objects.

If circumstances require, this limit will be exceeded. Specifically, if your incoming request rate requires more than *cache_mem* of memory to hold *In-transit* objects, Squid will exceed this limit to satisfy the new requests. When the load decreases, blocks will be freed until the high-water mark is reached. Thereafter, blocks will be used to store hot objects.

Note: This tag does not specify the maximum process size. It places a limit on one aspect of squid's memory usage. Squid uses memory for other things as well. Process will probably become twice or three times bigger than the value put here.

Arguments

size	Cache memory size

Example(s)

cache_mem 32 MB

TAG NAME	cache_swap_low, cache_swap_high
Description	This defines low- and high-water marks for cache object replacements
Build Option	Default
Usage	**cache_swap_low** percent(0-100) **cache_swap_high** percent(0-100)
Default	cache_swap_low 90 cache_swap_high 95

Synopsis

This tags define when the replacement should take place. Replacement begins when the swap (disk) usage is above the low-water mark and attempts to maintain utilization near the low-water mark. As swap utilization gets close to high-water mark object eviction becomes more aggressive. If utilization is close to the low-water mark less replacement is done each time.

Defaults are 90% and 95%. If you have a large cache, 5% could be hundreds of MB. If this is the case you may wish to set these numbers closer together.

Arguments

percent	low and high level in percentage

Example(s)

cache_swap_low 50
cache_swap_high 75

TAG NAME	maximum_object_size
Description	Defines maximum size for objects to be stored in the disk
Build Option	Default
Usage	**maximum_object_size** size
Default	object_size 4096 KB

Synopsis

Objects larger than this size will NOT be saved on disk. The value is specified in kilobytes, and the default is 4MB. If you wish to get a high BYTES hit ratio, you should probably increase this (one 32 MB object hit counts for 3200 10KB hits). Leave this value low if you wish to increase the speed more than what you want to save bandwidth.

Note: If using the LFUDA replacement policy you should increase this value to maximize the byte hit rate improvement of LFUDA! See replacement_policy below for a discussion of this policy.

Arguments

size	Maximum object size

Example(s)

maximum_object_size 320010 KB

TAG NAME	minimum_object_size
Description	Specifies the minimum object size below which will not be saved to the disk
Build Option	Default
Usage	**minimum_object_size** size
Default	minimum_object_size 0 KB

Synopsis

Objects smaller than this size will NOT be saved on disk. The value is specified in kilobytes, and the default is 0 KB, which means there is no minimum.

Arguments

size	Minimum object size

Example(s)

minimum_object_size 10 KB

TAG NAME	maximum_object_size_in_memory
Description	Defines maximum size of the object to be kept in memory cache
Build Option	Default
Usage	**maximum_object_size_in_memory** size
Default	maximum_object_size_in_memory 8 KB

Synopsis

Objects greater than the size specified in this tag will not be kept in the memory cache. This should be set high enough to keep objects accessed frequently in memory to improve performance at the same time low enough to keep larger objects from hoarding cache_mem.

Arguments

size	Maximum size of the object to be kept in memory cache

Example(s)

maximum_object_size_in_memory 100 KB

TAG NAME	ipcache_size, ipcache_low, ipcache_high

Description	The size of the cache used for IP addresses and the high and low water marks for the same
Build Option	Default
Usage	**ipcache_size** number of entries **ipcache_low** percent **ipcache_high** percent
Default	ipcache_size 1024 ipcache_low 90 ipcache_high 95

Synopsis

Defines the size of cache needed for caching ip address, also its low and high water marks.

Arguments

number of entries	Number of entries to be cached
percent	low and high level for the ipcache in percentage

Example(s)

ipcache_size 2048
ipcache_low 90
ipcache_high 95

TAG NAME	fqdncache_size
Description	Defines the size of in memory cache needed for fully qualified domain names
Build Option	Default
Usage	**fqdncache_size** number of entries
Default	fqdncache_size 1024

Synopsis

This is used to specify maximum number of entries for fully qualified domain names. Defaults to 1024, which is usually a safe value. In environments where DNS queries are slow, raising this may help.

Arguments

number of entries	Number of fully qualified domains to be cached

Example(s)

fqdncache_size 2048

TAG NAME	cache_replacement_policy
Description	The cache replacement policy parameter determines which objects are to be replaced when disk space is needed
Build Option	--enable-removal-policy
Usage	**cache_replacement_policy** policy
Default	cache_replacement_policy lru

Synopsis

Whenever space for new objects were not found in the disk, *cache_replacement_policy* tag determines which objects in the cache memory (disk) should be replaced.

The cache replacement policies is of four types. They are,

Policy	Explanation
lru	Squid's original list based LRU policy
heap GDSF	Greedy-Dual Size Frequency
heap LFUDA	Least Frequently Used with Dynamic Aging
heap LRU	LRU policy implemented using a heap

This applies to any cache_dir lines listed below this.

The *lru* policies keeps recently referenced objects.

The *heap GDSF* policy optimizes object hit rate by keeping smaller popular objects in cache so it has a better chance of getting a hit. It achieves a lower byte hit rate than *LFUDA* though since it evicts larger (possibly popular) objects.

The *heap LFUDA* policy keeps popular objects in cache regardless of their size and thus optimizes byte hit rate at the expense of hit rate since one large, popular object will prevent many smaller, slightly less popular objects from being cached.

Both policies utilize a dynamic aging mechanism that prevents cache pollution that can otherwise occur with frequency-based replacement policies.

For more information about the GDSF and LFUDA cache replacement policies see http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html.

Note: If using the LFUDA replacement policy you should increase the value of maximum_object_size above its default of 4096 KB to maximize the potential byte hit rate improvement of LFUDA.

Arguments

policy	One of the above mentioned policies

Example(s)

cache_replacement_policy heap LFUDA

TAG NAME	memory_replacement_policy
Description	Specifies the policy for object replacement in memory when space for new objects is not available
Build Option	Default
Usage	**memory_replacement_policy** policy
Default	memory_replacement_policy lru

Synopsis

Like *cache_replacement_policy*, this applies to memory space (RAM) for object replacement when the required space is not available for new objects.

Policies are same as cache_replacemen_policy.

Arguments

policy	One of the policies mentioned in cache_replacement_policy tag

Example(s)

memory_replacement_policy LFUDA

LOG FILE PATH NAMES AND CACHE DIRECTORIES

Squid provides a number of logs that can be used when debugging problems, and when measuring the effectiveness and identifying users and the sites they visit. Because Squid can be used to "snoop" on users browsing habits, one should carefully consider privacy laws in your region and more importantly be considerate to your users. That's being said, logs can be very valuable tools in insuring that your users get the best service possible from your cache.

TAG NAME	cache_dir

Description	This is used to define cache directory, its path, type and size
Build Option	Default
Usage	**cache_dir** Type Directory-Name Mbytes Level1 Level2 [options]
Default	cache_dir ufs /usr/local/Squid/var/cache 100 16 256

Synopsis

All objects which are to be cached are stored in the disk space defined by this tag. This defines the path to cache directory, cache directory name, type and size of the cache area.

Arguments

Type	Type specifies the kind of storage system to use. Only "ufs" is built by default. To enable any of the other storage systems see the **--enable-storeio** configure option. Type is one of the following: 1. **ufs** is the old well-known Squid storage format that has always been there. 2. **aufs** uses the same storage format as **ufs**, utilizing POSIX-threads to avoid blocking the main Squid process on disk-I/O.This was formerly known in Squid as async-io. 3. **diskd** uses the same storage format as **ufs**, utilizing a separate process to avoid blocking the main Squid process on disk-I/O.

Type	Usage
ufs	cache_dir ufs Directory-Name Mbytes L1 L2 [options]
aufs	cache_dir aufs Directory-Name Mbytes L1 L2 [options]s
diskd	cache_dir diskd Directory-Name Mbytes L1 L2 [options] [Q1=n] [Q2=n]

Directory-Name	Directory name is a top-level directory where cache swap files will be stored.If you want to use an entire disk or caching, then this can be the mount-point directory. The directory must exist and be writable by the Squid process. Squid will NOT create this directory for you.
Mbytes	Mbytes is the amount of disk space (in MB) to use under this directory. The default is 100 MB. Change this to suit your configuration
Level1	Number of first-level subdirectories which will be created under the Directory. The default is 16.
Level2	number of second-level subdirectories which will be created under each first-level directory. The default is 256.
Q1	number of unacknowledged I/O requests when Squid stops opening new files. If this many messages are in the queues, Squid won't open new files. Default is 64.
Q2	number of unacknowledged messages when Squid starts blocking. If this many messages are in the queues, Squid blocks until it receives some replies. Default is 72.

Option:

max-size=n	refers to the max object size this storedir supports. It is used to initially choose the storedir to dump the object.

Note: To make optimal use of the max-size limits you should order the *cache_dir* lines with the smallest max-size value first and the ones with no max-size specification last.

Example(s)

cache_dir ufs /cache_dir 5000 16 256

TAG NAME	logformat

Description	Defines the format for storing access logs in access.log file
Build Option	Default
Usage	**logformat** <name> <format specification>
Default	none

Synopsis

Using this, the default log format can be changed according to the requirement. This customizable format will be needed when you want to perform analysis on the logs stored in access.log file.

Arguments

name	Identifier holding the customized logformat
format specification	It is a string embedded with % format codes

% format codes all follow the same basic structure where all but the formatcode is optional. Output strings are automatically quoted as required according to their context and the output format modifiers are usually unneeded but can be specified if an explicit quoting format is desired. The logformat name should be added at the end of access log file in the access_log tag.

% ["|[|'|#] [-] [[0]width] [{argument}] formatcode

"	quoted string output format
[	Squid log quoted format as used by log_mime_hdrs
#	URL quoted output format
'	No automatic quoting
-	left aligned
width	field width. If starting with 0 then output is zero padded
{arg}	argument such as header name etc

Format codes:

>a	Client source IP address
>A	Client FQDN
<A	Server IP address or peer name
la	Local IP address (http_port)
lp	Local port number (http_port)
ts	Seconds since epoch
tu	subsecond time (milliseconds)
tl	Local time. Optional strftime format argument default %d/%b/%Y:%H:%M:S %z
tg	GMT time. Optional strftime format argument default %d/%b/%Y:%H:%M:S %z
tr	Response time (milliseconds)
>h	Request header. Optional header name argument on the format header[:[separator]element]
<h	Reply header. Optional header name argument as for >h
un	User name
ul	User login
ui	User ident
ue	User from external acl
Hs	HTTP status code
Ss	Squid request status (TCP_MISS etc)
Sh	Squid hierarchy status (DEFAULT_PARENT etc)
mt	MIME content type
rm	Request method (GET/POST etc)
ru	Request URL
rv	Request protocol version
et	Tag returned by external acl
ea	Log string returned by external acl
<st	Reply size including HTTP headers
<sH	Reply high offset sent

<sS	Upstream object size
%	a literal % character

Example(s)
logformat Squid %ts.%03tu %6tr %>a %Ss/%03Hs %<st %rm %ru %un %Sh/%<A %mt

TAG NAME	access_log
Description	Defines the file where client activities are logged
Build Option	Default
Usage	**access_log** <filepath> [<logformat name> [acl acl ...]] **access_log** none [acl acl ...]
Default	access_log /usr/local/Squid3.0pre3/var/logs/access.log

Synopsis
All the activities the client does gets logged in this file. Using this, analysis on the access made by the clients can be done.

Customization of the logging can be done using the acl's which logs only those clauses in the specified logformat. If no acl is specified, all requests will be logged to this file.

To disable logging of a request specify "none".

Arguments
filepath	Location of the file that stores the logs
logformat name	Identifier that holds the customized log formats
acl	Filter rules to match

Example(s)
access_log /var/logs/access.log

If logformat is configured, then define the *access_log* as shown to get the defined logformats.

access_log /var/logs/access.log common, where common is the logformat name defined above.

TAG NAME	cache_log
Description	This tag is used to define the Cache logging file
Build Option	Default
Usage	**cache_log** <filepath>/filename
Default	cache_store_log /usr/local/Squid/var/logs/store.log

Synopsis
This defines the path and the file where general information about your cache's behavior goes. This file contains startup configuration information, as well as assorted error information during Squid's operation. This file is a good place to look when a website is found to have problems running through the web cache. Entries here may point towards a potential solution.

Arguments
filepath	Specifies the location of the file
filename	Actual file where the log is gathered

Example(s)
cache_log /var/logs/cache.log

TAG NAME	cache_store_log

Description	Configures the location of the caches store log file
Build Option	Default
Usage	**cache_store_log** <filepath>/filename
Default	cache_store_log /usr/local/Squid3.0pre3/var/logs/store.log

Synopsis

This tag defines the location where the transaction log of all objects that are stored in the object store, as well as the time when the object get deleted. This file really doesn't have very much use on a production cache, and it primarily recommended for use in debugging. Therefore, it can be turned off by entering none in the entry field.

Arguments

filepath	Specifies the location of the file
filename	Actual file where the log is gathered

Example(s)

cache_store_log /var/cache/store.log

TAG NAME	cache_swap_log
Description	Defines the filename used in each store directory to store the web caches metadata
Build Option	Default
Usage	**cache_log** <filepath>/filename
Default	none

Synopsis

This tag defines the file where metadata of objects saved on disk. This is a form of index for the web cache object store. These metadata is used to rebuild the cache during startup. This is not a human readable log, and it is strongly recommended to leave it in its default location on each store directory.

Note: You must give a full filename, not just a directory. Since this is the index for the whole object list you CANNOT periodically rotate it!

Arguments

filepath	Specifies the location of the file
filename	Actual file where the log is gathered

Example(s)

cache_swap_log /var/cache/cache_swap.log

TAG NAME	emulate_httpd_log
escription	Allows you to specify that Squid write its access.log in HTTPD common log file format
Build Option	Default
Usage	**emulate_httpd_log on|off**
Default	emulate_httpd_log off

Synopsis

Squid write its access.log in HTTPD common log file format, such as that used by Apache and many other web servers. This allows you to parse the log and generate reports using a wider array of tools. However, this format does not provide several types of information specific to caches, and is generally less useful when tracking cache usage and solving problems. Because there are several effective tools for parsing and generating reports from the Squid standard access logs, it is usually preferable to leave this at its default of being off.

Arguments

on/off	Enable or disable this process

TAG NAME	log_ip_on_direct

Description	This tag enables/disables logging of IP adress/hostname in the access.log file
Build Option	Default
Usage	**log_ip_on_direct on|off**
Default	log_ip_on_direct on

Synopsis
By making this directive to on, logs the IP Address of the destination server in the access.log file. If you want the hostname to be logged, then configure the directive to off mode.

Arguments

on/off	Enable or disable this process

TAG NAME	mime_table
Description	Used to define the file and path to the file where Squid's mime table is located
Build Option	Default
Usage	**mime_table** path/filename
Default	mime_table /usr/local/Squid/etc/mime.conf

Synopsis
Squid uses the mime table defined by this tag

Arguments

path	Path for the file where mime table file is located
filename	File that contains mime table

Example(s)
mime_table /usr/local//mime.conf

TAG NAME	log_mime_hdrs
Description	Enables to log extra information about clients requests in the access log
Build Option	Default
Usage	**log_mime_hdrs on|off**
Default	log_mime_hdrs off

Synopsis
When enabled, causes Squid to log more information about the request in the access.log file. This causes Squid to also write the request and response MIME headers for every request. These will appear in brackets at the end of each access.log entry.

