ExactFile Evaluation

**This is one of a series of reviews of available programs for computing and verifying checksums. The information documented below is an account of my experience and contains my own comments and opinions. Your experience may be different depending on your needs and use of the program.*

Note: ExactFile is in beta, and is a replacement of FileCheckMD5. Updates have not been made for over two years, but as the creator says, it does what he needs it to do, and it is free.

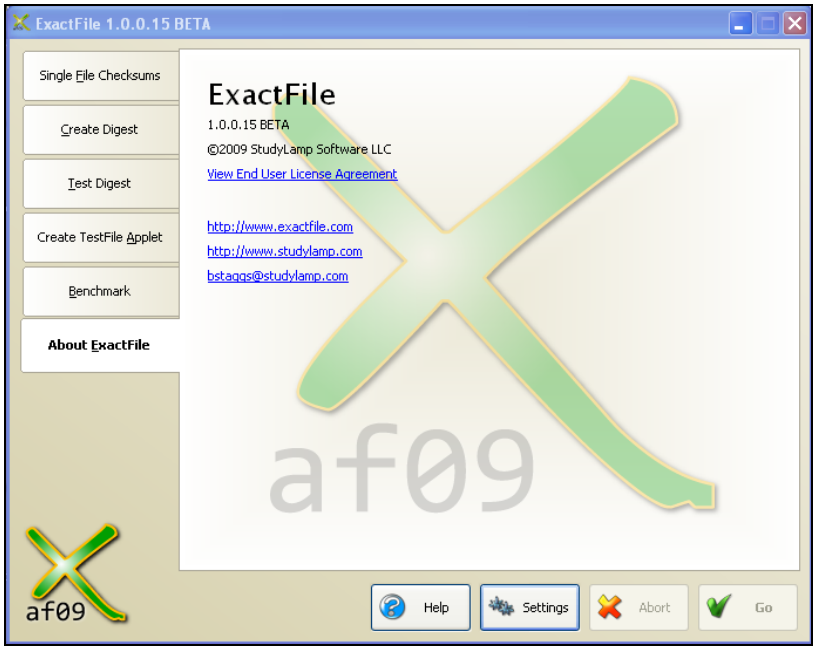
To begin, this program was downloaded from ExactFile's home page and installed by our company IT department onto my computer.

I began by briefly reviewing the materials available on ExactFile's website which provides information on the checksum algorithms it supports, how to calculate checksums on single files, on a folder, verification, creating a test file (for future verification), how to calculate computational speed of hash methods, and information on general settings.

ExactFile will let you calculate checksums on individual files as well as on a group of files and has the ability to calculate checksums using a variety of methods including MD5, and others in the SHA, RIPEMD, and TIGER families.

Overview

When you open the program you are brought to a screen with six tabs as shown below. Each tab is set up for a different task and will be described in more detail.



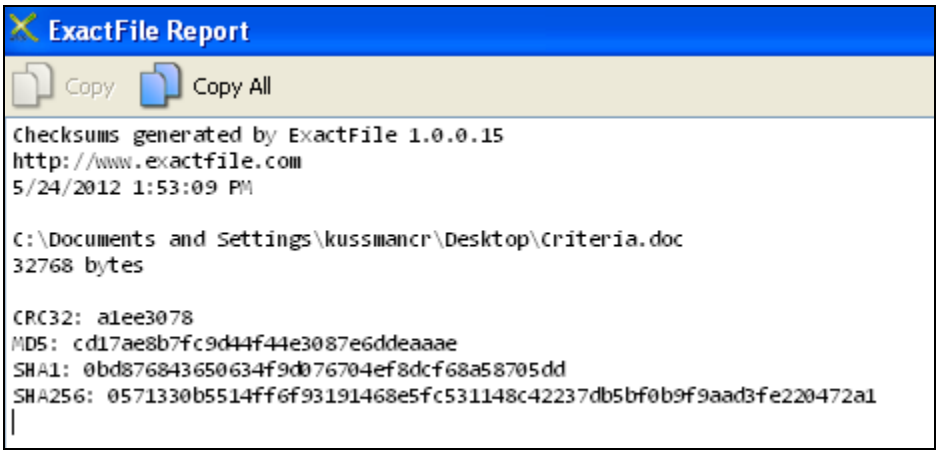
About ExactFile Tab

Single File Checksums Tab

To calculate a checksum on a single file use the Single File Checksum Tab.

