



Teldat WebCache

User's Guide

Copyright© Teldat-DM904-I Version 1.1, 04/2016 Teldat, S.A.

Legal Notice

Warranty

This publication is subject to change.

Teldat offers no warranty whatsoever for information contained in this manual.

Teldat is not liable for any direct, indirect, collateral, consequential or any other damage connected to the delivery, supply or use of this manual.

Table of Contents

Chapter 1	About This Guide	1
1.1	Supported Devices	1
1.2	Who should read this manual?	1
1.3	When should I read this manual?	1
1.4	What is in this manual?	1
1.5	What is not in this manual?	1
1.6	How is the information organized?	1
1.7	Technical Support	1
1.8	About OpenSource Software	2
Chapter 2	What is a cache and why is it useful?	3
2.1	Teldat WebCache application	3
2.1.1	Advantages	3
2.2	Scenarios	3
2.2.1	Direct Internet connection	3
2.2.2	WAN connection scenario	4
2.2.3	Collaborative scenario	4
Chapter 3	Application configuration	6
3.1	Web configuration	6
3.1.1	General application parameters	7
3.1.2	Access lists: Defining list of users, machines, domains etc.	9
3.1.3	HTTP rules: What are the requests to process by the cache engine?	12
3.1.4	Cache rules: Deciding what to cache	13
3.1.5	Always direct rules: forward request without using peers	14
3.1.6	Delay pools: Limiting the bandwidth usage per user/network consumption	14
3.1.7	Cache peers: Creating a simple hierarchy of neighbor caches	17
3.1.8	DNS servers	21
3.1.9	SSL configuration	21
3.2	Text configuration commands	25
3.2.1	Web Cache configuration	26
Chapter 4	WebCache Dashboard (Management Platform)	39
4.1	Introduction	39
4.2	Dashboard home section	39
4.2.1	Alerts level	40
4.2.2	Global service time	40

4.2.3	Global bandwidth saving	40
4.2.4	Service time distribution	41
4.3	Analysis section	41
4.3.1	Main analysis screen	42
4.3.2	Analyzing data for a single device	43
4.4	Rules section	46
4.5	Filtering	46
4.5.1	Selecting a time interval	47
4.6	Assigning plugin licenses	48
Appendix A	General HTTP Proxy configuration	49
A.1	HTTP proxy web configuration	49
A.1.1	Configuring an HTTP transparent proxy	49
A.1.2	Configuring an HTTP non transparent proxy	49
A.1.3	Configuring an HTTP and HTTPS transparent proxy	50
A.1.4	Configuring an HTTP and HTTPS non transparent proxy	50
A.1.5	Configuring application behind a corporative proxy for caching external traffic	50
A.1.6	Managing HTTP proxy logs	52
A.1.7	Always run the internal HTTP proxy	52
A.2	Text configuration commands	53
A.2.1	HTTP Proxy configuration	53
Appendix B	Certificates help for clients	56
B.1	Installing a certificate in the client's system	56
B.1.1	For Windows	56
B.1.2	For Linux	56
B.2	Import a certificate in the client's browser	56
B.2.1	For Internet Explorer	56
B.2.2	For Firefox	57
Appendix C	Troubleshooting.	58
C.1	Symptom: Your WebCache application is not caching anything	58
C.2	Symptom: The log server is not receiving the browsing data	60
C.3	Symptom: After importing or generating a new certificate you cannot access to HTTPS sites	60

Chapter 1 About This Guide

This is the User Guide for the WebCache application for the Teldat Atlas i6x.

1.1 Supported Devices

The information contained in this installation guide only applies to the Atlas i6x equipped with an internal storage device and with the WebCache application installed.

1.2 Who should read this manual?

This manual should be read by the user who needs to configure a WebCache in an Atlas i6x.

1.3 When should I read this manual?

Read this guide as soon as you are ready to configure your WebCache application. This manual shows several scenarios where WebCache is useful and how to configure its different parameters.

1.4 What is in this manual?

This User Guide contains the following information:

- Scenarios where the application is useful.
- Configuring the application using the Atlas i6x internal web.
- Configuring the application using the Teldat Management Platform.
- Usage and licensing of the Teldat Management Platform WebCache Dashboard plugin.
- Configuring the HTTP Proxy section inside the general configuration for the Atlas i6x Application Host.
- Troubleshooting.

1.5 What is not in this manual?

This user guide does not contain information relative to the Atlas i6x hardware nor is it intended as a comprehensive reference to all management operations available on the Management Platform, the Atlas i6x Application Host software and configuration, or other applications different from WebCache. It does not contain information on how to setup the device to connect to Internet. For information on configuring the device, please see the relevant manuals for the different protocols, which can be found at the following web site: www.teldat.com.

1.6 How is the information organized?

Chapter 1 is an introduction to what a WebCache application is and the different scenarios where it is useful to deploy it. Chapter 2 shows the different configuration methods for this application. Chapter 3 explains the usage of the Dashboard inside the Management Platform and to monitor the different WebCaches installed on your network.

In addition, there are some appendices to explain several additional features relative to WebCache, and troubleshooting.

1.7 Technical Support

Teldat, S.A. offers a technical support service. The device software can be regularly updated for maintenance reasons and for new features.

Contact information:

Web: www.teldat.com

Tel Nº: +34 918 076 565

Fax: +34 918 076 566

Email: support@teldat.com



Note

The manufacturer reserves the right to make changes and improvements in the appropriate features to either the software or hardware of this product, modifying the specifications of this manual without prior notice. The screen captures shown throughout the guide are provided as information guidelines only. Some small modifications may exist in the current software.

1.8 About OpenSource Software

Some software components of this product contain copyright software, which is licensed under the GPL, GFDL, LGPL and other open source licenses. You may obtain the complete corresponding source code from us for a period of three years, after our last shipment of said product, by downloading this free of charge from Teldat, S.A.. If you want to obtain the complete corresponding source code in a physical medium such as a CD-ROM, the cost of physically performing source distribution may be charged. This offer is valid to anyone who has this information.

For more information on the licenses for the installed software in the Application Host of an Atlas i6x, please see the *About section* on the device web configurator.

Chapter 2 What is a cache and why is it useful?

2.1 Teldat WebCache application

This well-known feature intercepts Internet traffic, which users on the enterprise branch office LAN are downloading, and stores a copy locally. This means the next user requesting the same information will receive it from the locally stored copy instead of downloading it from Internet again. This is a WAN optimization technique that saves bandwidth; the more users accessing the same information, the more bandwidth it saves.

Said application is useful for branch offices where local users frequently access a common set of information that can be cached. For instance, education is a very interesting market, since the students will normally download the same information on a given course: the course materials.

By using the Atlas i6x WebCache application, corporations can save investment on external dedicated web cache appliances at branch offices. An external appliance will have more performance (more storage space and more aggregated bandwidth serving capabilities) but will be more expensive and normally isn't justified (economically), especially for small branch offices.

Teldat proposes an application especially designed for the Atlas i6x device. This user guide explains how to configure the application and how to supervise the results on the Teldat Atlas i6x Management Platform.

2.1.1 Advantages

- *Productivity and performance are increased.* All contents viewed on your browser are stored on the local side so they can be used in successive accesses. Browsing speed is consequently increased, helping to optimize your employees' productivity.
- *WAN traffic is dramatically reduced.* All static browsed contents are stored when they are downloaded on first access. These contents are downloaded on successive accesses from the local storage device instead of being retrieved from the WAN.
- *Use an Internet policy that suits your needs.* It is possible to establish access priorities on your employee network, to assign delay pools, to prioritize the traffic, access lists and HTTP and cache rules.
- *Simplify your proxy configuration.* WebCache application has a simple configuration web based system where it is possible modify, in real time, your system parameters. This allows you to configure and manage said system without needing in-depth knowledge on usual cache applications, using simple access lists per user, domains, regular expressions...
- *Export all the browsing data.* All accesses carried out using WebCache can be sent to a remote server to be processed. This makes it easier to generate advanced reports using third party software.

2.2 Scenarios

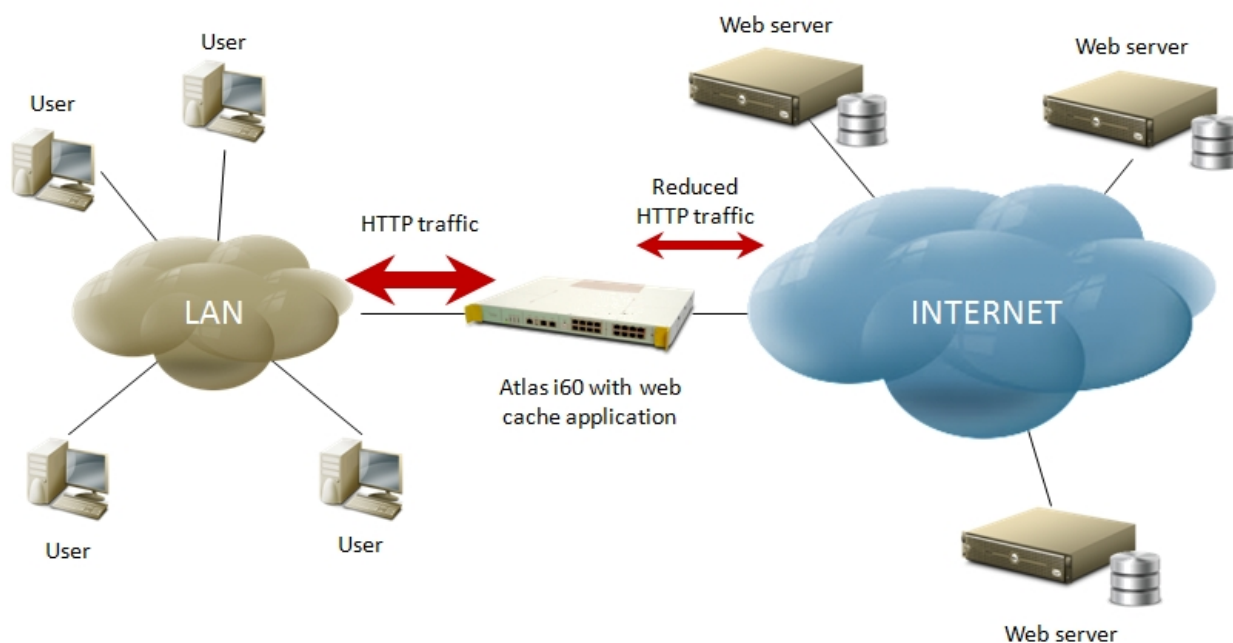
This section discusses the most common scenarios typically configured to optimize this kind of traffic applications.

For these scenarios, WebCache can be configured to be transparent to users (further configuration will not be required in user devices/browsers), or non-transparent, defining a port where the application is listening. For example, non-transparent mode is useful in several scenarios where the network administrator wants to cache only a single web server.

2.2.1 Direct Internet connection

The first scenario consists of an Atlas i6x directly connected to Internet. It caches all static requests and returns them in subsequent requests if a user is requesting the same content.

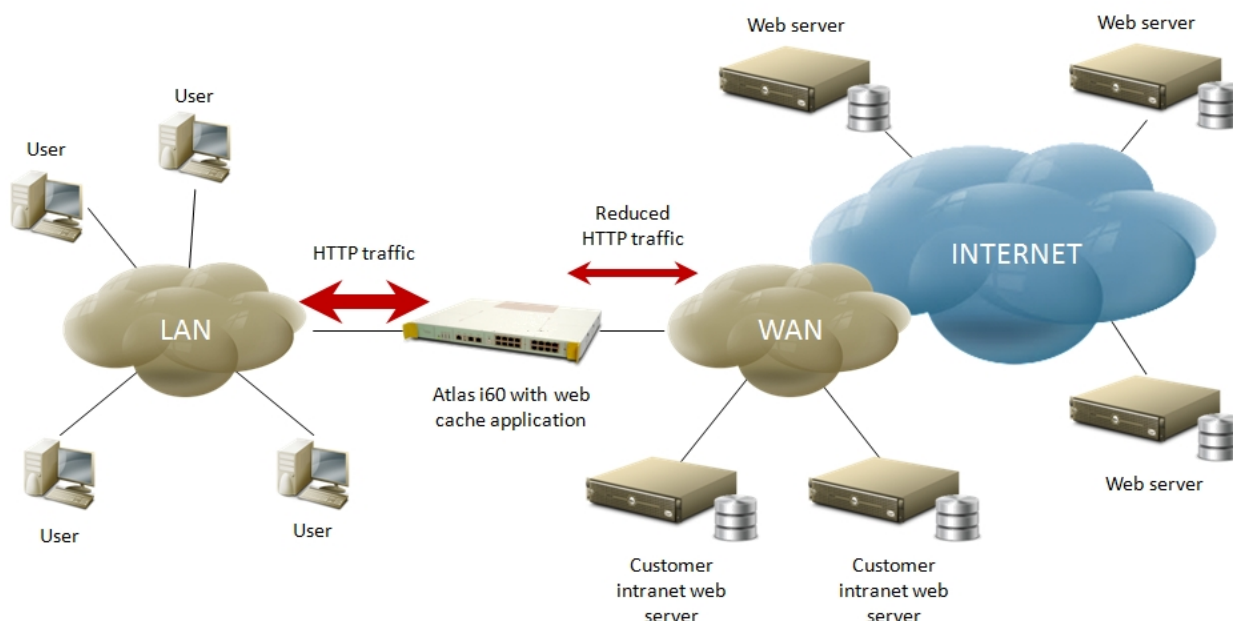
This reduces the bandwidth wasted in repeated requests, carried out by a single user or even different users in different time instants.

Fig. 2.1. Direct Internet connection scenario

2.2.2 WAN connection scenario

This scenario represents a typical configuration inside a corporation branch office. Usually WAN access is configured to interconnect all offices and isolate all internal communications securing connections over a VPN.

All HTTP requests are processed by a firewall or another proxy configured in the customer's central facilities. This application saves WAN bandwidth, serving local cached data to branch users, and generating only LAN traffic. The bottlenecks are usually the central connection to Internet and WAN speed. Both bottlenecks can be reduced by using this application.

Fig. 2.2. WAN connection scenario

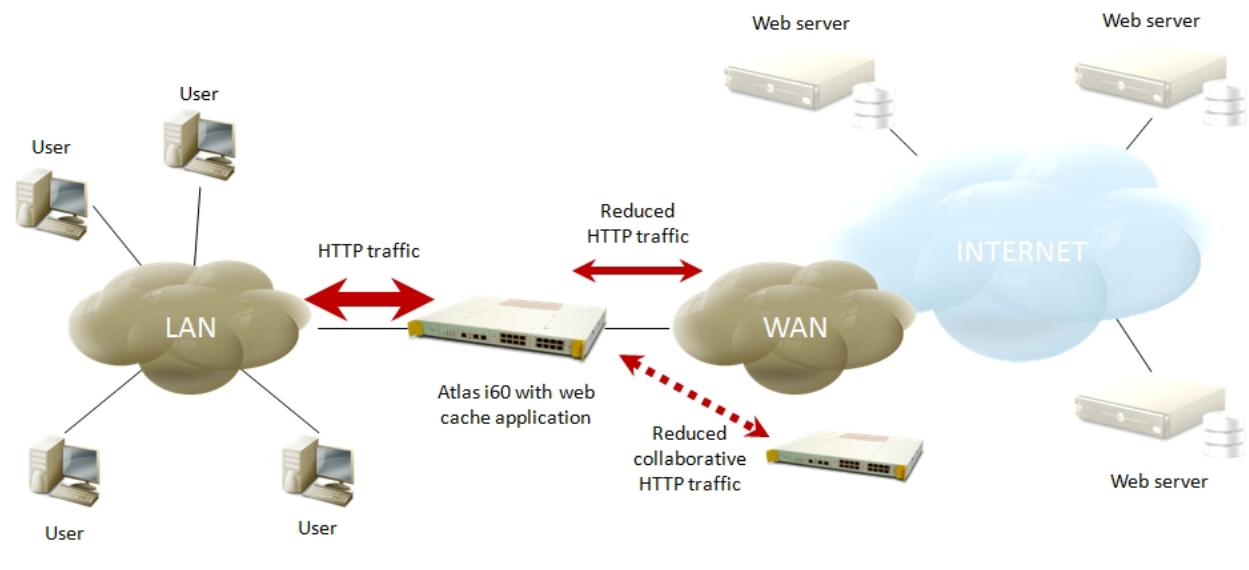
2.2.3 Collaborative scenario

If the previous scenario is used, use of multiple Atlas i6x with WebCache application installed can be configured.

This is useful to further reduce the bottleneck in central processing for all HTTP requests.

If a device does not find a request cached in local storage, it retransmits said request to a neighbor and, if this neighbor does not have said request cached either, it is retransmitted to the next neighbor and so on. If said request is not found in the neighborhood, it is finally retransmitted to Internet.