Arguments

on/off	Enable or disable this process

TAG NAME	useragent_log
Description	Using this tag, you can make Squid to write User-Agent field from HTTP requests to the filename specified in this tag
Build Option	--enable-useragent-log
Usage	**useragent_log** path/filename
Default	none

Synopsis
By default *useragent_log* is disabled.

Arguments

path	Path for the useragent log file
filename	File that contains useragent logs

Example(s)
useragent_log /var/logs/usragent.log

TAG NAME	referer_log
Description	Squid will write the Referer field from HTTP requests to the filename specified here
Build Option	--enable-referer-log
Usage	**referer_log** path/filename
Default	none

Synopsis
By default *referer_log* is disabled.

Arguments

path	Path for the referer log file
filename	File that contains useragent logs

Example(s)
referer_log /var/logs/referer.log

TAG NAME	pid_filename
Description	Used to define a filename where the process id of Squid is stored
Build Option	Default
Usage	**pid_filename** path/filename
Default	pid_filename /usr/local/Squid3.0pre3/var/logs/Squid.pid

Synopsis
If you don't want Squid to create this file enter none instead of filename.

Arguments

path	Path for the Squid pid file
filename	File that contains pid of Squid's process

Example(s)
pid_filename /usr/local/Squid.pid

TAG NAME	debug_options
Description	This provides a means to configure all of Squid's various debug sections
Build Option	Default
Usage	**debug_options** section, level
Default	debug_options ALL,1

Synopsis

Squid's debugging code has been divided into a number of sections, so that if there is a problem in one part of Squid debug logging can be made more verbose for just that section. The magic word "ALL" sets debugging levels for all sections. We recommend normally running with "ALL,1".

Arguments

section	Defines which section's information to be logged
level	Defines debugging levels (0-9)

Example(s)

debug_options ALL, 9

TAG NAME	log_fqdn
Description	Turn this on if you wish to log fully qualified domain names in the access.log
Build Option	Default
Usage	**log_fqdn on|off**
Default	log_fqdn off

Synopsis

This configures whether Squid will attempt to resolve the hostname, so the the fully qualified domain name can be logged. This can, in some cases, increase latency of requests.

Arguments

on/off	Enable or disable this process

TAG NAME	client_netmask
Description	Defines what portion of the requesting client IP is logged in the access.log
Build Option	Default
Usage	**client_netmask** netmask
Default	client_netmask 255.255.255.255

Synopsis

You can make the requesting clients IP to be logged as such or only the network part of the IP alone with the host part being zero. For privacy reasons it is often preferred to only log the network or subnet IP of the client. For example, a netmask of 255.255.255.0 will log the first three octets of the IP, and fill the last octet with a zero.

Arguments

netmask	Clients network mask

Example(s)

client_netmask 255.255.120.200

OPTIONS FOR EXTERNAL SUPPORT PROGRAMS

External support programs could be viewed as a simple means of modular design, allowing third parties to write modules to improve the features of Squid. That's being said, some of Squid's standard functionality is also provided by helper programs. The standard helper programs include dnsserver, pinger, and several authentication modules. Third party modules include redirectors, ad blockers, and additional authentication modules.

TAG NAME	ftp_user

Description	This is the email address Squid uses to login to remote FTP servers anonymously
Build Option	Default
Usage	**ftp_user** username
Default	ftp_user squid@

Synopsis

For login to some servers, an anonymous email address is to be used. This tag is used to provide the anonymous email address for the login. This can simply be a user name followed by an @ symbol, which your domain name can be automatically attached to. Or it can be a full email address. This should be something reasonable for your domain, such as wwwuser@mydomain.com, or in the domainless case first mentioned, squid@, which happens to be the default for this option.

Arguments

username	User name to be used while login

Example(s)

ftp_user squid@squid.visolve.com

TAG NAME	ftp_list_width
Description	The column width for auto-generated Web pages of FTP sites queried through Squid when Squid is in forward proxy mode
Build Option	Default
Usage	**ftp_list_width** number
Default	ftp_list_width 32

Synopsis

This tag gives some control over how Squid formats the resulting file lists. Squid provides limited FTP proxy features to allow browsers (even older, non-FTP aware browsers) to communicate with FTP servers.

Arguments

number	Column width

Example(s)

ftp_list_width 48

TAG NAME	ftp_passive
Description	If your firewall does not allow Squid to use passive connections, then turn off this option
Build Option	Default
Usage	**ftp_passive on|off**
Default	ftp_passive on

Synopsis

Enable or disable passive connections.

Arguments

on/off	Enable or disable this option

TAG NAME	ftp_sanitycheck

Description	Squid performs sanity checks of the addresses of FTP data connections ensure the data connection is to the requested server
Build Option	Default
Usage	**ftp_sanitycheck on|off**
Default	ftp_sanitycheck on

Synopsis

For security and data integrity reasons Squid by default performs sanity checks of the addresses of FTP data connections ensure the data connection is to the requested server. If you need to allow FTP connections to servers using another IP address for the data connection then turn this off.

Arguments

on/off	Enable or disable sanity checks

TAG NAME	check_hostnames
Description	For security and stability reasons Squid by default checks hostnames for Internet standard RFC compliance
Build Option	Default
Usage	**check_hostnames on|off**
Default	check_hostnames on

Synopsis

If you want Squid not to perform these checks then turn this directive off.

Arguments

on/off	Enable or disable hostname checks

TAG NAME	cache_dns_program
Description	This helper program is used for DNS resolution
Build Option	--disable-internal-dns
Usage	**cache_dns_program** program
Default	cache_dns_program /usr/local/Squid/libexec/dnsserver

Synopsis

Squid requires a non-blocking resolver for its queries, an external program called dnsserver is included in the standard distribution. This tag is used to specify the path for the external dnsserver program.

Arguments

program	Path and the external dnsserver program

Example(s)

cache_dns_program /usr/local/libexec/dnsserver

TAG NAME	dns_children

Description	The number of processes spawn to service DNS name lookups
Build Option	Default
Usage	**dns_children** number (1 to 32)
Default	dns_children 5

Synopsis
Specifies the number of external DNS resolver processes that will be started in order to serve requests. The default value of five is enough for many networks, however, if your Squid serves a large number of users, this value may need to be increased to avoid errors. However, increasing the number of processes also increases the load on system resources and may actually hinder performance if set too high. More than 10 is probably overkill.

Arguments
number	Number of dns children program

Example(s)
dns_children 10

TAG NAME	dns_retransmit_interval
Description	Defines the initial retransmit time interval for DNS queries
Build Option	Default
Usage	**dns_retransmit_interval** time-units
Default	dns_retransmit_interval 5 seconds

Synopsis
The interval is doubled each time all configured DNS servers have been tried.

Arguments
time-units	Retransmit time interval

Example(s)
dns_retransmit_interval 15 seconds

TAG NAME	dns_timeout
Description	This defines the DNS query timeout
Build Option	Default
Usage	**dns_timeout** time-units
Default	dns_timeout 5 minutes

Synopsis
If no response is received to a DNS query within this time then all DNS servers for the queried domain is assumed to be unavailable.

Arguments
time-units	DNS timeout period

Example(s)
dns_timeout 10 minutes

TAG NAME	dns_defnames

Description	Enable/disable the dnsserver to add the local domain name to single component host names
Build Option	Default
Usage	**dns_defnames on|off**
Default	dns_defnames off

Synopsis
Normally the 'dnsserver' disables the RES_DEFNAMES resolver option (see res_init(3)). This prevents caches in a hierarchy from interpreting single-component hostnames locally. To allow dnsserver to handle single-component names, enable this option.

Arguments
on/off	Enable or disable this option

TAG NAME	dns_nameservers
Description	Use this if you want to specify a list of DNS name servers (IP addresses) to use
Build Option	Default
Usage	**dns_nameservers** ip_address
Default	none

Synopsis
Normally defaults to resolve.conf, which simply means that Squid's parent DNS servers will be drawn from the /etc/resolve.conf file found on the system Squid runs on. It is possible to select other DNS servers if needed, for **e**xample to choose a more local caching DNS server, or a remote internet connected server.

Arguments
ip_address	IP address of the dns servers

Example(s)
dns_nameservers 10.0.0.1 192.172.0.4

TAG NAME	hosts_file
Description	Defines the location of the host-local IP name-address associations database
Build Option	Default
Usage	**host_file** path/filename
Default	hosts_file /etc/hosts

Synopsis
For Unix and Linux system this file is located at /etc/hosts

Arguments
path	Path to the file that contains the ip addresses
filename	File that contains the ip addresses

Example(s)
hosts_file /hosts

TAG NAME	diskd_program

Description	Specifies the location of the diskd executable
Build Option	Default
Usage	**diskd_program** path/filename
Default	diskd_program /usr/local/Squid/libexec/diskd

Synopsis
This tag is used to specify the location where diskd program is located

Note: This is only useful if you have compiled in diskd as one of the store io modules.

Arguments

path	Path where diskd program is located
filename	File that performs diskd operation

Example(s)
diskd_program /usr/local/libexec/diskd

TAG NAME	unlinkd_program
Description	Specifies the location where executable for file deletion process is stored
Build Option	Default
Usage	**unlinkd_program** path/filename
Default	unlinkd_program /usr/local/Squid/libexec/unlinkd

Synopsis
The name of the helper program that deletes, or unlinks old files in the cache to make room for newer objects.

Arguments

path	Path where the program is located
filename	File that performs the specified operation

Example(s)
unlinkd_program /usr/local/libexec/unlinkd

TAG NAME	pinger_program
Description	Specifies the location of the executable for the pinger process
Build Option	--enable-icmp
Usage	**pinger_program** path/filename
Default	pinger_program /usr/local/Squid/libexec/pinger

Synopsis
An external program that provides Squid with ICMP RTT information so that it can more effectively choose between multiple remote parent caches for request fulfillment.

Arguments

path	Path of the pinger executable program
filename	File that performs the pinger process

Example(s)
pinger_program /usr/local/libexec/pinger

TAG NAME	redirect_program

Description	Specifies the location of the executable for the URL redirector
Build Option	Default
Usage	**redirect_program** path/redirector
Default	none

Synopsis

This provides a method to export a request to an external program, and then to import that programs response and act as though the client sent the resulting request. To configure a redirector, enter the path to the redirector and the redirector filename in this tag. By default, a redirector is not used.

Arguments

path	Location of the redirector program
redirector	Executable file that performs the redirection process

Example(s)

redirect_program /usr/local/squirm/bin/squirm

TAG NAME	redirect_children
Description	Specifies the number of redirector processes to spawn
Build Option	Default
Usage	**redirect_children** number
Default	redirect_children 5

Synopsis

For the redirector program, this defines the number of redirector process to spawn. If you start too few Squid will have to wait for them to process a backlog of URLs, slowing it down. If you start too many they will use RAM and other system resources.

Arguments

number	Number of redirector process to spawn

Example(s)

redirect_children 15

TAG NAME	redirect_concurrency
Description	Defines the number of requests each redirector helper can handle in parallel
Build Option	Default
Usage	**redirect_concurrency** number
Default	redirect_concurrency 0

Synopsis

Defaults to 0 which indicates that the redirector is a old-style single threaded redirector.

Arguments

number	Number of requests to be handle

Example(s)

redirect_concurrency 10

TAG NAME	redirect_rewrites_host_header

Description	Enable/disable Squid rewriting any host header in redirected requests
Build Option	Default
Usage	**redirect_rewrites_host_header on|off**
Default	redirect_rewrites_host_header on

Synopsis

By default Squid rewrites any host header in redirected requests. If you want Squid not to perform this operation disable this option.

Note: If you are running a accelerator then this may not be a wanted effect of a redirector

Arguments

on/off	Enable /disable rewriting of host headers

TAG NAME	redirector_access
Description	Used to define the access lists which are to be redirected to the redirtor process
Build Option	Default
Usage	**redirector_access** allow|deny acl ...
Default	none

Synopsis

Some access lists which does not need redirection can be denied using this tag. By default all requests are sent to the redirector process.

Arguments

allow/deny	Allow or deny the access list
acl	List that to be allowed or denied

Example(s)

acl me src 172.16.1.35
redirector_access allow me

TAG NAME	auth_param
Description	Provides an interface to the external authentication interface within Squid
Build Option	Default
Usage	**auth_param** scheme parameter [setting]
Default	netdb_ping_period 5 minutes

Synopsis

This is used to pass parameters to the various authentication schemes making users to be authenticated in a number of ways. various schemes are explained below.

Scheme	Parameter	Explanation
basic	"program" cmdline	Specify the command for the external authenticator. Such a program reads a line containing "username password" and replies "OK" or "ERR" in an endless loop. If you use an authenticator, make sure you have 1 acl of type proxy_auth. By default, the basic authentication sheme is not used unless a program is specified. If you want to use the traditional proxy authentication, jump over to the ../auth_modules/NCSA directory and type: % make % make install Then, set this line to something like auth_param basic program /usr/local/Squid/bin/ncsa_auth /usr/local/Squid/etc/passwd

	"children" numberofchildren	The number of authenticator processes to spawn (no default). If you start too few Squid will have to wait for them to process a backlog of usercode/password verifications, slowing it down. When password verifications are done via a (slow) network you are likely to need lots of authenticator processes. auth_param basic children 5
	"concurrency" concurrency	The number of concurrent requests the helper can process. The default of 0 is used for helpers who only supports one request at a time. auth_param basic concurrency 0
	"realm" realmstring	Specifies the realm name which is to be reported to the client for the basic proxy authentication scheme (part of the text the user will see when prompted their username and password). There is no default. auth_param basic realm Squid proxy-caching web server
	"credentialsttl" timetolive	Specifies how long Squid assumes an externally validated username:password pair is valid for - in other words how often the helper program is called for that user. Set this low to force revalidation with short lived passwords. Note that setting this high does not impact your susceptability to replay attacks unless you are using an one-time password system (such as SecureID). If you are using such a system, you will be vulnerable to replay attacks unless you also use the max_user_ip ACL in an http_access rule.
digest	"program" cmdline	Specify the command for the external authenticator. Such a program reads a line containing "username":"realm" and replies with the appropriate H(A1) value base64 encoded. See rfc 2616 for the definition of H(A1). If you use an authenticator, make sure you have 1 acl of type proxy_auth. By default, authentication is not used. If you want to use build an authenticator, jump over to the ../digest_auth_modules directory and choose the authenticator to use. It's directory type % make % make install Then, set this line to something like auth_param digest program /usr/local/Squid/bin/digest_auth_pw /usr/local/Squid/etc/digpass
	"children" number of children	The number of authenticator processes to spawn (no default). If you start too few Squid will have to wait for them to process a backlog of H(A1) calculations, slowing it down. When the H(A1) calculations are done via a (slow) network you are likely to need lots of authenticator processes. auth_param digest children 5
	"realm" realmstring	Specifies the realm name which is to be reported to the client for the digest proxy authentication scheme (part of the text the user will see when prompted their username and password). There is no default. auth_param digest realm Squid proxy-caching web server
	"nonce_garbage_interval" timeinterval	Specifies the interval that nonces that have been issued to client_agent's are checked for validity.
NTLM	"program" cmdline	Specify the command for the external ntlm authenticator. Such a program reads a line containing the unencoded NEGOTIATE and replies with the ntlm CHALLENGE, then waits for the response and answers with "OK" or "ERR" in an endless loop. If you use an ntlm authenticator, make sure you have 1 acl of type proxy_auth. By default, the ntlm authenticator_program is not used. auth_param ntlm program /usr/local/Squid/bin/ntlm_auth

	"children" number of children	The number of authenticator processes to spawn (no default). If you start too few Squid will have to wait for them to process a backlog of credential verifications, slowing it down. When crendential verifications are done via a (slow) network you are likely to need lots of authenticator processes. auth_param ntlm children 5
	"max_challenge_reuses" number	The maximum number of times a challenge given by a ntlm authentication helper can be reused. Increasing this number increases your exposure to replay attacks on your network. 0 means use the challenge only once. (disable challenge caching) See max_ntlm_challenge_lifetime for more information. auth_param ntlm max_challenge_reuses 0
	"max_challenge_lifetime" timespan	The maximum time period that a ntlm challenge is reused over. The actual period will be the minimum of this time AND the number of reused challenges. auth_param ntlm max_challenge_lifetime 2 minutes

Note: Once an authentication scheme is fully configured, it can only be shutdown by shutting Squid down and restarting.

