NAME

netstiff − powerful and easy tool to check for Web and FTP updates

SYNOPSIS

netstiff [*options*] [*command*]

DESCRIPTION

Netstiff (formerly known as webdiff) is a powerful and easy-to-use tool which checks for Web page and/or FTP site updates.

For the Web, updates are recognized using several test criteria (`diff`, `html`, `size`, `date`, `md5sum`, `regexp`). The FTP update checker is only able to `diff` on directory listings and files and to compare `size` and `date` of files.

Without a given command, netstiff will check for updates of the specified URIs and then print the changes. If no configuration file exists, the configurator is launched instead.

Netstiff exits after all configured URIs are checked. Occuring warnings and errors leave a message in the log file (`˜/.netstiff/lastlog`) and on `stderr`. Use it with *cron* if you want to check for updates regularly.

COMMANDS

You can only pass one command to netstiff. It has to be the last argument in the argument list.

Commands may be shortened down to one character (e.g. **c** instead of **configure**). Leading dashes are ignored.

If you start netstiff without command, the **full** command will be used.

configure

Use this command if you want to start the configurator, the interactive configuration tool of netstiff. Of course, you may also edit the configuration file in `˜/.netstiff/config` by hand. Using the configurator is recommended if you are a new netstiff user, because it explains the possible test methods, validates your regexps, etc. Nevertheless, the configuration file format is very easy. See section **CONFIGURATION FILE**.
The configurator will not initialize the netstiff cache for added URIs, i.e. it will not download anything. To do so, you have to run `netstiff update` first. This is a feature.
If the config file does not exit, the configuration tool is started automatically.

diff Use this command if you want to see the differences between two versions of saved content (Web pages or meta data). See *diff*(1).

The version after the last **reset** (or the initial version) and the version of the last **update** will be compared.

full Use this command if you simply want netstiff to check for updates and print the diff.

full is a simple replacement for the following sequence:

```
netstiff update > /dev/null
netstiff diff
netstiff reset
```

help Use this command to get usage information about netstiff. To be honest, this manual page in conjunction with the configurator is a better documentation.

reset Use this command after you noticed all differences with the **diff** command (see above), so that **diff** will not show you the same changes again and again.

update Use this command if you want netstiff to fetch the data from the specified URIs and show you only those − one per line − that have changed since your last **update**.

version
This command will display version number and copyright.

OPTIONS

You may pass the following options.

--no-stderr, -S
Use this option to suppress warning and error messages on `stderr`. Thus the messages can only be seen in the log file.

--workdir DIR, -W DIR
Use this option if you want to specify another working directory. The working directory is the directory where netstiff reads the configuration file, stores the downloaded data and writes it logs. It defaults to `~/.netstiff`. See also section **BUGS**.

RESTRICTIONS

There is no special case to handle status codes other than 200. In practice, netstiff will neither follow redirections nor will it notice any 4xx or 5xx error code. The resulting error pages are treated as usual Web pages. No logged message. Please check on your own.

USAGE EXAMPLE

You want to add a new URI netstiff should check for updates.

```
    netstiff conf
```

The configurator is not described here. I know some weaknesses in usability, but you can get along with it.

When you are seeing your shell prompt again, you know that netstiff should retrieve an initial version of the Web page you specified.

```
    netstiff update
```

After some weeks in the sun you want to see if something has changed. So you let netstiff check for updates.

```
    netstiff update
```

It is printing an URI! Let’s see the changes!

```
    netstiff diff
```

Oh, it is so much, that it does not fit on a screen!

```
    netstiff d | pager
```

Now you are satisfied because you read all the changes. So you finally do

```
    netstiff reset
```

and netstiff forgets about the changes.

If this is the only way you want to use netstiff, you will only need to do

```
    netstiff | pager
```

That’s it!

CONFIGURATION FILE

There is no need to manually edit the configuration file *WORKDIR*`/config` (usually `~/.netstiff/config`), because `netstiff configure` should do the job. But sometimes it is easier to edit a simple file than to browse through menus, or you are writing another application that changes netstiff settings. So it is useful to describe the file format here.

RULES

- Whitespace at the begin and end of each line is ignored.
- Empty lines are ignored.
- A line beginning with # is regarded as comment.
- A line beginning with `+` is regarded as option. The `+` is followed by the *option name*, some whitespace and the *option value*.
- A line neither beginning with # nor `+` is regarded as URI. URIs without scheme (`https://`, `http://`, `ftp://`) are recognized as HTTP URIs.
- The configurator interprets a comment right above an URI as the title of the URI.
- Options always apply to the first URI above. Options without URI line above are *global options* and affect every URI that does not override these specific options.

CONFIGURATION OPTIONS

The following options are generally available:

test sets the test method (or test criteria).
See section **TEST METHODS** for a description. Defaults to `diff`.

timeout
sets the timeout (in seconds) for TCP connections.
Defaults to 20.