Fig. 2.3. Collaborative scenario



Chapter 3 Application configuration

This application, like other applications installed on the Atlas i6x device, is configurable in two ways: by using the Atlas i6x internal web and by using a text configuration or a configuration template inside the management platform, to simultaneously configure one or more devices.

3.1 Web configuration

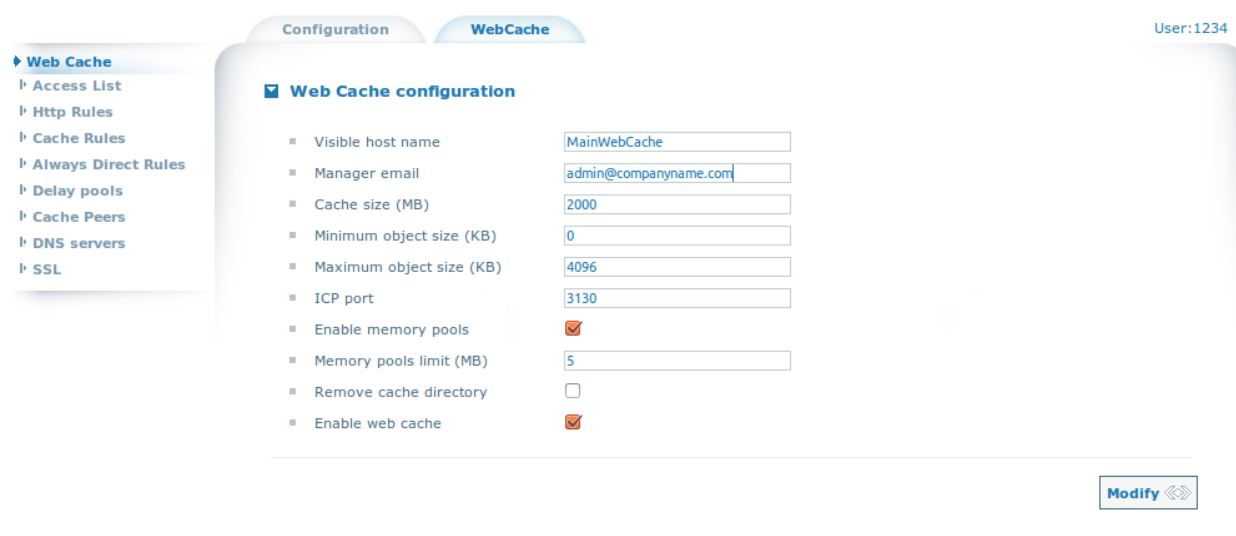
This application is represented in the main window as a cloud icon with an arrow. It is possible to click on the *WebCache* tab to access application configuration.

Fig. 3.1. Web configuration: Main window








When one of them is clicked, a new window opens. Click on WebCache (left hand menu) to enter the main configuration section.

Fig. 3.2. Web configuration: Main window



Note

Interface buttons: Summary

-  : Button to add a new entry in a table.
-  : Button to remove an entry from a table.
-  : Button to move an entry up in a table.
-  : Button to move an entry down in a table.
-  : Use this button to modify the value of the current section parameters. Please ensure you click on this icon before clicking on any other interface buttons, such as a table button, otherwise the whole page will refresh and you lose all your modifications.

3.1.1 General application parameters

The previous figure shows those general parameters that need to be configured. This section explains the meaning of each parameter and what it is used for.

3.1.1.1 Visible host name: Configuring the name of the WebCache engine

This parameter allows a user to specify a name for the engine. Said name is shown to users when a request cannot be processed and an error message, similar to the following figure, is shown in the browser.

Default is **WebCache**

Fig. 3.3. Error page

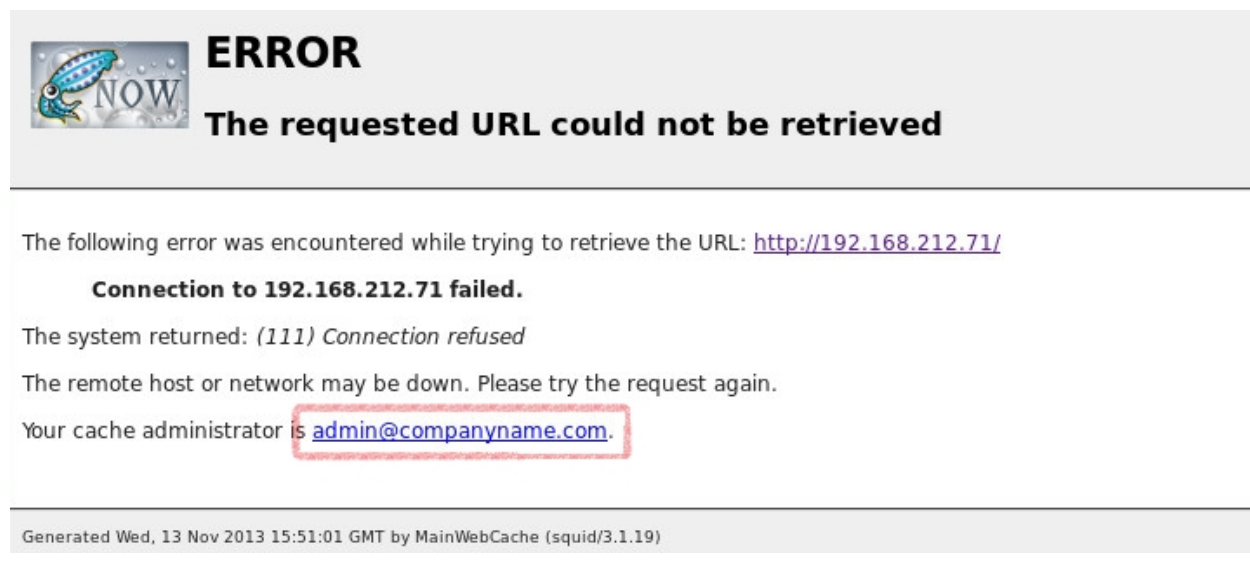


3.1.1.2 Manager email: Responsible for WebCache maintenance

Specifies an email address where end users should ask for help in cases of malfunction. This value is displayed on the error page too.

Default is **admin@cache.org**

Fig. 3.4. Error page



3.1.1.3 Cache size: Physical storage size for the internal cache

This parameter defines the size, in MB, which the engine needs to reserve for cached requests.

Default is **2000**



Note

The cache size value should not be the maximum size of the hard disk. Application performance depends on access time to stored cache files and physical memory (RAM) of the device.

This application holds an in-memory index of 10MB-RAM per GB-disk. Recommended maximum value for this parameter is about 20GB, provided it is not sharing the device with another application such as SiemSensor or Security. These applications demand a large amount of physical memory to run.

Additionally, there is another factor to keep in mind when you are configuring said parameter, which involves the kind of browsing your users usually carry out. Try with several values (no bigger than 20GB) to see if they increase performance when there are poor service times.

3.1.1.4 Minimum object size

This indicates minimum request size to cache. This value is represented in KB and should be a number greater or equal to 0. Use 0 to indicate no minimum object size.

This is useful so you don't cache smaller objects. Sometimes they are served more quickly by requesting them directly from the network (rather than reading these files from the cache stored on the disk) where the application is running. This is because an overload of small requests can occur, increasing hard disk access time, memory consumption...

For usual browsing pattern (visiting websites such as forums, news sites, etc.), this parameter may be configured to 0 KB (no lower limit).

Default is **0**

3.1.1.5 Maximum object size

This indicates maximum request size to cache. This value is represented in KB and should be a number greater and different to minimum object size value.

For usual browsing pattern (visiting websites such as forums, news sites, etc.), this parameter may be configured to 4096 KB.

Default is **4096**



Note

It is important to establish a small size for this parameter as, if a high value is set, the cache may be filled with large requests, discarding the smaller ones, and consequently not caching usual browsing data.

However, you may want to use your cache to store large files, such as ISO files, BLOBs, or similar, discarding small requests. In this case, a higher value should be established for said parameter.

As usual, parameterization of variables for any application always depends on what you really need.

3.1.1.6 ICP port: Configuring a collaborative cache network

The Internet Cache Protocol (ICP) is a protocol used for coordinating web caches. This application publishes a socket listening on a port, to retrieve external requests from other cache engines, to increase efficiency and reduce the number of requests sent to the target server.

This parameter is useful if you are using **cache peers** in another web cache and you want the Atlas i6x to receive re-

quests from said remote cache engine. See [Cache peers: Creating a simple hierarchy of neighbor caches](#) on page 17 for more details.

Default is **3130**

3.1.1.7 Enable memory pools

If set, this application keeps pools of allocated (but unused) memory available for future use. This is useful if you want to increase performance allowing the internal engine to retrieve cache requests directly from memory, instead of reading them from the hard disk.

However, if this option is enabled, it decreases the amount of free memory for the application host on the Atlas i6x. You may want to disable said option if other applications, needing a lot of RAM to run, are installed.

This option tends to be enabled.

Default is **enabled**

3.1.1.8 Memory pools limit (MB)

This parameter only works if memory pools are enabled. It represents the total amount of RAM reserved by the application for memory pools.

If you want to disable memory pools, please see [Enable memory pools](#) on page 9.

Default is **5**

3.1.1.9 Remove cache directory: Reestablishing initial state of internal cache

Sometimes you need to reestablish cache. Internally, cache memory is sorted, indexed and stored, but this indexation may be degraded due to undesired machine shutdowns (outages for instance). If application performance decreases without any reason, consider enabling this option.

When said option is enabled, the system internally removes the cache folder and creates a new one. Depending on total cache reserve size (see [Cache size: Physical storage size for the internal cache](#) on page 8), this process may take several minutes.

The checkbox for this option does not stay checked as it's only used to indicate the application should remove cache data. When the application receives this order, it executes indicated process and automatically unchecks said checkbox.

Cache reestablishment only works if WebCache engine is enabled. See the next option to enable said engine.

Default is **disabled**

3.1.1.10 Enable web cache

Check this box to enable the engine. If this box remains unchecked, the Atlas i6x does not serve any requests.

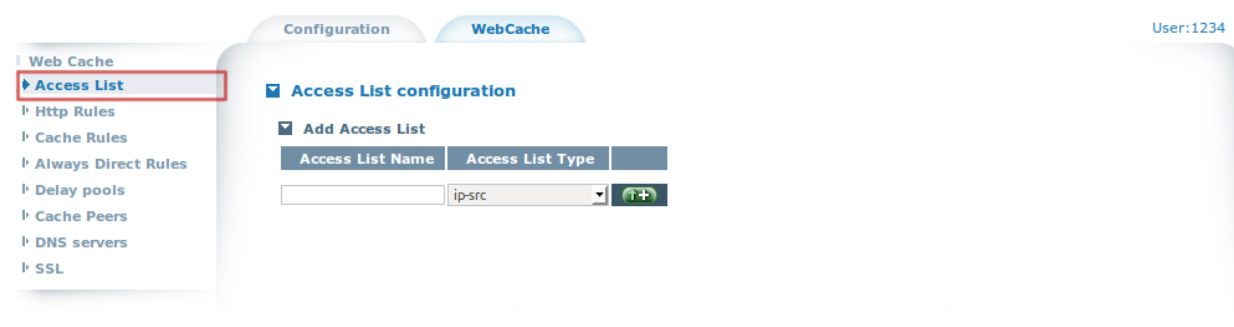
Default is **disabled**

3.1.2 Access lists: Defining list of users, machines, domains etc.

In this section (see figure below) you can define access lists to be applied to caching rules.

There are several ways to indicate what the application engine should process (filter, cache, no cache...). This is done by using access lists. If you apply an access list to a rule, it's only applied if an access list matches the processed HTTP request.

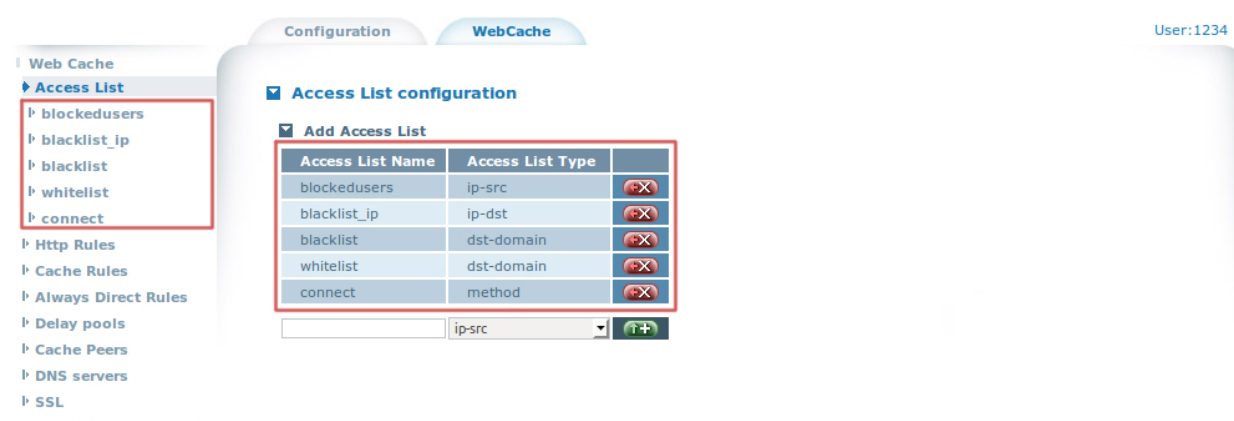
Fig. 3.5. Access lists section



To create a new access list, assign a unique name, select its type, and click on . When a new access list is correctly created, a new entry appears in the table with . If you press said icon, the access list is removed from this table *only if it is not being applied to any rule in the application*. Please ensure all rules relative to the access list are removed before removing said list.

For each access list configured on the system; an entry in the left hand menu appears under the Access List sub-menu. Click on any of these to configure access list parameters. Said parameters may be different for each type of access list.

Fig. 3.6. Access lists table



Full list of supported access lists:

- ip-src* : source (client) IP addresses.
- ip-dst* : destination (server) IP addresses.
- dst-domain* : destination (server) domain name.
- url-regex* : URL regular expression pattern matching.
- urlpath-regex* : URL-path regular expression pattern matching, leaves out the protocol and hostname.
- method* : HTTP request method (get, post).
- dst-port* : TCP connections destination port.

3.1.2.1 ip-src: source IP addresses

Select one or more subnets for this access list. You must specify a subnet IP and its mask.

Matching all *source* IPs belonging to specified subnets.

Fig. 3.7. Access list: ip-src

Add Source Subnet

Subnet IP	Subnet Mask	
172.16.5.0	255.255.255.0	
<input type="text"/>	<input type="text"/>	

3.1.2.2 ip-dst: destination IP addresses

Select one or more subnets for this access list. You must specify a subnet IP and its mask.

It matches all *destination* IPs belonging to specified subnets.

3.1.2.3 dst-domain: destination domain name

Select one or more domains to match. For example, to match all servers accessed using domain name somedomain.com, create an entry in the destination domain table including *.somedomain.com*. Note the period (.) before the domain name is used to match all subdomains ending with somedomain.com.

Fig. 3.8. Access list: dst-domain

Destination domain

Destination domain	
.facebook.com	
.youtube.com	
.linkedin.com	
.subdomain.domain.com	
<input type="text"/>	

3.1.2.4 url-regex: URL regular expression pattern matching

url_regex means to search the entire URL for the regular expression you specify. *Remember to check the "Case sensitive" checkbox to distinguish upper and lower cases.*

Default is enabled.

For example:

- Match a single address to access a specific URL:

```
^http://test.domain.com/path/$
```

- Match all URLs containing the word "adult":

```
adult
```

Fig. 3.9. Access list: url_regex

URL regular expressions access list configuration

Case sensitive



Add URL regular expression

Regular expression	
adult	
^http://test.domain.com/path/\$	
<input type="text"/>	

3.1.2.5 urlpath-regex: URL-path regular expression pattern matching

This option is the same as the previous one [url-regex: URL regular expression pattern matching](#) on page 11 without taking into consideration protocol and hostname.

3.1.2.6 Method: HTTP request method

Parameter used to specify what method should match this access list. There are two:

- GET: For GET requests (parameters included in the URL). Access list default value.
- POST: For POST requests (parameters included in the HTTP header).

3.1.2.7 dst-port: destination port

Select one or more ports for said access list.

Matches destination *TCP connection port*. E.g. To match HTTP traffic that goes to its destination through a direct connection, enter **port 80**. To match traffic that goes through a corporative proxy listening in port 8080 (after passing webcache), enter **8080** port, although final destination for the package, after passing corporative proxy, is a web server on port **80**.

3.1.3 HTTP rules: What are the requests to process by the cache engine?

This section define rules by using predeclared access lists to process HTTP traffic matching said lists.

You can define rules with multiple conditions. In the following figure for example, only requests matching a method access list selected in the first column and the destination domain access list selected in the second one, are allowed by the application.

Row 2 in the table discards all other requests not matching the first rows.