- Type in the file path or navigate to the file of interest in its current location.
- Select the one or more Hash Methods you would like to use.
- Click Go, at the bottom of the application.

This produces a report that includes the name of the program used (ExactFile), date, time, file location, size in bytes, and a list of the checksum values for each algorithm selected.



To save this information, select Copy All and paste into a program of your choice.

This method would work best when trying to verify the authenticity of a file, rather than for creating checksums, as saving the information is not automatic.

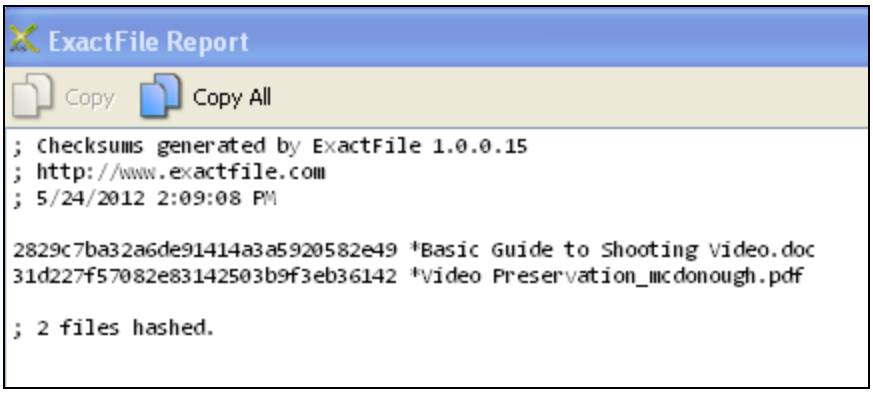
Create Digest Tab

A checksum digest is a report that records the checksum values. To create a checksum digest for files within a folder…

- Navigate to the folder location or drop and drag folder to top box.
- Click or unclick "Include subfolders" based on your need.
- Click or unclick "Include full paths in output" [remember if you plan on moving files later, do not select this feature].
- Click or unclick "Output to File", then select the location and name of the digest file. [If the circle arrow button is selected, the file will be placed in the same folder as the files the checksum digest is being run on and titled checksum.md5. You can move the file to another location and choose a different name by clicking the paper icon next to the circle arrow.]
- Select the Method and Format from the drop down list. [only one may be selected]

- Click Go.

This produces a report that includes the name of the program used (ExactFile), date, time, file name and checksum value selected, as well as the total number of files hashed. If selected, this file is automatically saved in the chosen location with the listed file name.



Note: If you name the file yourself, you must include the .md5 file extension (or whichever method you are using) if you want ExactFile to be able to use the digest to verify the files later.

Nested Folders
Creating a digest of nested folders, will produce one file with the checksums of all the files located in the top folder.

