

# disastrous, a del.icio.us link checker

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

First publication of this article on 17 May 2007. Last update on of 4 March 2008

<https://www.bortzmeyer.org/disastrous.html>

---

I just wrote and published yet another del.icio.us link checker.

A feature often requested by del.icio.us users is the ability to check periodically the links they bookmark to detect the broken ones (domains which disappeared, files that were moved or removed, etc). Although, in theory, Cool URIs don't change <<http://www.w3.org/Provider/Style/URI>>, in practice, it is not always the case.

The best place to put such a link checker are certainly inside del.icio.us itself. It could use the Yahoo Web crawlers to do so, since del.icio.us is now a subsidiary of Yahoo. But such a service does not exist yet, may be because the two systems are not actually merged.

So, in the mean time, several link checkers have been written (see the del.icio.us <<http://del.icio.us/help/thirdpartytools>> list or at the end of this article). What is the point of a new one, my **disastrous** program?

- free software ("free as in free speech, not free as in free beer"): source code available and you can modify and redistribute it.
- designed to be run unattended (typically from cron on an Unix machine). Several of its competitors can run only on the desktop, under the control of a human user.
- has a memory : it stores locally the result of the tests (in a SQLite database) and declares a link broken only after N tests in a row failed (N is configurable). It would be very bad, IMHO, to declare a link broken if there was only a temporary network glitch.
- tag the broken links. This is the most del.icio.us way to report a problem.

You can retrieve **disastrous** here : (en ligne sur <https://www.bortzmeyer.org/files/disastrous.py>). **disastrous** has only been seriously tested on Unix so users of MS-Windows systems should be careful. To install **disastrous**, you need a Python environment, the SQLite database engine and the `pysqlite` <<http://initd.org/pub/software/pysqlite>> Python module. The installation of these packages depend on the operating system you use so read the instructions for your system.

Then, run **disastrous** with the `-h` option to get help.

**disastrous** depends on a configuration file, `/.disastrousrc` on Unix (`disastrous.INI` in your default folder on Windows). A typical content is :

```
[disastrous]

# Your account at del.icio.us
name = smith
password = MySecretPassword

# The other options have sensible default values (displayed in the comment)
# but feel free to change them

# The string to use for tagging
# broken_tag = broken
# The number of tests failed in a row before we declare the link broken
# failed_tests_required = 3

# etc
```

Do not worry for the database, it will be created automatically the first time you run **disastrous**. If you want to see what's in the database, for debugging or by curiosity, you can do it from the SQL prompt, for instance :

```
% sqlite3 ~/.disastrous_db
SQLite version 3.5.6
Enter ".help" for instructions
sqlite> SELECT url FROM Bookmarks;
http://www.afnic.fr/
http://www.bortzmeyer.org/
...
```

If you run it on Unix from cron, as recommended, a possible configuration is :

```
30 3 * * * disastrous.py -d 2
```

It will run **disastrous** every day at 3:30 with the debug level set to 2. On MS-Windows, it can probably be run from the scheduler (Control Panel -> Performance and Maintenance -> Scheduled Tasks).

If you like SQL, the following request will find every bookmark which has been flagged as broken at least 3 times in a row :

```
-- Invoke with:
-- % sqlite3 ~/.disastrous_db < find-broken.sql

SELECT Tests.url, Bookmarks.valid, count(*) AS count FROM Bookmarks, Tests,
      (SELECT url, max(date) AS m from Tests WHERE result = 1 GROUP BY url) AS Last_ok
WHERE Tests.url=Last_ok.url AND result = 0 AND date > Last_ok.m AND in_use=1 AND
      Tests.url=Bookmarks.url
GROUP BY Tests.url HAVING count >= 3;
```

As far as I know, here are **disastrous'** competitors :

- **Dead.licious** <[http://www.malarkeysoftware.com/projects\\_dead-licious.html](http://www.malarkeysoftware.com/projects_dead-licious.html)>. Only for MacOS. Apparently non-free.
- **post checker** <[http://www.tiede.dk:8080/roller/kim/entry/del\\_icio\\_us\\_post\\_checker](http://www.tiede.dk:8080/roller/kim/entry/del_icio_us_post_checker)> Written in Java. Free and the code is available at <<http://code.google.com/p/delicious-post-checker/downloads/list>>.
- **Fresh delicious** <<http://freshdelicious.googlepages.com/>>. Non-free, it seems. <https://www.bortzmeyer.org/disastrous.html>