NAME

yaws − yet another webserver

SYNOPSIS

yaws [*OPTIONS*]

DESCRIPTION

Yaws is fast lightweight webserver. It can run as daemon or in interactive mode where it is possible to directly interact with the webserver. Yaws is particularly good at generating dynamic content. See the user docs for more information on that topic.

DAEMON/SERVER options

−i | --interactive

Interactive mode. This will start yaws in interactive mode with an erlang prompt. All error_logger messages will be written to the tty as well in this mode. Use this when developing yaws code.

−w | --winteractive

Cygwin inteactive mode (werl)

−D | --daemon

Daemon mode. This will start yaws as a daemon.

−-heart

This will cause the yaws system to be automatically restarted in case it should crash. This switch also requires the *--daemon* switch to be present.

−-heart-restart=C,T

This controls the number of restarts in a given time period that heart tolerates before refusing to restart Yaws. By default, heart allows up to 5 restarts within a 60 second period before refusing to restart Yaws again. This option allows up to *C* restarts in *T* seconds instead. To allow infinite restarts, set both *C* and *T* to 0. This switch automatically enables the *--heart* switch.

−-debug

Debug mode. This will produce some auxiliary error output for some error conditions. It will also start the otp sasl lib for additional error printouts.

−-nodebug

Non-debug mode. This is useful for running interactively via the -i option but without incurring the performance penalties of debug mode.

−-conf file

Use a different configuration file than the default. If the configuration parameter *config* is set, yaws use it as default configuration file. Else, The default configuration file when running as root is /etc/yaws/yaws.conf. When running as a non priviliged user, yaws will search for its configuration file in the following order. First in $HOME/yaws.conf, then in ./yaws.conf and finally in /etc/yaws/yaws.conf.

−-runmod module

Tells yaws to call *module:start/0* at startup. This makes it possible to startup user specific applications together with yaws.

−-pa path

Add path to the yaws system search path

−-tracetraf

Traffic trace mode. All traffic will be written to a trace file called trace.traffic in the log directory.

−-tracehttp

HTTP trace mode. All HTTP messages will be written to a trace file called trace.http in the log directory.

−-traceout

When yaws is put into trace mode using either --tracetraf or --tracehttp, traces are written to files. If we provide the --traceout flag, the trace will also be written to stdout.

−-trace Sames as --tracetraf --traceout. I.e. trace everything and write to stdout.

−-mnesiadir dir

Start Mnesia in directory <dir>

−-sname xxx

Start yaws as a distributed erlang node with name <xxx> using the unqualified hostname as nodename postfix

−-disable-kpoll

By default, yaws starts erlang with *+K true*. This flag reverses that.

−-name xxx

Start yaws as a distributed erlang node with name <xxx> using the fully qualified hostname as nodename postfix

−-proto_dist Mod

Use module Mod for erlang distribution. This is typically only used when we want to run erlang distribution over SSL.

−-erlarg STRING

Pass STRING as an additional argument to the "erl" program. If STRING comprises multiple words, you must quote it so that your shell passes it to yaws as a single argument. If STRING contains any single quote characters, you must quote each of them as well. For example, to pass the option *-env NAME O’Keeffe* to "erl" from a Bourne-compatible shell:

```
--erlarg "-env NAME O\’Keeffe"
```

−-id ID This flag sets the id. If we’re starting a daemon (or an interactive system) it gives the Yaws server the identity ID. This means that the server will write all internal files into the directory $HOME/.yaws/yaws/ID.

Yaws also creates a file called ${VARDIR}/run/yaws/ctl-${ID} which contains the portnumber the daemon is listening on for control request by the control command such as "yaws --hup" etc.

If we’re invoking a control command which should perform some control function on the daemon, we may have to give the --id flag also to the control command. If we don’t do this the control command may interact with the wrong daemon due to finding the wrong "ctl" file.

The daemon may also optionally specify the "id" in the yaws.conf configuration file.

−-umask MASK
Set the umask for the daemon to MASK.

CONTROL OPTIONS

The following list of options are are used to control the daemon from the "outside" while it is running.

−-hup [--id ID]
HUP the daemon. This forces the daemon to reread the configuration file. It also makes the daemon empty all its internal content caches. Hence when updating the doc root, HUPing the daemon is the fastest way to see the content updates.

−-stop [--id id]
Stop the daemon (called id)

−-ls Lists current ids and status of all yaws servers on localhost. In practice this amounts to a listdir in $HOME/.yaws/yaws - and check whether the different systems who has created files there are alive.

--status [--id id]
Query a running yaws daemon for its status, and print it.

--stats [--id id]
Query a running yaws daemon for its statistics, and print it.

--running-config [--id id]
Query a running yaws daemon for its current configuration, and print it. This can be useful when attempting to figure out how to set config in embedded mode. Configure yaws to you liking in non-embedded mode, run this command and use the output to populate the embedded mode records.

--load Modules [--id id]
Try to (re)load erlang modules into a running daemon. This is useful after modifying appmods or modules used by scripts.

--debug-dump [--id id]
Produce a debug dump on stdout. In particular this code lists what we refer to as suspicious processes. I.e. processes that might be hanging or processes that are "large" - hardcoded to 40k words.

--ctltrace [--id ID] http | traffic | off
Control the trace capabilities of a running yaws daemon. If the http or traffic option is given, the daemon will write a log for debug purposes into the logdir.

--wait-started[=T] [--id ID]
Waits at most 30 seconds for the server to start. Exits with 0 if server is running, 1 otherwise. Typically useful in test scripts. The default 30 seconds can be modified by appending =*T* to the option, where *T* is the desired number of seconds to wait for the server to start.

MISC OPTIONS

--check YawsFile [IncDirs ....]
Test compile a ‘.yaws’ file. Useful in Makefiles when we want to ensure that all .yaws files are syntactically correct

−-version
output version information and exit

ENVIRONMENT VARIABLES

HOME

Is used to determine where we write the temporary files. By default all tmp files end up in $HOME/.yaws. This includes the JIT files that are the result of processed .yaws files and also the so called control file that is used by the daemon to write the port number to which it is listening for control commands such as "yaws --status"

Thus HOME is the handle we use in the control commands to find the control file so that we know where to connect to.

YAWSHOME

Can be used to override the HOME variable. This is useful when we for example are running yaws under port binding programs such as authpriv.

It’s useful by distros that don’t want Yaws to write any files ever in the HOME directory of root.

AUTHOR

Written by Claes Wikstrom

SEE ALSO

yaws.conf(5) **erl**(1)