Fig. 3.10. HTTP rules table

Configuration WebCache User:1234

Web Cache
 Access List
Http Rules
 Cache Rules
 Always Direct Rules
 Delay pools
 Cache Peers
 DNS servers
 SSL

Http Rules configuration

Add Http Rules

Allow Traffic	Op	Access List	Op	Access List	Op	Access List
allow	yes	get	yes	cacheddomains	yes	none
deny	yes	all	yes	none	yes	none

allow yes all yes none yes none

In the table above, the column referencing what to do with matched requests is marked in red, the first condition of the row in green; second condition in blue and third condition in yellow. Fill out all conditions for each rule (you can use **none** access list so a condition is not used).

As you must use at least one condition per rule, access list for the first column cannot be set to **none**.

There is another special access list known as **all**. This matches all requests.

Rule precedence is controlled by the position of the row in the table. Upper rules are always applied before lower ones. Use and to move a rule in this table.

To cache all HTTP request traffic, declare only one rule in this table (allowing all):

Fig. 3.11. HTTP rules: Matching all requests

Add Http Rules

Allow Traffic	Op	Access List	Op	Access List	Op	Access List
allow	yes	all	yes	none	yes	none

allow yes all yes none yes none



Note

If there are no access lines present, default is deny request.

If none of the access lines finds a match, default is the opposite of the last line in the list. I.e. if the last line was deny, default is allow. Contrariwise, if the last line is allow, then default is deny. For these reasons, it's a good idea to have a **deny all** entry at the end of your access lists to avoid potential confusion.

3.1.4 Cache rules: Deciding what to cache

A list of access list elements, which, if matched and denied, means a request is not matched to the cache and a reply is not cached. Use this to when you want certain objects to never be cached.

Default is **allow everything to be cached**

Fig. 3.12. Cache rules table

The screenshot shows the 'WebCache' configuration page. On the left, a navigation menu has 'Cache Rules' highlighted with a red box. The main area is titled 'Cache Rules configuration' and contains a table for 'Add Cache Rules'.

Allow Traffic	Op	Access List	Op	Access List	Op	Access List
deny	yes	nocacheddomains	yes	none	yes	none
allow	yes	all	yes	none	yes	none

The table is used in the same way as HTTP rules tables. Please refer to [HTTP rules: What are the requests to process by the cache engine?](#) on page 12

3.1.5 Always direct rules: forward request without using peers

List of access lists to specify requests that should *always* be forwarded to source servers without using peers.

For example, to directly forward requests for local servers ignoring any parents or siblings, use an access list containing destination domains in a rule:

Fig. 3.13. Always direct rules

The screenshot shows the 'WebCache' configuration page. On the left, a navigation menu has 'Always Direct Rules' highlighted with a red box. The main area is titled 'Always Direct Rules configuration' and contains a table for 'Add Always Direct Rules'.

Allow Traffic	Op	Access List	Op	Access List	Op	Access List
allow	yes	dstdomains	yes	none	yes	none
allow	yes	all	yes	none	yes	none

The table is used in the same way as the HTTP rules tables. Please refer to [HTTP rules: What are the requests to process by the cache engine?](#) on page 12



Note

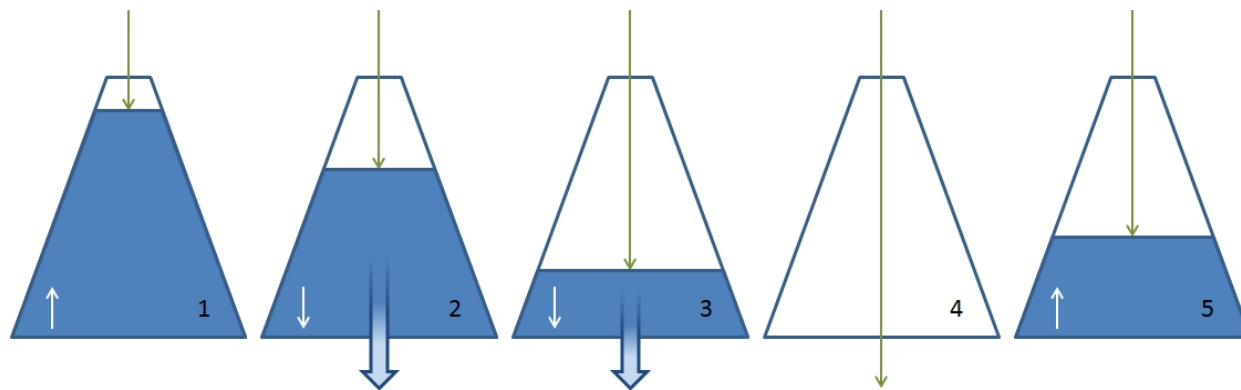
Always direct rules are not related to caching. The replies are cached as usual even if you use these rules. So replies are not cached, please see the cache rules [Cache rules: Deciding what to cache](#) on page 13.

3.1.6 Delay pools: Limiting the bandwidth usage per user/network consumption

Sometimes bandwidth needs to be restricted depending on network usage, for example, to prevent users from constantly downloading at maximum bandwidth speed. This helps prevent network congestion when users are consuming Internet connection bandwidth at the same time.

Delay pools are the best way to limit the bandwidth depending on the size of downloaded requests.

The following figure illustrates this concept:

Fig. 3.14. Delay pools illustration

To understand what a delay pool is, imagine a bucket. Said bucket contains data (blue inner color, figure 1). A user can consume data from the bucket at maximum bandwidth speed established by the administrator (or at maximum network speed if the administrator does not limit this parameter) (blue arrow, figures 2 and 3). When the bucket is empty, a user has his bandwidth speed reduced to a maximum speed allowed by the administrator (green arrow, figure 3). If the user stops bandwidth consumption, the bucket is progressively filled again, as quickly as permitted by a regeneration speed established by the proxy administrator (blue inner color, figure 5).

The white arrow represents the filling or emptying of the bucket.

The application allows you to define several delay pools, using access lists to match network traffic with the corresponding delay pool. First establish what the initial bucket level is (0-100 percent). Said bucket level is the initial level for all delay pools when the application starts. Subsequently, create delay pools using the delay pools table configuration page.

Fig. 3.15. Delay pools main configuration

Once the operation is complete, enter each delay pool to configure the parameters. Please see the following examples on parameters requiring configuration.

Delay initial bucket default level parameter is **50**.

3.1.6.1 Limiting the global bandwidth

In this example, the WebCache reduces bandwidth to 256KB/s (262144 bytes/s) when global traffic exceeds total bucket size, 100MB (1073741824 bytes).

Note: the network and individual settings are -1. This means infinite speed and bucket size for said parameters.

Fig. 3.16. Delay pools aggregate limit**Delay pool configuration**

▪ Agregate	<input type="text" value="262144"/>
▪ Agregate Maximum	<input type="text" value="1073741824"/>
▪ Network	<input type="text" value="-1"/>
▪ Network Maximum	<input type="text" value="-1"/>
▪ Individual	<input type="text" value="-1"/>
▪ Individual Maximum	<input type="text" value="-1"/>

3.1.6.2 Limiting the network bandwidth

To assign a delay pool to a network, play with different access lists these networks match.

If you only have one network or use the access list **All**, this generates the same effect as the aggregate example.

Here, configure a delay pool to limit bandwidth to 256KB/s (262144 bytes/s) when network traffic exceeds 100MB (1073741824 bytes).

Fig. 3.17. Delay pools network limit**Delay pool configuration**

▪ Agregate	<input type="text" value="-1"/>
▪ Agregate Maximum	<input type="text" value="-1"/>
▪ Network	<input type="text" value="262144"/>
▪ Network Maximum	<input type="text" value="1073741824"/>
▪ Individual	<input type="text" value="-1"/>
▪ Individual Maximum	<input type="text" value="-1"/>

Imagine assigning this delay pool to access list **network1**. Predefine access list network1 to include all subnet 192.168.212.x IPs.

Now, define another delay pool with 128KB/s (131072 bytes/s) and 50MB (536870912 bytes). This second delay pool is assigned to another access list, **network2**. This access list matches all subnet 192.168.213.x IPs.

Configuration of initial bucket level is to have 100% of the data available.

When WebCache starts, initial bucket for network1 users is 100MB, initial bucket for network2 is 50MB. If users begin to use the connection at maximum speed, when network2 users reach 50MB limit, their bandwidth is reduced to 128KB/s while network1 users can still download data at maximum speed. When network1 users reach 100MB of data, the proxy reduces their bandwidth to 256KB.

If some network stops data consumption, the data for said bucket is filled at the speed established as maximum for the network (network1 256KB/s, network2 128KB/s) until a 100% is reached again (100MB for network1, 50MB for network2). If the bucket is not full and a user begins to download data, maximum bandwidth is used until said network bucket is empty again.

3.1.6.3 Notes about delay pools

- You can combine several delay pools or even several types of limitations within a single delay pool, e.g., aggregate bucket and user bucket, assign -1 for network parameters.
- If you do not want to limit a parameter, use -1 as parameter value.
- All said values, bucket size and bandwidth speeds, are represented in bytes and bytes per second.
- If aggregate/network/users stop bandwidth consumption, said buckets are filled at the bandwidth established for each role. These sizes and bandwidths can be different.

Default for aggregate/network/users parameters is -1

3.1.7 Cache peers: Creating a simple hierarchy of neighbor caches

This is an interesting feature if you are deploying a large network of Atlas i6x with WebCache or other devices implementing Inter-Cache Protocol (ICP).

Said protocol increases hit rates, preventing object duplication. Usually, a large corporation has a single WebCache per branch office invisible to other WebCaches in the network. WebCaches look for HTTP requests directly in the source servers if a request has not been precached in the internal memory. If said protocol is enabled and the neighbor network configured, if a request is not found in the cache, the application will redirect the request to its peers instead of the central server.

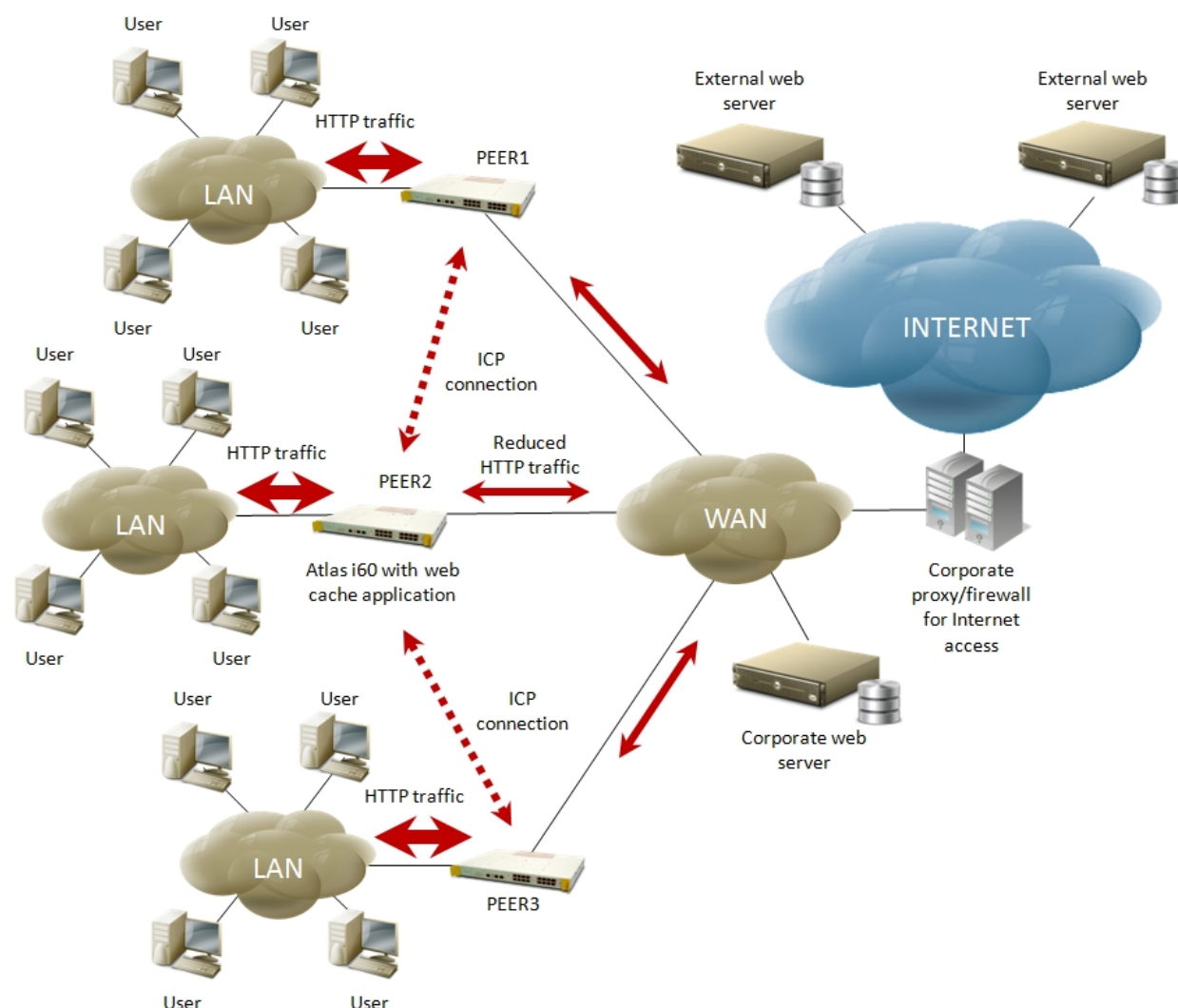
This generates interbranch cache with the following advantages:

- Reduces bottleneck at central servers as requests are distributed over the peer network. I.e. provides a better chance of a request being executed by another peer.
- Keeps objects near end users, reducing network latency.
- If a device needs to reestablish an internal cache, user experience is more efficient.
- If you include another WebCache application in the peers network, it adds more cache space to the overall hierarchy.

To use this feature, the administrator must choose, per device, its peers carefully, trying to connect peer neighbors with low connection latencies. There is no sense in having a connection, for example, between two peers with a low bandwidth over a VLAN connection, which is slower than a direct connection to Internet.

3.1.7.1 Cache peers example

In this example, we deploy a peers network with three different Atlas i6x routers with the WebCache application installed. They are connected over a WAN. Note in the following image there is a corporate proxy/firewall infrastructure configured to provide Internet access for corporate machines.

Fig. 3.18. Cache peers' example schema

If no cache peers are configured, the three remote branch offices alone are caching data. If network 1 user accesses the same static web content as network 3 user, the two requests are transmitted over the corporate proxy.

Configuring cache peers network allows the neighborhood to share caches and not send requests to the corporate proxy (imagine this corporate proxy is overloaded and cannot handle branch office traffic because local central users are overloading the Internet connection because they are carrying out backups or viewing online videos).

Now, imagine network 1 is close to network 2 (the same city) and the network 3 is on another continent. The logical structure to deploy is to configure device PEER1 and device PEER2 as peers, leaving PEER3 alone.

If someone in network 1 requests static HTTP data from the central server and it is not cached in the local cache (PEER1), said request is sent to the neighbor (PEER2). If this neighbor has the data cached, it forwards it to PEER1 and PEER1 caches and serves it. There is no connection to the central server. So, connection has been more efficient because PEER1 and PEER2 are closer than the central server, which may be in another different city, country, etc.

Imagine the three peers are in the same city and the central server is on another continent. If branch offices are configured as peers in the same city, they provide cached content more quickly in the neighborhood than by requesting it from the central server.

3.1.7.2 Cache peers configuration

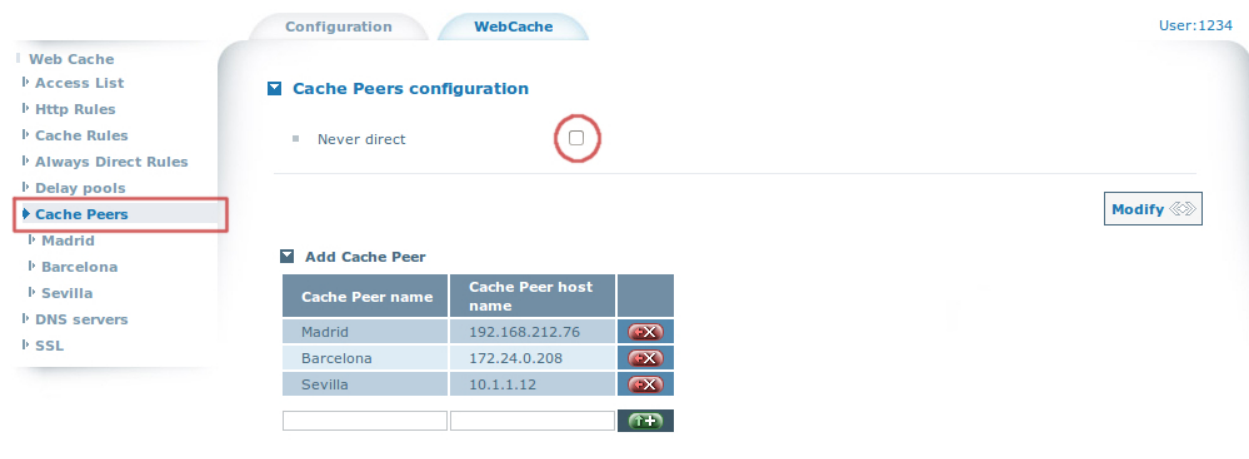
To configure cache peers, access Cache Peers section and configure several neighbors indicating their names and hostnames or IP addresses.

