

Httpproxy

Matthias Hopf

Copyright © Copyright1995 Matthias Hopf

COLLABORATORS

	<i>TITLE :</i> Httpproxy		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Matthias Hopf	July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Httpproxy	1
1.1	Httpproxy - Contents	1
1.2	Httpproxy - Introduction	1
1.3	Httpproxy - What is a proxy at all?	2
1.4	Httpproxy - Features	3
1.5	Httpproxy - Requirements	4
1.6	Httpproxy - Beta Version!	5
1.7	Httpproxy - Installation	5
1.8	Httpproxy - AMosaic configuration	6
1.9	Httpproxy - IBrowse configuration	7
1.10	Httpproxy - Weblink configuration	7
1.11	Httpproxy - Lynx configuration	7
1.12	Httpproxy - Emacs configuration	7
1.13	Httpproxy - Usage	8
1.14	Httpproxy - Example Settings	9
1.15	Httpproxy - Things left to do	9
1.16	Httpproxy - Bugs and Limitations	10
1.17	Httpproxy - Compiling	11
1.18	Httpproxy - Disclaimer	12
1.19	Httpproxy - Copyright and Distribution	13
1.20	Httpproxy - License	13
1.21	Httpproxy - Support	14
1.22	Httpproxy - Author	14

Chapter 1

Httpproxy

1.1 Httpproxy - Contents

Httpproxy 0.9 beta

A proxy supporting the http protocol
Usable on Amiga and (most) UNIX systems
Copyright © 1995 Matthias Hopf, EMailware
All Rights Reserved.

Introduction	What is Httpproxy?
Beginners Guide	What is a proxy at all???
Features	Why use such a sophisticated system?
Requirements	What does Httpproxy need?
Beta Version	*!Read!* I need your help...
Installation	No installer script, sorry.
Usage	All options.
Examples	My personal settings.
To do list	What would be cool?
Bugs & Limitations	What does not work?
Source	I know, it is ugly...
Compiling	All about the source.
History	What is new in this version?
Disclaimer	I'm not responsible for *anything*!
Copyright	And Distribution.
License	Use it and write EMail to me.
Support	What do *you* need?
Author	Where you can reach me.

1.2 Httpproxy - Introduction

Introduction

=====

You have AmiTCP V2.2 or greater installed and have access to a WWW-server?

You use AMosaic, IBrowse, Weblink (not right now ;-), Lynx or Emacs with WWW extension quiet often or at least sometimes, maybe even over a serial line?

You are annoyed of the long loading times of new pages or, even worse, inline graphics?

You don't like ARexx-Scripts like WWWCache.rexx or the like (I must admit, I never tried them :-]), because you

- 1) don't want the html pages to be modified,
- 2) they don't work for all links (maybe ?!),
- 3) you would like to have the pages updated in regular intervals or
- 4) can't use them, because you don't use AMosaic?

If any answer is no, you may save time and stop reading this, because you won't need Httpproxy.

Httpproxy is a caching proxy that has several basic configurations. These are explained in the features section.

With the use of Httpproxy you will be able to get any page you have already visited without any network traffic. That's much faster than polling the URLs from the net, of course...

1.3 Httpproxy - What is a proxy at all?

Beginners Guide

=====

I assume that you have some very basic knowledge on networking, e. g. you managed to get AmiTCP working on your system.

You used a WWW-browser like AMosaic before and know that the pages you are viewing do not reside on your own computer but are fetched from some specific servers all around the world. You already experienced, that waiting times are really annoying sometimes...

Now how about caching all pages you already viewed? Inline graphics for instance don't change very often, so it is absolutely useless to fetch them again and again from the remote server. The same happens to text files, but you won't recognize it that much there (they are usually much shorter).

That's the job of a proxy. A WWW-browser can be configured to send all URL requests to one specific server instead of sending it to the server that is stated in the URL.

For instance the document for the URL

```
http://wwwcip.informatik.uni-erlangen.de/user/mshopf
```

is fetched from the server "wwwcip.informatik.uni-erlangen.de" when you don't specify any proxy. This could take some time when you live in america or even australia...

Whenever a request is sent to the proxy, it checks whether it has already accessed the page and the cache entry is valid. In this case, the cache entry is sent to the browser and no network traffic occurs.

When there is no valid cache entry for the requested URL, the proxy contacts the server stated in the URL and fetches the document. The document is now cached and future requests to the same URL will make use of the cache.

Most internet providers do have a proxy running, too, it is wise to use it, though you can cache all files you already visited on your own computer. Maybe someone else already fetched the page you are demanding - this will speed things up.