Arguments

scheme	One of the above mentioned authentication scheme
parameter	various parameters for the schemes as listed above

Example(s)

auth_param basic program /usr/local/Squid/bin/ncsa_auth /usr/local/Squid/etc/passwd
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours

Recommended minimum configuration

auth_param digest program <uncomment and complete this line>
auth_param digest children 5
auth_param digest realm Squid proxy-caching web server
auth_param digest nonce_garbage_interval 5 minutes
auth_param digest nonce_max_duration 30 minutes
auth_param digest nonce_max_count 50

auth_param ntlm program <uncomment and complete this line to activate>
auth_param ntlm children 5
auth_param ntlm max_challenge_reuses 0
auth_param ntlm max_challenge_lifetime 2 minutes

auth_param basic program <uncomment and complete this line to activate>
auth_param basic children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours

TAG NAME	authenticate_cache_garbage_interval
Description	Defines the time period between garbage collection across the username cache
Build Option	Default
Usage	**authenticate_cache_garbage_interval** time
Default	authenticate_cache_garbage_interval 1 hour

Synopsis

This tag is used to specify the time period between garbage collection across the username cache.

Arguments

time	Specifies the time period

Example(s)

authenticate_cache_garbage_interval 2 hour

TAG NAME	authenticate_ttl
Description	Defines the time period for user & their credentials stay in the logged user cache since their last request
Build Option	Default
Usage	**authenticate_ttl** time
Default	authenticate_ttl 1hour

Synopsis

When the defined timeout reaches, then all user credentials that have passed their TTL are removed from memory.

Arguments

time	Time period of credentials stay

Example(s)

authenticate_ttl 2 hour

TAG NAME	authenticate_ip_ttl
Description	If you use proxy authentication and the max_user_ip ACL, this tag controls how long Squid remembers the IP addresses associated with each user
Build Option	Default
Usage	**authenticate_ip_ttl** time
Default	authenticate_ip_ttl 0 seconds

Synopsis

Use a small value (e.g., 60 seconds) if your users might change addresses quickly, as is the case with dialups. You might be safe using a larger value (e.g., 2 hours) in a corporate LAN environment with relatively static address assignments.

Arguments

time	Time period for which the ip addresses should be remembered

Example(s)

authenticate_ip_ttl 10 seconds

TAG NAME	external_acl_type
Description	This tag defines external acl classes using a helper program to look up the status
Build Option	Default
Usage	**external_acl_type** name [options] FORMAT.. path/helper [helper arguments..]
Default	none

Synopsis

This tag defines how the external acl classes using a helper program should look up the status.

Arguments

name	Name of the
path	Path to the external helper program
helper	Helper program

Options:

ttl=n	TTL in seconds for cached results (defaults to 3600 for 1 hour)
negative_ttl=n	TTL for cached negative lookups (default same as ttl)
children=n	Number of acl helper processes spawn to service external acl lookups of this type.
concurrency=n	concurrency level per process. Use 0 for old style helpers who can only process a single request at a time.
cache=n	result cache size, 0 is unbounded (default)
grace=n	Percentage remaining of TTL where a refresh of a cached entry should be initiated without needing to wait for a new reply. (default 0 for no grace period)

FORMAT specifications:

%LOGIN	Authenticated user login name
%IDENT	Ident user name
%SRC	Client IP
%SRCPORT	Client source port
%DST	Requested host
%PROTO	Requested protocol
%PORT	Requested port
%PATH	Requested URL path
%METHOD	Request method
%MYADDR	Squid interface address
%MYPORT	Squid http_port number
% USER_CERT_xx	SSL User certificate attribute xx
%USER_CA_xx	SSL User certificate CA attribute xx
%{Header}	HTTP request header
%{Hdr:member}	HTTP request header list member
%{Hdr:;member}	HTTP request header list member using ; as list separator. ; can be any non-alphanumeric character.

In addition, any string specified in the referencing acl will also be included in the helper request line, after the specified formats (see the "acl external" directive)

The helper receives lines per the above format specification, and returns lines starting with OK or ERR indicating the validity of the request and optionally followed by additional keywords with more details.

General result syntax: OK/ERR keyword=value ...

Defined Keywords	
ser=	The users name (login)
password=	The users password (for login= cache_peer option)
message=	Message describing the reason. Available as %o in error pages
tag=	Apply a tag to a request (for both ERR and OK results) Only sets a tag, does not alter existing tags.
log=	String to be logged in access.log. Available as %ea in logformat specifications

Keyword values need to be enclosed in quotes if they may contain whitespace, or the whitespace escaped using \. Any quotes or \ characters within the keyword value must be \ escaped.

Example(s)

auth_param basic program < put your authenticator here >
auth_param basic children 20
auth_param basic realm Squid proxy-caching web server

auth_param basic credentialsttl 1800 seconds
external_acl_type checkip children = 20 %LOGIN %SRC /usr/local/Squid/bin/checkip.pl
acl password external checkip
acl it src 172.16.20.1-172.16.20.199/255.255.255.255
http_access allow it password
Allows user if user belongs to a group that is allowed during a given time and using a given ip

---

OPTIONS FOR TUNING THE CACHE

This section describes the important parameters that determine Squid cache performance.

TAG NAME	wais_relay_host, wais_relay_port
Description	Defines WAIS host and port to relay WAIS requests
Build Option	Default
Usage	**wais_relay_host** hostname **wais_relay_port** portnumber
Default	wais_relay_host localhost wais_relay_port 8000

Synopsis

WAIS, or Wide Area Information System, is a system to catalog and search large amounts of data via a WAIS or WWW browser. This defaults to localhost and 8000.

Arguments

hostname	Machine name
portnumber	Port where to bind the socket

Example(s)

wais_relay_host localhost
wais_relay_port 8000

TAG NAME	request_header_max_size
Description	This specifies the maximum size for HTTP headers in a request
Build Option	Default
Usage	**request_header_max_size** size(KB)
Default	request_header_max_size 10 KB

Synopsis

Size of HTTP headers in a request can be controlled using this tag. Request headers are usually relatively small (about 512 bytes). Placing a limit on the request header size will catch certain bugs (for example with persistent connections) and possibly buffer-overflow or denial-of-service attacks.

Arguments

size	Maximum size of request header

Example(s)

request_header_max_size 100 KB

TAG NAME	request_body_max_size

Description	Specifies the maximum size for an HTTP request body
Build Option	Default
Usage	**request_body_max_size** size(KB)
Default	request_body_max_size 0 KB

Synopsis

This is the maximum size of a PUT/POST request. A user who attempts to send a request with a body larger than this limit receives an "Invalid Request" error message. If you set this parameter to a zero (the default), there will be no limit imposed.

Arguments

size	Maximum size of request body

Example(s)

request_body_max_size 10 KB

TAG NAME	refresh_pattern
Description	Used to define the manner how Squid treats the objects in the cache
Build Option	Default
Usage	**refresh_pattern [-i]** regex min percent max [options]
Default	-

Synopsis

The way how the objects in the cache be refreshed is defined using this tag. By default, regular expressions are CASE-SENSITIVE. To make them case-insensitive, use the -i option.

Basically a cached object is:

FRESH	if expires < now, else STALE
STALE	if age > max
FRESH	if lm-factor < percent, else STALE
FRESH	if age < min
else	STALE

The *refresh_pattern* lines are checked in the order listed here. The first entry which matches is used. If none of the entries match, then the default will be used.

Arguments

regex	regular expression
Min	time (in minutes), an object without an explicit expire time should be considered fresh.
percent	percentage of the objects age (time since last modification age) an object without explicit expire time will be considered fresh.
Max	upper limit on how long objects without an explicit expiry time will be considered fresh.

Options:

override-expire	enforces min age even if the server sent a Expires: header. Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems which it causes.
override-lastmod	enforces min age even on objects that was modified recently.
reload-into-ims	changes client no-cache or ``reload'' to If-Modified-Since requests. Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems which it causes.
ignore-reload	ignores a client no-cache or ``reload'' header. Doing this VIOLATES the HTTP standard. Enabling this feature could make you liable for problems which it causes.

Example(s)

```
refresh_pattern ^ftp:       1440    20%    10080
refresh_pattern ^gopher:    1440    0%     1440
refresh_pattern .           0       20%    4320
```

TAG NAME	quick_abort_min, quick_abort_max, quick_abort_pct

Description	Signals the cache how to continue downloads during abort signals sent by the clients
Buid Option	Default
Usage	**quick_abort_min** size **quick_abort_max** size **quick_abort_pct** percent
Default	quick_abort_min 16 KB quick_abort_max 16 KB quick_abort_pct 95

Synopsis

The cache by default continues downloading aborted requests which are almost completed (less than 16 KB remaining). This may be undesirable on slow (e.g. SLIP) links and/or very busy caches. Impatient users may tie up file descriptors and bandwidth by repeatedly requesting and immediately aborting
downloads.

Arguments

size	Minimum and maximum transfer size
percent	Percentage of transfer

When the user aborts a request, Squid will check the *quick_abort* values to the amount of data transferred until then.
If the transfer has less than *quick_abort_min* KB remaining, it will finish the retrieval.
If the transfer has more than *quick_abort_max* KB remaining, it will abort the retrieval.
If more than *quick_abort_pct* of the transfer has completed, it will finish the retrieval.
If you do not want any retrieval to continue after the client has aborted, set both *quick_abort_min* and *quick_abort_max* to '0 KB'.
If you want retrievals to always continue if they are being cached then set *quick_abort_min* to '-1 KB'.

Example(s)

quick_abort_min 30 KB
quick_abort_max 30 KB
quick_abort_pct 80

TAG NAME	read_ahead_gap
Description	Define the amount of data the cache will buffer ahead of what has been sent to the client when retrieving an object from another server
Buid Option	Default
Usage	**read_ahead_gap** buffer-size
Default	read_ahead_gap 16 KB

Synopsis

This tag determines the prefetch cache buffer size for holding objects from another server while sending to the client.

Arguments

buffer-size	Size of the cache buffer

Example(s)

read_ahead_gap 30 KB

TAG NAME	negative_ttl

Description	Defines Time-to-Live (TTL) for failed requests
Buid Option	Default
Usage	**negative_ttl** time-units
Default	negative_ttl 5 minutes

Synopsis
Certain types of failures (such as "connection refused" and "404 Not Found") are negatively-cached for a configurable amount of time. The default is 5 minutes. Note that this is different from negative caching of DNS lookups.

Arguments
time-units	Timeout for negatively cached objects

Example(s)
negative_ttl 1 minutes

TAG NAME	positive_dns_ttl
Description	Defines Time-to-Live (TTL) for positive caching of successful DNS lookups
Buid Option	Default
Usage	**positive_dns_ttl** time-units
Default	positive_dns_ttl 6 hours

Synopsis
For positive caching of successful DNS lookups, this defines Time-to-Live period. Default is 6 hours (360 minutes). If you want to minimize the use of Squid's ipcache, set this to 1, not 0.

Arguments
time-units	Timeout for positive cachings

Example(s)
positive_dns_ttl 24 hours

TAG NAME	negative_dns_ttl
Description	Time-to-Live (TTL) for negative caching of failed DNS lookups
Buid Option	Default
Usage	**negative_dns_ttl** time-units
Default	negative_dns_ttl 5 minutes

Synopsis
Sometimes DNS lookups may get failed. This parameter defines the Time-To-Live period for failed DNS lookups. Normally this will be a small value.

Arguments
time-units	Timeout period

Example(s)
negative_dns_ttl 1 minutes

TAG NAME	range_offset_limit

Description	Sets a upper limit on how far into the file a Range request may be to cause Squid to prefetch the whole file
Buid Option	Default
Usage	**range_offset_limit** bytes
Default	range_offset_limit 0 KB

Synopsis

If beyond this limit then Squid forwards the Range request as it is and the result is NOT cached.

This is to stop a far ahead range request (lets say start at 17MB) from making Squid fetch the whole object up to that point before sending anything to the client.

A value of -1 causes Squid to always fetch the object from the beginning so that it may cache the result. (2.0 style)
A value of 0 causes Squid to never fetch more than the client requested. (default)

Arguments

bytes	Upper limit for the range request

Example(s)

range_offset_limit 17 MB

TIMEOUT

Timeout parameters in Squid can be based on overall connection timeouts, peer-specific timeouts, site/domain-specific timeouts, request-specific timeouts etc. Proper setting of timeout values is critical to optimal Squid performance. Relevant parameters for timeout settings are listed here.

TAG NAME	connect_timeout
Description	An option to force Squid to close connections after a specified time
Build Option	Default
Usage	**connect_timeout** time-units
Default	connect_timeout 2 minutes

Synopsis

Some systems (notably older Linux versions) can not be relied upon to time out connect requests. For this reason, this option specifies the timeout for how long Squid should wait for the connection to complete. This value defaults to 120 seconds (2 minutes).

Arguments

time-units	Connection timeout period

Example(s)

connect_timeout 180 seconds

TAG NAME	peer_connect_timeout
Description	This parameter specifies how long to wait for a pending TCP connection to a peer cache
Build Option	Default
Usage	**peer_connect_timeout** time-units
Default	peer_connect_timeout 30 seconds

Synopsis
default is 30 seconds. You may also set different timeout values for individual neighbors with the 'connect-timeout' option on a cache_peer line.

Note: Setting of *peer_connect_timeout* to more than 30 seconds will be a performance issue.

Arguments

time-units	Time to wait for pending TCP connection

Example(s)
peer_connect_timeout 45 seconds

TAG NAME	read_timeout
Description	Used to set the timeout period for server-side connections
Build Option	Default
Usage	**read_timeout** time-units
Default	read_timeout 15 minutes

Synopsis
On each successful read() request the timeout is reset to this amount. If no data is read within this period of time, the request is aborted and logged with ERR_READ_TIMEOUT.

Arguments

time-units	Reset time duration

Example(s)
read_timeout 10 minutes

TAG NAME	request_timeout
Description	Defines the timeout for HTTP requests from clients
Build Option	Default
Usage	**request_timeout** time-units
Default	request_timeout 5 minutes

Synopsis
Using this, instruct Squid to wait for an HTTP request after initial connection establishment. By default the value is 5 minutes.

Arguments

time-units	Wait time period after initial connection establishment

Example(s)
request_timeout 8 minutes

TAG NAME	persistent_request_timeout
Description	This defines the time period to wait for the next HTTP request on a persistent connection after the previous request completes
Build Option	Default
Usage	**persistent_request_timeout** time-units
Default	persistent_request_timeout 1 minute

Synopsis
This tag defines the time period between completion of a HTTP request and starting of the next request on persistent connection.