If you do not want to allow this peer to directly connect to Internet to request data that has not cached, check check-

box *Never direct*. If said checkbox is checked, all requests are sent to the neighbors, but never to Internet. Consequently, at least one WebCache must be configured with said option disabled in the peers neighborhood.

Default for never direct parameter is **disabled**

Fig. 3.19. Cache peers general configuration



For each configured cache peer, a new option appears in the left hand menu under Cache Peers. Each suboption is used to configure specific parameters for each configured peer.

Use to add and to remove a cache peer.

3.1.7.3 Particular options for cache peers

Each cache peer has specific parameters to be configured. The following figure represents all said parameters:

Fig. 3.20. Cache peers particular parameters configuration

- Cache Peer host name
- Cache Peer type
- Cache Peer proxy port
- Cache Peer icp port
- Set proxy only option
- Set default option
- Set no query option
- Set no digest option
- Set no netdb exchange option
- Transparent authentication

3.1.7.3.1 Cache peer host name

Indicates IP address or hostname to access said proxy from local WebCache.

3.1.73.2 Cache peer type

This option establishes how to use this peer to request uncached data. Permitted values are:

- (a) *Parent*: Cache, which can be used as a last-resort if a peer cannot be located through any of the peer-selection methods. If more than one is specified, only the first is used.
- (b) *Sibling*: Cache is in the same hierarchical level. All requests are sent to the siblings.
- (c) *Multicast*: A multicast ICP server joins multicast groups addresses to receive messages. This is useful so ICP requests are not generated per neighbor in the network. Bear in mind the response is always unicast.

Default is **parent**.



Note

When you use a multicast peer, hostname must be a multicast address. Ensure this group is not being used by another application. Additionally, ensure all members of the multicast group are using the same ports for ICP and HTTP requests.

3.1.73.3 Cache peer proxy port

This is the port where all HTTP traffic is requested. Please note, said port is configurable in the general Atlas i6x configuration: HTTP Proxy.

Default **3128**.

3.1.73.4 Cache peer icp port

This is the port where all ICP requests are sent for the peer.

Make sure you configure your sibling and/or child caches if you decide to use a nonstandard port.

Default is **3130**.

3.1.73.5 Set proxy only option

Enable this checkbox so objects retrieved from the peer are not locally stored.

Default is **disabled**.

3.1.73.6 Set default option

Enable this checkbox to specify this cache peer should be used as a last resort in a scenario where other peers cannot be contacted.

Default is **disabled**.

3.1.73.7 Set no query option

Enable this checkbox to disable queries to this cache peer.

Default is **disabled**.

3.1.73.8 Set no digest option

Enable this checkbox to disable requests for cache digests.

Default is **disabled**.

3.1.73.9 Set no netdb exchange option

Enable this checkbox to disable requesting ICMP RTT database (NetDB).

Default is **disabled**.

3.1.7.3.10 Transparent authentication

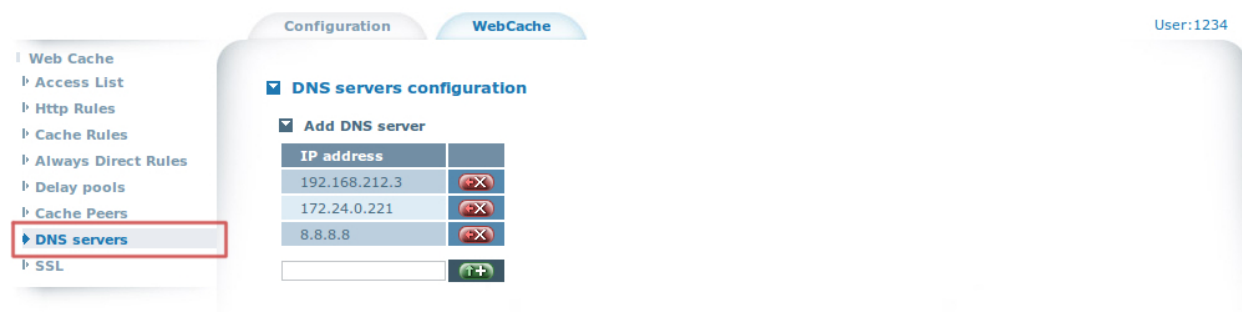
Enable this checkbox to send login details received from client to this peer. Use to transparently authenticate clients with the cache peer (as there is no WebCache).

Default is **disabled**.

3.1.8 DNS servers

This section allows you to specify one or more DNS servers. If you do not add any DNS servers, then DNS servers system is used instead. Remember hostapplication DNS servers system is configured in **feature vlii** in Atlas i6x routing (CIT) configuration.

Fig. 3.21. DNS servers' configuration



Use  to add and  to remove a DNS server.

3.1.9 SSL configuration

This section allows you to configure SSL parameters for the application. SSL parameters are used for caching HTTPS traffic.

For HTTPS caching, **HTTPS Proxy** must be enabled in the HTTP Proxy Configuration section. See [HTTP proxy web configuration](#) on page 49 for further information.

HTTPS caching: basic mechanism. When you request a URL using HTTPS, you negotiate an SSL session with the server and receive a certificate signed by a Certificate Authority (CA) you trust. If the certificate is self-signed, or you don't trust in the CA, you are asked to trust or not in the issuer of the certificate.

When WebCache is configured to cache SSL traffic, it is not the client but the WebCache which negotiates the SSL session with the server. WebCache receives the certificate from the server, which it can trust or not. If the application doesn't trust it, it sends the client an error page reporting the event. If the application does trust, with its own certificate CA (which the customer has to trust), it dynamically generates a certificate for the requested site. WebCache stores said certificate in the certificates database and passes it to the client in the same way as any server.

If the client examines the certificate he has received, he'll see a certificate for the requested site, signed by a Certificate Authority. This CA can be: a certificate the administrator has imported in the WebCache configuration or the WebCache Root Certificate Authority, if said WebCache certificate has been generated by the application instead of being imported by the administrator.

Http Rules, Cache Rules and Always Direct Rules operate with SSL traffic in the same way as described in their sections. There are however, some some limitations for SSL. Due to the cipher nature of the SSL traffic it isn't possible to match a received HTTPS message with a list containing domains or regular expressions. If HTTP Proxy is configured in non transparent mode, the SSL traffic comes through an HTTP tunnel that is started with a CONNECT package indicating the destination domain. Access lists can match with the URL of this CONNECT package too.

There are several SSL parameters to be configured, as you can see in the figure. All of them are described in this section. In the SSL section (configuration web) there is also a status button with the said parameters.

Fig. 3.22. SSL configuration

SSL configuration
Show status

- Private Key

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAwSCTwclIh0g/oM6ETjA6Xw+wBSbz4fBH93FmaeREuZc07P6R
2EyxatG/40zk36kWHYYkcIu4Mp5MalN5WQlqcqxLDB4vrXbXf0zWnW+Kd1j3wo
4+8B24IKQZgvlEs01OnHW8cswDLbVvIUZYsJZJWqJB9bsYUpcFU6gg+bdWF97G
rSfLgbHzNIbGcJTEfVLUppKKF9+lQ7wPR0uTB4f9WE1VjIUCYdPqSmd+lAd3F2A0r
mJMSylWwu+ZgchCHhIZIECmuBFilwggAR4JF6IM59CU0HQX9m9wcNiW0y/u//IAf
L7cAtBztZSHKXec79qcks0mpDGxyt4l0MzbKGwIDAQABAoIBAQCltPZ0XD6UzoZo
3JxHN4oyVFXgFEkeskRNUwJkHdY9b3JDrRByKWC6xksKsghrZ5Tw/KYFqH0yW0Ms
k4AYEcCofdv9yP6/88luMM2tfyActn9LseSeprpwQYl0Hyy00ee1NWKCP0Svzy
mWkBJlU0JzRydyDMugmlxWfY7B704k3NREAQ75Kcsaw3HFPLEda7hfTqbSLpEDV
/FovJkRBU7/m59tukXFQ5fwNG8Vqb9tBqjeycC1LAGl0bsWfCWtAM5Bq85xfSV4j
vfCzY4HuFl7kJIRVILJRimkVZUX5MwKaJ8nxD0e5iBCX1A0CSLcxmW/a061lvQ5
kz2IBoo5AoGBAPxBiZDgSLaquiLj8IUsI4B9gqrb+qLwZ097056dfLB1nJkTLt
gkTLWohkNgu/teVNIp5R8tk49wUtue0ssm91K5IHnK5h3spCk0iCHLfiGFSTwCN
+0zAr8eRofrZLVmCLEAw+JtZs5PXgIAR7gj/P0+Bdt43QsG8lHsQJzS/AoGBAMP+
YFqwp07HmMee94uXhXMCr7aXUU9yd/J0+i5hGjhhbKCKJ5Cj4623ArIkGhW6WX
ASmh1x1WLXhdp8E836ETbz6SB65CjLS0RvTomE0dsgtU31s43j+wn3SJY9q04yN+
9WHsIkLBe1+heKX0eeUX6p/Urd8VTdZdonNGSfWlAoGAbHOAwVir6LzHqsqNtLNO
```
- Public Key

```
-----BEGIN CERTIFICATE-----
MIID1zCCAr+gAwIBAgIHaHvN2bvzBjANBgkqhkiG9w0BAQsFADCBgTELMAKGA1UE
BhMCRVMxZzANBGNVBAcMBk1hZHJpZDEVMBMGA1UECgwMQ2FjaGUGRW5naW5lMR0w
GAYDVQQQLDBFXZVjYWN0ZSBTZW51cm10eTEuMCwGA1UEAwVlV2ViY2FjaGUGUm9v
dCB0ZDZl0aWZpY2F0aW9uIEF1dGhvcml0eTAeFw0xNTA2MDg5NzQwMDdaFw00NTA1
MzExNzQwMDdaMG4xCzAJBgNVBAYTAkVTMQ8wDQYDVQQHDAZNYWRyaWQxTATBgNV
BAoMDENhY2hlIEVuz2luZTEaMBGGA1UECwwRV2ViY2FjaGUGU2VjZDkxJm1p1vndx
BgNVBAMMEldlYmNhY2h1IDc1N18wMDE1NzCCASIwDQYJKoZIhvcNAQEBBQADggEP
ADCCAQoCggEBAMEgk8HJSIdIP6D0hE4w0l8PsAUm8+HwR/dxZmnkRLmXDuz+kdhM
sWrRv+NM5N+pFh2GJHCLuDKeTGpTeVkJanKsVywweL68QYV39M1p1vindY98K0Pv
AduCCKGyL5RLDpTpx1vHLMay21b4r12GbCWSVqiQfW7GLj3H10oI4Pm3Vhfexq0n
y4Gx8zSGxncUxH157jyihffpU08D0TrkweH/VhNVYyFAMHT6kpnfpWndxdgDq5iT
EspVsLvmYHIQh4YmSBAPrgRYpcIIAEeCreiD0fQ1NB0F/ZvcHDYltMv7v/yAHy+3
ALQc7WUhy13gu/anJLDppQxscrcrTjM2yhsCAwEAANmMG0wHQYDVR00BBYEFF4e
VeGQLbAM3zUj2UbKlJakAcaUMB8GA1UdIwQYMBaAFHxY+OJYQypIxfHyVd8Bvd7M
BqnmMBIGA1UdEwEB/wQIMAYBAf8CAQAwDgYDVR0PAQH/BAQDAgGMA0GCSqSqsIb3
DQEBChUAA4IBAQCysNSACBgHz0No01fG5S7x0w62Jyp1ro00okxaguxlgjd0VFD
3DvBl6AcF/fQxI7AcPHNl1b0bWRs6phdyH1x9tZhdKDueqrDKFljuxPqloAR/d
```
- Certificates cache size (MB)
- Max. SSL processes
- Remove certificates database
- Generate certificate

Modify

If you click on the status button, information on SSL configuration appears. First, you can see the certificate generation information. This displays *imported* when the certificates public and private keys have been entered by the user using the appropriate text boxes; *autogenerated* when the certificate has been generated by WebCache using the generate certificate option in the SSL configuration section. The following is the WebCache Root Certification Authority certificate public key. This is the root CA used to generate a WebCache certificate when the WebCache certificate generation option is selected. This is common to all WebCache applications whatever devices they are installed in. If you import said certificate in a client device, it will trust in all auto-generated certificates for all WebCache applications. The status section shows detailed information on the WebCache certificate (said public and private keys are available on the SSL configuration page).

Fig. 3.23. SSL status

SSL status
Show conf

- Certificate generation status Autogenerated
- Root CA public key for autogenerated certificates


```

-----BEGIN CERTIFICATE-----
MIID7zCCAtegAwIBAgIJANJrNjI0q7rYMA0GCSqGSIb3DQEBCwUAMIGBMQswCQYD
VQQGEwJFUzEPMA0GA1UEBwwGTWfkcmlkMRUwEwYDVQQKDAxDYWN0ZS5BfmdpbmUx
GjAYBgNVBAsMEVdlYmNhY2h1IFN1Y3VyaXR5MS4wLWVhY2F1dDQ0VXZjZjYWN0ZS5B
b290IENlcnRpZmljYXRpb24gQXV0aG9yaXR5MCAXDTE1MDUyMjA4MTgxOjFoYDZlW
NjUwNTA5MDgxODE4WjCBgTELMAkGA1UEBhMCRVMxZDzANBgNVBAMkMk1hZHZjZDEp
ZDEuMCwGA1UEAwVlV2ViY2FjaGUGUm9vdCB0ZXJ0aWZpY2F0aW9uIEF1dGhvcml0
eTCCASIdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMwyfBI1ohChj67FWp7s
wGNvhuPnbM+BX+Y1SgtuFJ9skkZKjd0Y9qF2bao729nZCaDwrFpJ9wAWbml7iYUz
3LXAAxazcn0HBgWbWkHiSi21/HC+MQJfSEHiDcrI07yMIZJZuGSisR/SLUnoIwRe
1iJfxV1kK0cbvmhgMJYGut8Ee7jjC6QzW16//20Noalxaf30pvztZ5XiukAKZi8S
2B47vVXR9FFqLGNbPFk8PYYkGzk4eVmHCrjgtHYPK4Kcm1EUpt9C5tg6u0gg8ZwG
YbIgxqZHT0u9tU90Eiw9qTRwnRj0JY/uFwrHVK+6V1DzLFm6xFDcb0Q5331FLMAc
FacCAwEAAaNmMGQwHQYDVR00BBYEFHxY+OJUQypIfxHyVd8Bvd7MBqnmMB8GA1Ud
IwQYMBaAFHxY+OJUQypIfxHyVd8Bvd7MBqnmMBIGA1UdEwEB/wQIMAYBAf8CAQIw
DgYDVR0PAQH/BAQDAgGMA0GCSqGSIb3DQEBCwUAA4IBAQCChzTm2Y0UYgrTueBEe
+Waizev9vnf0HwnVzYKat0zab4HRSZrLFgon2Rzd3JnViat6G07NMoo2xsD0YncJ

```
- Certificate information

Certificate:

Data:

