

Forum:[Hack Forums](#)

Thread: [BTCH - Bytes to C++ Header \(.h file\) - Convert ANY File to Array of Bytes Include](#)

Hackers - February 23, 2024, 7:25 am

BTCH

A small utility tool that will take any input file, convert to array of bytes, and save to a .h output file. You can then include this header file in any of your programs, to have access to the bytes array & the data size. This is what allows you to then load the data and either save as a file, to a process, etc.

BTCH stands for Bytes To C++ Header. It's a very small program but might be useful to you. I am releasing this for free as it is easy to make and needs adjusting to implement any sort of encryption / intellectual property protections.

For testing purposes when you use BTCH it will create the .h header file (or whatever your output file name is) AND it will also re-create the original file as TEST-original (ex: TEST-original.exe). This is so you can make sure there are no problems with converting to/from byte array & original.

Get The Sauce

Here is the single file source code for your byte array header template creator:

Spoiler ([Click to View](#))

Code #include <iostream>

#include <fstream>

#include <vector>

// Function to convert byte array back to original file

```
void testConversion(const std::vector<unsigned char>& data, const std::string& originalFileName) {  
    std::string testFileName = "TEST-" + originalFileName;  
    std::ofstream outputFile(testFileName, std::ios::binary);
```

```
    if (!outputFile.is_open()) {  
        std::cerr << "Failed to create " << testFileName << std::endl;  
        return;  
    }
```

```
    for (auto byte : data) {  
        outputFile.put(static_cast<char>(byte));  
    }
```

```
    std::cout << "Test file created successfully: " << testFileName << std::endl;  
}
```

```
int main(int argc, char* argv[]) {  
    if (argc != 3) {
```

```

std::cerr << "Usage: " << argv[0] << " <input.original> <output.h>" << std::endl;
return 1;
}

std::ifstream originalFile(argv[1], std::ios::binary);
if (!originalFile.is_open()) {
    std::cerr << "Failed to open " << argv[1] << std::endl;
    return 1;
}

std::vector<unsigned char> buffer(std::istreambuf_iterator<char>(originalFile), {});

std::ofstream headerFile(argv[2]);
if (!headerFile.is_open()) {
    std::cerr << "Failed to create " << argv[2] << std::endl;
    return 1;
}

headerFile << "#ifndef PAYLOAD_DATA_H\n#define PAYLOAD_DATA_H\n\n";
headerFile << "unsigned char payloadData[] = {";
for (size_t i = 0; i < buffer.size(); ++i) {
    headerFile << "0x" << std::hex << (int)buffer[i];
    if (i < buffer.size() - 1) headerFile << ", ";
    if ((i + 1) % 12 == 0) headerFile << "\n";
}
headerFile << "};\n\n";
headerFile << "unsigned int payloadDataSize = sizeof(payloadData);\n\n";
headerFile << "#endif // PAYLOAD_DATA_H\n";

std::cout << "Conversion completed successfully." << std::endl;

// Call test function to verify conversion integrity
testConversion(buffer, argv[1]);

return 0;
}

```

You can also view the repo here:

<https://github.com/LNodesL/BytesToCppHeader>

Compiling Example:

Code g++ main.cpp -o BTCH -static

Usage Example:

Code ./BTCH original.exe payloadData.h

Dannisthethird - February 23, 2024, 7:30 am

Awesome work! you are showing great advances with C++ keep going man! <3

Hackers - February 23, 2024, 8:37 am

(Feb 23, 2024 08:30 AM)Dannisthethird Wrote: [\[CITE\]](#) Awesome work! you are showing great advances with C++ keep going man! <3

Thank you, I am learning as fast as possible but also trying to make sure I don't move too quickly so I can retain all the info I am learning.

This is something I had to create a couple weeks ago for my program and I've now basically re-created it 3X at this point across different codebases so I decided to release this early version here. I will probably also release a more general tool that converts to bytes array and prints them out in terminal, or to an output file, but without any of the templating (for the .h header file) so that people can use that for other things easily.

One thing I'm really enjoying is creating mini tools to release on HF that can be used in combination together.

For example, you can take my SFED-AES-256 program (encrypt & decrypt single file, command line usage) and encrypt original.exe. Then you could run that through BTCH to convert to array of bytes & header file. Then you can re-use SFED-AES-256 in runtime to decrypt the payload before recreating the exe or pushing to a process.

<https://hackforums.net/showthread.php?tid=6264320>

Can also do the same thing with my tool SFEDXOR which is same usage but with XOR instead of AES256:

<https://hackforums.net/showthread.php?tid=6264286>

My plan is to release more and more mini tools as I build my own programs. If someone is watching close enough I'm sure they can take my mini tools now & in the future to create their own builders too if they wanted to.

Phantom_NiNja - February 26, 2024, 2:55 pm

(Feb 23, 2024 08:25 AM)nodes Wrote: [\[CITE\]](#) BTCH

A small utility tool that will take any input file, convert to array of bytes, and save to a .h output file. You can then include this header file in any of your programs, to have access to the bytes array & the data size. This is what allows you to then load the data and either save as a file, to a process, etc.

