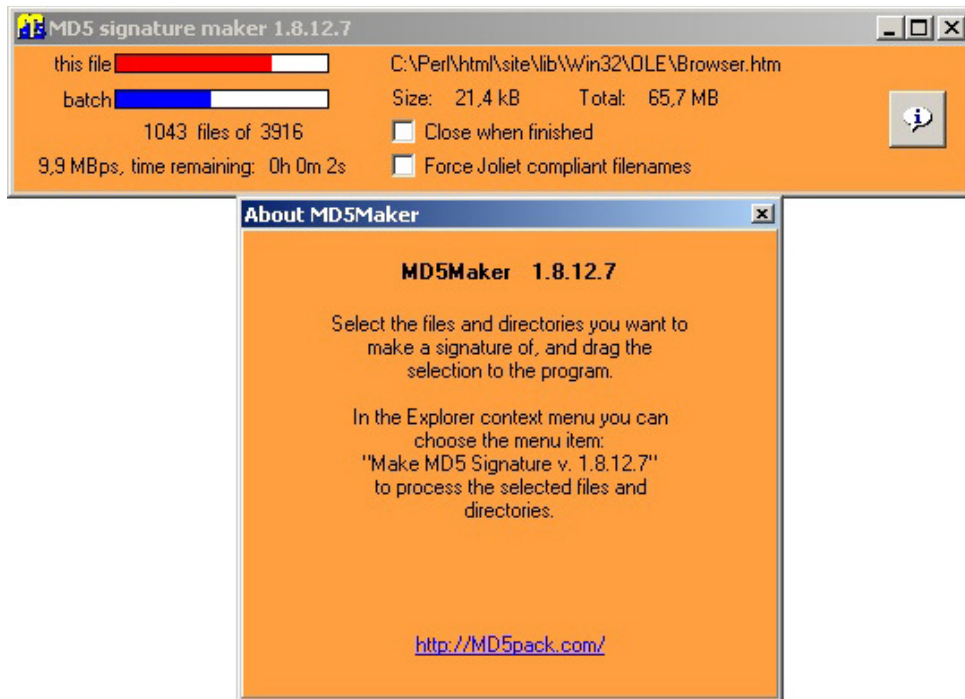


MD5Maker



MD5Maker makes so-called MD5 signature (or 'hash') files.

After data has been copied - from internet or from one type of media to another type - it is good practice to ascertain that the copy is identical to the original.

To check this, make a signature of the original (or the copy) and Verify it with the copy (or the original). The md5 signature file should be saved to the original directory and afterwards copied to the same directory of the copy. It is accomplished easily when you make the signature before you make the copy. With read-only media this is the best way to do it, since you cannot save an md5 signature file to such media. ¹⁾

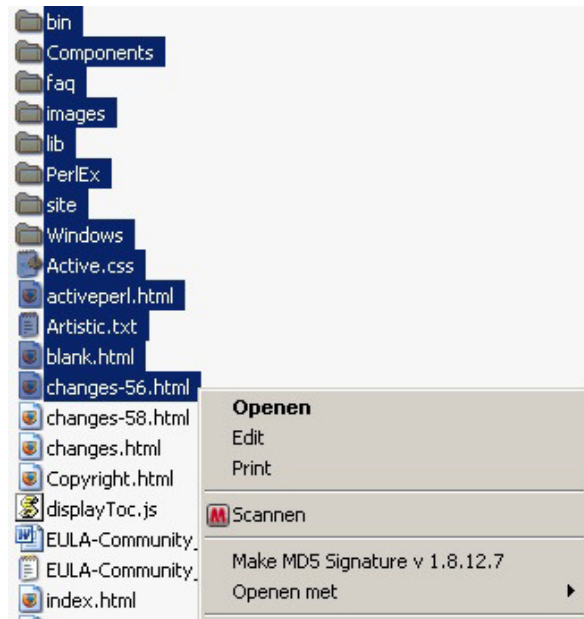
Often signatures are accompanying files posted to the web. You only have to run the signature through MD5Verify to see if the retrieved file is intact.

=====

USAGE

The program starts by right-clicking in the Windows Explorer on a selection of files and/or directories and choosing the "Make MD5 Signature v n.n.n.n" menu item in the context menu that appears (where "n.n.n.n" represents the current version of MD5Maker). You can also 'drag and drop' such a selection on the opened program. Signatures are calculated for all files in the root directory and in all of the subdirectories. Relative paths are added where necessary. The list is written to disc and has the extension .md5.

=====



NOTE

When a new version of MD5Maker should be installed, the old version first has to be de-installed. This is initiated by the 'installer'. If, after de-installation has finished, the file MD5ext1.dll has not been erased, and if this file cannot be erased manually, just log off and then log on again. MD5ext1.dll can then be removed manually.