```

Version: 3 (0x2)
Serial Number:
68:75:67:d9:bb:f3:06
Signature Algorithm: sha256WithRSAEncryption
Issuer: C=ES, L=Madrid, O=Cache Engine, OU=Webcache Secur
Validity
Not Before: Jun  8 17:40:07 2015 GMT
Not After : May 31 17:40:07 2045 GMT
Subject: C=ES, L=Madrid, O=Cache Engine, OU=Webcache Secu
Subject Public Key Info:
Public Key Algorithm: rsaEncryption
Public-Key: (2048 bit)
Modulus:
00:c1:20:93:c1:c9:48:87:48:3f:a0:ce:84:4e:30:
3a:5f:0f:b0:05:26:f3:e1:f0:47:f7:71:66:69:e4:

```

3.1.9.1 Private key

The WebCache certificate private key is available from this text box. This can be either generated by WebCache or entered by the administrator. If you want to enter a new one, simply copy said private key to the text box. The application checks the public and private key modules to ensure they match, i.e. you have to enter your new public and private keys before clicking the **modify** button to save them. Contrariwise, the application displays an error message. Said application also checks the entered text corresponds to a certificate and is not empty. Checking is only executed when WebCache is enabled and **Enable HTTPS Proxy** is enabled in the HTTP Proxy configuration section. If WebCache or **Enable HTTPS Proxy** are disabled when data is entered, checking only occurs once the former are enabled.

3.1.9.2 Public key

In the same way as the private key, the WebCache certificate public key is available from the text box. This can be either generated by WebCache or entered by the administrator. If you want to enter a new one, simply copy said public key to the text box. The application checks the public and private key modules to ensure they match, i.e. you have to enter your new public and private keys before clicking the **modify** button to save them. Contrariwise, the application displays an error message. Said application also checks the entered text corresponds to a certificate and is not empty. Checking is only executed when WebCache is enabled and **Enable HTTPS Proxy** is enabled in the HTTP Proxy configuration section. If WebCache or **Enable HTTPS Proxy** are disabled when data is entered, checking only occurs once the former are enabled.

3.1.9.3 Certificates cache size

Maximum size (in MB) for the dynamically generated SSL certificates storage. Default 4 MB, which is normally sufficient for 1000 certificates. If WebCache is used in busy environments, this may need to be increased.

Default is 4

3.1.9.4 Maximum number of SSL processes

The maximum number of processes spawn to service SSL server. You need at least one. The default value is 5. If WebCache is used in busy environments this may need to be increased. The maximum this may be safely set to is 32.

Default is 5

3.1.9.5 Remove certificates database

Use this option to clean the WebCache certificates database. This database contains dynamically generated certificates passed to clients for each site they access. It doesn't remove either the WebCache certificate (whose public and private key is on the SSL configuration page) or the WebCache Root Certification Authority certificate (whose public key is available on the SSL status page).

Check this option when you import or create a new WebCache certificate to remove all old dynamically generated certificates signed by the old WebCache certificate (no longer available).

Default is **disabled**



Note

Remove the certificates database after you import or generate a new certificate to clean all final certificates signed by the old certificate.

3.1.9.6 Generate certificate

Check this option to generate a certificate for WebCache. After clicking in the modify button you see the new public and private keys in the text boxes of the WebCache SSL configuration page. You can see detailed information about your new certificate in the status page. The common name of this certificate contains the serial number of the device where the application is installed. The certificate is signed by the WebCache Root Certificate Authority, whose public key is available in the status page.

Default is **disabled**

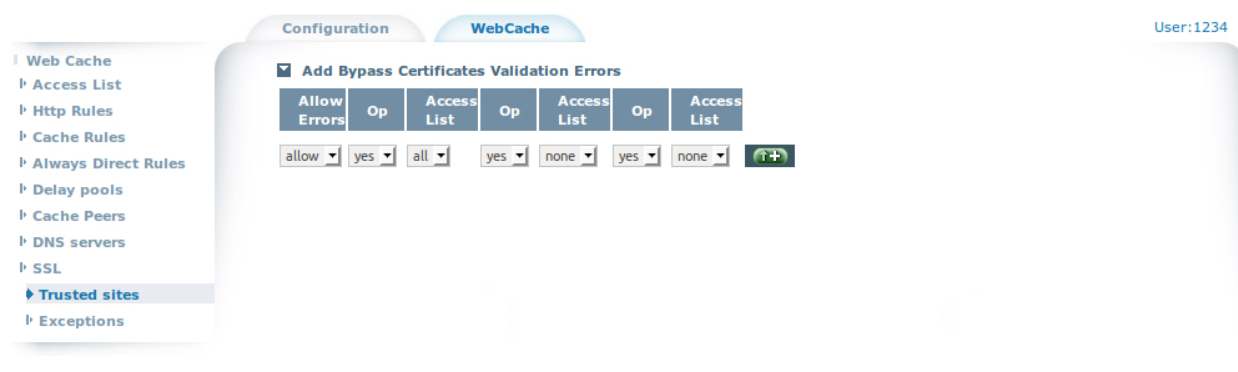
3.1.9.7 Trusted sites

Access this section from the configuration web by clicking on the **Trusted sites** tag on the left menu. Said menu appears by selecting SSL.

When WebCache sends a request to a URL, it receives a certificate from said site. The application can trust the certificate (or not) depending on whether a specific Certification Authority is installed and said certification information is correct. WebCache can be configured (by a user) for situations where suspicious certificates appear. Suspicious certificates are common, especially when a site provides a self-signed certificate.

Access lists can be added here, in the same way as in the sections for HTTP Rules, Cache Rules or Connection Rules. When **Trusted sites** is empty, WebCache is configured to ignore certificate validation (**default** behavior) so, should a suspicious certificate appear, WebCache automatically trusts it (this is transparent to the client). This is recommended configuration.

If there is an entry in the table, WebCache reports a suspicious certificate it doesn't trust when said certificate is provided by a site and permitted in the table. The browser asks the user if he trusts the certificate and, if he does, said user can access the site. If the site is **deny0**, WebCache immediately terminates communications and sends the client an error message referencing the event.

Fig. 3.24. SSL trusted sites

3.1.9.8 Exceptions

Access this section from the configuration web by clicking on **Exceptions** on the left side menu. Said menu opens on selecting SSL.

Here, you can access lists you predefined in the **Access Lists** section. Access lists added to this section are not cached; the client receives the original certificate from the server (not the WebCache certificate). Only access lists containing IPs address are considered for exceptions. As SSL traffic is ciphered, it isn't possible to match a received HTTP message with a list containing domains or regular expressions. If HTTP Proxy is configured in non transparent mode, SSL traffic comes through an HTTP tunnel started with a CONNECT package, indicating destination domain. Access lists can match the CONNECT package URL as well.

Access lists included in this SSL Exceptions lists are only considered when WebCache is configured to cache SSL traffic. Otherwise they have no effect.



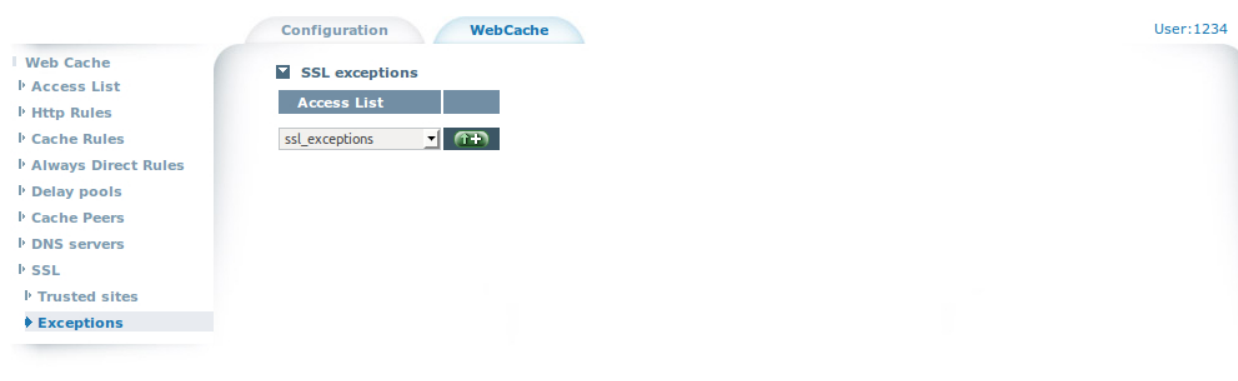
To add a new access list to the SSL exceptions table, select an access list from existing ones (they are shown in a combo button) and click the right button . It is not possible to add an empty access list. To delete an access list from the exceptions table, click on delete  located on the right side of the chosen access list name.

Fig. 3.25. SSL exceptions

3.2 Text configuration commands

In this section all configuration directives allowed in the application text configuration are described.



Note

The configuration directives should be sent in a single text file to the device through the Atlas i6x management portal.

If a statement does not appear in the configuration text, the engine will use the default value.

3.2.1 Web Cache configuration

```
webcache
```

Top level configuration directive.

3.2.1.1 Visible host name

```
visible-hostname <value>
```

This parameter allows the user to specify a name for the engine.

Default is WebCache.

3.2.1.2 Manager email

```
cache-manager <value>
```

Responsible for WebCache maintenance.

Default is admin@cache.org

3.2.1.3 Cache size (MB)

```
size <value>
```

Amount of megabytes for the internal cache.

Default is 2000.

3.2.1.4 Minimum object size (KB)

```
minimum-object-size <value>
```

Minimum request size to cache.

Default is 0.

3.2.1.5 Maximum object size (KB)

```
maximum-object-size <value>
```

Maximum request size to cache.

Default is 4096.

3.2.1.6 ICP port

```
icp-port <value>
```

Internet Cache Protocol (ICP) port for cache peers.

Default is 3130.

3.2.1.7 Enable memory pools

```
memory-pools-enable
```

If set, the application keeps pools of allocated memory available for future use.

Default is **enabled**. To disable this feature enter **no memory-pools-enable** in the configuration text.

3.2.1.8 Memory pools limit (MB)

```
memory-pools-limit <value>
```

Total amount of RAM that is reserved by the application for memory caching.

Default is **5**.

3.2.1.9 Remove cache directory

```
remove-cachedir
```

Reestablishes the initial state of the internal cache.

This is a special parameter. It is automatically disabled when the process of removing the cache directory is completed. To enable this feature enter **remove-cachedir** statement in the text configuration.

3.2.1.10 Enable web cache

```
enable
```

Enables WebCache engine.

Default is **disabled**. Enter **enable** in the configuration text to enable the engine.

3.2.1.11 Access List configuration

```
accesslist
```

Access list configuration section.

3.2.1.11.1 Add Access List

```
add name <value> type <value>
```

Add an entry in the access lists table:

- *Access List Name* : Name of the access list
- *Access List Type* : Type of the access list:
 - *ip-src* : Source (client) IP addresses.
 - *ip-dst* : Destination (server) IP addresses.
 - *dst-domain* : Destination (server) domain name.
 - *url-regex* : URL regular expression pattern matching.
 - *urlpath-regex* : URL-path regular expression pattern matching.
 - *method* : HTTP request method.
 - *dst-port* : TCP connection destination port.

3.2.1.11.2 Source Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.2.1 Add Source Subnet

```
add subnetip <value> subnetmask <value>
```

Add a new subentry:

- *Subnet IP* : Network IP address (I.E. 192.268.0.0).
- *Subnet Mask* : Network mask (I.E. 255.255.255.0).

3.2.1.11.3 Destination Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.3.1 Add Destination Subnet

```
add subnetip <value> subnetmask <value>
```

Add a new subentry:

- *Subnet IP* : Network IP address (I.E. 192.268.0.0).
- *Subnet Mask* : Network mask (I.E. 255.255.255.0).

3.2.1.11.4 Destination domain Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.4.1 Destination domain

```
add dstdomain <value>
```

Add a new subentry:

- *Destination domain* : Destination domain address.

3.2.1.11.5 URL Regular Expressions Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.5.1 Case sensitive

```
case-sensitive
```

Distinct use of uppercase and lowercase letters.

Default is **enabled**. Enter **no case-sensitive** in the configuration text to disable this feature.

3.2.1.11.5.2 Add URL regular expression

```
add url-regex <value>
```

Add a new subentry:

- *Regular expression* : Regular expression to match.

3.2.1.11.6 URL Path Regular Expressions Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.6.1 Case sensitive

```
case-sensitive
```

Distinct use of uppercase and lowercase letters.

Default is **enabled**. Enter **no case-sensitive** in the configuration text to disable this feature.

3.2.1.11.6.2 Add URL path regular expression

```
add urlpath-regex <value>
```

Add a new subentry:

- *Regular expression* : Regular expression to match.

3.2.1.11.7 Method Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.7.1 Method

```
type <value>
```

Entry type:

- *GET* : GET requests (Request parameters in the URL).
- *POST* : POST requests (Request parameters in the body).

Default is **GET**.



Note

This statement is mandatory and must always appear in the text configuration when this entry type is used.

3.2.1.11.8 Destination port Access List configuration

```
name <value>
```

Entry name.

3.2.1.11.8.1 Destination port

```
add dstport <value>
```

Add a new subentry.

- *Destination port* : TCP connection destination port.

3.2.1.12 Http Rules configuration

```
rules-http
```

HTTP Rules configuration.

3.2.1.12.1 Add Http Rules

```
add allow-traffic <value> operator <value> access-list
    <value> operator <value> access-list <value>
    operator <value> access-list <value>
```

Add a new HTTP rule:

- **Allow Traffic** : Allow traffic:
 - *allow* : Traffic matching this rule is allowed.
 - *deny* : Traffic matching this rule is not allowed.
- **Op** : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- **Access List** : Access list to apply:
 - *all* : Apply to all HTTP request.
 - *access_list* : Select the name of a predefined access list.
- **Op** : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- **Access List** : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.
- **Op** : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- **Access List** : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.

3.2.1.13 Cache Rules configuration

```
rules-cache
```

Cache rules configuration.

3.2.1.13.1 Add Cache Rules

```
add allow-traffic <value> operator <value> access-list
    <value> operator <value> access-list <value>
    operator <value> access-list <value>
```

Add a new cache rule:

- **Allow Traffic** : Allow traffic:

- *allow* : Traffic matching this rule is allowed.
- *deny* : Traffic matching this rule is not allowed.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.

3.2.1.14 Always Direct Rules configuration

```
rules-alwaysdirect
```

Always direct rules configuration.

3.2.1.14.1 Add Always Direct Rules

```
add allow-traffic <value> operator <value> access-list
<value> operator <value> access-list <value>
operator <value> access-list <value>
```

Add a new always direct rule:

- *Allow Traffic* : Allow traffic:
 - *allow* : Traffic matching this rule is allowed.
 - *deny* : Traffic matching this rule is not allowed.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:

- *none* : Apply to none of the HTTP requests.
- *all* : Apply to all HTTP requests.
- *access_list* : Select the name of a predefined access list.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.

3.2.1.15 Delay pools configuration

```
delaypools
```

Delay pools configuration.

3.2.1.15.1 Delay initial bucket level

```
initial-bucket-level <value>
```

Initial percent of the bucket level.

Default is **50**.

3.2.1.15.2 Add delay pool

```
add name <value>
```

Add a new delay pool rule:

- *Name* : Name of the new delay pool rule.

3.2.1.15.3 Delay pool configuration

```
name <value>
```

Entry name.

3.2.1.15.3.1 Aggregate

```
aggregatespeed <value>
```

Bandwidth aggregation speed (Bytes/s).

Default is **-1**.

3.2.1.15.3.2 Aggregate Maximum

```
aggregatemaximum <value>
```

Bandwidth with maximum aggregation (Bytes).

Default is **-1**.

3.2.1.15.3.3 Network

```
networkspeed <value>
```

Network bandwidth speed (Bytes/s).

Default is -1.

3.2.1.15.3.4 Network Maximum

```
networkmaximum <value>
```

Maximum network bandwidth (Bytes).

Default is -1.

3.2.1.15.3.5 Individual

```
individualspeed <value>
```

Individual bandwidth speed (Bytes/s).

Default is -1.

3.2.1.15.3.6 Individual Maximum

```
individualmaximum <value>
```

Maximum individual bandwidth (Bytes).

Default is -1.

3.2.1.15.3.7 Add access list

```
adddelayaccess allow-pool <value> operator <value>
  access-list <value> operator <value> access-list
  <value> operator <value> access-list <value>
```

Access list to apply:

- *Allow pool* : Allow traffic:
 - *allow* : Traffic matching this rule is allowed.
 - *deny* : Traffic matching this rule is not allowed.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.
- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.

- *Op* : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- *Access List* : Access list to apply:
 - *none* : Apply to none of the HTTP requests.
 - *all* : Apply to all HTTP requests.
 - *access_list* : Select the name of a predefined access list.

3.2.1.16 Cache Peers configuration

```
cachepeers
```

Cache peers configuration.

3.2.1.16.1 Never direct

```
never-direct
```

Never use direct connections.

Default is **disabled**.

3.2.1.16.2 Add Cache Peer

```
add name <value> hostname <value>
```

Add a new cache peer:

- *Cache Peer name* : Specify the name of the cache peer.
- *Cache Peer host name* : Specify the host name/IP of the cache peer.

3.2.1.16.3 Cache Peer configuration

```
name <value>
```

Entry name.

3.2.1.16.3.1 Cache Peer host name

```
hostname <value>
```

Cache peer host name/IP.

3.2.1.16.3.2 Cache Peer type

```
type <value>
```

Entry type:

- *parent* : Parent cache peer.
- *sibling* : Sibling cache peer.
- *multicast* : Multicast cache peer.

Default is **parent**.

3.2.1.16.3.3 Cache Peer proxy port

```
proxy-port <value>
```

Cache peer proxy port.

Default is **3128**.

3.2.1.16.3.4 Cache Peer icp port

```
icp-port <value>
```

Cache peer ICP port.

Default is **3130**.

3.2.1.16.3.5 Set proxy only option

```
proxy-only
```

Do not locally store objects fetched from the peer.

Default is **disabled**.

3.2.1.16.3.6 Set default option

```
default
```

Use this entry as the default cache peer.

Default is **disabled**.

3.2.1.16.3.7 Set no query option

```
query-no
```

Do not query requests to this cache peer.

Default is **disabled**.

3.2.1.16.3.8 Set no digest option