BTCH stands for Bytes To C++ Header. It's a very small program but might be useful to you. I am releasing this for free as it is easy to make and needs adjusting to implement any sort of encryption / intellectual property protections.

For testing purposes when you use BTCH it will create the .h header file (or whatever your output file name is) AND it will also re-create the original file as TEST-original (ex: TEST-original.exe). This is so you can make sure there are no problems with converting to/from byte array & original.

Get The Sauce

Here is the single file source code for your byte array header template creator:

Spoiler ([Click to View](#))

```
Code #include <iostream>
```

```
#include <fstream>
```

```
#include <vector>
```

```
// Function to convert byte array back to original file
```

```
void testConversion(const std::vector<unsigned char>& data, const std::string& originalFileName) {  
    std::string testFileName = "TEST-" + originalFileName;  
    std::ofstream outputFile(testFileName, std::ios::binary);
```

```
    if (!outputFile.is_open()) {  
        std::cerr << "Failed to create " << testFileName << std::endl;  
        return;
```

```

    }

    for (auto byte : data) {
        outputFile.put(static_cast<char>(byte));
    }

    std::cout << "Test file created successfully: " << testFileName << std::endl;
}

int main(int argc, char* argv[]) {
    if (argc != 3) {
        std::cerr << "Usage: " << argv[0] << " <input.original> <output.h>" << std::endl;
        return 1;
    }

    std::ifstream originalFile(argv[1], std::ios::binary);
    if (!originalFile.is_open()) {
        std::cerr << "Failed to open " << argv[1] << std::endl;
        return 1;
    }

    std::vector<unsigned char> buffer(std::istreambuf_iterator<char>(originalFile), {});

    std::ofstream headerFile(argv[2]);
    if (!headerFile.is_open()) {
        std::cerr << "Failed to create " << argv[2] << std::endl;
        return 1;
    }

    headerFile << "#ifndef PAYLOAD_DATA_H\n#define PAYLOAD_DATA_H\n\n";
    headerFile << "unsigned char payloadData[] = {";
    for (size_t i = 0; i < buffer.size(); ++i) {
        headerFile << "0x" << std::hex << (int)buffer[i];
        if (i < buffer.size() - 1) headerFile << ", ";
        if ((i + 1) % 12 == 0) headerFile << "\n";
    }
    headerFile << "};\n\n";
    headerFile << "unsigned int payloadDataSize = sizeof(payloadData);\n\n";
    headerFile << "#endif // PAYLOAD_DATA_H\n";

    std::cout << "Conversion completed successfully." << std::endl;

    // Call test function to verify conversion integrity
    testConversion(buffer, argv[1]);

    return 0;
}

```

You can also view the repo here:

<https://github.com/LNodesL/BytesToCppHeader>

Compiling Example:

Code g++ main.cpp -o BTCH -static

Usage Example:

Code ./BTCH original.exe payloadData.h
Why would you want to convert a file to bytes?

Hackers - February 27, 2024, 7:38 am

(Feb 26, 2024 03:55 PM)randommanty89 Wrote: [\[CITE\]](#) Why would you want to convert a file to bytes?

When you save a file as binary or an array of bytes it's very easy to then include it in another file and use it in your code. For example, you could use a byte array include to then save it back to a file, push to a process, etc. Lots of options depending on the format of your file before you use BTCH or after.

Think of it as similar to a resource, but without using resources and being included in your sourcecode from a .h file as a data source. Up to the dev why they do it this way, or how they use it. BTCH is just handling the conversion & template file creation for you. For example perhaps you have a program that compiles another program - this could be used in program1 to handle the conversion and header file generating automatically during that process.

Stanley - March 16, 2024, 4:29 pm

I'm Stanley HF's helper bot. The OP (original poster) of this thread wants everyone to see this page and has elected to bump it to the top of the showthread listing. Since HF doesn't want a ton of LQP (Low Quality Posts) this is the best method of bumping your content. There is a small ?yte fee for each bump. Any OP can bump any of their threads on HF from the linked icon on the top of their first post.

If you haven't already read the first post of this thread we encourage you to do so.

Thanks,
Stanley

Hackers - April 28, 2024, 9:58 pm

This tool should still be useful to many :)

I used it numerous times in my other programs, really glad I made this. There are other tools (HxD) that can handle this too, but honestly for automated header creation, BTCH is fantastic & I don't mean to toot my own horn!

nucleo_type09 - April 29, 2024, 5:54 am

bro you're seriously advancing in c++ :)

Hackers - April 29, 2024, 6:07 am

(Apr 29, 2024 05:54 AM)nucleo_type09 Wrote: [\[CITE\]](#) bro you're seriously advancing in c++ :) thanks, I appreciate that! This was actually one of the first things I did before the more advanced stuff ;)

nucleo_type09 - April 29, 2024, 6:10 am

(Apr 29, 2024 06:07 AM)nodes Wrote: [\[CITE\]](#) thanks, I appreciate that! This was actually one of the first things I did before the more advanced stuff ;) really? so this was some of the first things you wrote in c++?

