About URL rewriting

URL rewriting is a function of EZproxy. Users have several options for how URLs will be written, learn more about the options and how to select one.

Background

For the purpose of this document, the term proxy server is narrowed in scope to refer only to proxy servers between web browsers and web servers.

One of the original goals of web proxy servers was to accelerate access and conserve bandwidth. Browsers were configured to direct all their outgoing requests to the proxy server. The proxy server received all requests for documents, and when possible would maintain local copies of these documents after they were retrieved. As people accessed common web sites, the proxy server could immediately return documents from its own cached copies, avoiding the need to send every request out over the institution's link to the Internet. Not only were these common documents retrieved faster, but requests for other documents were also processed more quickly since the Internet link was no longer bogged down retrieving the common documents.

When a browser makes a request through a proxy server, the web server receives the request from the proxy server, not the user's workstation. When remote users direct their browsers to use a proxy server, their requests are likewise seen by the web server as coming from the proxy server. Since many library databases permit access based on the IP address of the request, the use of the proxy server provides automatic authentication for remote users.

From browser to proxy

There are three major ways that determine if a browser will use a proxy server: transparent proxying, browser configuration, and URL rewriting.

In some network configurations, a network router may be configured to reroute all web traffic through a proxy server. This has the advantage that no browser configuration is required. When such changes are made without prior announcement, they may also cut off access to specific databases if the proxy server's IP address has not been given to the remote database vendor.

Users may be required to configure their browsers to use a proxy server. For machines at your institution, you may configure all browsers to use your proxy server for all web access. For remote users, it is common to use an autoconfiguration file, which tells the users' browsers which web sites should use your proxy server, so that only requests to your database vendors are routed through your proxy server. There are three main problems with using a standard proxy server for remote users: browser configuration directions must be provided for every version of every browser on every operating system you want to support, users may be unable to access your proxy server due to proxy servers at their own sites or firewall restrictions, and users who access databases from multiple institutions must change the browser each time they want to use your proxy server instead of another institutions proxy server.
URL rewriting proxy servers such as EZproxy require no browser configuration. These proxy servers change the URLs in web pages so that requests for web pages from licensed databases are routed back to the proxy server.

**URL rewriting strategies**

URL rewriting proxy servers use specific strategies to map "real URLs" so the user's browser automatically directs all requests back to the proxy server. For these mapping examples, the rewriting proxy server is called ezproxy.yourlib.org and the database servers are called www.somedb.com, search.somedb.com, and www.otherdb.com.

Assume that the initial URL to access Some Database is:

```
http://www.somedb.com/index.html
```

One way to map this URL is:

```
```

Here, the mapping is to remove the "http://" from the beginning of the "real URL" and to put http://ezproxy.yourlib.org/ in front. This mapping is simple to follow, and was the first attempted in the first prototype of EZproxy. Unfortunately, this approach does not adapt itself well to certain standards for web servers, including the handling of certain cookies, so it is of limited value and is inadequate to meet general purpose proxying of library databases.

For any rewriting solution, a wider range of options can be supported if everything after the host name portion of the URL can be left as-is (the /index.html in this example). Therefore, it is preferable for proxy solutions to manipulate only the host name portion of the URL (www.somedb.com in this example).

The original strategy used in EZproxy was to map each combination of web server host name/port number to a unique port number on the EZproxy server. Under this strategy, www.somedb.com might be assigned 2050, www.somedb.com:180 might be assigned 2051, and search.somedb.com might be assigned 2052. Under this scheme, since 2050 represents www.somedb.com, our sample URL of:

```
http://www.somedb.com/index.html
```

is mapped to:

```
```

This strategy works well, but poses a few difficulties. Corporate sites often block access to these non-standard port numbers (web server normally use only ports 80 and 443). At institutions that run EZproxy, it may be difficult to configure firewalls to support the range of ports required.

**Proxy by hostname**

To overcome the restrictions of port mapping, proxy by hostname was introduced. Instead of using port numbers to represent remote web servers, unique host names are used. Our sample URL of:

```
http://www.somedb.com/index.html
```

is mapped to:

```
```

This strategy works well, but poses a few difficulties. Corporate sites often block access to these non-standard port numbers (web server normally use only ports 80 and 443). At institutions that run EZproxy, it may be difficult to configure firewalls to support the range of ports required.
maps to:


This mapping allows EZproxy to operate using only the standard web server ports. This eliminates the non-standard port firewall issues for corporate sites, and simplifies firewall configuration at your institution’s site since only one or two ports must be allowed through to the EZproxy server. It also reduces the resource requirements on your EZproxy server.

To support host-name based rewriting, your domain name service (DNS) administrator must make two entries. If your EZproxy server used the IP address 192.168.10.15 and was named ezproxy.yourlib.org, the two entries would look like:

- ezproxy.yourlib.org. IN A 192.168.10.15
- *.ezproxy.yourlib.org. IN A 192.168.10.15

If you manage DNS on Windows, see Proxy By Hostname Windows DNS Configuration for the steps required to create these entries.

The first entry is standard, but the second entry is unusual. This second entry indicates that any host name that ends in .ezproxy.yourlib.org should be associated with the IP address 192.168.10.15. This wildcard mapping is the key to allowing host name proxying to work. The entry is perfectly legitimate and is part of RFC 882, one of the original RFCs for domain name service. If your DNS server is unable to support this entry, your DNS administrator may be able to delegate control of this name to another DNS server that supports this entry, perhaps running on the same server as your EZproxy server.

For any questions regarding DNS configuration, please send an email to support@oclc.org.

**Converting from proxy by port to proxy by hostname**

Instructions for converting from proxy by port to proxy by hostname appear at Proxy By Hostname Configuration.