```
digest-no
```

Disable request of cache digests.

Default is **disabled**.

3.2.1.16.3.9 Set no netdb exchange option

```
netdb-exchange-no
```

Disable requesting ICMP RTT database (NetDB).

Default is **disabled**.

3.2.1.16.3.10 Transparent authentication

```
transparent-auth
```

Send login details received from client to this peer.

Default is **disabled**.

3.2.1.17 DNS servers configuration

```
dns-servers
```

DNS servers' configuration.

3.2.1.17.1 Add DNS server

```
add dns <value>
```

Add a new DNS server:

- *IP address* : DNS server IP address.

3.2.1.18 SSL configuration

```
ssl-config
```

SSL configuration.

3.2.1.18.1 Certificate

```
begin-multiline-command
  certificate privatekey ^---><value begin>
  ...
  ...
  ...
  <value end><---^ publickey ^---><value begin>
  ...
  ...
  ...
  <value end><---^
end-multiline-command
```

Certificate, private and public keys.

- *privatekey* : Private key.
- *publickey* : Public key.

3.2.1.18.2 Certificates cache size

```
ssl-cache-size
```

Maximum size (in MB) of the dynamically generated SSL certificates storage.

Default is 4.

3.2.1.18.3 Maximum number of SSL processes

```
max-ssl-processes
```

The maximum number of processes spawn to service SSL server.

Default is 5.

3.2.1.18.4 Delete certificates database

```
remove-cert-db
```


Remove the dynamically generated certificates database.

3.2.1.18.5 Generate certificate

```
generate-cert
```

Generate a certificate for WebCache application.

3.2.1.18.6 Trusted sites

```
rules-bypass-cert-errors
```

Bypass certificate validation errors rules.

3.2.1.18.6.1 Add Rules

```
add allow-traffic <value> operator <value> access-list
    <value> operator <value> access-list <value>
    operator <value> access-list <value>
```

Add a new SSL certificate validation error bypass rule:

- **Allow Traffic** : Allow bypass certificate validation errors:
 - *allow* : Traffic matching this rule will bypass certificate validation errors.
 - *deny* : Traffic matching this rule will not bypass certificate validation errors.
- **Op** : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- **Access List** : Access list to apply:
 - *all* : Apply to all requests.
 - *access_list* : Select the name of a predefined access list.
- **Op** : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- **Access List** : Access list to apply:
 - *none* : Apply to none of the requests.
 - *all* : Apply to all requests.
 - *access_list* : Select the name of a predefined access list.
- **Op** : Rule operator:
 - *yes* : Yes/Affirmation.
 - *no* : No/Denial.
- **Access List** : Access list to apply:
 - *none* : Apply to none of the requests.
 - *all* : Apply to all requests.
 - *access_list* : Select the name of a predefined access list.

3.2.1.18.7 Exceptions

```
rules-ssl-exceptions
```

SSL caching exceptions, passes original certificate to the client.

3.2.1.18.7.1 Add Exceptions