Arguments

time-units	Time duration between the requests

Example(s)
persistent_request_timeout 1 minute

TAG NAME	client_lifetime
Description	The time limit Squid sets for a client to remain connected to the cache process
Build Option	Default
Usage	**client_lifetime** time-units
Default	client_lifetime 1 day

Synopsis
This defines the maximum amount of time that a client (browser) is allowed to remain connected to the cache process. This is merely a safeguard against clients that disappear without properly shutting down. It is designed to prevent a large number of sockets from being tied up in a CLOSE_WAIT state. The default for this option is 1440 minutes, or 1 day.

Note: The default value is intended to be much larger than any client would ever need to be connected to your cache. You should probably change client_lifetime only as a last resort. If you seem to have many client connections tying up file descriptors, we recommend first tuning the read_timeout, request_timeout, pconn_timeout and quick_abort values. If the more file descriptors are in use then the memory in use will also increase, which is also a performance issue.

Arguments

time-units	Client lifetime with the cache

Example(s)
client_lifetime 1000 minutes

TAG NAME	half_closed_clients
Description	Defines Squid's behavior towards some types of clients that close the sending side of a connection while leaving the receiving side open
Build Option	Default
Usage	**half_closed_clients** on/off
Default	half_closed_clients on

Synopsis
Turning this option off will cause Squid to immediately close connections when a read(2) returns "no more data to read". It's usually safe to leave this at the default value of on.

Arguments

on/off	Enable or disable this action

Example(s)
half_closed_clients off

TAG NAME	pconn_timeout

Description	Defines the timeout value for persistent connections
Build Option	Default
Usage	**pconn_timeout** time-units
Default	pconn_timeout 120 seconds

Synopsis

When this timeout is set, Squid will close persistent connections if they are idle for this amount of time. Persistent connections will be disabled entirely if this option is set to a value less than 10 seconds. The default is 120 seconds.

Arguments

time-units	Time period to wait for closing idle connections

Example(s)

pconn_timeout 2 minutes

TAG NAME	ident_timeout
Description	Maximum time to wait for IDENT lookups to complete
Build Option	Default
Usage	**ident_timeout** time-units
Default	ident_timeout 10 seconds

Synopsis

The timeout, usually in seconds. If this is set too high, you may be susceptible to denial or service from having too many outstanding ident requests. The default for this is 10 seconds.

Arguments

time-units	Time duration to wait during ident lookups

Example(s)

ident_timeout 5 seconds

TAG NAME	shutdown_lifetime
Description	This is the time Squid allows for existing connections to continue after it has received a shutdown signal
Build Option	Default
Usage	**shutdown_lifetime** time-units
Default	shutdown_lifetime 30 seconds

Synopsis

When SIGTERM or SIGHUP is received, the cache is put into "shutdown pending" mode until all active sockets are closed. It will stop accepting new connections immediately, but connections already in progress will continue to be served for this amount of time. Defaults to 30 seconds, which is a good safe value. Any active clients after this many seconds will receive a "timeout" message.

Note: If this time is set to be too low then some file descriptors may remain open which will be a performance issue in memory usage.

Arguments

time-units	Time period of "shutdown pending" mode

Example(s)

shutdown_lifetime 20 seconds

ACCESS CONTROLS

Access control settings are among the most important features of Squid. You can configure Squid to set filters for various entities and at different granularities (e.g. filters for specific protocols, filters for certain types of commands, filters for specific routers, filters for specified domains, etc).

TAG NAME	acl
Description	Used to define an Access List
Build Option	Default
Usage	**acl** aclname acltype string1 ... **acl** aclname acltype "file"
Default	Required minimum configuration for normal functioning

Synopsis

The first field represents the name of the ACL, which is simply an assigned name, that can be just about anything the user chooses. The second field is the type of the ACL, which can be one of a number of choices, that indicates to Squid what part of a request should be matched against for this ACL. The possible types include the requesting clients address, the Web server address or host name, a regular expression matching the URL, and many more. The final field is the actual string to match. Depending on what the ACL type is, this may be an IP address, a series of IP addresses, a URL, a host name, etc.

When using "file", the file should contain one item per line.

Arguments

aclname	Name of the acl
acltype	Type of acl
string	String to match
file	File that containing strings

ACL TYPE	src
Description	The IP address of the requesting client, or the clients IP Address
Build Option	Default
Usage	**acl** aclname **src** ip-address/netmask ... (clients IP address) **acl** aclname **src** addr1-addr2/netmask ... (range of addresses)

Synopsis

Matching done based on clients ip address

Arguments

aclname	Access list name
ip-address	Clients ip address
netmask	Network mask
add1-addr2	Range of ip address

Example(s)

acl network1 src 172.16.1.0/24 - refers to the whole Network with address 172.16.1.0
acl source1 src 172.16.1.25/32 - refers to a single source
acl srcrange src 172.16.1.25-172.16.1.35/32 - refers range of IP Addresses from 172.16.1.25-172.16.1.35

ACL TYPE	dst
Description	Same as src but looks for destination IP Address
Build Option	Default
Usage	**acl** aclname **dst** ip-address/netmask ... (URL host's IP address)

Synopsis
First Squid will dns-lookup for IPAddress from the domain-name, which is in request header. Then this acl is interpreted.

Arguments

aclname	Access list name
ip-address	ip-address of the oirgin server/machine
netmask	Network mask

Example(s)
acl dest dst 63.194.81.67/32

ACL TYPE	myip
Description	The local IP address on which the client connection exists.
Build Option	Default
Usage	**acl** aclname **myip** ip-address/netmask ... (local socket IP address)

Synopsis
Using this acl type, we can specify the local IP address on which the clients connections exists. This allows ACLs to be constructed that only match one physical network, if multiple interfaces are present on the proxy, among other things.

Arguments

aclname	Access list name
ip-address	ip address of the local machine
netmask	Network mask

Example(s)
acl myip1 myip 172.16.1.53/32

ACL TYPE	srcdomain
Description	This acl type provides matches against the client domain name
Build Option	Default
Usage	**acl** aclname **srcdomain** .domain-name ...

Synopsis
Matching can be either a single domain name, or a list or domain names, or the path to a file that contains a list of domain names. If a path to a file, it must be surrounded by parentheses. This ACL type can increase the latency, and decrease throughput significantly on a loaded cache, as it must perform an address-to-name lookup for each request, so it is usually preferable to use the Client IP Address type.

Note: Here "." is more important.

Arguments

aclname	Access list name
domain-name	Clients domain name

Example(s)
acl mydomain srcdomain .kovaiteam.com

ACL TYPE	dstdomain

Description	This refers to destination domain i.e. the source domain where the origin server is located
Build Option	Default
Usage	**acl** aclname **dstdomain** .domain-name ...

Synopsis
When matching to be done based on the destination servers domain, you can use this acl type.

Note: Here "." is more important.

Arguments

aclname	Access list name
domain-name	Destination domain

Example(s)
acl domain1 dstdomain .kovaiteam.com
This looks for *.kovaiteam.com from URL

ACL TYPE	srcdom_regex
Description	Matches against the client domain name
Build Option	Default
Usage	**acl** aclname **srcdom_regex [-i]** pattern ...

Synopsis
Matching can be either a single domain name, or a list of domain names, or a path to a file that contains a list of domain names. If a path to a file ia specified, it must be surrounded parentheses.

Arguments

aclname	Access list name
pattern	Pattern to match

Example(s)
acl aclname srcdom_regex kovai - This looks for the word kovai from the client domain name

ACL TYPE	dstdom_regex
Description	Provides match against destination domain
Build Option	Default
Usage	**acl** aclname **dstdom_regex [-i]** pattern ...

Synopsis
Matching done on destination domain based on regular expression.

Arguments

aclname	Access list name
pattern	Patern to match

Example(s)
acl domain1 dstdom_regex redhat - This looks for the word redhat from the client's reqest.

ACL TYPE	time

Description	This type is just what it sounds like, providing a means to create ACLs that are active during certain times of the day or certain days of the week
Build Option	Default
Usage	**acl** aclname **time** [day-abbrevs] [h1:m1-h2:m2]

Synopsis

Provides timed filter settings. This feature is often used to block some types of content or some sections of the internet during business or class hours. Many companies block pornography, entertainment, sports, and other clearly non-work related sites during business hours, but then unblock them after hours. This might improve workplace efficiency in some situations (or it might just offend the employees). This ACL type allows you to enter days of the week and a time range, or select all hours of the selected days.

Day-abbrevs:	
S	Sunday
M	Monday
T	Tuesday
W	Wednesday
H	Thursday
F	Friday
A	Saturday
Time:	
h1:m1 must be less than h2:m2	

Arguments

aclname	Access list name
day-abbrevs	Days list (as shown above)
h1:m1-h2:m2	from and to time

Example(s)

acl acltime time M 9:00-17:00 - acltime refers day of Monday and time from 9:00 to 17:00.

ACL TYPE	url_regex
Description	Matches using a regular expression on the complete URL
Build Option	Default
Usage	**acl** aclname **url_regex [-i]** pattern

Synopsis

This ACL can be used to provide access control based on parts of the URL or a case insensitive match of the URL, and much more. The regular expressions used in Squid are provided by the GNU Regex library which is documented in the section 7 and 3 regex manpages. Regular expressions are also discussed briefly in a nice article by Guido Socher at LinuxFocus.

Arguments

aclname	Access list name
pattern	Matching to be done for

Example(s)

acl reject url_regex -i ^http://www.google.co.in/index.html
reject contains the entire url http://www.google.co.in/index.html. If in the http_access if you deny reject, it will deny only the url http://www.google.co.in/index.html but allows the url http://www.google.co.in

ACL TYPE	urlpath_regex

Description	This ACL matches on the URL path minus any protocol, port, and host name information
Build Option	Default
Usage	**acl** aclname **urlpath_regex [-i]** pattern

Synopsis
This acl type expects for the particular pattern alone from the given URLs. If found the words, it will match it.

Arguments

aclname	Access list name
pattern	Pattern which is expected for match

Example(s)

acl reject url_regex -i index.html
reject contains the pattern index.html. If in the http_access if you deny reject, then for any url containing the pattern index.html will be denied.

ACL TYPE	port
Description	This ACL matches on the destination port for the request
Build Option	Default
Usage	**acl** aclname **port** port-number **acl** aclname **port** range

Synopsis
Matching done on the destination port.

Arguments

aclname	Access list name
port-number	Destination port number
range	Range of port numbers

Example(s)
The following allows access only to the destination 172.16.1.115:80 from network 172.16.1.0
acl acceleratedhost dst 172.16.1.115/255.255.255.255
acl acceleratedport port 80
acl mynet src 172.16.1.0/255.255.255.0
http_access allow acceleratedhost acceleratedport mynet
http_access deny all

ACL TYPE	myport
Description	This provides match against local socket TCP port
Build Option	Default
Usage	**acl** aclname **myport** port-numbers ...

Synopsis
Matching done on local interface port.

Arguments

aclname	Access list name
port-numbers	Local interface port numbers

Example(s)
acl mp myport 3128
http_access deny 3128
http_access allow all
These will deny all requests coming to the port 3128. Allows requests coming at all other ports.

ACL TYPE	proto
Description	This ACL matches on the protocol of the request
Build Option	Default
Usage	**acl** aclname **proto** protocol

Synopsis
Matching done based on protocol used in the request such as FTP, HTTP, ICP, etc.

Arguments

aclname	Access list name
protocol	Protocol to be matched

Example(s)
acl myproto proto HTTP FTP - This refer protocols HTTP and FTP

ACL TYPE	method
Description	This ACL type matches the HTTP method in the request headers. This includes the methods GET, PUT, etc
Build Option	Default
Usage	**acl** aclname **method** method-type

Synopsis
Matching done based on the HTTP request method like GET, PUT, etc.

Arguments

aclname	Access list name
method-type	HTTP request method

Example(s)
acl getmethod method GET POST - This refers get and post methods only.

ACL TYPE	browser
Description	A regular expression that matches the clients browser type based on the user agent header
Build Option	Default
Usage	**acl** aclname **browser** [-i] regexp

Synopsis
This allows for ACL's to operate based on the browser type in use, for example, using this ACL type, one could create an ACL for Netscape users and another for Internet Explorer users.

Arguments

aclname	Access list name
regexp	Browser name

Example(s)
acl aclname browser MOZILLA
This refers to the requests, which are coming from the browsers who have "MOZILLA" keyword in the user-agent header.

ACL TYPE	ident

Description	Provides string matching on user name
Build Option	Default
Usage	**acl** aclname **ident** username ...

Synopsis

Matching based on iden lookups.

Note: Need an external **ident server** server running on the client machines.

Arguments

aclname	Access list name
username	Username to be matched

Example(s)

You can use ident to allow specific users access to your cache. In your Squid.conf configuration file you would write something like this,
ident_lookup_access allow all
acl friends ident kim lisa frank joe
http_access allow friends
http_access deny all

ACL TYPE	ident_regex
Description	Provides string match on ident output
Build Option	Default
Usage	**acl** aclname **ident_regex [-i]** pattern

Synopsis

Same as ident but matching is done on ident output.

Note: Need an external **ident server** server running on the client machines.

Arguments

aclname	Access list name
pattern	Pattern to be matched

Example(s)

In your Squid.conf configuration file you would write something like this,
ident_lookup-access allow all
acl friends ident_regex joe
This looks for the pattern "joe" in username

ACL TYPE	src_as, dst_as
Description	Source Autonomous System Number is another AS related ACL type, and matches on the AS number of the source of the request Destination Autonomous System Number is the AS number of the server being queried
Build Option	Default
Usage	**acl** aclname **src_as** number **acl** aclname **dst_as** number

Synopsis

The autonomous system number ACL types are generally only used in Cache Peer, or ICP, access restrictions. Autonomous system numbers are used in organizations that have multiple internet links and routers operating under a single administrative authority using the same *gateway protocol*. Routing decisions are then based on knowledge of the AS in addition to other possible data.

Arguments

aclname	Access list name
number	AS numbers to be matched

Example(s)

An example for routing all requests for AS#1241 and only those to mycache.mydomain.net
acl asexample dst_as 1241
cache_peer_access mycache.mydomain.net allow asexample
cache_peer_access mycache_mydomain.net deny all

ACL TYPE	proxy_auth
Description	This ACL type calls an external authenticator process to decide whether the request will be allowed
Build Option	Default
Usage	**acl** aclname **proxy_auth [-i]** username ...

Synopsis

Some of the authenticator helper programs available for Squid are PAM, NCSA, UNIX passwd, SMB, NTLM, etc. Note that authentication cannot work on a transparent proxy or HTTP accelerator. The HTTP protocol does not provide for two authentication stages (one local and one on remote Web sites). So in order to use an authenticator, your proxy must operate as a traditional proxy, where a client will respond appropriately to a proxy authentication request as well as external Web server authentication requests.

Note: *proxy_auth* can't be used in a transparent proxy. It collides with any authentication done by origin servers. It may seem like it works at first, but it doesn't. When a Proxy-Authentication header is sent but it is not needed during ACL checking the username is NOT logged in access.log.

Arguments

aclname	Access list name
username	User name to be authenticated

Example(s)

acl ACLAUTH proxy_auth ramesh senthil muthu
http_access allow ACLAUTH
http_access deny all
The above configuration will allow only ramesh, senthil and muthu if they give valid username and password.

