

Forum: [Hack Forums](#)

Thread: [C# MD5 & SHA256 File Hash Checker with WinForms](#)

SIM-587985 - September 22, 2022, 6:19 pm

MD5 & SHA256 File Hash Checker using Winforms (C#) By: Wardyn (Well I made the file after learning how)

GitHub: <https://github.com/CPYAY/FileHashChecker>

VirusTotal:

<https://www.virustotal.com/gui/file-analysis/NWQ2ODJhOGU1MDU5NGZmNTJjNDNiOGNjOTMzZW E5ZmE6MTY2Mzg5NjUxNg==>

It says 3 malware when I scan the zip but I uploaded from my visual studio and its hella basic, so look at it if you wanna be safe.

In the backup ZIP file is the MD5 only version, but I made it where it gets the SHA256 as well. This changes with the name of the program, so I thought it could help prove authenticity somewhere.

This is my first C# program, thanks for checking it out.

Basically, I am hoping to figure out how to use this to scan through ClamAVs hash database and maybe check virus total.

I kinda have an idea so don't give me any spoilers. It is no fun if yall do it for me.

If you like it make sure to click the heart.

MavenGradle - September 22, 2022, 8:09 pm

I will congratulate you on the originality of the project, because for starters it's not malware.

But, you will want to work on your programming style.

For instance, it would be better to give meaningful names to your variables and objects. Calling a textbox textbox1 is not saying much. Instead, you could go for the name txtFilePath, so that we can immediately see that it's a textbox and also know what it's purpose is.

As the KISS principle and OOP says: your classes should be limited to doing one thing. That is why it would be better to introduce a HashCalculator class in your solution and put your hash calculation functions there, so that you can use your code across multiple forms or in other parts of your code in a later stage. Do also give your forms a proper name, like frmHashCalculator for example.

Another thing you can improve upon is error handling and logging.

And one more note regarding git repositories: you placed a zip file inside your git repo. That does not belong there.

SIM-587985 - September 23, 2022, 6:08 am

(Sep 22, 2022 08:09 PM)MavenGradle Wrote: [\[CITE\]](#) I will congratulate you on the originality of the project, because for starters it's not malware.

But, you will want to work on your programming style.

For instance, it would be better to give meaningful names to your variables and objects. Calling a textbox `textbox1` is not saying much. Instead, you could go for the name `txtFilePath`, so that we can immediately see that it's a textbox and also know what its purpose is.

As the KISS principle and OOP says: your classes should be limited to doing one thing. That is why it would be better to introduce a `HashCalculator` class in your solution and put your hash calculation functions there, so that you can use your code across multiple forms or in other parts of your code in a later stage. Do also give your forms a proper name, like `frmHashCalculator` for example.

Another thing you can improve upon is error handling and logging.

And one more note regarding git repositories: you placed a zip file inside your git repo. That does not belong there.

Thanks!

I honestly just don't know how to delete something on GitHub, it automatically uploaded everything I had in my computer folder.

I will go through and try to dress up everything and make it look nice before trying to move forward.

In my head, I guess I knew that I should name stuff, it is just so much easier to remember as `textbox1` etc.

How Can I handle errors better?

And in other programming languages I can use "goto" or "call". Is there anything similar in C#?

MavenGradle - September 23, 2022, 8:23 am

(Sep 23, 2022 06:08 AM)Wardyn Wrote: [\[CITE\]](#) How Can I handle errors better?

And in other programming languages I can use "goto" or "call". Is there anything similar in C#?

Error handling is done by catching or throwing exceptions that make sense. For example, passing a null value in a function that does not compute with null values should throw a null reference / pointer exception.

Catching exceptions makes sense for functions that you know you can recover from. For example, if an integer variable with a value of zero is passed to a function that divides another integer value by said value parameter and this zero value was passed in by the end user: in such a case, you can catch, log and display this exception with a message that dividing by zero makes the end user an idiot.

Fun little gotcha: diving by zero does not throw exceptions when working with floating point numbers according to the IEEE 754 specs.

Some programmers like to try-catch all functions and all exceptions but this is an extremely bad practice because some error are unrecoverable. An OOM exception (out of memory) is such an example.

Remember that error handling is the opposite of pokemon: you do not have to catch them all. Only the ones you truly need.

As to the goto functionalities of programming languages:

The programming languages may provide this feature but this is simply not done unless you write code in ASM. Under no sane circumstances should you use a goto in C# or almost any other programming language for that matter, because it is universally accepted as a code smell (e.g. badly constructed code). As such I will simply state that you should know about their existence but never,

ever use a goto in real life; you might as well use Excel as a database. Please don't!