

# e-httpxfer

e-tunity

2002

## Abstract

*e-httpxfer* is a HTTP request transfer utility. This document describes how the uses of *e-httpxfer*, how it is installed, and how it is configured.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Invocation of e-httpxfer</b>	<b>2</b>
<b>3</b>	<b>Installing e-httpxfer</b>	<b>3</b>
3.1	Compilation and installation . . . . .	3
<b>4</b>	<b>Configuration</b>	<b>3</b>
4.1	Network configuration . . . . .	3
4.1.1	Configuring the inetd mode . . . . .	3
4.1.2	Configuring the daemon mode . . . . .	4
4.2	The configuration file . . . . .	4
<b>5</b>	<b>Performance issues</b>	<b>5</b>

## 1 Introduction

*e-httpxfer* is a utility which is used to catch HTTP request, to re-route those requests to a given (pre-defined) location, to collect all data there, and to send back the data to the originator.

The general working of *e-httpxfer* is explained below:

- A requestor, typically a browser, sends a HTTP request to a server, where *e-httpxfer* is installed.
- *e-httpxfer* receives the request, and builds its own connection with a given webserver. The webserver that *e-httpxfer* connects to, is defined by the configuration of *e-httpxfer* and by the settings in the original request.
- Via its own connection, the request is forwarded to the true webserver and an answer is collected.
- The answer is then sent back to the requestor.

Using this approach, *e-httpxfer* is ideal as a handler on a public system, and can be used to handle requests to multiple "hidden" webservers, which themselves may be located e.g. in a LAN, unreachable by a direct connection.

*e-httpxfer* can be best described as a very light-weight and configurable proxy for HTTP traffic.

## 2 Invocation of e-httpxfer

Throughout this document, *flags* of *e-httpxfer* will be mentioned. The full explanation is given here.

The invocation of *e-httpxfer* is *e-httpxfer flags*. No non-flag arguments are allowed. The supported flags are shown when *e-httpxfer* is started with an argument, as in *e-httpxfer x*. The flags are also discussed below.

- *-afile*: Logs the activity to *file*. The activity is shown as:
  - The date in ANSI format (*YYYY-MM-DD*),
  - The time in ANSI format (*HH:MM:SS*),
  - The original requested host as received by *e-httpxfer*,
  - The host and port that handled the request prior to sending back the contents to the browser,
  - The HTTP request itself (*GET, POST, etc.*).
- *-cretries*: Instructs *e-httpxfer* to compensate *retries* times when trying to get a response from the webserver. The default is (of course) 1 – i.e., try just once.

You might want to use this flag when *e-httpxfer* is requesting data from a non-stable webserver, e.g., a webserver in a testing environment, which is re-started often. Restarting the webserver would otherwise break the request transmission and *e-httpxfer* would report a *read error*.

- *-dport*: Instructs *e-httpxfer* to "daemonize" while binding to the specified port. The default is that *e-httpxfer* will run once (single-shot) and will use *stdin* and *stdout* as its inputs.
- *-resource*: Instructs *e-httpxfer* to load *resource* as its resource file, instead of */usr/e/etc/e-httpxfer.rc*. (The resource -or configuration-file is further discussed below.)

- `-skip,skip,skip,...`: Instructs `e-httpxfer` to skip the indicated HTTP header lines. This means that when the lines are seen in the request, the lines are **not** sent to the true webserver. For example, "smarter" browsers will try to determine whether or not to request a page, by supplying a header line that starts with `If-Modified-Since`. The webserver is supposed to send the requested information only if it was modified after the given date. By suppressing this line, the webserver will **always** send the requested information.
- `-timeout`: By default, `e-httpxfer` uses a timeout of 300 seconds (5 minutes). If the true webserver doesn't respond by then, then `e-httpxfer` assumes that the request cannot be fulfilled. The flag `-t` defines this timeout in seconds.
- `-v`: This makes the run more "verbose", `e-httpxfer` logs its actions in `/tmp/e-httpxfer.log`. Enabling logging is a good way to see what header lines are received and how `e-httpxfer` handles them.
- `-V`: This instructs `e-httpxfer` to show its version ID and to stop.

## 3 Installing e-httpxfer

### 3.1 Compilation and installation

To build `e-httpxfer`, the following e-tunity packages are necessary:

- The source tree for `e-httpxfer` itself,
- `e-lib`, version 3.00 or better,
- `misc`, version 1.21m or better.

Once the support packages `e-lib`, `epp-lib` and `misc` are available and installed, `e-httpxfer` compiles and installs out-of-the-box.

## 4 Configuration

Configuring and using `e-httpxfer` is a two-step process: first, `e-httpxfer` is bound to the HTTP port of a server (port 80). Second, its internal configuration file must be adapted.

### 4.1 Network configuration

`e-httpxfer` knows two operating modes: an `inetd`-based single shot mode, and a server (daemon) mode.

#### 4.1.1 Configuring the inetd mode

To enable `e-httpxfer` as a "service" at port 80, two files must be changed by the superuser (`root`): `/etc/services` and `/etc/inetd.conf`. The changes in `/etc/services` are:

```
1
2     # service  port/type
3     httpxfer  80/tcp
4
```

Note that other predefined services for port 80, type tcp, may co-exist as far as the file `/etc/services` is concerned.