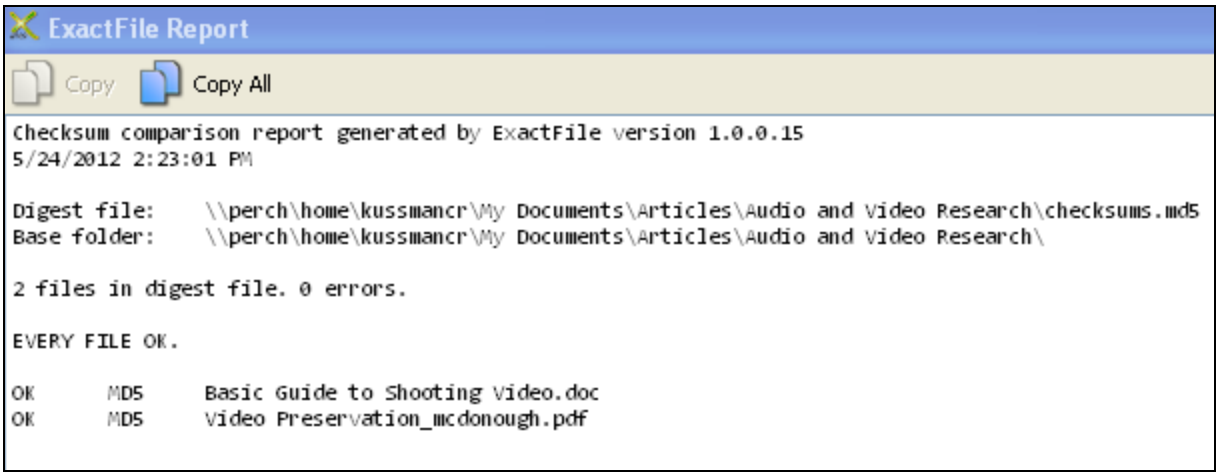
Opening and Viewing Checksum Digests
To open a checksum digest to view the hash values themselves, you must use a text editor or another program. To verify checksums, use the Test Digest Tab.

Test Digest Tab

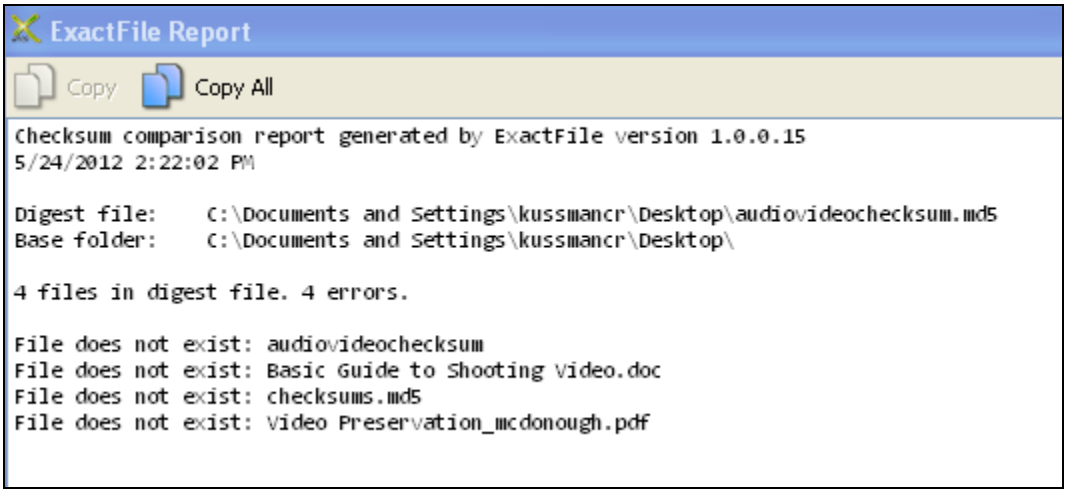
This tab is used to verify checksums.

Double Clicking on a Digest File
When you double click on a file with .md5, .sha1, or other hash codes as the file extension, the Test Digest is automatically opened and run. [This program can read digests created by other programs as well as its own digest files. It was verifying digests created with FastSum.]

If this is what you have done, a report will be created that lists the file names (if verbose report is selected) and states if the files are the same as before or not. The hash file number is not listed in this report. It just verifies if the files are OK or not.



This Verbose Report shows that the files are OK.