Hackers - April 29, 2024, 6:19 am

(Apr 29, 2024 06:10 AM)nucleo_type09 Wrote: [\[CITE\]](#) really? so this was some of the first things you wrote in c++?

not the very first but yeah because all it's doing is just reading a file, converting to bytes and saving to a file. It's a small program all things considered. It took me a bit to get it working obviously but

wasn't that difficult once I understand some of the types needed to do this.

nucleo_type09 - April 29, 2024, 7:09 am

(Apr 29, 2024 06:19 AM)nodes Wrote: [\[CITE\]](#) not the very first but yeah because all it's doing is just reading a file, converting to bytes and saving to a file. It's a small program all things considered. It took me a bit to get it working obviously but wasn't that difficult once I understand some of the types needed to do this.

nice one bro, just i would add const where possible and TC blocks to handle errors, that's all, it pretty good by itself but adding those will make it even better :)

boyah1337 - April 29, 2024, 9:29 pm

(Apr 29, 2024 07:09 AM)nucleo_type09 Wrote: [\[CITE\]](#) -
Lord have mercy.

Code static const void testConversion(const std::vector<unsigned char> *data, const std::string *originalFileName) noexcept

Congratz you have made it worse, exposing pointers is a big no!

Why would you wanna do that in the first place? the whole point of std::string, std::vector is to prevent all this pointer bullshit.

Stating that a function is not throwing and throwing inside (With catch) is crazy work.

Code throw std::runtime_error("data or original file name not supplied!");

Your whole exception logic is flawed, exceptions are a big NO in production code, if you can avoid them do not use them, they are not efficient and very costly. Use instead std::optional or std::expected.

Code if (!writeDestFile.is_open() || !writeDestFile.good())

No sense. Both do the same thing in this context.

Code for (size_t i = 0; i < buffer.size(); ++i)

{
NO NO NO.

Code const std::vector<unsigned char> buffer(std::istreambuf_iterator<char>(readSourceFile), {});
Instead of regular for loop, use std::span and for each.

Code std::string sourceFile = "", destFile = "";

Why std::string? You are just creating another copy for no reason.

You are not manipulating anything, it is read only in your case. Use std::string_view.

Code for (const auto &byte : *data)

outputFile.put(static_cast<char>(byte));

This is not necessary since this type is very small. do it by value. (Reason is it can be passed directly into a register)

Hackers - April 30, 2024, 5:42 am

(Apr 29, 2024 09:29 PM)boyah1337 Wrote: [\[CITE\]](#) Lord have mercy.

[snipped]

While I appreciate you giving nucleo feedback in your post, I encourage you to make your own version of this and post a thread and then link to it so we can compare and see your improvements

from nucleo's code. He's a good guy - your post came across as a little harsh in my opinion.

Nucleo, for future please also make your own thread & link to it.
It makes it cleaner and lets people discuss each codebase better.

boyah1337 - April 30, 2024, 6:54 am

(Apr 30, 2024 05:42 AM)nodes Wrote: [\[CITE\]](#) While I appreciate you giving nucleo feedback in your post, I encourage you to make your own version of this and post a thread and then link to it so we can compare and see your improvements from nucleo's code. He's a good guy - your post came across as a little harsh in my opinion.

Nucleo, for future please also make your own thread & link to it.
It makes it cleaner and lets people discuss each codebase better.
No, I don't need to post a thread. You provide code, I review it. If you don't want to get criticism don't post code, simple. His code is bad, There is no way around it.

Whenever you guys get your real job, you will understand. I am just trying to be nice.

Hackers - April 30, 2024, 6:55 am

(Apr 30, 2024 06:54 AM)boyah1337 Wrote: [\[CITE\]](#) No, I don't need to post a thread. You provide code, I review it.

Whenever you guys get your real job, will you tell the architect, write your own code? Or will you fix your own?

Lol...

"get your real job"

Kid, you probably make what I made 10 years ago... You are coming across as rude.

You don't get to come in, make 0 threads, then claim to be some sort of godly sys architect. Prove it with your own code, not by being slightly rude to people who do contribute.

boyah1337 - April 30, 2024, 6:58 am

(Apr 30, 2024 06:55 AM)nodes Wrote: [\[CITE\]](#) Lol...

"get your real job"

Kid, you probably make what I made 10 years ago... You are coming across as rude.

You don't get to come in, make 0 threads, then claim to be some sort of godly sys architect. Prove it with your own code, not by being slightly rude to people who do contribute.

You are probably some web coder judging by your native skills. There is no need to get mad, In a real job, you will get a code review from a senior member and you will get 200 comments if needed. Stop being a kid about it and embrace your mistakes.

Age has nothing to do with knowledge, you will also learn that in a few years in the industry.

Hackers - November 19, 2024, 12:50 pm

If anyone wants to use this & has questions, just let me know :) !

Hackers - March 16, 2026, 8:42 am

If anyone has any questions, let me know!