NAME

testcurl.pl − (automatically) test curl

SYNOPSIS

testcurl.pl [options] [dir] > output

DESCRIPTION

testcurl.pl is the master script to use for automatic testing of curl off git or daily snapshots. It is written for the purpose of being run from a crontab job or similar at a regular interval. The output is suitable to be mailed to curl-autocompile@haxx.se to be dealt with automatically (make sure the subject includes the word "autobuild" as the mail gets silently discarded otherwise). The most current build status (with a reasonable backlog) will be published on the curl site, at https://curl.haxx.se/dev/builds.html

options may be omitted. See *--setup* for what happens then.

dir is a curl source dir, possibly a daily snapshot one. Using this will make testcurl.pl skip the 'buildconf' stage and thus it removes the dependency on automake, autoconf, libtool, GNU m4 and possibly a few other things.

testcurl.pl will run 'buildconf' (or similar), run configure, build curl and libcurl in a separate build directory and then run 'make test' to test the fresh build.

OPTIONS

--configure=[options]
: Configure options passed to configure.

--crosscompile
: This is a cross-compile. Makes *testcurl.pl* skip a few things.

--desc=[desc]
: Description of your test system. Displayed on the build summary page on the weba site.

--email=[email]
: Set email address to report as. Displayed in the build logs on the site.

--mktarball=[command]
: Generic command to run after completed test.

--name=[name]
: Set name to report as. Displayed in the build summary on the site.

--nobuildconf
: Don't run buildconf. Useful when many builds use the same source tree, as then only one need to do this. Also, if multiple processes run tests simultaneously on the same source tree (like several hosts on a NFS mounted dir), simultaneous buildconf invokes may cause problems. (Added in 7.14.1)

--nogitpull
: Don't update from git even though it is a git tree. Useful to still be able to test even though your network is down, or similar.

--runtestopts=[options]
: Options that is passed to the runtests.pl script. Useful for disabling valgrind by force, and similar.

--setup=[file name]
: File name to read setup from (deprecated). The old style of providing info. If info is missing when testcurl.pl is started, it will prompt you and then store the info in a 'setup' file, which it will look for on each invoke. Use *--name*, *--email*, *--configure* and *--desc* instead.

--target=[your os]
: Specify your target environment. Recognized strings include 'vc', 'mingw32', 'borland' and 'netware'.

INITIAL SETUP

First you make a checkout from git (or you write a script that downloads daily snapshots automatically, find inspiration in https://curl.haxx.se/dev/autocurl.txt):

```
$ mkdir daily-curl
$ cd daily-curl
$ git clone https://github.com/curl/curl.git
```

With the curl sources checked out, or downloaded, you can start testing right away. If you want to use *testcurl.pl* without command line arguments and to have it store and remember the config in its 'setup' file, then start it manually now and fill in the answers to the questions it prompts you for:

```
$ ./curl/tests/testcurl.pl
```

Now you are ready to go. If you let the script run, it will perform a full cycle and spit out lots of output. Mail us that output as described above.

CRONTAB EXAMPLE

The crontab could include something like this:

```
0 4 * * * cd daily-curl && ./testit.sh
```

Where testit.sh is a shell script that could look similar to this:

```
mail="mail -s autobuild curl-autocompile@haxx.se"
name="--name=whoami"
email="--email=iamme@nowhere"
desc='"--desc=supermachine Turbo 2000"'
testprog="perl ./curl/tests/testcurl.pl $name $email $desc"
opts1="--configure=--enable-debug"
opts2="--configure=--enable-ipv6"

# run first test
$testprog $opts1 | $mail

# run second test
$testprog $opts2 | $mail
```