=====

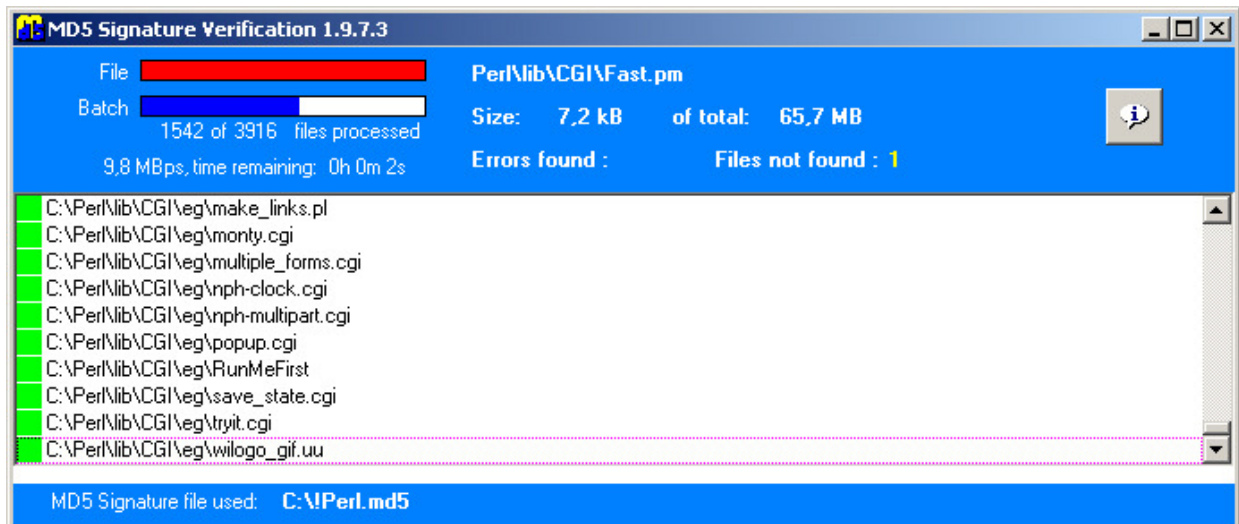
MD5Verify

MD5Verify only processes files with the extension ".MD5". The program starts by double clicking an md5 file, or by drag and drop a selection to the open program. The verification process goes as follows. MD5Verify calculates again the MD5 signatures of all files present in the MD5 file list and compares these to the signatures in the list. The results are put in a list shown in a listbox. To the left of the file names, a colored rectangle is placed. There are three possible outcomes:

GREEN the signatures verify: the files are identical.

RED the signatures differ: the files are not identical.

YELLOW the file in the list could not be found.



Invalid signatures (in general lines in the md5 file that do not conform to the format) are considered to be a comment line. These lines are not displayed in the list, but the number of these lines is shown in **BLACK** in the top panel.

To view the list of files that didn't verify, press the RED filter button. The YELLOW button shows the files not found.

More than one copy of the program can be running simultaneously.

=====

WHAT IS AN MD5 SIGNATURE?

A signature is made by processing the binary contents of a file in a well known and precisely defined manner. This results in a hexadecimal string of 32 characters (= 128 bits). It has been proved that it is extremely unlikely that two different files have the same signatures ²⁾. A small difference between two files causes their signatures to differ greatly, so it is easy to see they do differ indeed.

The execution of the program takes place in two phases. In phase 1 a complete list of files present in the selection is built and their lengths are summed. In phase 2 the list of signatures is built and written to file. This file can be used to later check whether changes have taken place to the original files or to check a copy of the files. The program can be stopped prematurely in phase 2 only.

=====

THE FORMAT OF AN MD5 SIGNATURE FILE.

These are text files containing one or more lines having the following structure:

```
88326ff1342bde33514f7aa857ccac3c *changes-56.html
91d85a8995c8f4c0b321e89337a5c155 *bin\c2ph.html
c0772f20ad09e116c40ae74f442a6459 *bin\cpan.html
eccab9c6975996018756f0361ac7f369 *bin\dprofpp.html
```

Each line starts with a set of 32 hexadecimal characters (only 0-9 and a-f are allowed). Upper or lower case can be used. Then the characters 'space' and 'asterisk' follow. Finally, the file name follows, preceded by a 'relative path' if the file was in a subdirectory.

In the first line of the example the file 'changes-56.html' is situated in the selected directory itself. In

the second line, 'bin\' is the relative path, and 'c2ph.html' is the file name. This file is situated in the directory 'bin'.

A special form of md5 signature is now and again posted together with a file. It has, for example, the following filename: foo.exe.md5, and has the following format:

```
fae1db8f3d3012c7b1af10beddc3cc12
```

It is the signature of the file foo.exe. MD5Verify is able to process this type of signatures.

=====

1) There is a way around this problem. If you make a signature file of data on a CD-R and save it to your hard drive, a text editor will show that the drive letter is added in front of each file name:

```
539d5b568f4cb4bd1a80143a6c371e7e *G:\bin\cpan.html
```

By replacing this drive letter by the one that belongs to the drive you used to copy the data to, you still can verify the copy. This works both ways!

=====

2) It is possible for two non-identical files to have the same MD5 hash. An example can be found at <http://stols.com/net/collision/collision!.zip>. The files 'abc2.bin' and 'def2.bin' are 128 bytes in size only and have different bits in 6 places. Nonetheless, they have identical MD5 hashes. Ref.:

<http://www.reussirsurlenet.fr/v2/sections.php?op=viewarticle&artid=9>

http://www.doxpara.com/md5_someday.pdf

<http://www.codeproject.com/KB/security/HackingMd5.aspx>

=====