The changes in `/etc/inetd.conf` are:

```

1
2     # service net type state.max user path to binary
3     httpxfer stream tcp nowait.1000 root /usr/e/bin/e-httpxfer e-httpxfer
4

```

Note that in `inetd.conf`, only one service for port 80 may be present. E.g., if `services` also defines a service `www`, for port 80, then the line starting with `www` in `/etc/inetd.conf` must be either deleted or commented out.

Once the files `/etc/services` and `/etc/inetd.conf` are changed, the daemon `inetd` must be restarted, typically using `killall -1 inetd`.

#### 4.1.2 Configuring the daemon mode

The startup of `e-httpxfer` as daemon only requires the presence of a flag `-dport`, thereby instructing `e-httpxfer` to bind to the specified port. Often a start script in or under `/etc/rc.d/` will be necessary.

## 4.2 The configuration file

`e-httpxfer`'s configuration file by default resides in `/usr/e/etc/e-httpxfer.rc`, though a different path may be specified using the switch `"-r alternate-file"`. If needed, this switch must be stated in `/etc/inetd.conf` or in `e-httpxfer`'s start script.

The configuration file is a list of two- or three column entries. The first column specifies a host setting received HTTP request. When `e-httpxfer` matches this host setting, then the second column entry is taken as the webserver to connect to. When present, a third column specifies options to `e-httprequest` that apply to only this host entry.

An example is given below:

```

1
2     # Requests for...      should go to...      using optional flags
3     e-tunity.com          192.168.1.50
4     www.e-tunity.com      192.168.1.50
5
6     e-tunity.com:81       192.168.1.50:81
7
8     e-tunity.net          192.168.1.100        /usr/e/bin/e-httpxfer \
9                             e-httpxfer -v -a /tmp/act-net.log
10    www.e-tunity.net      192.168.1.100        /usr/e/bin/e-httpxfer \
11                             e-httpxfer -v -a /tmp/act-net.log
12
13    e-tunity.nl           192.168.1.100:81
14    www.e-tunity.nl       192.168.1.100:81
15
16    default               192.168.1.80
17

```

The sample entries are:

- Requests for `e-tunity.com` and `www.e-tunity.com` are handled via the webserver at `192.168.1.50`. Both host specifications do not specify a port, so that the default HTTP port 80 applies. In this example, `192.168.1.50` would typically be a webserver in a private LAN.
- Requests for `e-tunity.com:81` also go to `192.168.1.50`, but must be handled there via port 81. Note that the configuration file does not provide for `www.e-tunity.com:81`.
- Requests for `e-tunity.net` (or when prefixed with `www`), go to `192.168.1.100`. These entries also supply a third column, which is formulated a-la the `/etc/inetd.conf` entry. In these two entries the flags `-v` and `-a`

`/tmp/act-net.log` are supplied. The flag `-v` makes `e-httpxfer` more "verbose" and logs actions in `/tmp/e-httpxfer.log`; the flag `-a` must be followed by a file name, and instructs `e-httpxfer` to log its activity in the given file.

This entry also shows how "long" lines can be wrapped using \-continuation.

- Requests for `e-tunity.nl` (or when prefixed with `www`) go to the same webserver as the requests for `e-tunity.net`, but to port 81 instead of the default port.
- The last entry is the *default*: all requests that do not match any of the above, go to `192.168.1.80`.

This example of course assumes that the requests to `e-tunity.com`, `e-tunity.net` etc., all arrive at the server where `e-httpxfer` is installed. This is a DNS-issue and is not further discussed here.

## 5 Performance issues

As far as the performance is concerned, the following rules of thumb apply.

- `e-httpxfer` can be configured to run via `inetd`, the Internet daemon. This means that for each request (even for just a simple gif), `e-httpxfer` is started. This fact is nowever no performance bottleneck; `e-httpxfer` is so small and compact that the multiple starts do not slow down the system.  
If you still suspect that the "single-shot" approach is a bottleneck, please switch to daemon mode.
- However, performance gains can be achieved by specifying the "true webserver" in the configuration file (this is the second column) as an IP address, instead of a hostname. This particularly applies when non-local hosts are involved. E.g., having requests handled by `www.altavista.com` will be significantly slower than having them handled by `209.73.164.92`.  
Note again that each HTML page will typically contain multiple elements, all pointing to the same host (such as `/images/whatever.gif`, `/css/sheet.css`, framed pages). That means that `e-httpxfer` will be re-started many times for just one page: having the handler address as an IP number helps a lot.
- `e-httpxfer` will happily log what is is doing to `/tmp/e-httpxfer.log` and will ensure that the log doesn't clog the disk: the log is "rotated" when it reaches 100K, and no more than 5 rotated logs are kept around. However, this also means that every run of `e-httpxfer` may involve log rotation, compression etc.. Therefore, once `e-httpxfer` is tested, the flag `-v` in the activation command should be removed.