```
add ssl-exception <value>
```

Add a new SSL exception:

- *ssl-exception* : Select the name of a predefined access list.

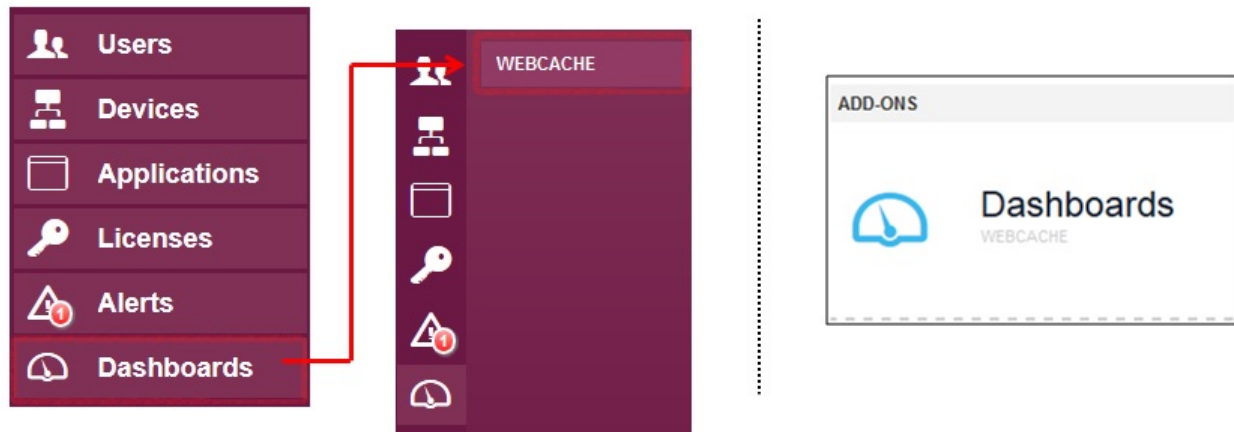
Chapter 4 WebCache Dashboard (Management Platform)

4.1 Introduction

The internal parameter values of this application are periodically sent to the management platform.

If you have a WebCache dashboard plugin license, you can associate said license with the desired Atlas i6x in the platform, providing in your user environment can access said plugin. The plugin summarizes all the data of your licensed devices with the installed WebCache application.

Fig. 4.1. Accessing the dashboard of the WebCache plugin



To access the dashboard, select **Dashboards / WebCache** from the left hand side of the management platform menu, or click on the WebCache dashboard icon located on the home screen.

4.2 Dashboard home section

This section shows an overall summary of the devices with the WebCache application assigned to this plugin. Said summary lets you check several status parameters simultaneously.



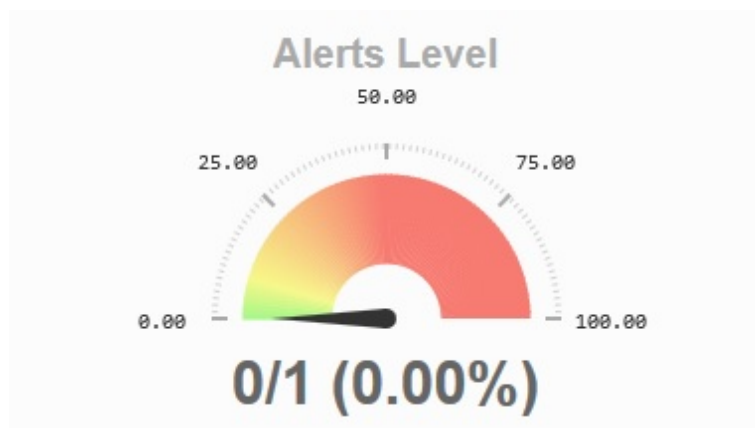
Note

The WebCache plugin home screen shows the average of the latest values received from the devices.

Please note, if a device stops sending values, the engine will use the latest value received from that device, even if it was received a while ago.

4.2.1 Alerts level

Fig. 4.2. Overall alert level

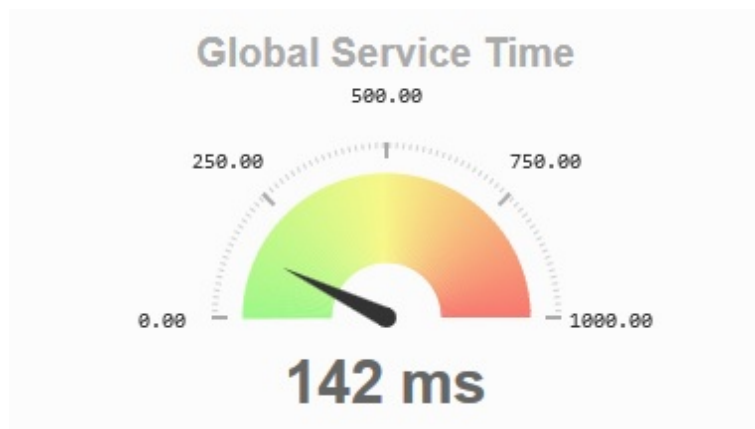


This figure shows a gauge indicating a percent of how many devices are presenting an alert. To generate alerts, **rules** must be defined. See [Rules section](#) on page 46 for more details.

Green represents low alert and red a high one.

4.2.2 Global service time

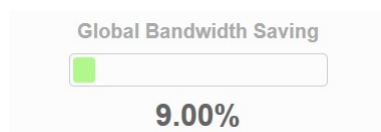
Fig. 4.3. Overall alert level



Global service time is the time between the user request and the reception of a response, from the internal cache or from the requested server. This figure is important to perceive the global health of your WebCache service in your network. This time, in milliseconds, marks real user browsing experience. This value should always be low. If it is high, such as 500ms (half a second), the users experience slow HTTP browsing speed.

4.2.3 Global bandwidth saving

Fig. 4.4. Global bandwidth saving

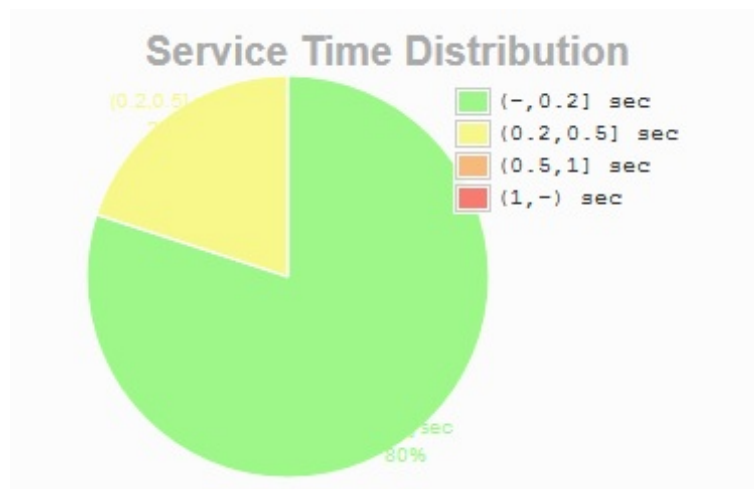


This bar shows how much (in percent) bandwidth you are saving due to cache hits. Said percentage is usually relatively small due to several factors:

- Users of a network are not usually browsing the same webpages. However, there are some scenarios where they are browsing the same contents and this value may be higher (schools, learning and training classrooms, some shops where a catalog is published to be viewed by customers through standalone kiosks...)
- A lot of web servers forbid users to cache the content of their pages (I.e. with the no-cache header in the HTTP response).
- Obviously, this value depends on the minimum and maximum cache object size configured in your WebCache applications.
- It also depends on the cache size configured.
- This value is calculated when there is HTTP traffic in your network. So when the users are not browsing and there is no web traffic, this value could present invalid values (100% bandwidth saving for example).

4.2.4 Service time distribution

Fig. 4.5. Service time distribution



The service time distribution graph shows the percent of devices depending on their service time. Said service time is measured in seconds: green represents best service time and red the worst.

Further explanations on service time are provided [Global service time](#) on page 40.

4.3 Analysis section

For a more detailed analysis on the status of your devices with WebCache, select the second tab on the dashboard, **Analysis**.

4.3.1 Main analysis screen

Fig. 4.6. Analysis main table

Analysis									
WEBCACHE STATUS									
<input checked="" type="checkbox"/> Show only last events									
	Device	Date	Median Service Time	Hits as % of all requests	Hits as % of bytes sent	HTTP requests	HTTP requests per minute	Mean Object Size	
	757/00149 Internet Access	2013-11-21 07:14:57	87 ms	32%	0.1%	33512	114.0	23.00 KB	

Page 1 of 1 Results by page: 10

Depending on the predefined rules for this plugin, devices are shown with a status icon:

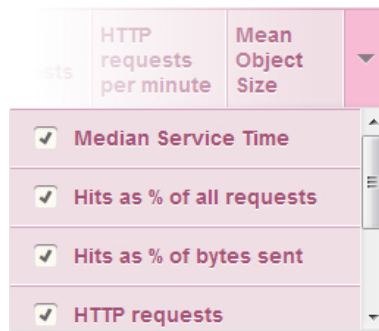
- : Device without alerts.
- : The device presents a warning alert.
- : The device presents an error/critical alert.

In the previous table, there is an entry per device with a valid license for the WebCache plugin assigned. In the example, only one device is being monitored.

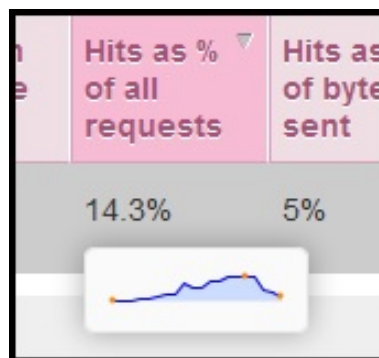
For each row in the table, in addition to the serial number and the device name, the most representative parameter values concerning WebCache appear. These parameters are:

- Date:** Last status data post for WebCache sent by the device.
- Median service time:** Average time between the user request and response.
- Hit as % of all requests:** % of requests found in the cache memory and served to the user without requesting them from the destination server.
- Hit as % of bytes sent:** % of bytes served to the users, which have not been transferred over Internet because they are in the cache.
- HTTP requests:** Total of user requests processed by WebCache.
- HTTP requests per minute:** Number of user requests processed by WebCache per minute.
- Mean object size:** This value indicates what the mean object size is. This value could be as low as the minimum object size value and as high as the maximum object size. See sections [Minimum object size](#) on page 8 and [Maximum object size](#) on page 8 for further details on the object sizes.

Click on the header on the tables column to order the rows depending on the selected field. You can also click on the arrow (on the right of the header) to select what fields to show in the table.

Fig. 4.7. Fields selection

For an instant preview of the evolution of a parameter, move your mouse pointer over the desired field and a graph with the data appears. This feature only works with numerical data.

Fig. 4.8. Preview of the data evolution

Finally, if you want to show all the events received from the device, uncheck **Show only last events**.

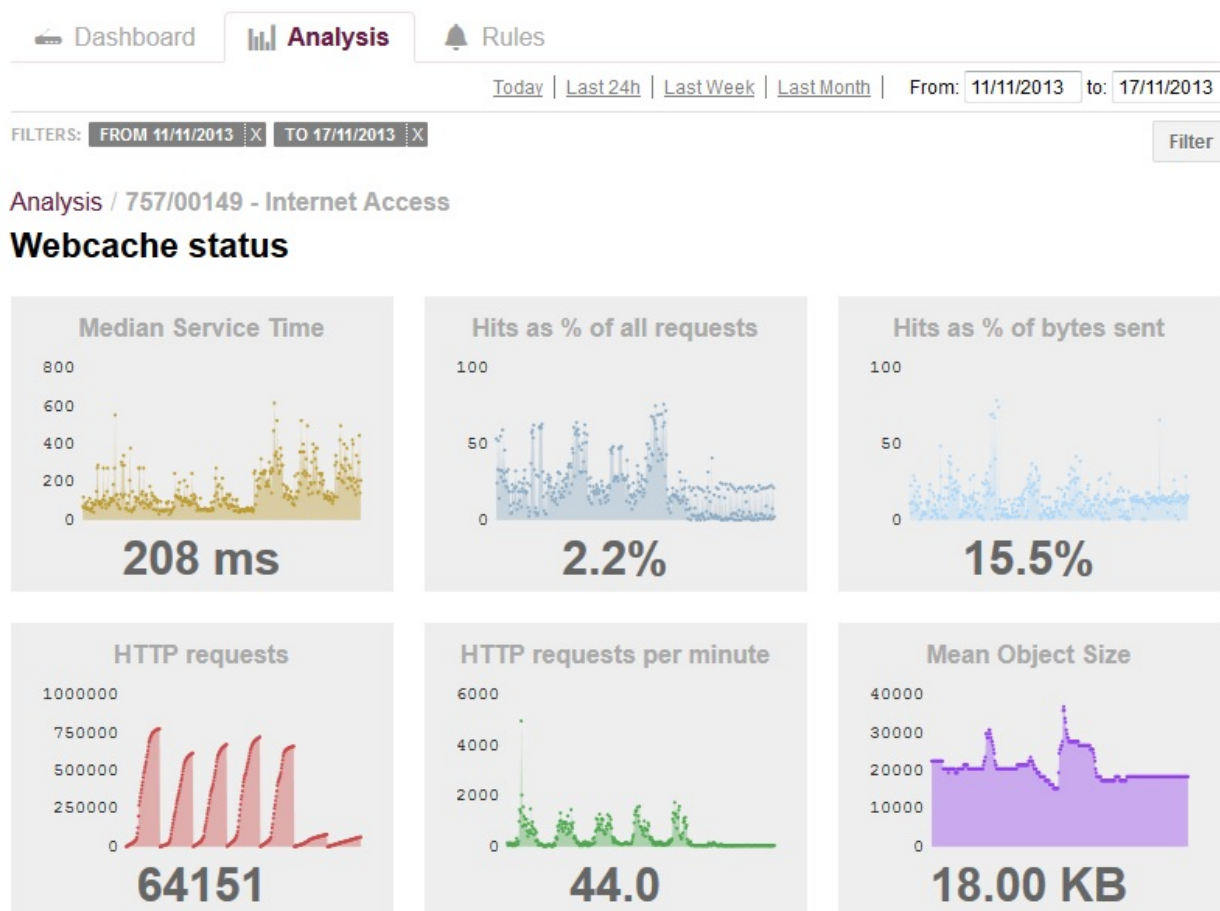
4.3.2 Analyzing data for a single device

Click on a device on the main screen of the analysis section. A new section is shown containing the data for said device.

This data can also be filtered. See [Filtering](#) on page 46 for more information.

Several graphs are generated to show the user what the evolution of a parameter is in a selected period of time. This period could be today, over the last 24h, last week, last month or a user defined period between two dates.

Fig. 4.9. Analyzing a single device



This feature allows the user to check if the WebCache installed on this device is useful and operating correctly. In this example, the device corresponds to a typical company WebCache, where a week interval has been selected. As you can see in the figure, during the first five days (from Monday to Friday), the graphs are showing more web connection usage than the final two days (Saturday and Sunday). The graphs are similar for each day, for example, HTTP requests maximum grows at the same speed per day (except at the weekend).

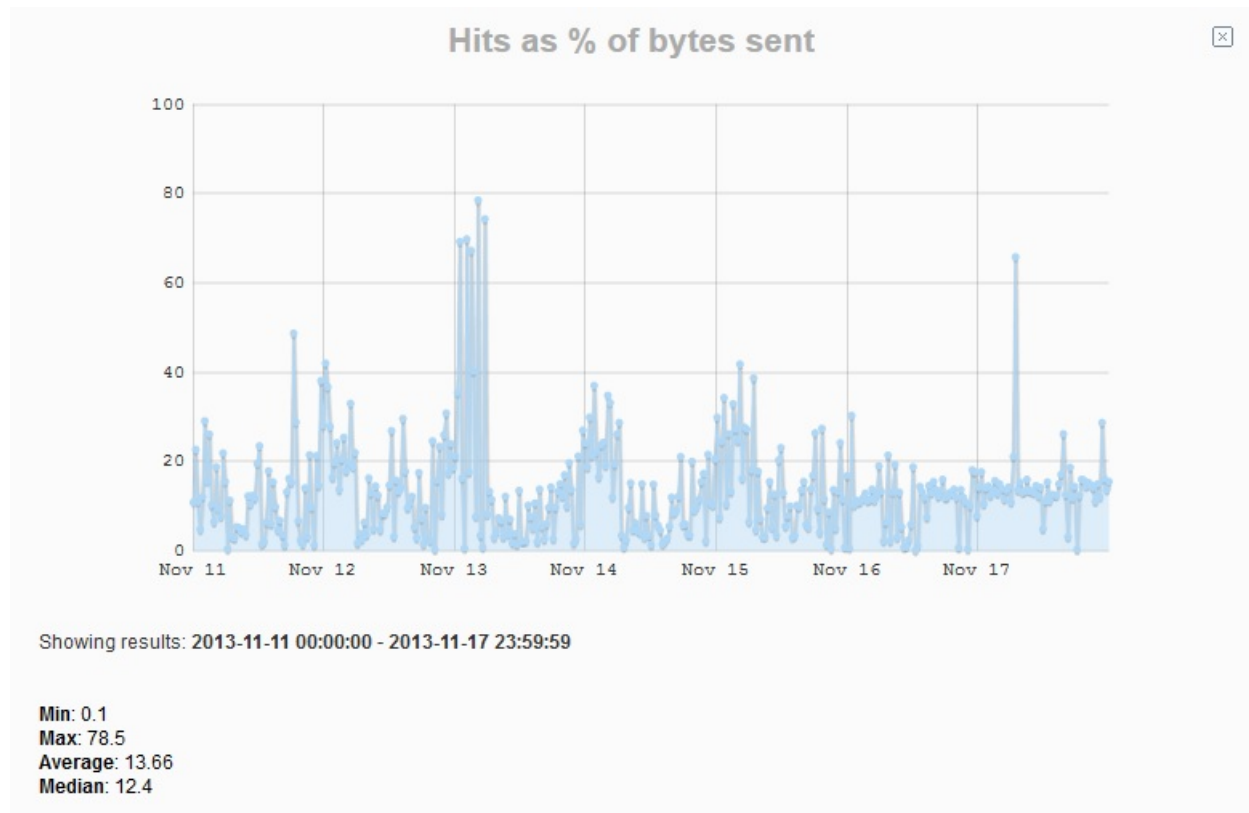


Note

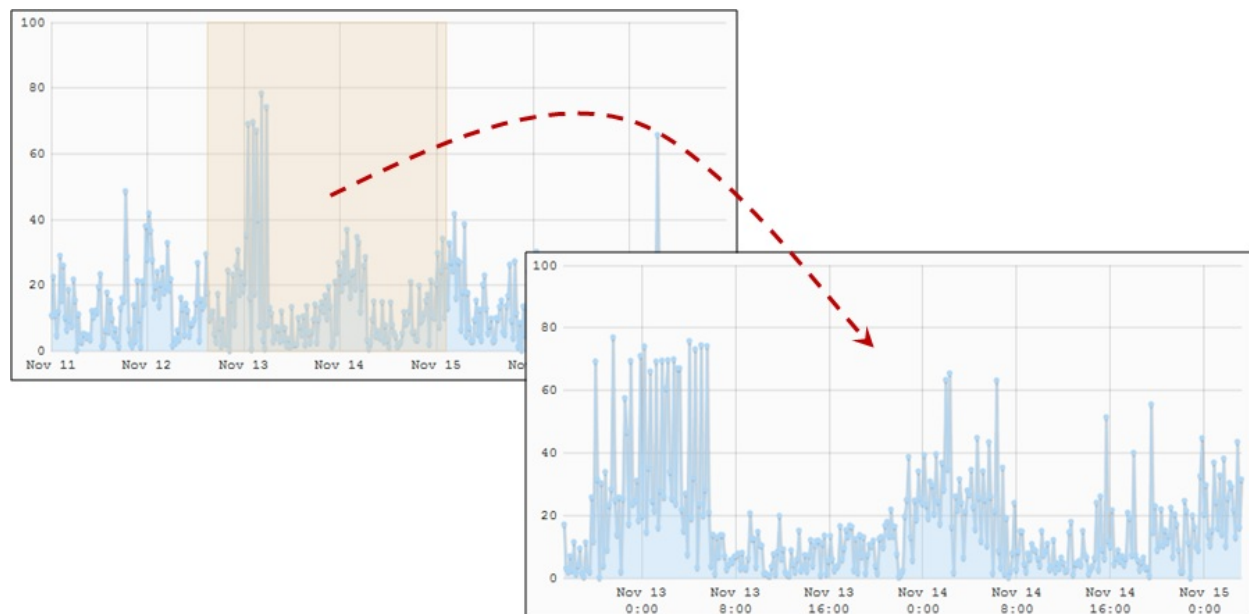
The HTTP requests graph is accumulative. This is why this parameter is automatically reinitialized per day to allow the user to compare the differences between days.

4.3.2.1 View a more detailed graph

You can click on a graph to see more details. When a graph is clicked, a new screen is shown with the entire graph and some representative values. For this example, we have selected the **Hits as % of bytes sent** graph.

Fig. 4.10. Hits as % of bytes sent detailed graph

When a detailed graph is shown, you still can apply filters to view more detailed information. Because the graph is interactive, it is possible to apply **from date** and **to date** filters by clicking on the graph and selecting an interval. The system automatically refines the results with the new filters.

Fig. 4.11. Ingraph selection

At the bottom of the graph, depending on the selected graph, several parameters are shown as a median, the maximum and minimum value, the average... All said data is calculated depending on the selected time interval.

To go back to the graphs selection screen, use the close icon

**Note**

The features on the detailed graph, such as the graphical time interval selection, are the same as the other detailed graphs published on the management platform. For example, the device health monitoring (HDD, Memory and CPU).

4.4 Rules section

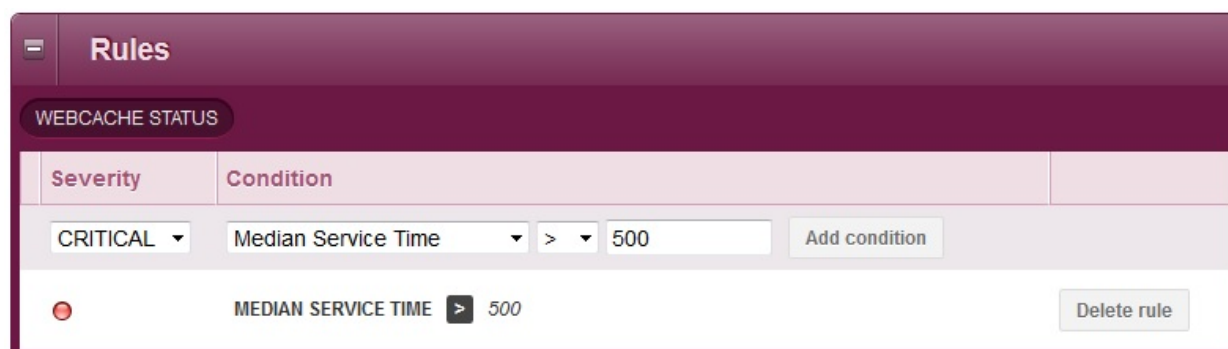
Define rules to generate alerts depending on the current values of the WebCache application, installed in your devices, with this dashboard plugin enabled.

To define a rule, select the severity grade and a condition to trigger the alert:

- Severity:
 - Warning.
 - Critical.
- Condition:
 - Left conditional:
 - Mean service time.
 - Hits as a % of all requests.
 - Hits as a % of bytes sent.
 - HTTP requests.
 - HTTP requests per minute.
 - Mean object size.
 - Conditional operator:
 - > (greater than).
 - < (lower than).
 - = (equal to).
 - != (not equal to).
 - Right conditional: <value>

For example, if you wish to trigger an error/critical alert when users, behind the proxy running on a device, are receiving their responses with a mean service time greater than 500ms, create the following rule:

Fig. 4.12. Defining rules



4.5 Filtering

On the home screen and analysis screen (on the dashboard) you can filter devices, tags or groups of devices to show in your graphs.


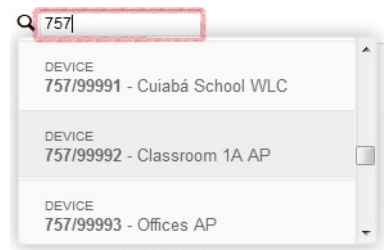
Use the text box represented by  (magnifying glass). When you type any text in said box, the system automatically searches for devices, groups or tags matching this text. Subsequently, a list of the matched objects is shown, allowing the user to select one of the items from the list.

Fig. 4.13. Devices/groups/tags filtering



When an item on the list is selected, a new tag is added to the filter bar. Note the different colors for each tag depending on the type of the selected filters (devices, groups or tags).

Fig. 4.14. Filter bar



To remove a filter, click on the "X".

4.5.1 Selecting a time interval



Note

This only applies to the *analysis screen*.

If you want to filter using a time interval, use the time selection interface on the right of the filter textbox.

Fig. 4.15. Selecting time intervals for filtering



You can select the following time intervals:

- Today.
- Last 24 hours.
- Last week.
- Last month.
- Between two dates (use the text boxes "from" and the "to" to select a date).

When a time filter is selected, a new filter tag is inserted in the filter bar. Remember it is possible to remove the selected filter by clicking on the filter tag **X** button.

4.6 Assigning plugin licenses

First, access the license section in the management portal using the left hand menu option or the home icon *Licenses*.

Here there is a table containing the available licenses: select the *plugin: Dashboard WebCache* license,



Fig. 4.16. Dashboard WebCache licenses

Manage licenses			
Type	Validity	Available	Status
Managed devices	365 days	0 of 1 	 Valid license 
Applications <i>webcache</i>	No expiration	3 of 5 	 Valid license 
Managed devices	365 days	9 of 10 	 Valid license 
Applications <i>siemensor</i>	No expiration	9 of 10 	 Valid license 
Plugin <i>Dashboard Webcache</i>	365 days	10 of 10 	 Valid license 

If you have available unassigned licenses, a message at the top of the table appears with a link to assign said licenses to devices. If you click on the link, a text box where you can search for devices, groups or tags, is shown. Said search box behaves in the same way the device filtering search field. See the [Filtering](#) on page 46 for more information.

Click on **Assign devices** to confirm your selection. This action cannot be undone.

Fig. 4.17. Dashboard WebCache assigned licenses

Devices assigned to the license		
Assign date	Serial Num.	Name
8 available licenses Assign Devices		
 21/11/2013 16:14:25	757/00128	Main WebCache device
 21/11/2013 16:16:31	725/00103	Secondary WebCache device
Page 1 of 1		Results by page: 10 



Note

If you do not have licenses available, please contact your distributor to purchase a new set of Dashboard WebCache Licenses. These licenses have an expiry date. The plugin cannot access any device without a valid license being assigned.

Appendix A General HTTP Proxy configuration

Because the internal HTTP proxy engine is used for several purposes (by the application host), it is installed as a general package in the system core. This is a common engine for several applications such as URL content filtering feature for Security application and the WebCache application.

These general engine options should be configured outside WebCache configuration (they are always available in the application host) whether WebCache has been installed or not.

Fig. A.1. General HTTP proxy configuration

The screenshot displays the 'HTTP Proxy configuration' page. On the left, a sidebar lists navigation categories: General, Management, Monitoring, Traffic Control, Router Settings, HTTP Proxy (highlighted), HTTP Server, and E-mail. The main panel has two tabs: 'Configuration' (active) and 'WebCache'. The 'HTTP Proxy configuration' section includes the following settings:

- Proxy ports: 80
- Transparent:
- First Internal port: 8060
- Second Internal port: 3128
- Forward ports: (empty)
- Internal HTTP proxy forward port: 3127
- Enable HTTPS Proxy:
- HTTPS Proxy ports: 443
- Internal HTTPS port: 3129
- Disable log:
- Remote log server IP: (empty)
- Enable remote log server:
- Interval: 900
- Run Always:

Buttons for 'Help', 'Show status', and 'Modify' are visible.

A.1 HTTP proxy web configuration

A.1.1 Configuring an HTTP transparent proxy

To configure a transparent proxy so nothing changes for end user browsers, enable checkbox **Transparent** and reference proxy port **80**.

HTTP Proxy port is 80 by default. This is used to internally redirect RTMPT traffic (RTMP video over HTTP) and web requests.

Please do not modify the proxy port default value if you want to use transparent proxy.

The port configured in **First internal port** is only used for transparent mode, internally redirecting HTTP traffic. This parameter can be modified as you may be using an application (installed in the application host) listening in that port for some operations. This is an advanced setting and should not be modified without good reason,

A.1.2 Configuring an HTTP non transparent proxy

Disable **transparent** checkbox and establish the desired listening port in the **Second internal port** value. This is the port you have configured (in the browsers) for the users to allow Internet browsing.



Note

Do not use **First internal port** on your browser configuration when configuring a non transparent proxy!

A.1.3 Configuring an HTTP and HTTPS transparent proxy

First, follow the instructions given in [Configuring an HTTP transparent proxy](#) on page 49 to create a transparent proxy HTTP.

If you also want an HTTPS proxy, check the **Enable HTTPS Proxy** checkbox. With HTTPS proxy enabled, WebCache must be properly configured with a certificate. If WebCache is enabled but the certificate is incorrect (or there aren't any) and you enable HTTPS Proxy, a warning message appears on the configuration screen. Please note, despite the fact HTTPS Proxy is in a different section to WebCache, enabling the former without a certificate doesn't make sense.

There are two ports related to HTTPS Proxy: **HTTPS proxy port** and **HTTPS internal port**. Default for **HTTPS Proxy port** is 443 (standard port for HTTPS traffic). Traffic transmitted through said port is diverted to the applications host and then redirected to the **Internal HTTPS port**, where the cache engine is listening.

A.1.4 Configuring an HTTP and HTTPS non transparent proxy

Follow the instructions given in [Configuring an HTTP non transparent proxy](#) on page 49 section to configure a non transparent proxy HTTP.

There are two options to configure your browser and HTTP Proxy:

The first: configure the browser to use an IP address for proxy (applications host address) and two different ports: one for HTTP traffic and one for HTTPS traffic. Here, configure proxy HTTP in said browser with the **Second internal port** value and the browser proxy HTTPS port with **Internal HTTPS port**. Check the **Enable HTTPS proxy** checkbox.

The second option is to configure the browser to use only one IP address and port as proxy. Said address must be the application host address. Selected port must be the same as the **Second internal port** and **Internal HTTPS port**. Check the **Enable HTTPS proxy** checkbox

Please note, even if you are configuring browser proxy parameters to transmit through application internal ports, the ports traffic appearing in the **Proxy Ports** and **HTTPS Proxy Ports input** boxes will be diverted to the applications host (**traffic-control** enabled). If you don't want this to occur, change the ports values (80 and 443 by default) to different ones.



Note

HTTPS Proxy Ports are only diverted if **Enable HTTPS proxy** is checked.

A.1.5 Configuring application behind a corporative proxy for caching external traffic

With the right configuration, WebCache application is able to work behind a corporative proxy and cache traffic going through it. To do this, create a cache-peer (see [Cache peers: Creating a simple hierarchy of neighbor caches](#) on page 17 section) and configure it with the following options:

- never direct: *enable*
- Cache peer host name: *the corporative proxy address*
- Cache peer type: *parent*
- Cache peer proxy port: *the port of the corporative proxy*
- Cache peer icp port: *0*
- Set proxy only option: *disable*
- Set default option: *disable*
- Set no query option: *enable*
- Set no digest option: *enable*
- Set no netdb exchange option: *enable*

- Transparent authentication: *enable*

Fig. A.2. Configuring application behind a corporate proxy

The screenshot shows the 'WebCache' configuration page. On the left is a navigation menu with options like 'Web Cache', 'Access List', 'Http Rules', 'Cache Rules', 'Always Direct Rules', 'Delay pools', 'Cache Peers', 'proxycorp', 'DNS servers', and 'SSL'. The main area is titled 'WebCache' and contains a list of configuration items:

- Cache Peer host name: 192.168.212.76
- Cache Peer type: parent
- Cache Peer proxy port: 8080
- Cache Peer icp port: 0
- Set proxy only option:
- Set default option:
- Set no query option:
- Set no digest option:
- Set no netdb exchange option:
- Transparent authentication:

A 'Modify' button with a double-headed arrow icon is located at the bottom right of the configuration area.

Now, you have several configuration options depending on what traffic you want to cache. The above parameters must be set to an option.

A.1.5.1 Caching internal traffic

If you have a corporate proxy in your network but only want to cache internal traffic, do not configure the cache-peer: configure WebCache as if corporate proxy doesn't exist.

Continue with the instructions given in previous sections in this appendix

A.1.5.2 Caching external traffic

To cache external traffic only, configure the **Proxy ports** WebCache parameter with the corporate proxy port to divert said port traffic to the application. Subsequently, also configure the **Forward ports** WebCache parameter with said corporate proxy port. When is port is added to the **Forward ports** parameter, traffic for said port is redirected to the **Internal HTTP proxy forward port** (you can modify this port if there are any applications or services already listening on it). This configuration doesn't need to modify anything in the client browser. As this is a transparent configuration, the **Transparent** option must be checked. The **Enable HTTP proxy** checkbox should be unchecked.

Through this configuration, ebcache can cache external HTTP traffic but not external HTTPS. External HTTPS traffic, however, is transparently routed to its destination.

A.1.5.3 Caching internal and external traffic

If you want to cache internal and external traffic enter two ports in the Proxy ports text-box: port used for internal traffic (usually 80) and the corporate proxy port used to route external requests. The list of ports must be entered separated by commas. If you enter repeated ports or blank spaces between commas, an error message will appear. Subsequently, in the **Forward ports** parameter, enter the port (from the **Proxy ports** list) corresponding to the corporate proxy.

You can also cache internal HTTPS traffic by marking the **Enable HTTPS proxy** option. WebCache can't cache external HTTPS traffic, however it's routed transparently.

Lastly, indicate the port, in the WebCache configuration, to use for traffic to the corporate proxy and the port to send traffic to the internal network. To do this, create a destination port type access list (**dst-port**); then enter the internal traffic ports in the access list (usually 80 and also 443 if caching HTTPS). For further information, please see [Access lists: Defining list of users, machines, domains etc.](#) on page 9 Subsequently, enter **Always Direct Rules** (in WebCache configuration) and add the created access list to the table. This tells the WebCache that traffic bound to said ports (internal traffic) is direct i.e. doesn't pass through the configured parent cache peer (corporate proxy)

**Note**

Parameters *Proxy ports*, *Forward ports* and *HTTPS proxy ports* admit several ports. Enter said ports as a list of elements separated by commas, e.g.