The following options only affect HTTP URIs:

client set the user-agent string.
Some web sites check the HTTP header field *User-Agent* and display different content for different agents. By setting this field you can pretend to use Mozilla Firefox, for example. Because many log analyzer tools for webmasters display statistics about that field, you may spread the word about netstiff by setting this variable to the truth: `netstiff`. ;-)
Example: `+ client Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.12) Gecko/20080208 Galeon/2.0.4`
This option is not set by default.

lang sets the accepted languages.
Internationalized web sites offer there contents in different languages and may check the HTTP header field *Accept-Language*. It contains a list of languages (and sometimes extra information like associated countries) sorted by priority. The best way to get a good value is to copy and paste it from the preferences of your web browser.
Example: `de,en;q=0.9`
This option is not set by default.

proxy sets HTTP proxy host and port. Must be in the form `host:port`. Will fail if no port is given.

range sets the range (in bytes) to get from a server.
Use this option if you are only interested in the changes within a small region of a big file on a HTTP server. Examples are `12000-12500` or `13000-` (till the end).
The Range feature is not supported by all web servers or for every content. That means, that some web servers send the whole content instead of only the given range.
This option is not set by default.

referer sets the referrer.
Some web sites check the HTTP header field *Referer* and refuse to display the wished contents if it is not appropriately set. When clicking on a link in an ordinary web browser, the referrer is set to the URI, where you clicked on the link. By setting this option to an URI, you can pretend clicking on a link on the web page of this URI. Please do not use this option to 'advertise' your own homepage (so-called *referer spamming*).

This option is not set by default.

The following options only affect the test method `html`:

htmlcmd

sets the command that is used to produce non-HTML human-readable output. The command will be run on a temporary file.
Doing many experiments I got my best results using `+ htmlcmd lynx -nolist -dump`. Other dumpers had features, like justified text or well-formatted tables, that turned out to be disadvantages when looking at the diffs.
This option is not set by default. If you use the `html` test method then, a very simple mechanism will hide HTML tags. It is possible to get good results doing that, but it is not likely and thus not recommended to leave this option unset.

The following options only affect the test methods `diff` and `html`:

start, **end**

Motivation: Many modern or CMS-generated web pages have a dynamic and a static part. For example, at the beginning of a web page there is always a randomly chosen citation the author liked. At the end there is a calendar showing the current date, a weather analysis for the next days, and some other useless stuff. The information you want to monitor for changes (the *static part*) is situated between those dynamic parts. It is very often possible to figure out *textual anchors*, that indicate the start or the end of the static part.
Using this options you can set regular expressions to that anchors. For example, if the last entry of the navigation bar is *Imprint* and thereafter comes the static part, set `+ start /Imprint/`. I hope, you can imagine analogous examples for the `end` option.
Note, that the regular expressions act on the unprocessed input (e.g. HTML source code), also when using the `html` test method.
These options are not set by default.

The following options only affect FTP URIs:

passive is a boolean option (value `true` or `false`, case-insensitive). Passive mode (PASV) will not be used on FTP connections iff set to `false`.
Defaults to `true`.

EXAMPLE

```
# this is my netstiff config file
+ test      html
+ htmlcmd   lynx -nolist -dump
+ client    netstiff
+ lang      de_DE
+ timeout   6

# local usage statistics
http://localhost/stats.php
  + start   /Statistics/
  + end     /Generating page took/

# sbeyer’s homepage
http://pkqs.net/~sbeyer/

# buggy scripts test
http://localhost/buggyscripts/test.cgi
  + test /Internal Server Error/

# muetze’s funny videos
ftp://foo:duff23@muetze.localnet/funnyvideos/
```

```
+ passive false
```

TEST METHODS

The following test methods can be used:

date On HTTP URIs, this method downloads the HTTP header to check when the file has last been modified. To make this feature work, the server should response the *Last-Modified* header entity. This behaviour can become useless when fetching some dynamic web sites.
On FTP URIs, this method requests the last modification date of the file on the FTP site to check when the file has last been modified.

diff This method downloads the HTTP content, FTP file or FTP directory listing and saves the two last versions. Later you can use `netstiff diff` to take a look at a diff of these versions.

html This method acts like `diff`, but assumes to get HTML input and preprocesses it to it more human-readable.
See also the description of the `htmlcmd` option in section **CONFIGURATION FILE / CONFIGURATION OPTIONS**.
This method is not available on FTP URIs.

md5sum
This method downloads the HTTP header to check if the MD5 sum has changed. The server should response the *Content-MD5* header entity to make this method work.
Use this method on big binary files on HTTP sites and only if the server supports it. (`netstiff` will tell you.)
This method is not available on FTP URIs.

size On HTTP URIs, this method downloads the HTTP header to check if the file size has changed. This feature needs the server to response the *Content-Length* header entity.
On FTP URIs, this method requests the size of the file on the FTP site to check if it has changed.

/regexp/
This method downloads the HTTP content and checks if the given regular expression matches or not. The URI is prompted (when using **update**) iff this match status has changed.
This method is not available on FTP URIs.

RETURN VALUE

The number of errors are returned. So exit code 0 is success.

BUGS

The regular expression stuff is using the *eval* function of Ruby. This means that you are able to do non-regex-related stuff using special strings as ‘regular expressions’. This is a big security issue when using netstiff as a backend for e.g. Web applications. So do NOT do it and NEVER start netstiff on foreign, unchecked configurations (**-W** can be dangerous).

Feel free to send feedback, bug reports, etc.

AUTHOR AND COPYRIGHT

© 2004, 2007-2008 Stephan Beyer `<s−beyer@gmx.net>`, GNU GPL