ACL TYPE	proxy_auth_regex
Description	As above, this ACL calls an external authenticator process, but allows regex pattern or case insensitive matches
Build Option	Default
Usage	**acl** aclname **proxy_auth_regex [-i]** pattern

Synopsis

Matching based on regular expressions using external authentication process.

Arguments

aclname	Access list name
pattern	Pattern to be matched

Example(s)

acl ACLAUTH proxy_auth_regex -i mesh

ACL TYPE	snmp_community string ...
Description	Provides matching against community string to limit access to your SNMP Agent
Build Option	Default
Usage	**acl** aclname **snmp_community** string ...

Synopsis
Matching done on snmp community strings.

Arguments

aclname	Access list name
string	String to be matched

Example(s)
acl snmppublic snmp_community public

ACL TYPE	maxconn
Description	Matches when the client's IP address has more than the specified number of HTTP connections established
Build Option	Default
Usage	**acl** aclname **maxconn** number

Synopsis
Matching is true when the defined number of HTTP connections has been established for a client.

Arguments

aclname	Access list name
number	Maximum number of HTTP connections

Example(s)
acl someuser src 1.2.3.4
acl twoconn maxconn 5
http_access deny someuser twoconn
http_access allow !twoconn
The above configuration will allow a maximum of 5 http connection to access.

ACL TYPE	max_user_ip
Description	This will be matched when the same user attempts to log in for more than the specified number of times from different ip addresses
Build Option	Default
Usage	**acl** aclname **max_user_ip** [-s] number

Synopsis
If -s is specified then the limit is strict, denying browsing from any further IP addresses until the ttl has expired. Without -s Squid will just annoy the user by "randomly" denying requests. (the counter is then reset each time the limit is reached and a request is denied). The authenticate_ip_ttl parameter controls the timeout on the ip entries.

Note: 1. Any authentication program must be running for this. NCSA will a simple configuration.
2. In acceleration mode or where there is mesh of child proxies, clients may appear to come from multiple addresses if they are going through proxy farms, so a limit of 1 may cause user problems.

Arguments

aclname	Access list name
number	Number of times to login

Example(s)
authenticate_ip_ttl 2 hours
acl maxuser max_user_ip -s 2

http_access deny maxuser
The same user can log to a maximum of 2 times from two different machines and when he tries to login for the third time from a third system, he will not be allowed to browse.

ACL TYPE	req_mime_type
Description	Match against the mime type of the request generated by the client
Build Option	Default
Usage	**acl** aclname **req_mime_type** mime_type ...

Synopsis
Using this you can detect file upload or some types HTTP tunneling requests.

Arguments

aclname	Access list name
mime_type	Mime type at the requesting side

Example(s)
acl mymime req_mime_type text
This acl looks for the pattern "text" in request mime header.

Note
This does NOT match the reply. You cannot use this to match the returned file type.

ACL TYPE	rep_mime_type
Description	Match against the mime type of the reply received by Squid
Build Option	Default
Usage	**acl** aclname **rep_mime_type** mime_type ...

Synopsis
Also, using this, you can detect file download or some types HTTP tunneling requests.

This has no effect in http_access rules. It only has effect in rules that affect the reply data stream such as http_reply_access.

Arguments

aclname	Access list name
mime_type	Mime type at the receiving side

Example(s)
acl m2 rep_mime_type ^application/pdf$
http_reply_access deny m2
This blocks access to application/pdf mime types.

ACL TYPE	arp
Description	Ethernet (MAC) address matching
Build Option	--enable-arp-acl
Usage	**acl** aclname **arp** mac-address

Synopsis
This option only works for clients on the same local subnet, and only for certain platforms. Linux, Solaris, and some BSD variants are the supported operating systems for this type of ACL. This ACL can provide a somewhat secure method of access control, because MAC addresses are usually harder to spoof than IP addresses, and you can guarantee that your clients are on the local network (otherwise no ARP resolution can take place).

Arguments

aclname	Access list name
mac-address	Physical address to be matched

Example(s)
acl ACLARP arp 11:12:13:14:15:16
ACLARP refers MACADDRESS of the ethernet 11:12:13:14:15:16

Note
Squid can only determine the MAC address for clients that are on the same subnet. If the client is on a different subnet, then Squid cannot find out its MAC address.

ACL TYPE	external
Description	Provides match against external ACL lookup via a helper class defined by the external_acl_type tag
Build Option	Default
Usage	**acl** aclname **external** class_name [arguments...]

Synopsis
Provides match against external ACL lookup via a helper class defined by the external_acl_type tag

Arguments

aclname	Access list name
class_name	Defined by external_acl_type tag

Example(s)
auth_param basic program < put your authenticator here >
auth_param basic children 20
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 1800 seconds
external_acl_type checkip children=20 %LOGIN %SRC /usr/local/Squid/bin/checkip.pl
acl password external checkip
acl it src 172.16.20.1-172.16.20.199/255.255.255.255
http_access allow it password
Allows user if user belongs to a group that is allowed during a given time and using a given ip.

Recommended minimum acl configuration
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80
acl Safe_ports port 21
acl Safe_ports port 443 563
acl Safe_ports port 70
acl Safe_ports port 210
acl Safe_ports port 1025-65535
acl Safe_ports port 280
acl Safe_ports port 488
acl Safe_ports port 591
acl Safe_ports port 777
acl CONNECT method CONNECT

TAG NAME	http_access
Description	Using this, you can allow or deny the access lists defined by acl
Build Option	Default
Usage	**http_access allow|deny [!]** aclname ...
Default	http_access deny all

Synopsis

This is used for filtering based on the acl matchings. If none of the "access" lines cause a match, the default is the opposite of the last line in the list. If the last line was deny, then the default is allow. Conversely, if the last line is allow, the default will be deny. For these reasons, it is a good idea to have an "deny all" or "allow all" entry at the end of your access lists to avoid potential confusion.

Arguments

allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)

To allow http_access for only one machine with MAC Address 00:08:c7:9f:34:41
To restrict access to work hours (9am - 5pm, Monday to Friday) from IP 192.168.2/24
Can i use multitime access control list for different users for different timing
Rules are read from top to bottom

Note

The deny all line is very important. After all the http_access rules, if access isn't denied, it's ALLOWED !! So, specifying a LOT of http_access allow rules, and forget the deny all after them, is the same of NOTHING. If access isn't allowed by one of your rules, the default action (ALLOW) will be triggered. So, don't forget the deny all rule AFTER all the rules.
And, finally, don't forget rules are read from top to bottom. The first rule matched will be used. Other rules won't be applied.

Recommended minimum http_access configuration

```
http_access allow manager localhost
http_access deny manager
http_access deny !Safe_ports
http_access deny CONNECT !SSL_ports
http_access deny all
```

TAG NAME	http_reply_access
Description	This is complementary to http_access which allows or denies clients replies
Build Option	Default
Usage	**http_reply_access allow|deny [!]** aclname ...
Default	http_reply_access allow all

Synopsis

This is used for filtering based on the acl matchings but on the client requests reply. If none of the access lines cause a match, then the opposite of the last line will apply. Thus it is good practice to end the rules with an "allow all" or "deny all" entry.

Arguments

allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)

```
acl reject urlpath_regex i home
http_reply_access deny reject
```

TAG NAME	icp_access

Description	Allowing or Denying access to the ICP port based on defined access lists
Build Option	Default
Usage	**icp_access allow|deny [!]** aclname ...
Default	icp_access deny all

Synopsis
This tag controls icp access on defined access lists.

Arguments
allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)
icp_access allow all
Allows ICP queries from everyone.

TAG NAME	miss_access
Description	Used to force your neighbours to use you as sibling instead of parent
Build Option	Default
Usage	**miss_access allow|deny [!]** aclname ...
Default	miss_access allow all

Synopsis
This tag forces the neighbouring peers to treat you as sibling instead of parent.

Arguments
allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)
acl localclients src 172.16.0.0/16
miss_access allow localclients
miss_access deny !localclients
This means that only your local clients are allowed to fetch MISSES and all other clients can only fetch HITS.

TAG NAME	cache_peer_access
Description	Similar to cache_peer_domain but provides more flexibility by using ACL elements
Build Option	Default
Usage	**cache_peer_access** cache-host **allow|deny** [!]aclname ...
Default	none

Synopsis
The syntax is identical to http_access and the other lists of ACL elements. See http_access for further reference.

Arguments
allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)
The following example could be used, if we want all requests from a specific IP address range to go to a specific cache server (for accounting purposes, for example). Here, all the requests from the 10.0.1.* range are passed to proxy.visolve.com, but all other requests are handled directly.

Using acls to select peers,
acl myNet src 10.0.0.0/255.255.255.0
acl cusNet src 10.0.1.0/255.255.255.0
acl all src 0.0.0.0/0.0.0.0
cache_peer proxy.visolve.com parent 3128 3130

cache_peer_access proxy.visolve.com allow custNet
cache_peer_access proxy.visolve.com deny all

TAG NAME	ident_lookup_access
Description	A list of ACL elements which, if matched, cause an ident (RFC 931) lookup to be performed for this request
Build Option	Default
Usage	**ident_lookup_access allow|deny** aclname
Default	ident_lookup_access deny all

Synopsis

This tag allows or denies ident lookups an matching the access lists.

Note: Only src type ACL checks are fully supported. A src_domain ACL might work at times, but it will not always provide the correct result. This option may be disabled by using --disable-ident-lookups with the configure script.

Arguments

allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)

To enable ident lookups for specific client addresses, you can follow this example,
acl ident_aware_hosts src 198.168.1.0/255.255.255.0
ident_lookup_access allow ident_aware_hosts
ident_lookup_access deny all

TAG NAME	tcp_outgoing_tos
Description	Allows you to select a TOS/Diffserv value to mark outgoing connections with, based on the username or source address making the request
Build Option	Default
Usage	**tcp_outgoing_tos** ds-field [!]aclname ...
Default	none

Synopsis

The TOS/DSCP byte must be exactly that - a byte, value 0 - 255, or "default" to use whatever default your host has. Processing proceeds in the order specified, and stops at first fully matching line.

Arguments

ds-fiels	Outgoing TOS value
aclname	Identifier that contains the list to match on

Example(s)

acl good_service_net src 10.0.1.0/255.255.255.0
tcp_outgoing_tos 0x20 good_service_net
Here, good_service_net uses the TOS value 0x20

TAG NAME	tcp_outgoing_address

Description	Allows you to map requests to different outgoing IP addresses based on the username or source address of the user making the request
Build Option	Default
Usage	**tcp_outgoing_address** ipaddr [[!]aclname] ...
Default	none

Synopsis
Processing proceeds in the order specified, and stops at first fully matching line.

Arguments

ipaddr	Outgoing ip address
aclname	Access lists

Example(s)
acl normal_net src 172.16.1.0/24
tcp_outgoing_address 172.16.1.53 normal_net
Here requests from machines in network 172.16.1.0 will be sent as request from 172.16.1.53 to the origin server.

TAG NAME	reply_body_max_size
Description	This option specifies the maximum size of a reply body
Build Option	Default
Usage	**reply_body_max_size** size [acl acl...]
Default	none

Synopsis
Using this you can prevent users from downloading very large files, such as MP3's and movies.

Note: 1. Downstream caches probably can not detect a partial reply if there is no content-length header, so they will cache partial responses and give them out as hits. You should NOT use this option if you have downstream caches.
2. A maximum size smaller than the size of Squid's error messages will cause an infinite loop and crash Squid. Ensure that the smallest non-zero value you use is greater that the maximum header size plus the size of your largest error page.

Arguments

size	Maximum reply body size
acl	Access lists on which this functions during match

Example(s)
acl site url_regex -i ^http://www.visolve.com
reply_body_max_size 5 KB site
Here, the reply contains content-length. Its size is checked with the specified value. If it is greater then the specified range the an error page is displayed only for this site while access to other sites are allowed.

TAG NAME	log_access
Description	This options allows you to control which requests gets logged to access.log
Build Option	Default
Usage	**log_access allow|deny** acl acl...
Default	none

Synopsis

Sometimes you will not be interested in certain access to be logged in the access.log file. This can be implemented using this tag as follows.

Arguments

allow/deny	Allow or deny on matching the acl
aclname	Access list to be allowed/denied on match

Example(s)

acl google url_regex ^http://www.google.co.in
log_access deny google
access_log /usr/local/Squid3.0pre3/var/logs/access.log common google
This will not log access to http://www.google.co.in into the access.log file.

Example(s)

(1) To allow http_access for only one machine with MAC Address 00:08:c7:9f:34:41

To use MAC address in ACL rules. Configure with option -enable-arp-acl.

```
acl all src 0.0.0.0/0.0.0.0
acl pl800_arp arp 00:08:c7:9f:34:41
http_access allow pl800_arp
http_access deny all
```

(2) To restrict access to work hours (9am - 5pm, Monday to Friday) from IP 192.168.2/24

```
acl ip_acl src 192.168.2.0/24
acl time_acl time M T W H F 9:00-17:00
http_access allow ip_acl time_acl
http_access deny all
```

(3) Can i use multitime access control list for different users for different timing.

Acl Defnitions,

```
acl abc src 172.161.163.85
acl xyz src 172.161.163.86
acl asd src 172.161.163.87
acl morning time 06:00-11:00
acl lunch time 14:00-14:30
acl evening time 16:25-23:59
```

Access Controls,

```
http_access allow abc morning
http_access allow xyz morning lunch
http_access allow asd lunch
```

This is wrong. The description follows:
Here access line "http_access allow xyz morning lunch" will not work. So ACLs are interpreted like this ...

```
http_access RULE statement1 AND statement2 AND statement3 OR
http_access ACTION statement1 AND statement2 AND statement3 OR
.....
```

So, the ACL "http_access allow xyz morning lunch" will never work, as pointed, because at any given time, morning AND lunch will ALWAYS be false, because both morning and lunch will NEVER be true at the same time. As one of them is false, and acl uses AND logical statement, 0/1 AND 0 will always be 0 (false).

That's because this line is in two. If now read,

```
http_access allow xyz AND morning OR
http_access allow xyz lunch
```

If request comes from xyz, and we're in one of the allowed time, one of the rules will match TRUE. The other will obviously match FALSE. TRUE OR FALSE will be TRUE, and access will be permitted.

Finally Access Control looks...

```
http_access allow abc morning
http_access allow xyz morning
http_access allow xyz lunch
http_access allow asd lunch
http_access deny all
```

(4) Rules are read from top to bottom. The first rule matched will be used. Other rules won't be applied.

```
http_access allow xyz morning
http_access deny xyz
http_access allow xyz lunch
```

If xyz tries to access something in the morning, access will be granted. But if he tries to access something at lunchtime, access will be denied. It will be denied by the deny xyz rule, that was matched before the 'xyz lunch' rule.

ADMINISTRATIVE PARAMETERS

The parameters in this section allow the Squid admin to specify, for example, which users and groups have the right to run Squid, what host name should be displayed while displaying errors, which users have the authority to view Cache activity details, etc.

TAG NAME	cache_mgr
Description	This is to specify email address of the administrator of this cache
Build Option	Default
Usage	**cache_mgr** mail_id
Default	cache_mgr webmaster

Synopsis

This is the address which will be added to any error pages that are displayed to clients. Defaults to either webmaster.

Arguments

mail_id	Mail id to be displayed

Example(s)

cache_mgr Squidadmin@visolve.com

TAG NAME	cache_effective_user, cache_effective_group
Description	The user name and group name Squid will operate as
Build Option	Default
Usage	**cache_effective_user** username **cache_effective_group** groupname
Default	cache_effective_user nobody cache_effective_group nogroup

Synopsis

Squid is designed to start as root but very soon after drop to the user/group specified here. This allows you to restrict, for security reasons, the permissions that Squid will have when operating. By default, Squid will operate as either nobody user and the nogroup group.