```
80, 8080, 8081
```

A.1.6 Managing HTTP proxy logs

A.1.6.1 Disable logs

Enable the **Disable logs** checkbox so no logs, relative to the HTTP proxy, and other related applications such as WebCache, are stored in the application host internal storage.

A.1.6.2 Sending logs to a remote server

Internal cache logs are only locally accessible if you have console access. This is only possible if your Atlas i6x has a development license. If you have said license, access through an SSH terminal connected to the application host IP. The logs are stored in the directory

```
/var/log/squid3
```

However, there is an option to configure the engine to send access logs to a remote server, using a remote log server such as *syslog-ng* or *rsyslog*.

If you have a remote log server configured in your network, you may want to enable **Enable remote log server** option and establish the IP address for your remote log server in the **Remote log server IP** parameter.

This feature allows the administrator to store all logs in a central server and to process them with his favorite software. There is a lot of software available to process access log data, and the majority of them are opensource.

Because the WebCache application engine is based on Squid3 opensource software, all access logs are stored and transferred to remote servers in a well known format for the thousands of applications used for passing them.

**Note**

The application host uses port UDP 514, the standard port for syslog remote logging.

A.1.6.3 Changing the dashboard monitor interval

The application periodically sends information on cache constants to the management platform to be analyzed. Default interval for data sending is 15 minutes (900 seconds).

From time to time sending more information is useful if more accuracy is needed in central servers. However, but if this value is lower, it may overload both the network and management platform, so use it carefully.

You may not want, however, to generate monitoring data over your network. In this case, there is no way to disable data transmission, but you can always use a higher value, 24h for example. (86400 seconds).

**Note**

This option is configured in seconds.

A.1.7 Always run the internal HTTP proxy

To enable internal HTTP proxy (even if you do not have applications installed using this feature), check this option. This forces the application host to reconfigure access lists in the communications configuration (CIT) if **Traffic control** is correctly configured. All traffic corresponding to configured ports for this section, is redirected to the application host.

**Note**

We do not recommend you enable this option if you do not have HTTP applications installed in the application host.

A.2 Text configuration commands

In this section all configuration directives, allowed in the application text configuration, are described.

**Note**

Configuration directives should be sent in a single text file to the device through the Atlas i6x management portal.

If a statement does not appear in the configuration text, the engine takes the default value.

A.2.1 HTTP Proxy configuration

```
http-proxy
```

Top level configuration directive.

A.2.1.1 Proxy ports

```
proxyport <value>
```

This parameter allows a user to specify the proxy port or a list of proxy ports. If the value is a list of ports, the elements must be separated by commas (,,). The list must be between quotation marks ("").

Default is **80**.

A.2.1.2 Transparent

```
transparent
```

This parameter enables the proxy transparent mode.

Default is **disabled**.

A.2.1.3 First internal port

```
firstinternalproxyport <value>
```

First internal proxy port.

Default is **8060**.

A.2.1.4 Second internal proxy port

```
secondinternalproxyport <value>
```

Second internal proxy port.

Default is **3128**.

A.2.1.5 Forward ports

```
externalforwardproxyport <value>
```

This parameter allows a user to specify a **Forward port** or a list of Forward ports. If the value is a list of ports, the elements must be separated by commas(.). The list must be between quotation marks (""). This parameter is empty by default.

Default is

A.2.1.6 Internal HTTP proxy forward port

```
forwardproxyport <value>
```

Internal HTTP proxy forward port.

Default is **3127**.

A.2.1.7 Enable HTTPS proxy

```
enable-https-proxy
```

Enables the HTTPS proxy.

Default is **disabled**.

A.2.1.8 HTTPS proxy ports

```
sslproxyport <value>
```

This parameter allows a user to specify an HTTPS proxy port or a list of HTTPS proxy ports. If the value is a list of ports, the elements must be separated by commas. The list must be between quotation marks ("").

Default is **443**.

A.2.1.9 Internal HTTPS port

```
internalsslproxyport <value>
```

Internal HTTPS port.

Default is **3129**.

A.2.1.10 Disable logs

```
disable-log
```

Enable **Disable logs** so no logs, relative to the HTTP proxy and other related applications such as WebCache, are stored in the application host internal storage.

Default is **disabled**.

A.2.1.11 Remote log server IP

```
remote-log-server-ip <value>
```

Remote log server IP address.

A.2.1.12 Enable remote log server

```
remote-log-server-enable
```

Enable the use of a remote log server.

Default is **disabled**.

A.2.1.13 Monitor interval

```
interval <value>
```

Interval for sending monitoring information to the server.

Default is **900**.

A.2.1.14 Run always

```
runalways
```

Enable to always run internal HTTP proxy.

Default is **disabled**.

Appendix B Certificates help for clients

In this section some common operations related to handling clients certification are described, step by step.

B.1 Installing a certificate in the client's system

How to install a certificate in the client's system.

B.1.1 For Windows

- (a) Obtaining the certificate public key:
 - (a) If a certificate has been generated using the WebCache facility, you'll find the WebCache Root Certification Authority certificate public key on the SSL configuration status page. You can also use the WebCache certificate public key (in SSL configuration). This allows you to trust the application stored in *that* particular device, but not all WebCache applications
 - (b) If the certificate has been imported, ask the certificate generator for the public key.
- (b) Copy the certificate public key to a file and save it with a **.crt** extension.
- (c) Execute the **.crt** file by double clicking on it and follow the instructions to install the certificate in your Windows system. You can install it in the Trusted Root Certification Authorities certificate storage.

B.1.2 For Linux

- (a) Obtaining the certificate public key:
 - (a) If a certificate has been generated using the WebCache facility, you'll find the WebCache Root Certification Authority certificate public key on the SSL configuration status page. You can also use the WebCache certificate public key (in SSL configuration). This allows you to trust the application stored in *that* particular device, but not all WebCache applications.
 - (b) If the certificate has been imported, ask the certificate generator for the public key.
- (b) Copy the certificate public key to a file with extension **.crt**.
- (c) Import the certificate in your keystores using `keytool` utility.

```
keytool -import -trustcacerts -keystore /usr/lib/jvm/java-1.6.0-openjdk-amd64/jre/lib/security/cacerts -storepass mypass -
```

You have to replace in the example the `cacerts` path with yours and "mypass" with the password of your keystore.

- (d) If you changed the certificate and you want to replace it with the new one, delete the old using command:

```
keytool -delete -alias webcache -keystore /usr/lib/jvm/java-1.6.0-openjdk-amd64/jre/lib/security/cacerts -storepass mypass -
```

And then, import the new one as in step 3.

- (e) Check the certificate has been imported successfully by using command:

```
keytool -list -keystore /usr/lib/jvm/java-1.6.0-openjdk-amd64/jre/lib/security/cacerts
```

This shows you all certificates stored in the keystore.

B.2 Import a certificate in the client's browser

How to import a certificate in the client browser.

B.2.1 For Internet Explorer

If you have installed the certificate in your Windows system (as described in [For Windows](#) on page 56), you don't need to configure anything else.

B.2.2 For Firefox

- (a) Go to the **preferences** section, **advanced** configuration, **Certificates** tag.
- (b) Click on **See certificates**.
- (c) Select **Authorities** tag.
- (d) Click **Import** and select the **.crt** file with the public key from your file tree.
- (e) When you are asked, tell the system you trust said CA.
- (f) Restart Firefox.

Appendix C Troubleshooting

Below, there are several common situations described and their solutions. They may help you search for issues when your users cannot browse behind an Atlas i6x WebCache application.

We assume your WebCache application is installed in the Atlas i6x Application Host and the device is accessible over the network.

C.1 Symptom: Your WebCache application is not caching anything

If you suspect your application is not caching because you do not have Internet access or said access is restricted, please carry out the following:

- (a) Check your router is correctly configured to send HTTP traffic to the Application Host. If you enter Atlas i6x CLI, in dynamic configuration (process 5), there should be a **traffic-divert** command in the feature vli section:

```
application traffic-divert access-list 101
```

There are two ways to insert this line:

- *Configuring your application to automatically configure the communications configuration:*

Go to Application Host web configuration (by default user 1234, password 1234) and select **Configuration**.

Then select **Traffic control**. Then configure the credentials to allow the Application Host to access your communications configuration.

Ensure your access lists are unique and the numbers you have chosen are not in use. Check your credentials are valid (your system administrator can configure them) and this feature is enabled:

Fig. C.1. Traffic control

To find out the status of said feature, click on **Show**. If this section has been correctly configured, **Router management connection** should be **Established**.

Fig. C.2. Traffic control status

- *Configuring the communications configuration manually:*

Enter CLI communications and manually configure your VLI feature. Information on this is in the Atlas i6x user guide.

To summarize the necessary directives you need to get the WebCache application running, refer to the following configuration entries:

```

...

feature access-lists
; -- Access Lists user configuration --
access-list 101
description DONT_MODIFY_vli_traffic_divert
;
entry 1 description httpproxy_pre
entry 1 default
entry 1 permit
entry 1 destination port-range 80 80
entry 1 protocol tcp
;
entry 2 default
entry 2 deny
;
exit
;
exit

...

feature vli
; -- VLI configuration --
application address <app_host_ip_address_value>
application traffic-divert access-list 101
application management-platform address
    <app_mng_platform_ip_value>
application dns-server <dns_server_ip>
;
exit
;
...

```

- (b) Check your application is enabled in the WebCache configuration main window:

Fig. C.3. Enable WebCache application

Web Cache configuration Show status ⇌

▪ Visible host name	<input type="text" value="WebCache"/>
▪ Manager email	<input type="text" value="fcamacho@teldat.com"/>
▪ Cache size (MB)	<input type="text" value="2000"/>
▪ Minimum object size (KB)	<input type="text" value="0"/>
▪ Maximum object size (KB)	<input type="text" value="4096"/>
▪ ICP port	<input type="text" value="3130"/>
▪ Enable memory pools	<input checked="" type="checkbox"/>
▪ Memory pools limit (MB)	<input type="text" value="5"/>
▪ Remove cache directory	<input type="checkbox"/>
▪ Enable web cache	<input checked="" type="checkbox"/>

- (c) Remove your cached contents. They may have vanished or broken if your device has been accidentally rebooted, due to a power outage for example:

Fig. C.4. Remove cached contents

Web Cache configuration
[Show status](#)

▪ Visible host name	<input type="text" value="WebCache"/>
▪ Manager email	<input type="text" value="fcamacho@teldat.com"/>
▪ Cache size (MB)	<input type="text" value="2000"/>
▪ Minimum object size (KB)	<input type="text" value="0"/>
▪ Maximum object size (KB)	<input type="text" value="4096"/>
▪ ICP port	<input type="text" value="3130"/>
▪ Enable memory pools	<input checked="" type="checkbox"/>
▪ Memory pools limit (MB)	<input type="text" value="5"/>
▪ Remove cache directory	<input type="checkbox"/>
▪ Enable web cache	<input checked="" type="checkbox"/>

[Modify](#)

- (d) If your proxy is configured to be a non transparent proxy, ensure your browsers are configured to access Internet using your WebCache application.

Please refer to your browser for help to configure a proxy connection.

- (e) Check the **HTTP Proxy** configuration in the Application Host configuration tab.

Ensure HTTP proxy port is 80, when configuring a transparent proxy, and the second port if you are configuring a non-transparent proxy.

- (f) You must have at least one DNS server configured in your application host or WebCache application. See [DNS servers](#) on page 21 for further information.
- (g) If you have configured cache peers, check there is at least one neighbor with **Never direct** disabled. See [Cache peers configuration](#) on page 18.
- (h) Check your HTTP rules, Cache Rules and Always Direct Rules. Refer to [HTTP rules: What are the requests to process by the cache engine?](#) on page 12 , [Cache rules: Deciding what to cache](#) on page 13 and [Always direct rules: forward request without using peers](#) on page 14.
- (i) Finally, check the error you are receiving in your browser. If said error does not contain information on the application (see [error figure](#) in [General application parameters](#) on page 7), then it may be different problem.

The WebCache application always shows its own error. Nevertheless, it could present other errors when there is congesting or overload. In this case, check that you are not generating more traffic than router maximum, especially if you are using other applications in your Application Hosts.

C.2 Symptom: The log server is not receiving the browsing data

- Enter the configuration tab in the application host web. In HTTP Proxy submenu, check **Remote log server IP** is correctly configured and option **Enable remote log server** is enabled.
- Is the server reachable from the network where the router is connected? Try to execute a `ping` to the server from a host in this network.
- If the remote log server is defined by a host name instead of an IP address, ensure you have correctly configured DNS servers in your device.
- Finally, remember access log information is transmitted by the Atlas i6x to the remote log server using port **UDP 514**.

C.3 Symptom: After importing or generating a new certificate you cannot access to HTTPS sites

The WebCache certificates database may contain certificates signed by an old WebCache certificate. Carry out the following to remove the certificates database.

- (a) Enter in the SSL section of the WebCache configuration.
- (b) Check **Remove certificates database** checkbox.
- (c) Click **modify** and wait. This operation may take several seconds before you can browse normally.