

Fast Start Instructions.

BUILDING CURL-LOADER.

Build it using general C development environment with bash, gcc (3 or 4 series), make, etc on a linux machine.

Building pre-requirements are:

1. openssl binaries;
2. openssl development package with include files (on debian package libssl-dev);
3. ncurses development package.

Adjust Makefile variables to point to the openssl headers and libraries. If you want to specify an openssl development directory with include files (e.g. crypto.h), export environment variable OPENSSLDIR with the value of that directory.

For example: `$export OPENSSLDIR=the-full-path-to-the-directory`

Another known issue is libidn.so, which means, that some linux distributions do have some libidn.so.11, but not libidn.so. Resolve it by creating a softlink.

Run the following commands from your (hopefully) bash linux shell:

```
$tar zxvf curl-loader-<version>.tar.gz
```

```
$cd curl-loader-<version>
```

```
$make
```

By default, we are building both libcurl and curl-loader without optimization and with debugging -g option. To build with optimization and without debugging, please, run:

```
$make cleanall
```

```
$make optimize=1 debug=0
```

If still any building issues, please, fill you free to contact us for assistance.

LOADING CONFIGURATION.

To run the load create a configuration file to be passed to curl-loader by the -f command line option, e.g.

```
#curl-loader -f ./conf-user/user_batch.conf
```

For more examples, please, look at the files in "conf-examples" directory. You may copy an example file and edit it.

Another option is to start with running "\$make menuconfig" configuration GUI, which requires ncurses development package on your system. The dialog window will guide you and create your configuration file in conf-user directory. A limitation of the menu-guided configuration, is that it enables to create only a single UAS URL.

ENVIRONMENT AND SYSTEM.

Running hundreds and thousands of clients, please, do not forget:

- to increase limit of descriptors (sockets) by running e.g.

```
#ulimit -n 10000;
```

- optionally, to set reuse of sockets in time-wait state: by setting

```
#echo 1 > /proc/sys/net/ipv4/tcp_tw_recycle and/or
```

```
#echo 1 > /proc/sys/net/ipv4/tcp_tw_reuse;
```

In some cases you may need to increase the system limits for open descriptors (sockets).

USAGE:

run as a root user:

```
#./curl-loader -f <configuration filename> [other options]
```

For the details about [other options], please see FAQs.

If curl-loader on start reports any problems your configuration file, please, "don't panic" - just fix it.

For more details, please, look into the FAQs page.