Note: If these tags are not configured properly, then Squid will have problems while starting.

Arguments

username	Username for Squid process
groupname	Group name for Squid process

Example(s)

cache_effective_user Squid

cache_effective_group Squid

TAG NAME	visible_hostname
Description	The host name that Squid will advertise itself on
Build Option	Default
Usage	**visible_hostname** anyname
Default	none

Synopsis

This effects the host name that Squid uses when serving error messages. This option may need to be configured in cache clusters if you receive IP-Forwarding errors.

Note: If not configured, Squid will not start.

Arguments

anyname	Name of the Squid machine

Example(s)

visible_hostname SYS-CO1

TAG NAME	unique_hostname
Description	Used to set a unique host name for Squid to report in cache clusters in order to allow detection of forwarding loops
Build Option	Default
Usage	**unique_hostname** hostname
Default	none

Synopsis

If you want to have multiple machines with the same visible_hostname then you must give each machine a different *unique_hostname* so that forwarding loops can be detected. In brief, Just set *visible_hostname* to the address the clients connects to, and *unique_hostname* to the externally visible address of each proxy. (address == registered domain name).

Arguments

hostname	Unique name of the Squid machine

Example(s)

unique_hostname www.kovaiteam.com

TAG NAME	hostname_aliases
Description	Used to list of other DNS names that your cache has
Build Option	Default
Usage	**hostname_aliases** name
Default	none

Synopsis

There may be situations where you system or cache have more than one DNS names. In such situations you may specify the DNS names in this tag.

Arguments

name	Alias name

Example(s)

hostname_aliases rose

CACHE REGISTRATION SERVICE

This section contains configurations needed for the (optional) cache announcement service. This service is provided to help cache administrators locate one another in order to join or create cache hierarchies. An *announcement* message is sent (via UDP) to the registration service by Squid. By default, the announcement message is NOT SENT unless you enable it with *announce_period* below.

All current information is processed regularly and made available on the Web at http://www.ircache.net/Cache/Tracker/.

TAG NAME	announce_period
Description	Defines the rate of sending announcement messages
Built Option	Default
Usage	**announce_period** units
Default	announce_period 0

Synopsis

This tag refers to the frequency at which Squid will send announcement messages to the announce host. Defaults to 0 which stops sending announcement messages.

Arguments

units	Announce time period

Example(s)

announce_period 10

TAG NAME	announce_host
Description	Used to define the host address to which Squid will send announcement message to participate in the cache hierarchy
Built Option	Default
Usage	**announce_host** hostname
Default	announce_host tracker.ircache.net

Synopsis

announce_host defines the host for sending announcement messages to get participated in the cache hierarchy.

Arguments

hostname	Host name for announcement message

Example(s)

announce_host cache.ircache.net

TAG NAME	announce_port
Description	Port through which Squid sends announcement message to participate in the cache hierarchy
Built Option	Default
Usage	**announce_port** portnumber
Default	announce_port 3131

Synopsis

announce_port defines the port to send announcement message for participating in the cache hierarchy.

Arguments

portnumber	Port where Squid binds the socket

Example(s)

announce_port 3132

TAG NAME	announce_file

Description	Defines the file whose contents to be sent along with the announcements
Built Option	Default
Usage	**announce_file** filename
Default	-

Synopsis
announce_file contains message to be sent with announcements.

Arguments
filename	File whose content to be sent

Example(s)
announce_file /usr/local/file1

MISCELLANEOUS

This section covers configurations that could not be explicitly bundled in with any of the previous categories. Examples of features covered here are limiting the growth of log files, displaying customized information to clients upon error conditions or access denial, defining memory pools for Squid, network management by enabling SNMP, co-ordination with neighbor caches by enabling WCCP, directing the requests either to the origin server or to the neighbor cache, etc.

TAG NAME	dns_testnames
Description	This points to a number of hosts that Squid can use to test if DNS service is working properly on your network
Build Option	Default
Usage	**dns_testnames** url
Default	dns_testnames netscape.com internic.net nlanr.net microsoft.com

Synopsis
If DNS isn't working properly, Squid will not be able to service requests, so it will refuse to start, with a brief message regarding why in the cache.log. It is recommended that you select two or more host names on the internet and one or two host names on your intranet, assuming you have one and Squid is expected to service it. By default, the *dns_testnames* directive checks a few well known and popular sites: netscape.com, internic.net, nlanr.net, and microsoft.com.

Arguments
url	Sites on which DNS test to be done

Example(s)
dns_testnames visolve.com

TAG NAME	logfile_rotate
Description	Used to specify the number of old rotated log files Squid will keep
Build Option	Default
Usage	**logfile_rotate** number
Default	logfile_rotate 10

Synopsis
The value in this tag defines number of rotated log files to be generated. This defaults to 10, which means Squid will keep 10 old log files before overwriting the oldest. **Squid -k rotate** is the command line usage to implement this function.

Arguments
number	Number of rotations

Example(s)
logfile_rotate 5

TAG NAME	append_domain
Description	The domain that Squid will append to requests that are not possibly fully qualified domain names (more precisely, those that have no dots in them)
Build Option	Default
Usage	**append_domain** domainname
Default	none

Synopsis
Using this tag, you can append the domain names to the requests that are not fully qualified domains names.

Note: append_domain must begin with a period.

Arguments

dimainname	Domain name to be appended

Example(s)
append_domain .cgi.com

TAG NAME	tcp_recv_bufsize
Description	Defines the size of the buffer used for TCP packets being received
Build Option	Default
Usage	**tcp_recv_bufsize** bytes
Default	tcp_recv_bufsize 0 bytes

Synopsis
When defined to a non-zero value, this determines the TCP packets receiving buffer size. By default Squid uses whatever the default buffer size for your operating system is. This is done by setting its value to zero.

Arguments

bytes	Buffer size

Example(s)
tcp_recv_bufsize 500 bytes

TAG NAME	err_html_text
Description	Provides a means to automatically add some extra information to Squid's error pages
Build Option	Default
Usage	**err_htmal_text** text
Default	none

Synopsis
You can add HTML or plain text comments or links here, which will be added to the error messages displayed to clients. To include this in your error messages, you must rewrite the error template files (found in the "errors" directory). Wherever you want the *err_html_text* line to appear, insert a %L tag in the error template file.

Arguments

text	Message to be displayed

Example(s)
err_html_text squid@visolve.com
Consider you want to display this mail Id when access denied error occurs, then edit the corresponding file (ERR_ACCESS_DENIED in '$prefix/etc/errors' directory) with %L where this mail Id should be displayed.

TAG NAME	email_err_data

Description	If enabled, information about the occurred error will be included in the mailto links of the ERR pages (if %W is set) so that the email body contains the data
Build Option	Default
Usage	**email_err_data on|off**
Default	email_err_data on

Synopsis

Enabling this feature, information about the occurred error will be included in the mailto links of the ERR pages
Syntax is <A HREF="mailto:%w%W">%w</A>

Arguments

on/off	Enable or disable

TAG NAME	deny_info
Description	Used to define a customized error page for the requests which gets denied by http_access rules
Build Option	Default
Usage	**deny_info** err_page_name acl **deny_info** link acl
Default	none

Synopsis

You might have defied certain rules which filters access to certain domains. While accessing those domains, Squid normally displays a default error page. Using this tag, we can define a customized error page.

Arguments

err_page_name	Customized error page to be displayed
acl	acl for which the page to be displayed
link	Link to be displayed on deny access

Example(s)

acl test1 urlpath_regex -i .index.html
http_access deny test1
deny_info http://www.google.co.in test1
On satisfying http_access, instead of the default error page, the site google will be loaded.

TAG NAME	memory_pools
Description	This allows Squid to keep memory that it has allocated (but no longer needs), so that it will not need to reallocate memory in the future
Build Option	Default
Usage	**memory_pools on|off**
Default	memory_pools on

Synopsis

Memory pools can improve performance to a small margin by allocating memory, but may need to be turned off if memory is at a premium on your system. This option defaults to on.

Arguments

on/off	Enable or disable memory pool feature

TAG NAME	memory_pools_limit

Description	The amount of memory Squid will keep allocated, assuming the Keep memory for future use option is turned on
Build Option	Default
Usage	**memory_pools_limit** bytes
Default	none

Synopsis

Any non-zero value to this tag will instruct Squid not to keep more than that amount allocated, and if Squid requires more memory than that to fulfill a request, it will use your system's malloc library. Squid does not pre-allocate memory, so it is safe to set this reasonably high. If your Squid runs on a dedicated host, it is probably wisest to leave it to its default of unlimited. If it must share the system with other server processes (like Apache or Sendmail) then it might be appropriate to limit it somewhat.

Arguments

bytes	Memory pool limit size

Example(s)

memory_pools_limit 50 MB

TAG NAME	via
Description	Enable/disable via header
Build Option	Default
Usage	**via on|off**
Default	via on

Synopsis

If set (default), Squid will include a Via header in requests and replies as required by RFC2616.

Arguments

on/off	Enable/disable via header

TAG NAME	forwarded_for
Description	This option allows you to choose whether Squid will report the host name of the system that originally made the request to the origin server
Build Option	Default
Usage	**forwarded_for on|off**
Default	forwarded_for on

Synopsis

If set, Squid will include your system's IP address or name in the HTTP requests it forwards. By default it looks like this: **X-Forwarded-For: 192.1.2.3**

If you disable this, it will appear as **X-Forwarded-For: unknown**

Arguments

on/off	Enable or disable

TAG NAME	log_icp_queries

Description	Dictates whether Squid will log ICP requests
Build Option	Default
Usage	**log_icp_queries on|off**
Default	log_icp_queries on

Synopsis
When you fell if ICP loads are very high, you can disable it otherwise you can enable for logging ICP requests.

Arguments
on/off	Enable or disable logging ICP queries

TAG NAME	icp_hit_stale
Description	Enable/disable to return ICP_HIT for stale cache objects
Build Option	Default
Usage	**icp_hit_stale on|off**
Default	icp_hit_stale off

Synopsis
If you want to return ICP_HIT for stale cache objects, set this option to 'on'. If you have sibling relationships with caches in other administrative domains, this should be 'off'. If you only have sibling relationships with caches under your control, then it is probably okay to set this to 'on'. If set to 'on', then your siblings should use the option "allow-miss" on their cache_peer lines for connecting to you.

Arguments
on/off	Enable or disable

TAG NAME	minimum_direct_hops
Description	Define minimum number of direct hops after which it directs Squid to do direct fetches
Build Option	Default
Usage	**minimum_direct_hops** number
Default	minimum_direct_hops 4

Synopsis
When using ICMP pinging features of Squid to determine distance to peers and origin servers, this configures when Squid should prefer going direct over a peer. This parameter plays a role in deciding latency.

Arguments
number	Number of hops

Example(s)
minimum_direct_hops 10

TAG NAME	minimum_direct_rtt
Description	Defines minimum rtt after which it directs Squid to do direct fetches
Build Option	Default
Usage	**minimum_direct_rtt** timeunits
Default	minimum_direct_rtt 400

Synopsis
If using the ICMP pinging stuff, do direct fetches for sites which are no more than this many rtt milliseconds away.

Arguments
timeunits	Round Trip Time

Example(s)
minimum_direct_rtt 200

TAG NAME	cachemgr_passwd
Description	Specify passwords for cachemgr operations
Build Option	Default
Usage	**cachemgr_passwd** password action action ...
Default	none

Synopsis

By using this we can have secured administration over the Squid.

Actions:

5min, 60min, asndb, authenticator, cbdata, client_list, comm_incoming, config *, counters, delay, digest_stats, dns, events, filedescriptors, fqdncache, histograms, http_headers, info, io, ipcache, mem, menu, netdb, non_peers, objects, offline_toggle *, pconn, peer_select, redirector, refresh, server_list, shutdown *, store_digest, storedir, utilization, via_headers, vm_objects.

* Indicates actions which will not be performed without a valid password, others can be performed if not listed here.

To disable an action, set the password to "disable". To allow performing an action without a password, set the password to "none".

Use the keyword "all" to set the same password for all actions.

Arguments

password	Password for the action
action	Action as described above

Example(s)

cachemgr_passwd secret shutdown

TAG NAME	store_avg_object_size
Description	Average object size, used to estimate number of objects your cache can hold.
Build Option	Default
Usage	**store_avg_object_size** size(Kbytes)
Default	store_avg_object_size 13 KB

Synopsis

To Estimate the number of objects your cache can hold: NUM_OBJ = cache_swap / store_avg_object_size where, cache_swap is the size of the cache.

Arguments

size	Size of the object

Example(s)

store_avg_object_size 100 KB

TAG NAME	store_objects_per_bucket
Description	Defines the number of objects in each store hash table
Build Option	Default
Usage	**store_objects_per_bucket** number
Default	store_objects_per_bucket 20

Synopsis
Target number of objects per bucket in the store hash table. Lowering this value increases the total number of buckets and also the storage maintenance rate.

Arguments

number	Number of objects

Example(s)
store_objects_per_bucket 50

TAG NAME	client_db
Description	Allows you to choose whether Squid will keep statistics regarding each individual client
Build Option	Default
Usage	**client_db on|off**
Default	client_db on

Synopsis
If you want to disable collecting per-client statistics, then turn off client_db here.

Arguments

on/off	Enable or disable collecting client statistics

TAG NAME	netdb_low, netdb_high
Description	Defines low and high water marks for the ICMP measurement database
Build Option	Default
Usage	**netdb_low** number **netdb_high** number
Default	netdb_low 900 netdb_high 1000

Synopsis
These measurements are counts and not percentage. The defaults are 900 and 1000. When the high water mark is reached, database entries will be deleted until the low mark is reached.

Arguments

number	Number of entries

Example(s)
netdb_low 500
netdb_high 800

TAG NAME	netdb_ping_period
Description	Defines minimum period for measuring a site
Build Option	Default
Usage	**netdb_ping_period** timeunits
Default	netdb_ping_period 5 minutes

Synopsis
When this is defined, there will be at least this much delay between successive pings to the same network. The default is five minutes.

Arguments

timeunits	Time period between successive pings

Example(s)
netdb_ping_period 15 minutes

TAG NAME	query_icmp
Description	Enabling this option, makes Squid to ask your peers to include ICMP data in their ICP replies
Build Option	--enable-icmp
Usage	**query_icmp on|off**
Default	query_icmp off

Synopsis

If your peer has configured Squid (during compilation) with '--enable-icmp' then that peer will send ICMP pings to origin server sites of the URLs it receives. If you enable this option then the ICP replies from that peer will include the ICMP data (if available). Then, when choosing a parent cache, Squid will choose the parent with the minimal RTT to the origin server. When this happens, the hierarchy field of the access.log will be "CLOSEST_PARENT_MISS". This option is off by default.

Arguments

on/off	Enable or disable this option

TAG NAME	test_reachability
Description	When this is 'on', ICP MISS replies will be ICP_MISS_NOFETCH instead of ICP_MISS if the target host is NOT in the ICMP database, or has a zero RTT
Build Option	Default
Usage	**test_reachability on|off**
Default	test_reachability off

Synopsis

If the target host is NOT in the ICMP database, or has a zero RTT, enabling this tag, ICP MISS replies will be ICP_MISS_NOFETCH instead of ICP_MISS.