Some internet providers don't like their customers to have full internet access or don't have enough free routable host addresses and therefore install a firewall. Most times on these firewalls is running a proxy, too.

1.4 Httpproxy - Features

Features

=====

First of all Httpproxy is a proxy and does exactly what it is supposed to do.

It caches all visited URLs and expires and deletes its cache files after a given timeout or after an immediate second request (within 10 seconds by default). This method will change in future versions.

Httpproxy has several basic configurations:

1) proxy for http: URLs to remote machines

In this mode only http: requests are accepted. Httpproxy does not have other protocols built in than HTTP.

2) proxy for all non-interactive requests (http:, ftp:, gopher:, wais:, ..) to another proxy (only caching)

In this mode almost all requests are possible and are forwarded to another proxy specified by you.

This mode will be referred to as 'proxyproxy' mode for obvious reasons.

3) proxy for already visited URLs in offline mode

While you are offline (only sensible for dial-in connections, of course) you are able to browse through all documents you already visited. Httpproxy will inform you whenever a page is outdated, but you will be able to receive this expired one by immediately reloading the document.

4) queuer for new URLs in offline mode and auto spider in online mode

While you are offline you are able to click on links or enter URLs you don't have visited already. Those URLs are queued and the next time you are online these pages are loaded automatically, up to four simultaneously.

I plan to add an auto spider option that automatically gets all sub-URLs of a requested URL on request, but that is yet to come.

Httpproxy is compilable on Amiga® computers and most UNIX systems. This may be interesting for people running NetBSD or Linux?! Of course Httpproxy may be run on any other system it compiles. For this source is included.

Httpproxy may have even more features in the future, but that depends very much on you and your response!

Please note also that this version is still beta software!

1.5 Httpproxy - Requirements

Requirements

=====

- An Amiga® or any UNIX system
 - On Amiga Kickstart 2.0 should be enough, but it is only tested on an A4000 with Kickstart 3.0 so far. Perhaps it works with Kickstart 1.3, too (does AmiTCP run on Kick 1.3 ???)
 - AmiTCP 2.2 or greater installed and running
 - Some Mbytes of harddisk space (you don't want to run AmiTCP from floppy, do you?) for the cached data
 - Some Mbytes of memory. Right now Httpproxy is relative memory hungry (ca. 213 Kbyte static memory for a maximum of 1024 cache entries) AmiTCP, AMosaic and MUI will take away the rest...
 - Ability to connect to a WWW-server ;-)
 - that's all...
-

1.6 Httpproxy - Beta Version!

Beta Version!
=====

Note! This is still beta!

Though I will never claim any code of me (other than HelloWorld.c ;-) to be bug free, beta software is even less bug free...

However Httpproxy never ever crashed on my computer and when it freezed it was always interruptable with Ctrl-C (Gee, the first >10Kbyte proggy I ever wrote that didn't send any Guru at all to me B-). The latest version has been working fine on my computer for over a month yet.

You should always run it on a small partition, though, there's a big chance it will invalidate your harddisk when any programm (maybe AMosaic?) crashes while Httpproxy saves its data to a cache file. But that problem can not be solved at all...

Be carefull on first time configuration. On startup Httpproxy deletes all cache files that are older than specified. All cache files start with '@' or '_', so most of your files won't even be touched.

Please help me to find any bugs left in the code:

If you find Httpproxy crashing your computer or behaving strange on some URLs (please note the Limitations section) or in some environments, please feel free to contact me. EMail is preferred.

Please specify as much information as possible as this will greatly reduce debugging time. A statement of the form "Httpproxy doesn't work" is really useless!

Note also that english is not my native language. I'm sure this guide file contains several severe misspellings and gramatical errors. Feel free to correct me (shame on me!).

Thank you for helping me to improve Httpproxy!

1.7 Httpproxy - Installation

Installation
=====

That's the easy part:

Copy the according executable to any directory in your path (e.g. AmiTCP:bin)

There are several to choose from:

- httpproxy000 Choose this one when you're running on a 68000 system
- httpproxy020 Choose this one when you're running on a >=68020 system
- httpproxy.db Choose this one when you want to have lots of debugging output (in order to send it to me ==)

httpproxy does not need a mathematical coprocessor.

Make a new directory for cache files. You will specify this directory on startup with the 'cache' option.

Note that debugging output slows the program down. Debugging files tend to grow very fast, too...

Now that's the hard one...

First of all you have to tell your browsers how to contact the proxy.