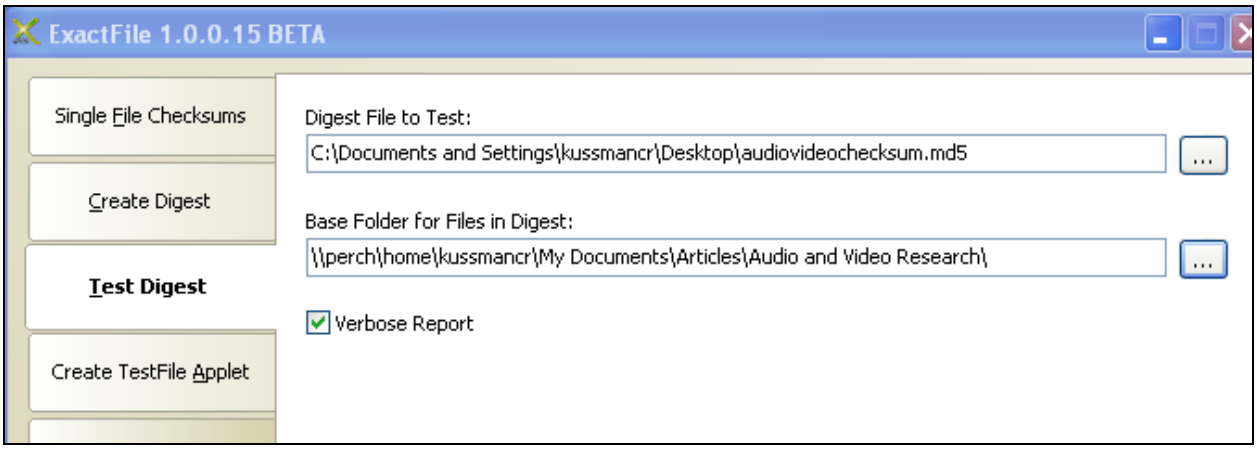


This Verbose Report shows that the files do not exist.

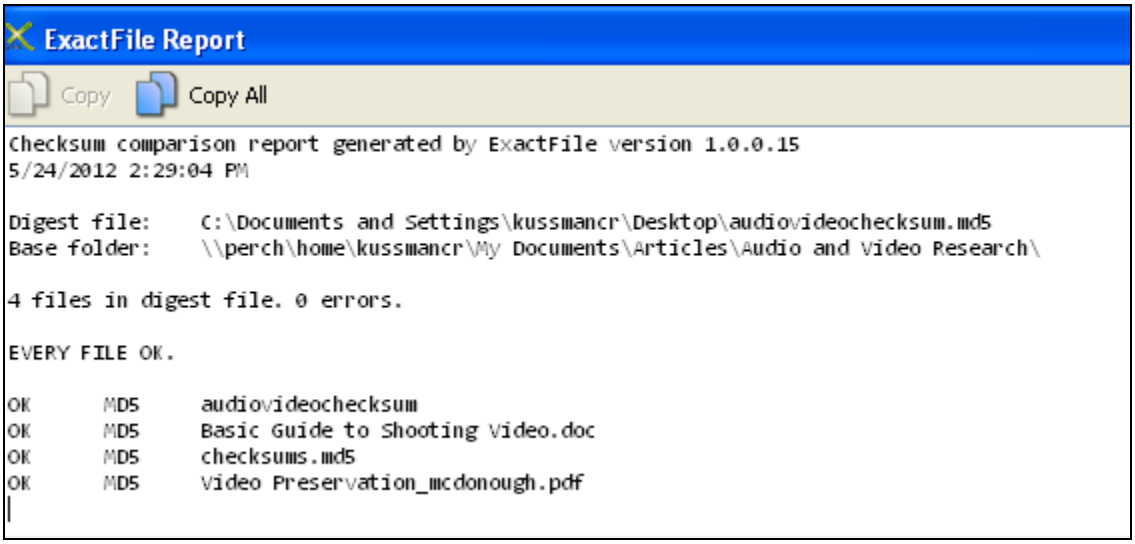
Running Test Digest from ExactFile Application

You can also run this verification from the Test Digest tab itself. This allows you to select a digest file as well as the location of the file. This is useful if the digest files are not kept in the same folder as the files.

In the example above, when double clicking on the digest file on my desktop, the report told me that the files did not exist. So as shown in the screen shot below, I was able to select the same digest file, and then point to the correct location of the files. This returned an "all ok" report.