Arguments

on/off	Enable or disable

TAG NAME	buffered_logs
Description	Buffering and unbuffering can be done while writing cache.log with stdio functions using this tag
Build Option	Default
Usage	**buffered_logs on|off**
Default	buffered_logs off

Synopsis

Buffering it can speed up the writing slightly. By default it will be unbuffered.

Arguments

on/off	Enable or disable buffering

TAG NAME	reload_into_ims
Description	When you enable this option, client no-cache or reload requests will be changed to If-Modified-Since requests
Build Option	Default
Usage	**reload_into_ims on|off**
Default	reload_into_ims off

Synopsis

This tag is used to change clients no-cache or reload requests to IMS(if-modified sequence).

Note: Enabling this feature could make you liable for problems which it causes.

Arguments

on/off	Enable or disable

TAG NAME	always_direct
Description	Here you can use ACL elements to specify requests which should ALWAYS be forwarded directly to origin servers
Build Option	Default
Usage	**always_direct allow|deny [!]**aclname ...
Default	none

Synopsis

Allows you to easily pick which ACL matches will not be cached. Requests that match the selected ACLs will always be answered from the origin server. Example below explains the tag to a clear extent.

Arguments

allow/deny	Allow or deny direct access
aclname	Access list on which this should act

Example(s)

acl local-servers dstdomain my.domain.net
always_direct allow local-servers

To always forward FTP requests directly, use
acl FTP proto FTP
always_direct allow FTP

TAG NAME	never_direct
Description	With *never_direct* you can use ACL elements to specify requests which should NEVER be forwarded directly to origin servers
Build Option	Default
Usage	**never_direct allow|deny [!]** aclname ...
Default	none

Synopsis

never_direct is the opposite of always_direct. By default all requests are not forwarded directly to the origin server.

The following example explains this tag.

Arguments

allow/deny	Deny or allow direct access
aclname	Access list on which this should act

Example(s)

To force the use of a proxy for all requests, except those in your local domain use something like
acl local-servers dstdomain .foo.net
acl all src 0.0.0.0/0.0.0.0
never_direct deny local-servers
never_direct allow all

or if Squid is inside a firewall and there is local intranet
servers inside the firewall then use something like:

acl local-intranet dstdomain .foo.net
acl local-external dstdomain external.foo.net
always_direct deny local-external
always_direct allow local-intranet
never_direct allow all

TAG NAME	header_access

Description	This creates a list of ACLs for each header, allowing you very fine-tuned header mangling
Build Option	Default
Usage	**header_access allow|deny** header_name ...
Default	none

Synopsis

This option replaces the old 'anonymize_headers' and the older 'http_anonymizer' option with something that is much more configurable. This new method creates a list of ACLs for each header, allowing you very fine-tuned header mangling.

You can only specify known headers for the header name. Other headers are reclassified as 'Other'. You can also refer to all the headers with 'All'.

Arguments

allow/deny	Allow or deny access for the specified header name
header_name	Header name

Example(s)

header_access Proxy-Connection allow all

TAG NAME	header_replace
Description	This option allows you to change the contents of headers denied with header_access above
Build Option	Default
Usage	**header_replace** header_name message
Default	none

Synopsis

For headers denied with header_access, this tag allows to replace the content of the header using the message specified This is done by replacing them with some fixed string. This replaces the old fake_user_agent option. By default, headers are removed if denied.

Arguments

header_name	Header for which content to be changed
message	Content to be replaced with the message specified here

Example(s)

header_replace User-Agent Nutscrape/1.0 (CP/M; 8-bit)

TAG NAME	icon_directory
Description	Used to specify the path to icon deirectory
Build Option	Default
Usage	**icon_directory** path/directoryname
Default	icon_directory /usr/local/Squid/share/icons

Synopsis

This tag is used to indicate the icon directory.

Arguments

path/directoryname	Location path and name of the directory

Example(s)

icon_directory /usr/local/icons

TAG NAME	error_directory

Description	Defines path to your own error directory
Build Option	Default
Usage	**error_directory** path/directoryname
Default	error_directory /usr/local/Squid/share/errors/English

Synopsis
Used to specify location and name of the error directory used.

Arguments
path/directoryname	Location path and name of the directory

Example(s)
error_directory /usr/local/error

TAG NAME	maximum_single_addr_tries
Description	This sets the maximum number of connection attempts for a host that has only one address
Build Option	Default
Usage	**maximum_single_addr_tries** number
Default	maximum_single_addr_tries 3

Synopsis
If the host has more number of address (for multiple-address hosts), each address is tried once. The default value is three tries, the (not recommended) maximum is 255 tries.

Note: A warning message will be generated if it is set to a value greater than ten.

Arguments
number	Number of tries

Example(s)
maximum_single_addr_tries 5

TAG NAME	snmp_port
Description	Squid serves statistics and status information via SNMP defined using this tag
Build Option	--enable-snmp
Usage	**snmp_port** port_number
Default	snmp_port 3401

Synopsis
By default it listens to port 3401 on the machine. If you don't wish to use SNMP, set this to "0".

Arguments
port_number	Port where Squid binds the socket

Example(s)
snmp_port 3401

TAG NAME	snmp_access

Description	Access to SNMP port is controlled using this tag
Build Option	--enable-snmp
Usage	**snmp_access allow|deny [!]**aclname ...
Default	snmp_port 3401

Synopsis
All access to the agent is denied by default.

Arguments
allow/deny	Allow or deny access
aclname	Access list this should act on

Example(s)
snmp_access allow snmppublic localhost
snmp_access deny all

TAG NAME	snmp_incoming_address, snmp_outgoing_address
Description	Defines the interface for snmp incoming and outgoing requests
Build Option	--enable-snmp
Usage	**snmp_incoming_address** ip_address **snmp_outgoing_address** ip_address
Default	snmp_incoming_address 0.0.0.0 snmp_outgoing_address 255.255.255.255

Synopsis
The default *snmp_incoming_address* (0.0.0.0) is to listen on all available network interfaces. If *snmp_outgoing_address* is set to 255.255.255.255 (the default) then it will use the same socket as *snmp_incoming_address*. Only change this if you want to have SNMP replies sent using another address than where this Squid listens for SNMP queries.

Note: *snmp_incoming_address* and *snmp_outgoing_address* can not have the same value since they both use port 3401.

Arguments
ip_address	Incoming and outgoing interface address

Example(s)
snmp_incoming_address 172.16.1.35
snmp_outgoing_address 172.16.1.36

TAG NAME	as_whois_server
Description	This is to query AS numbers
Build Option	Default
Usage	**as_whois_server** server_name
Default	as_whois_server whois.ra.net

Synopsis
AS numbers are queried only when Squid starts up, not for every request.

Arguments
server_name	Server name for which AS numbers to be queried

Example(s)
as_whois_server ra.net

TAG NAME	wccp_router

Description	To define your WCCP "home" router for Squid
Build Option	Default
Usage	**wccp_router** ip_address
Default	wccp_router 0.0.0.0

Synopsis
Setting the *wccp_router* to 0.0.0.0 (the default) disables WCCP.

Arguments
ip_address	ip address of the router

Example(s)
wccp_router 172.16.1.100

TAG NAME	wccp_version
Description	Used to specify the version of Cisco IOS used in the Router
Build Option	Default
Usage	**wccp_version** version_number
Default	wccp_version 4

Synopsis
According to some users, Cisco IOS 11.2 only supports WCCP version 3. If you're using that version of IOS, change this value to 3.

Arguments
version_number	IOS version number

Example(s)
wccp_router 172.16.1.100

TAG NAME	wccp_incoming_address, wccp_outgoing_address
Description	Defines the interface through which WCCP requests will be sent and received
Build Option	Default
Usage	**wccp_incoming_address** ip_ddress **wccp_outgoing_address** ip_ddress
Default	wccp_incoming_address 0.0.0.0 wccp_outgoing_address 255.255.255.255

Synopsis
wccp_incoming_address - Use this option if you require WCCP messages to be received on only one interface. Do NOT use this option if you're unsure
how many interfaces you have, or if you know you have only one interface.
wccp_outgoing_address - Use this option if you require WCCP messages to be sent out on only one interface. Do NOT use this option if you're unsure
how many interfaces you have, or if you know you have only one interface.

The default behavior is to not bind to any specific address.

Arguments
ip_address	Incoming and outgoing ip_address

Example(s)
wccp_incoming_address 172.16.1.36
wccp_outgoing_address 172.16.1.35

Note
*wccp_incoming_address*and *wccp_outgoing_address* can not have the same value since they both use port 2048.

DELAY POOL PARAMETERS

Conceptually, delay pools are bandwidth limiters - "pools" of bandwidth that drain out as people browse the Web, and fill up at the rate specified - this can be thought of as a leaky bucket that is continually being filled. This is useful when bandwidth charges are in place, if we want to reduce bandwidth usage for web traffic.

Delay Pools can do wonders when combined with ACLs. These tags permit us to limit the bandwidth of certain requests, based on any criteria. Delay behavior is selected by ACLs (low and high priority traffic, staff Vs students or student Vs authenticated student or so on). In ISPs, delay pools can be implemented in a particular network to improve the quality of service. To enable this, Squid needs to be configured with the --enable-delay-pools option.

TAG NAME	delay_pools
Description	Used to specify number of delay pools
Built Option	--enable-delay-pools
Usage	**delay_pools** number
Default	delay_pools 0

Synopsis

This represents the number of delay pools to be used. For example, if you have one class 2 delay pool and one class 3 delays pool, you have a total of 2 delay pools.

Arguments

number	Number of delay pools

Example(s)

delay_pools 5

TAG NAME	delay_class
Description	This defines the class of each delay pool
Built Option	--enable-delay-pools
Usage	**delay_class** pool-number class-number
Default	none

Synopsis

Class of the delay pool used is defined using this tag. There must be exactly one *delay_class* line for each delay pool. There are five categories of delay classes.

class 1	Everything is limited by a single aggregate bucket.
class 2	Everything is limited by a single aggregate bucket as well as an "individual" bucket chosen from bits 25 through 32 of the IP address.
class 3	Everything is limited by a single aggregate bucket as well as a "network" bucket chosen from bits 17 through 24 of the IP address and a "individual" bucket chosen from bits 17 through 32 of the IP address.
class 4	Everything in a class 3 delay pool, with an additional limit on a per user basis. This only takes effect if the username is established in advance - by forcing authentication in your http_access rules.
class 5	Requests are grouped according their tag (see external_acl_type tag= reply).

If an IP address is a.b.c.d

-> bits 25 through 32 are "d"
-> bits 17 through 24 are "c"
-> bits 17 through 32 are "c * 256 + d"

Arguments

pool-number	Delay pool number
class-number	Delay class number

Example(s)

delay_pools 2
delay_class 1 2   (pool 1 is a class 2 pool)

delay_class 2 3 (pool 2 is a class 3 pool)

TAG NAME	delay_access
Description	This is used to determine which delay pool a request falls into
Built Option	--enable-delay-pools
Usage	**delay_access** delay_pool **allow/deny** domainname
Default	none

Synopsis

The first matched delay pool is always used, i.e., if a request falls into delay pool number one, no more delay are checked, otherwise the rest are checked
in order of their delay pool number until they have all been checked.

Arguments

delay_pool	Delay pool number
allow/deny	Allow or deny access
domainname	Domain name on which this should act

Example(s)

If you want some_big_clients in delay pool 1 and lotsa_little_clients in delay pool 2:

```
delay_access 1 allow some_big_clients
delay_access 1 deny all
delay_access 2 allow lotsa_little_clients
delay_access 2 deny all
delay_access 3 allow authenticated_clients
```

TAG NAME	delay_parameters
Description	Defines the parameters for a delay pool
Built Option	--enable-delay-pools
Usage	**delay_parameters** pool aggregate (In general). For detailed format refer usage syntax bellow
Default	none

Synopsis

Using this tag, delay parameters for each each delay pool has a number of "buckets" associated with it, as explained in the description of delay_class.

Usage syntax for each class:	
class 1	delay_parameters pool aggregate
class 2	delay_parameters pool aggregate individual
class 3	delay_parameters pool aggregate network individual
class 4	delay_parameters pool aggregate network individual user
class 5	delay_parameters pool tag

A pair of delay parameters is written restore/maximum, where restore is the number of bytes (not bits - modem and network speeds are usually quoted in
bits) per second placed into the bucket, and maximum is the maximum number of bytes which can be in the bucket at any time.

Arguments

pool	Delay pool number - ie, a number between 1 and the number specified in delay_pools as used in delay_class lines.
aggregate	the "delay parameters" for the aggregate bucket (class 1, 2, 3).
individual	the "delay parameters" for the network buckets (class 3).
user	user on which this condition is applied
tag	the delay parameters for the tag buckets (class 5).

Example(s)

If delay pool number 1 is a class 2 delay pool is being used to strictly limit each host to 64kbps (plus overheads), with no overall limit, the

usage is,
delay_parameters 1 -1/-1 8000/8000

For a class 4 delay pool, each user will be limited to 128 Kbs no matter how many workstations they are logged into:
delay_parameters 4 32000/32000 8000/8000 600/64000 16000/16000

TAG NAME	delay_initial_bucket_level
Description	Used to determine how much data is put in each bucket when Squid starts, is reconfigured, or first notices a host accessing it
Built Option	--enable-delay-pools
Usage	**delay_initial_bucket_level** percent(0-100)
Default	delay_initial_bucket_level 50

Synopsis

The initial bucket percentage is used to determine how much is put in each bucket when Squid starts, is reconfigured, or first notices a host accessing it. In class 2 and class 3, individual hosts and networks only have buckets associated with them once they have been "seen" by Squid

Arguments

percent	Initial bucket level in percentage

Example(s)

delay_initial_bucket_level 20

TAG NAME	incoming_icp_average, incoming_http_average, incoming_dns_average, min_icp_poll_cnt, min_dns_poll_cnt, min_http_poll_cnt
Description	Using these tags, average number of ICP, HTTP requests, their polling rates can be specified
Built Option	--enable-delay-pools
Usage	**Tagname** number
Default	incoming_icp_average 6 incoming_http_average 4 incoming_dns_average 4 min_icp_poll_cnt 8 min_dns_poll_cnt 8 min_http_poll_cnt 8

Synopsis

INCOMING sockets are the ICP and HTTP ports. Squid need to check these fairly regularly, but how often? When the load increases, Squid want to check the incoming sockets more often. If Squid have a lot of incoming ICP, then it needs to check these sockets more than if we just have HTTP. These change of algorithms by Squid are decided by these tags.

Arguments

Number	Number to change the algorithm used by Squid

Example(s)

incoming_icp_average 3
incoming_http_average 2
incoming_dns_average 3
min_icp_poll_cnt 8
min_dns_poll_cnt 6
min_http_poll_cnt 6

TAG NAME	max_open_disk_fds

Description	Defines number of file descriptors to be handled directly
Built Option	Default
Usage	**max_open_disk_fds** number
Default	max_open_disk_fds 0

Synopsis

To avoid having disk as the I/O bottleneck Squid can optionally bypass the on-disk cache if more than this amount of disk file descriptors are open.
A value of 0 indicates no limit.

Arguments

number	Maximum number of file descriptors

Example(s)

max_open_disk_fds 5

TAG NAME	offline_mode
Description	When enabled, Squid will never try to validate cached objects.
Built Option	Default
Usage	**offline_mode on|off**
Default	offline_mode off

Synopsis

offline_mode gives access to more cached information than the proposed feature would allow (stale cached versions, where the origin server should have been contacted).