- AMosaic
- IBrowse
- Weblink
- Lynx
- Emacs with WWW module

Then you have to start Httpproxy, whenever you start AmiTCP. The options depend on your startup mode (online or offline), however.

In my configuration, for example, I have two separate startnet scripts that are copied into AmiTCP:bin on startup time (in fact there's a lot more to do, but that's too much to explain here). These startnet scripts execute a startservices script (which is mode dependent, too) which runs Httpproxy in the background.

This section is not complete right now.

Is there anybody willing to build an installer script? I would be glad to see one...

1.8 Httpproxy - AMosaic configuration

Configuration for AMosaic
=====

Just set the environment variable 'http_proxy' to

```
http://localhost:PORT/
```

with PORT beeing the proxy port.

In the standard configuration (without 'port' option specified) this would result in

```
http://localhost:8080/
```

When you run httpproxy in proxyproxy mode you may set the environment variables 'ftp_proxy', 'gopher_proxy' and 'wais_proxy' to the same value.

1.9 Httpproxy - IBrowse configuration

Configuration for IBrowse
=====

Sorry, I couldn't download the IBrowse beta executable so far...
(the line to www.omnipresence.com is really busy :-()

May be it is configured the same way as AMosaic.

1.10 Httpproxy - Weblink configuration

Configuration for Weblink
=====

Sorry, I don't know enough right know about the coming Weblink AmiTCP network module (Sorry, Jesse, when I missed something ;-). Perhaps it is possible to programm a new api module that interacts directly with Httpproxy. But that is yet to come.

1.11 Httpproxy - Lynx configuration

Configuration for Lynx
=====

Sorry, I don't know enough about Lynx and its Amiga port. Perhaps someone else could send me a note?

1.12 Httpproxy - Emacs configuration

Configuration for Emacs
=====

Sorry, I don't know enough about Emacs and its Amiga port. Perhaps someone else could send me a note?

1.13 Httpproxy - Usage

All known Options

=====

Usage: httpproxy ?

```
httpproxy    [proxy PROXYHOST PROXYPORT] [port PORT] [cache DIR]
             [del SECONDS] [expire SECONDS] [reload SECONDS] [log FILE]
             [unread] [offline] [get]
```

(I know, this is no standard Amiga template...)

- proxy PROXYHOST PROXYPORT
Specifies a proxyproxy. If you don't know the port number, try both 80 and 8080. These are the standard values. It can be usefull to specify a proxyproxy even in offline mode as this will enable queueing of non-http requests.
 - port PORT
Specifies the port Httpproxy should listen on. This value is needed when you configure your browser.
 - cache DIR
Specifies the cache directory you created on installing Httpproxy. The program changes its lokal directory to this place, so if you don't run it in the background you will notice some strange effects on Amigas. For instance the shell will think it is still in the old directory, but 'cd' will prove that you are not.
 - del SECONDS
Specifies the minimum age of cache files that are deleted on startup. Right now only URLs that are reloaded in online mode will be deleted once the program is running.
 - expire SECONDS
Specifies the minimum age of cache files that are marked as invalid. Requests for invalid URLs are queued in offline mode and reloaded in online mode.
 - reload SECONDS
Specifies the maximum (!) number of seconds between two (identical) requests to be interpreted as a reload request.
 - log FILE
The logfile. Specify nil: if you don't want one.
 - unread
Specify this switch when you want interrupted transfers to be continued by the proxy (e.g. you interrupt the transfer in the AMosaic window, but the proxy will keep getting data).
 - offline
Specify this switch when you are in offline mode. Don't specify both 'offline' and 'get'.
-

- get

Specify this switch when you want the proxy to auto load queued URLs. Does only work in online mode, of course.

1.14 Httpproxy - Example Settings

Example Settings

=====

These are my settings in offline mode:

Delete cache entries older than two months, expire cache entries older than one week, allow ftp:// and co. queueing.

Sometimes I use the debug version of httpproxy, so I redirect stdout to a file. Note that Httpproxy won't print anything to stdout but debugging information.