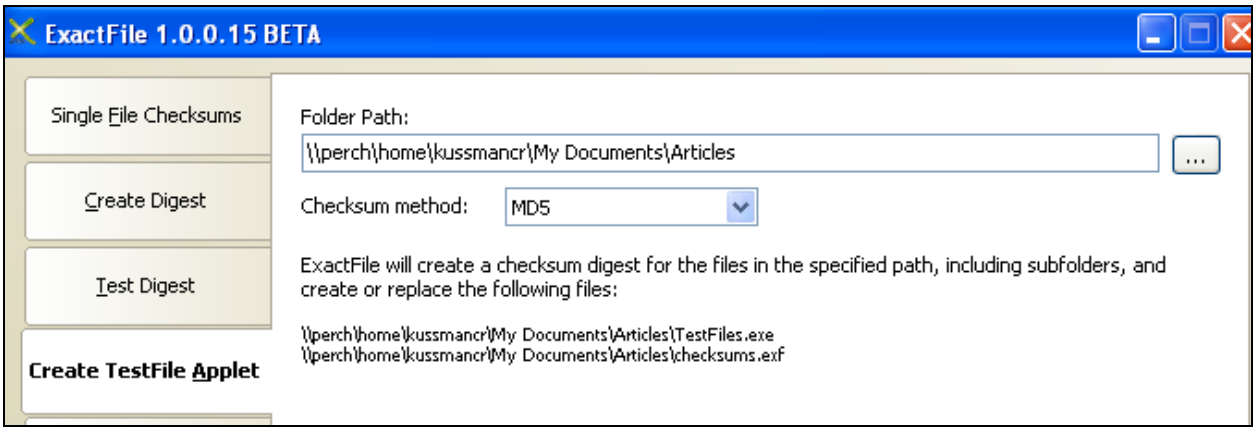
Ability to select a digest file in one location and point to the files in a separate location.



The files are now all OK.

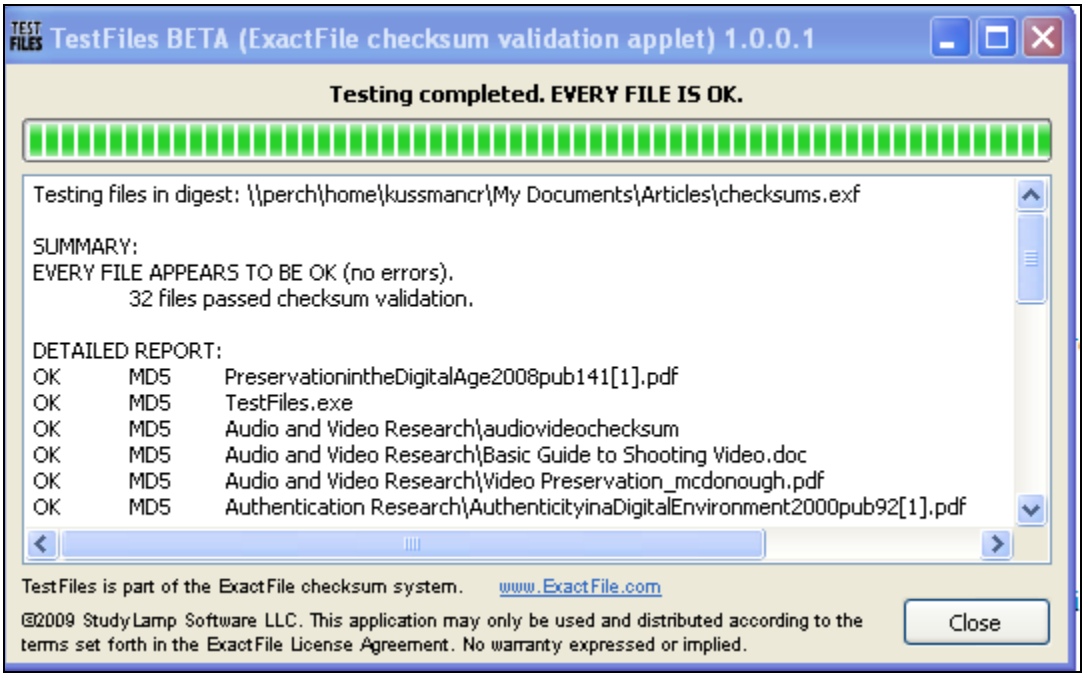
Create TestFile Applet Tab

The TestFile Applet was created specifically for use when burning CDs, however it can be used in any situation in which the receiver of files has no other way to verify the files checksums. When using this tab, a digest is created in the same way as if you were using the Create Digest tab, but is also creates a TestFile.exe file that can be added to the CD when burning the files.



