Downloads

Here you can find the sources, pre compiled 'binaries' and further additional information about GoAccess....

Stable

GoAccess `v1.2` is a major release that gives the ability to parse multiple log files in real-time. It also adds the ability to pipe data in w/o closing the pipe and thus allowing a great amount of flexibility such as the ability to use `tail -f` and still have real-time stats. In addition to GeoIP2 support and other features and fixes.

For the complete list of features and changes, see the **Releases Notes** page.

Download (https://tar.goaccess.io/goaccess-1.2.tar.gz)

Release Notes (/release-notes)

Testing

If you would like to be on top of all the changes occuring to **GoAccess**, feel free to use the GitHub master branch and build from it. Instructions on how to build GoAccess from Github can be found at the bottom of the Download (/download#build) page.

This is where all the development goes. If you already have Git installed, you can get the latest development version via Git.

GitHub (https://github.com/allinurl/goaccess)

Installation

Installing GoAccess is pretty easy. Just download, extract and compile it with:

```
$ wget https://tar.goaccess.io/goaccess-1.2.tar.gz
$ tar -xzvf goaccess-1.2.tar.gz
$ cd goaccess-1.2/
$ ./configure --enable-utf8 --enable-geoip=legacy
$ make
# make install
```

Note: The only dependency is ncurses.
Info: If outputting real-time HTML, ensure port 7890 is opened. See more details (/faq#html-report).

Configure Options

Multiple options can be used to configure GoAccess. For a complete up- to-date list of configure options, run ./configure --help

`--enable-debug`
Compile with debugging symbols and turn off compiler optimizations.

`--enable-utf8`
Compile with wide character support. Ncursesw is required.

`--enable-geoip=<legacy|mmdb>`
Compile with GeoLocation support. MaxMind's GeoIP is required. `legacy` will utilize the original GeoIP databases. `mmdb` will utilize the enhanced GeoIP2 databases.

`--enable-tcb=<memhash|btree>`
Compile with Tokyo Cabinet storage support. `memhash` will utilize Tokyo Cabinet's in-memory hash database. `btree` will utilize Tokyo Cabinet's on-disk B+ Tree database.

`--disable-zlib`
Disable zlib compression on B+ Tree database.

`--disable-bzip`
Disable bzip2 compression on B+ Tree database.

`--with-getline`
Dynamically expands line buffer in order to parse full line requests instead of using a fixed size buffer of 4096.

`--with-openssl`
Compile GoAccess with OpenSSL support for its WebSocket server.

Distributions

It is easiest to install GoAccess on Linux using the preferred package manager of your Linux distribution. Please note that not all distributions will have the lastest version of GoAccess available

Debian/Ubuntu

```
# apt-get install goaccess
```

Note: This may not always give you the latest stable version. To make sure that you're running the **latest stable** version of GoAccess see option below.

Official GoAccess' Debian/Ubuntu Repository

```
$ echo "deb http://deb.goaccess.io/ $(lsb_release -cs) main" | sudo tee -a /etc/apt/sources.list.d/goaccess.list
$ wget -O - https://deb.goaccess.io/gnugpg.key | sudo apt-key add -
$ sudo apt-get update
$ sudo apt-get install goaccess
```

Notes

- For on-disk support (Trusty+ or Wheezy+), run: `sudo apt-get install goaccess-tcb`.
- `.deb` packages in the official repo are available through `https` as well. You may need to install `apt-transport-https`.

Fedora

```
# yum install goaccess
```

Arch Linux

```
# pacman -S goaccess
```

Gentoo

```
# emerge net-analyzer/goaccess
```

OS X / Homebrew

```
# brew install goaccess
```

Note: If GeoIP is needed, append `--with-libmaxminddb` to the command above. You will need to download the City/Country database from MaxMind (http://dev.maxmind.com/geoip/geoip2/geolite2/) and use the option `--geoip-database` to specify the database.

FreeBSD

```
# pkg install sysutils/goaccess
```

OpenBSD

```
# pkg_add goaccess
```

pkgsrc (NetBSD, Solaris, SmartOS, ...)

```
# pkgin install goaccess
```

Slackware

```
# curl https://slackbuilds.org/slackbuilds/14.1/system/goaccess.tar.gz | tar xvz
# cd goaccess/
# GEOIP=yes ./goaccess.SlackBuild
```

Windows

GoAccess can be used in Windows through Cygwin. See packages (/faq#installation) to install.

Distribution Packages

GoAccess has minimal requirements, it's written in C and requires **only** ncurses. However, below is a table of some optional dependencies in some distros to build GoAccess from source.

Distro	NCurses	GeoIP (optional)	GeoIP2 (optional)	Tokyo Cabinet (optional)	OpenSSL (optional

Distro	NCurses	GeoIP (optional)	GeoIP2 (optional)	Tokyo Cabinet (optional)	OpenSSL (optional
Ubuntu/Debian	libncursesw5-dev	libgeoip-dev	libmaxminddb-dev	libtokyocabinet-dev	libssl-dev
Fedora/RHEL/CentOS	ncurses-devel	geoip-devel	libmaxminddb-devel	tokyocabinet-devel	openssl-devel
Arch Linux	ncurses	geoip	libmaxminddb	compile from source (/faq#tokyo-src)	openssl
Gentoo	sys-libs/ncurses	dev-libs/geoip	dev-libs/libmaxminddb	dev-db/tokyocabinet	dev-libs/open
Slackware	ncurses	GeoIP	libmaxminddb	tokyocabinet	openssl

Note: You may need to install tools like gcc, `make` , etc for compiling/building software from source. e.g., `base-devel` , `build-essential` , `"Development Tools"` .

Docker

Before running your own GoAccess Docker container, first create a config file in /srv/goaccess/data. You can start one from scratch or use the one from config/goaccess.conf (https://raw.githubusercontent.com/allinurl/goaccess/master/config/goaccess.conf) as a starting point and change it as needed.

A minimal GoAccess configuration file for a Docker container with a real-time HTML report would need at least the following options to be set `log-format` , `log-file` , `output` , `real-time-html and ws-url` .

Once you have your configuration file all set, clone the repo from Github:

```
$ git clone https://github.com/allinurl/goaccess.git goaccess && cd $_
```

and then build and run the image as follows:

```
docker build . -t allinurl/goaccess
docker run --restart=always -d -p 7890:7890 \
  -v "/srv/goaccess/data:/srv/data"         \
  -v "/srv/goaccess/html:/srv/report"       \
  -v "/var/log/apache2:/srv/logs"           \
  --name=goaccess allinurl/goaccess
```

Note: You may need to replace `/var/log/apache2` according to the location of your webserver access logs.

If everything goes fine, the generated report should live under `/srv/goaccess/html/` .

If you you made changes to the config file after building the image, you don't have to rebuild from scratch. Simply restart the container:

```
docker restart goaccess
```

For more details, please see the Docker section on README.md (https://github.com/allinurl/goaccess#docker) file

Build from GitHub (Development)

```
$ git clone https://github.com/allinurl/goaccess.git
$ cd goaccess
$ autoreconf -fi
$ ./configure --enable-geoip --enable-utf8
$ make
# make install
```

MIT Licensed — Patches, suggestions, and comments are welcome Tweet Follow @goaccess **Star** **7,228**

Last Modified: 2018-06-13