Note that .-files in the cache directory are simply ignored (no warning message).

```
-----
run httpproxy >>Cache:ProxyCache/.Httpproxy.debug cache Cache:ProxyCache
    proxy proxy.rrze.uni-erlangen.de 80 del 5184000
    expire 604800 log Logs:Httpproxy.log offline
-----
```

These are my settings in online mode:

Basically the same as above, but be online and get queued documents.

```
-----
run httpproxy >>Cache:ProxyCache/.Httpproxy.debug cache Cache:ProxyCache
    proxy proxy.rrze.uni-erlangen.de 80 del 5184000
    expire 604800 log Logs:Httpproxy.log get unread
-----
```

Another example:

Just be a caching proxy and listen on port 80 (standard httpd port), expire and delete cache files after one day:

```
-----
run httpproxy <>nil: cache Cache:ProxyCache port 80 log nil:
-----
```

1.15 Httpproxy - Things left to do

Todo - List

=====

in order of importance for me:

- better handling of documents that are expired (more options for the user)
- special URL `http://proxy:0/...` requests to inform user about queued and auto-requested documents (and perhaps structures) and cache information
- user configurable document specific expire times (e.g. don't expire any URLs that match `#!.jpg` or `#!.gif`)
That includes URL patterns that will never be cached (e.g. `#!.lha`)
- working POST method
- access control (at least a simple one)
- autoget of user defined URLs when they are expired (only on startup)
- save full URL requests in url cache file, send full URL auto request (may be this will never implemented as it can be a security problem)
- only keep on caching interrupted requests, when `> xx%` (on `Content-Length:` known) or `> yyy Bytes` are already read
- auto spider
- better caching system and dynamic cache tables (even a totally different system with hash tables)
- save cache table on regular shutdown -> speedup on starting
- automatic addition of `Content-Length:` fields wherever possible
- check of `Expire:`, `Proxy:` fields
- other protocols (ftp, may be gopher)
- reload on immediate second request needs work for cases where multiple recipients are involved

1.16 Httpproxy - Bugs and Limitations

Known Bugs and Limitations

=====

- The POST method does not work right now. Luckily most forms use the GET method and normal documents are always fetched with the GET method.
If your browser tells you that the host could't be contacted, take a look at the source of the current document. You may find a `'action=post'` tag inside a form description...
 - Not all HTTP specifications are observed right now. This will change in
-

the future. But Httpproxy should work with all (tolerable) servers.

- The caching file system should be local (not nfs mounted), because the cache file descriptors are not included in the select() list (the amiga version of select() doesn't support standard file descriptors here =(), and because of this the file operations are supposed to return immediately or at least very fast.
- don't specify any URLs that would contact the proxy itself. It will lock up, because the URLs are not checked for this and:
- no timeouts for connections are implemented right now.
- no more than 2^{31} requests allowed per startup, no startup exactly (the second counts...) 136 years after first startup allowed without cache flushing :-P (this won't be fixed...)
- error messages are cached as standard cache entries right now (they should not be valid in online mode)
- logmessages should be reworked
- some untested cases...
- queued requests will only contain the URL, user-agent: and accept: */* (see Todo list)
- my difftime() macro is brocken right now when compiling with AmiTCP 4.0 demo API

1.17 Httpproxy - Compiling

Compiling

=====

Httpproxy was compiled with SAS C V6.56 with near data and code, full optimizing and no stack check and/or extension code (it should never ever need more than 2 Kbytes of stack) and the AmiTCP V2.2 api libraries. These libraries miss <syslog.h>, but it can easily be added by an

```
echo >netinclude:syslog.h "#include <sys/syslog.h>"
```

The AmiTCP4.0 demo contains this syslog.h already.

Httpproxy compiles fine with the AmiTCP4.0 demo API, but the calculation of time differences in Httpproxy is brocken right now. It's no problem to run the V2.2 API version under AmiTCP4.0, though.

It should be fully ansi compliant except for the socket api.

The source code is not the best I've ever written however (in fact it was a four days hack plus additional bug fixes...).

Recompiling should be straight forward with Ansi compatible compilers and include files. Because of this Httpproxy won't link properly on SunOS4.x

systems for instance right now (it needs memmove).

Use gcc to compile on UNIX systems.

Compiling was tested on SunOS4.x, Solaris5.x (I *won't* call this SunOS!), HpUX, Irix. The executable was not tested, though, but it should work as I learned socket programming on exact these machines.

On some UNIX system you may have to link with libsocket, libnet or libnsl or something similar.

There are no signals or special files used at all, so no problems should occur from semantical differences (SysV <-> BSD).

Please mail me *any* experiences with compiling and running Httpproxy.

1.18 Httpproxy - Disclaimer

Disclaimer

=====

In short:

I was not, I am not and I will be not be responsible for anything this or other programs written by me or by others do or don't do.

It's your own risk, to use, not to use, to copy, modify, keep, archive, delete this program or other data or to do anything that can be done to a program or to anything else with this program or with anything else.

Ufff...

And note it's still beta!

However it works fine for me ==)

THERE IS NO WARRANTY FOR THIS PROGRAM TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHERE OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDER AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH

DAMAGES.