Applet Files

Once both files are on the CD, this TestFile.exe can be opened and run, and just like the Test Digest Tab of ExactFile. TestFile checks the files on the CD against the previously generated checksum values to verify that the items have not changed.

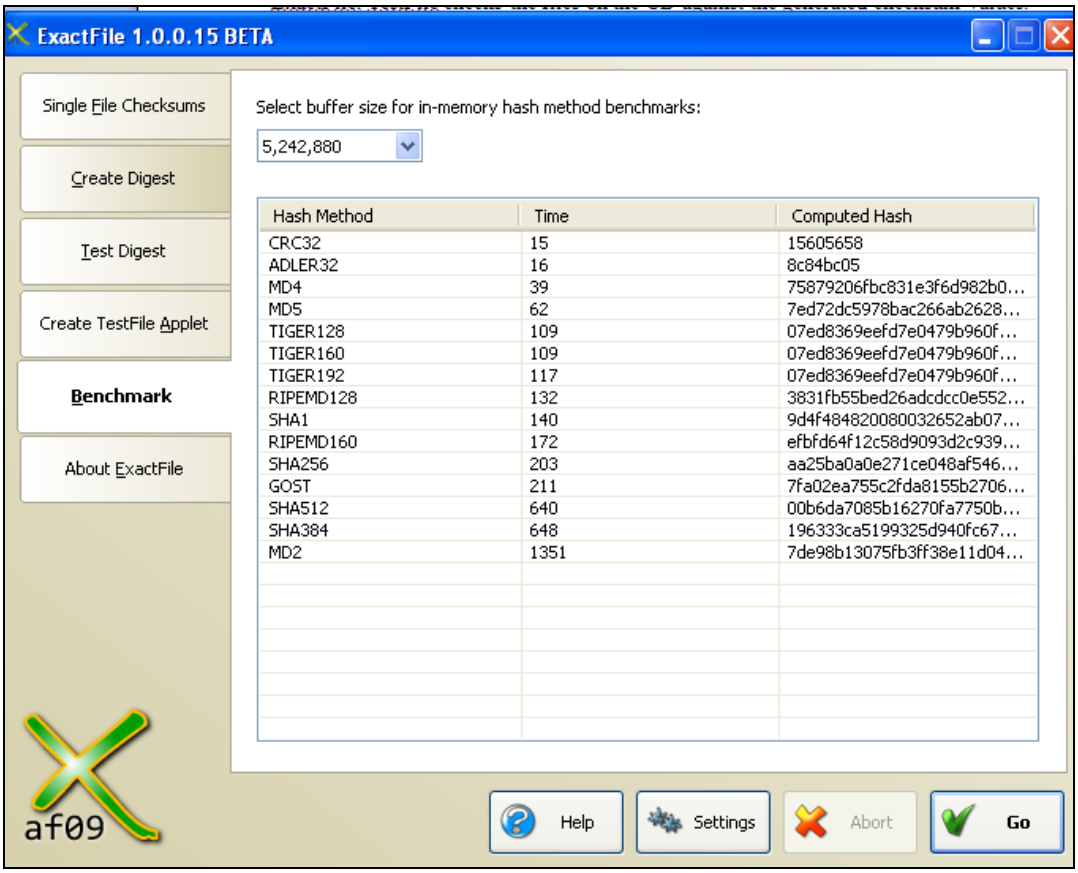


TestFile.exe report

[This may also be useful when sending files to others via other methods who may or may not have a way to verify checksum digests.]

Benchmark Tab

Benchmark is a tool that can be used to determine how much time it may take to compute the various different hash methods based on buffer size.



Other Information

Settings

The settings is limited to selecting the number of processor cores to use and to the font and size of the reports.

Help Menu

There is a help menu at the bottom of that screen that can be accessed in one click for information on how to use each tab.

EXF Command Line Tool

There is also a command line tool available for those who wish for this application.