Arguments

on/off	Enable or disable *offline_mode* feature

TAG NAME	uri_whitespace
Description	Used to specify the action of Squid when the requests that have whitespace characters in the URI
Built Option	Default
Usage	**uri_whitespace** action
Default	uri_whitespace strip

Synopsis

When the requested URL's contains whitespaces, them this tag is used to specify the action of Squid on that URL's. Actions are shown in the table below.

Actions:	
strip	The whitespace characters are stripped out of the URL. This is the behavior recommended by RFC2396.
deny	The request is denied. The user receives an "Invalid Request" message.
allow	The request is allowed and the URI is not changed. The whitespace characters remain in the URI. Note the whitespace is passed to redirector processes if they are in use.
encode	The request is allowed and the whitespace characters are encoded according to RFC1738. This could be considered a violation of the HTTP/1.1 RFC because proxies are not allowed to rewrite URI's.
chop	The request is allowed and the URI is chopped at the first whitespace. This might also be considered a violation.

Arguments

acion	Action of Squid on identifying the white spaces

Example(s)

uri_whitespace deny

TAG NAME	broken_posts

Description	A list of ACL elements which, if matched, causes Squid to send an extra CRLF pair after the body of a PUT/POST request
Built Option	Default
Usage	**broken_posts allow|deny** aclname ...
Default	none

Synopsis

Squid will send an extra CLRF pair after the body of a PUT/POST request for the access list specified is matched. Some HTTP servers has broken implementations of PUT/POST, and rely on an extra CRLF pair sent by some WWW clients.

Arguments

allow/deny	Allow or deny access list
aclname	Access list name

Example(s)

acl buggy_server url_regex ^http://....
broken_posts allow buggy_server

TAG NAME	mcast_miss_addr
Description	When enabled,every "cache miss" URL will be sent out on the specified multicast address
Built Option	-DMULTICAST_MISS_STREAM
Usage	**mcast_miss_addr** ip_address
Default	mcast_miss_addr 255.255.255.255

Synopsis

You will be needing the "cache miss" URL to be sent on a specified multicast address. This tag provides the option.

Note: Do not enable this option unless you are are absolutely certain you understand what you are doing.

Arguments

ip_address	ip address through which the URL to be sent

Example(s)

mcast_miss_addr 172.16.1.255

TAG NAME	mcast_miss_ttl
Description	Defines time-to-live value for packets multicasted when multicasting off cache miss URLs is enabled
Built Option	-DMULTICAST_MISS_TTL
Usage	**mcast_miss_ttl** time-units
Default	mcast_miss_ttl 16

Synopsis

The value specified in this tag specifies the time-to-live period for packets multicated when multicasting off cache miss URLs is enabled. By default this is set to 'site scope', i.e. 16.

Arguments

time-units	Time to Live period

Example(s)

mcast_miss_ttl 10

TAG NAME	mcast_miss_port

Description	Used to define the port number to be used in conjunction with mcast_miss_addr.
Built Option	-DMULTICAST_MISS_STREAM
Usage	**mcast_miss_port** portnumber
Default	mcast_miss_port 3135

Synopsis
Port to be used for *mcast_miss_addr*.

Note: This tag is used only when you enable *mcast_miss_addr*.

Arguments
portnumber	Port number on which Squid binds the socket

Example(s)
mcast_miss_port 3100

TAG NAME	mcast_miss_encode_key
Description	This is the encryption key used in the multicast miss stream
Built Option	-DMULTICAST_MISS_STREAM
Usage	**mcast_miss_encode_key** key
Default	mcast_miss_encode_key XXXXXXXXXXXXXXXX

Synopsis
The URLs that are sent in the multicast miss stream are encrypted. This is the encryption key.

Arguments
key	Encription key to be used

TAG NAME	nonhierarchical_direct
Description	Enable/disable Squid to send non-hierarchial requests to parents
Built Option	Default
Usage	**nonhierarchical_direct on|off**
Default	nonhierarchical_direct on

Synopsis
By default, Squid will send any non-hierarchical requests (matching hierarchy_stoplist or not cacheable request type) direct to origin servers. If you set this to off, then Squid will prefer to send these requests to parents. Note that in most configurations, by turning this off you will only add latency to this request without any improvement in global hit ratio. If you are inside a firewall then see never_direct instead of this directive.

Arguments
on/off	Enable or disable sending non-hierarchal requests

TAG NAME	prefer_direct
Description	For enabling Squid to use parent if direct going is failed
Built Option	Default
Usage	**prefer_direct on|off**
Default	prefer_direct off

Synopsis
Normally Squid tries to use parents for most requests. If you by some reason like it to first try going direct and only use a parent if going direct fails then set this to on.

By combining *nonhierarchical_direct off* and *prefer_direct on* you can set up Squid to use a parent as a backup path if going direct fails.

Arguments
on/off	Enable or disable *preferer_direct* option

TAG NAME	strip_query_terms
Description	For tripping query items before logging
Built Option	Default
Usage	**strip_query_terms on|off**
Default	strip_query_terms on

Synopsis

Squid by default does not log query parameters. These parameters are however forwarded to the server verbatim. If we want to enable logging of query parameters, the *strip_query_terms* directive can be used.

By default, Squid strips query terms from requested URLs before logging. This protects your user's privacy

Arguments

on/off	Enable or disable query parameters from logging

TAG NAME	coredump_dir
Description	Squid leaves core files in the directory specified
Built Option	Default
Usage	**coredump_dir** directory
Default	coredump_dir none

Synopsis

By default Squid leaves core files in the directory from where it was started. If you set *coredump_dir* to a directory that exists, Squid will chdir() to that directory at startup and coredump files will be left there.

Arguments

directory	Directory for used for core dump

Example(s)

coredump_dir /usr/local

TAG NAME	redirector_bypass
Description	Used for bypassing the request
Built Option	Default
Usage	**redirector_bypass on|off**
Default	redirector_bypass off

Synopsis

When this is 'on', a request will not go through the redirector if all redirectors are busy. If this is 'off' and the redirector queue grows too large, Squid will exit with a FATAL error and ask you to increase the number of redirectors. You should only enable this if the redirectors are not critical to your caching system. If you use redirectors for access control, and you enable this option, then users may have access to pages that they should not be allowed to request.

Arguments

on/off	Enable or disable *redirector_bypass*

TAG NAME	ignore_unknown_nameservers

Description	Enable or disable responses from unknown nameservers
Built Option	Default
Usage	**ignore_unknown_nameservers on|off**
Default	ignore_unknown_nameservers on

Synopsis
By default Squid checks that DNS responses are received from the same IP addresses that they are sent to. If they don't match, Squid ignores the response and writes a warning message to cache.log. You can allow responses from unknown nameservers by setting this option to 'off'.

Arguments
on/off	Enable or disable

TAG NAME	digest_generation
Description	This controls whether the server will generate a Cache Digest of its contents
Built Option	--enable-cache-digests
Usage	**digest_generation on|off**
Default	digest_generation on

Synopsis
This tag enables or disable the server generating a cache digest of its contents. By default, Cache Digest generation is enabled if Squid is compiled with USE_CACHE_DIGESTS defined.

Arguments
on/off	Enable or disable the server generating a cache digest of its contents

TAG NAME	digest_bits_per_entry
Description	Defines number of bits of server's cache digest to be associated with the digest entry
Built Option	--enable-cache-digests
Usage	**digest_bits_per_entry** number
Default	digest_bits_per_entry 5

Synopsis
This is the number of bits of the server's Cache Digest which will be associated with the Digest entry for a given HTTP Method and URL (public key) combination.

Arguments
number	Number of bits per entry

Example(s)
digest_bits_per_entry 5

TAG NAME	digest_rebuild_period
Description	This is the number of seconds between Cache Digest rebuilds
Built Option	--enable-cache-digests
Usage	**digest_rebuild_period** time(seconds)
Default	digest_rebuild_period 1 hour

Synopsis
This tag defines the time period between successive cache digest rebuilds.

Arguments
time	Time period between rebuilds

Example(s)
digest_rebuild_period 2 hour

TAG NAME	digest_rewrite_period
Description	This is the number of seconds between Cache Digest writes to disk.
Built Option	--enable-cache-digests
Usage	**digest_rewrite_period** time(seconds)
Default	digest_rewrite_period 1 hour

Synopsis
This tag specifies the time period between successive writing to disk by cache digest .

Arguments

time	Time period between successive writes

Example(s)
digest_rewrite_period 2 hour

TAG NAME	digest_swapout_chunk_size
Description	This is the number of bytes of the Cache Digest to write to disk at a time
Built Option	--enable-cache-digests
Usage	**digest_swapout_chunk_size** bytes
Default	digest_swapout_chunk_size 4096 bytes

Synopsis
Using this tag, total number of bytes to be written to the disk at a time by the cache digest is specified.

Arguments

bytes	Total number of bytes to be written to the disk in single time

Example(s)
digest_swapout_chunk_size 2048 bytes

TAG NAME	digest_rebuild_chunk_percentage
Description	This specifies the percentage of the Cache Digest to be scanned at a time
Built Option	--enable-cache-digests
Usage	**digest_rebuild_chunk_percentage** percent(0-100)
Default	digest_rebuild_chunk_percentage 10

Synopsis
Using this tag, we can specify the percentage of the cache disgest to be scanned at a time.

Arguments

percent	Percentage of cache digest to be scanned at a time

Example(s)
digest_rebuild_chunk_percentage 20

TAG NAME	chroot

Description	Use this to have Squid do a chroot() while initializing
Built Option	Default
Usage	**chroot**
Default	none

Synopsis
Squid by default does not fully drop root privileges because it may be required during reconfigure.So use this directive to have Squid do a chroot() while initializing. This also causes Squid to fully drop root privileges after initializing . Squid only drops all root privilegies when chroot_dir is used. Without chroot_dir it runs as root with effective user nobody. This means, for example, that if you use a HTTP port less than 1024 and try to reconfigure, you will get an error.

Example(s)
chroot

TAG NAME	client_persistent_connections, server_persistent_connections
Description	Enable/disable persistent connection support for clients and servers
Built Option	Default
Usage	**client_persistent_connections on|off** **server_persistent_connections on|off**
Default	client_persistent_connections on server_persistent_connections on

Synopsis
By default, Squid uses persistent connections (when allowed) with its clients and servers. You can use these options to disable persistent connections with clients and/or servers.

Arguments

on/off	Enable or disable persistent connections

TAG NAME	pipeline_prefetch
Description	Used to boost the performance of pipelined requests to closer match that of a non-proxied environment
Built Option	Default
Usage	**pipeline_prefetch on|off**
Default	pipeline_prefetch off

Synopsis
Squid can try to fetch up to two requests in parallell from a pipeline. Defaults to off for bandwidth management and access logging reasons.

Arguments

on/off	Enable or disable pipeline prefetch

TAG NAME	extension_methods

Description	You can add up to 20 additional request "extension" methods here for enabling Squid to allow access unknown methods
Built Option	Default
Usage	**extension_methods** methods
Default	none

Synopsis
Squid only knows about standardized HTTP request methods. Unknown methods are denied, unless you add them to this tag.

Arguments
methods	New methods

Example(s)
extension_methods SEARCH

TAG NAME	request_entities
Description	Set this directive to on if you have clients which insists on sending request entities in GET or HEAD requests
Built Option	Default
Usage	**request_entities on|off**
Default	request_entities off

Synopsis
Squid defaults to deny GET and HEAD requests with request entities, as the meaning of such requests are undefined in the HTTP standard even if not explicitly forbidden. Set this directive to on if you have clients which insists on sending request entities in GET or HEAD requests

Arguments
on/off	Enable or disable

TAG NAME	high_response_time_warning
Description	Enables Squid to print a WARNING to get the administrators attention
Built Option	Default
Usage	**high_response_time_warning** time(msec)
Default	high_response_time_warning 0

Synopsis
If the one-minute median response time exceeds this value, Squid prints a WARNING with debug level 0 to get the administrators attention. The value is in milliseconds.

Arguments
time	Time after which warning is printed

Example(s)
high_response_time_warning 20

TAG NAME	high_page_fault_warning

Description	Enables Squid to print a WARNING to get the administrators attention
Built Option	Default
Usage	**high_page_fault_warning** time
Default	high_page_fault_warning 0

Synopsis
If the one-minute average page fault rate exceeds this value, Squid prints a WARNING with debug level 0 to get the administrators attention. The value is in page faults per second.

Arguments
time	Time after which warning is printed

Example(s)
high_page_fault_warning 10

TAG NAME	high_memory_warning
Description	Enables Squid to print a WARNING to get the administrators attention
Built Option	--enable-snmp
Usage	**high_memory_warning** number
Default	high_memory_warning 0

Synopsis
If the memory usage (as determined by mallinfo) exceeds value, Squid prints a WARNING with debug level 0 to get the administrators attention.

Arguments
time	Time after which warning is printed

Example(s)
high_memory_warning 20

TAG NAME	store_dir_select_algorithm
Description	Used to specify the algorithm for store directory selection
Built Option	--enable-snmp
Usage	**store_dir_select_algorithm** algorithm-type
Default	store_dir_select_algorithm least-load

Synopsis
As there are more number of store directories, this tag allos you to specify the algorithm by which Squid will select the store directories.

Arguments
algorithm-type	Algorithm to be used

Example(s)
store_dir_select_algorithm round-robin

TAG NAME	ie_refresh

Description	Turning this on provides a partial fix to the problem with Microsoft Internet Explorer up until version 5.5 Service Pack 1 which has an issue with transparent proxies, wherein it is impossible to force a refresh
Built Option	Default
Usage	**ie_refresh on|off**
Default	ie_refresh off

Synopsis

Turning this on provides a partial fix to the problem, by causing all IMS-REFRESH requests from older IE versions to check the origin server for fresh content. This reduces hit ratio by some amount (~10% in my experience), but allows users to actually get fresh content when they want it. Note that because Squid cannot tell if the user is using 5.5 or 5.5SP1, the behavior of 5.5 is unchanged from old versions of Squid (i.e. a forced refresh is impossible). Newer versions of IE will, hopefully, continue to have the new behavior and will be handled based on that assumption. This option defaults to the old Squid behavior, which is better for hit ratios but worse for clients using IE, if they need to be able to force fresh content.

Arguments

on/off	Enable or disable this feature

TAG NAME	vary_ignore_expire
Description	This option enables Squid to ignore, immediate expiry time with no cache-control header when requested by a HTTP/1.0 client
Built Option	Default
Usage	**vary_ignore_expire on|off**
Default	vary_ignore_expire off

Synopsis

Many HTTP servers supporting Vary gives such objects immediate expiry time with no cache-control header when requested by a HTTP/1.0 client. This tag enables Squid to ignore such expiry times until HTTP/1.1 is fully implemented.

Note: This may eventually cause some varying objects not intended for caching to get cached.

Arguments

on/off	Enable or disable *vary_ignore_expire* feature

TAG NAME	sleep_after_fork
Description	When this is set to a non-zero value, the main Squid process sleeps the specified number of microseconds after a fork() system call
Built Option	Default
Usage	**sleep_after_fork** time(microseconds)
Default	sleep_after_fork 0

Synopsis

This sleep may help the situation where your system reports fork() failures due to lack of (virtual) memory. Note, however, that if you have lot of child processes, then these sleep delays will add up and your Squid will not service requests for some amount of time until all the child processes have been started.

Arguments

time	Sleep time period

Example(s)

sleep_after_fork 20