1.19 Httpproxy - Copyright and Distribution

Copyright and Distribution

=====

Httpproxy is Copyright ©1995 by Matthias Hopf (Yes, that's me B^)

All Rights Reserved.

This program is EMailware, *almost* freeware.

It is *not* public domain. The source is, though provided, still copyright by me.

You may modify or reverse engineer (though not necessary ;-) the program for your own use or for others as long as you don't charge anything for it. But the original copyright messages in the code must not be changed! Don't redistribute changed versions of the program without my explicit permission (a pgp signed emailed permission will be sufficient for this program).

The archive may only be distributed in unmodified form. No files may be added, changed or removed. That includes .displayme and similar files from mailboxes. Noone may charge for this archive more than normal media and duplication fees. Distribution is allowed in all forms except where noted.

Urban Mueller is explicitly allowed to include this program in archive or in ready to run form on any Aminet CD. Other parties have to contact me prior to including the program on CDs in any form or on disks or ftp sites in non-archive form.

All legal matters are subject to change without notice.

Written permissions by me can override all Copyright, Distribution, Disclaimer and Licence notes stated here.

For this program pgp signed EMail have the same status as written notes.

1.20 Httpproxy - License

License

=====

You are hereby granted to use Httpproxy when you agree and only when you agree on the following conditions before and while using it:

- You will use the program at your own risk.
- You accept the Copyright and Distribution restrictions.
- You will send an EMail notice to me that you are using my program.

Snailmail and postcards are welcome even more, of course, but bug reports are preferred by EMail.

Come on, a line 'I'm using your marvelous/great/nice/buggy/sh*tting Httpproxy' would suffice though more would be nice. :)

As you have access to a WWW-Server it normally should be no problem to you to send a short EMail to me...

- You will send an EMail notice to me specifying the approximate number of hosts and users when you install Httpproxy on a system on which several users have perhaps simultaneously access to the proxy or several hosts are involved. Note that multiple users may result in some problems right now.

Please note that I won't upload newer versions when I don't get any response from this one.

Don't be afraid to tell me that you don't like or can't use Httpproxy. I just want to hear your *real* opinion. But I hope you like it.

1.21 Httpproxy - Support

Support
=====

As I stated several times before I won't put updates into Aminet when I don't receive any response to this version.

So go right now into your favourite mailer and send an EMail to me.

I'm open for bug reports, likes, dislikes, opinions, improvements.

This program will only develop the way *you* like when *you* send your comments about it to me.

There are several things to do, but order is not fixed. I just sorted all features that came into my mind like I think they are important.

1.22 Httpproxy - Author

Author
=====

Matthias Hopf
Neckstraße 4
91052 Erlangen

email: mshopf@immd4.informatik.uni-erlangen.de

I'm a student of computer science in Erlangen, Germany.

Feel free to EMail or to snailmail me ==-D

You're welcome on my WWW home page.

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.6ui (Amiga)

```
mQCNAi7TlyUAAAEEOyEU5wh5wNu1Pu5w+UGDPidBMlg3pPWXjkZf9vWGLu4vX14
DgAMwiGBcJlWE4KkiRabY9WmZdBjhuy2ECftZn8UBAxxxItgkN3pe8HObI7KWCdn
/ct1X9bzUymnAAZ46trFk6A3mwb6j5tYjKQJ14WZUPwDB5zsWD9QXspNMJEBAUR
tDVNYXR0aG1hcyBIb3BmIDxtc2hvcGZAY2lwLmluZm9ybWF0aWsudW5pLWVybGFu
Z2VuLmRlPokAlQIFEC8+Qm/5F+bfpcmz1QEBEZoEAJ0gIeV4XAMvRCuv6m8p37WT
Wfg0uqzeRXVZaWf4JLep3wzrlXl jtn3QhTXabrCwGLg9QXEKyPsZCglImr3uwcnc
90LjC9L2nmRVYzCFAgtu7tkW/xAPLU2iIiycu6XnUWmB5F4zNDTBbImANhoiGVhm
srhoPyXTLZcOyZia6ElR
=IV/W
```

-----END PGP PUBLIC KEY BLOCK-----

```
--
_ //          | Matthias Hopf - "Hoeppe" | _ _
\\ //   Amiga | student of computer science | _|cience |-iction
 \X/ by conviction |          in Erlangen/Germany | by belief in Future
```

Email: mshopf@immd4.informatik.uni-erlangen.de
Aminet: ftpamiga@epix.rrze.uni-erlangen.de
WWW: http://wwwcip.informatik.uni-erlangen.de/user/mshopf
