

Forum:[Hack Forums](#)

Thread: [How to Load file as RawData to memory c++](#)

H4x10r - July 23, 2020, 1:32 am

Hey Folks,

need a bit of help here !

normaly i got a file as unsigned char rawData[bla] = 0x34 etc.

writing this file to binary and execute it ok fine.

But how is it possible to download a file to an unsigned char rawData[bla] in memory ?

so i could use it later on, so i could stop carrying the rawData in the bin , i could load it whn i need it.

thanks in adv.

Xora - July 23, 2020, 2:42 am

Code ...

```
#include <WinInet.h>
#pragma comment(lib, "WinInet.lib")
#include <urlmon.h>
#pragma comment(lib, "urlmon.lib")
```

```
typedef struct
```

```
{
    LPVOID Base;
    DWORD SizeInKilobytes;
} URL_FILE;
```

```
URL_FILE
```

```
UFDDownload(
    LPSTR URL,
    DWORD SizeInKilobytes
```

```
)
```

```
{
    URL_FILE File;
    memset(&File,
        0,
        sizeof(File));
```

```
if (!URL || '\0' == URL[0] || !SizeInKilobytes)
    return File;
```

```
HINTERNET Internet = InternetOpenA(
```

```
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
Chrome/84.0.4147.89 Safari/537.36",
```

```
    INTERNET_OPEN_TYPE_PRECONFIG,
```

```
    NULL,
```

```
    NULL,
```

```

    0
);

if (!Internet)
    return File;

HINTERNET InternetConnection = InternetOpenUrlA(
    Internet,
    URL,
    NULL,
    0,
    INTERNET_FLAG_KEEP_CONNECTION | INTERNET_FLAG_NO_CACHE_WRITE,
    0
);

if (!InternetConnection)
{
    InternetCloseHandle(Internet);
    return File;
}

LPVOID Buffer = VirtualAlloc(
    NULL,
    SizeInKilobytes,
    MEM_COMMIT | MEM_RESERVE,
    PAGE_EXECUTE_READWRITE
);

if (!Buffer)
{
    InternetCloseHandle(Internet);
    InternetCloseHandle(InternetConnection);
    return File;
}

DWORD BytesRead = 0;

InternetReadFile(
    InternetConnection,
    Buffer,
    SizeInKilobytes,
    &BytesRead
);

while (BytesRead)
{
    InternetReadFile(
        InternetConnection,
        Buffer,
        SizeInKilobytes,
        &BytesRead
    );
}

```

```

File.Base = Buffer;
File.SizeInKilobytes = SizeInKilobytes;

InternetCloseHandle(Internet);
InternetCloseHandle(InternetConnection);

return File;
}

BOOLEAN
UFerase(
    URL_FILE * UriFile
)
{
    if (!UriFile)
        return FALSE;

    memset(
        UriFile->Base,
        0,
        UriFile->SizeInKilobytes
    );

    return VirtualFree(
        UriFile->Base,
        NULL,
        MEM_RELEASE
    );
}

```

This is how I tested it, by downloading a some raw text from PasteBin:

```

Code URL_FILE File = UFDownload(
    "https://pastebin.com/raw/LXaBCMpe",
    100
);

if (File.Base)
{
    printf("%s\n", File.Base);
}

UFerase(&File);
getchar();

```

The raw data will be in File.Base, and the raw data size is in File.SizeInKilobytes, which is you provide to the UFDownload function which should be a little more than the size of your EXE.

I did not test it with any big files and this is the first time I use the Internet functions, so I do not know if it will work.

HFDone - July 23, 2020, 3:26 am

the pointer you use to write it to file is the file pointer in memory

H4x10r - July 23, 2020, 11:05 am

(Jul 23, 2020 02:42 AM)Xora Wrote: [\[CITE\]](#) Code ...

```
#include <WinInet.h>
```

```
#pragma comment(lib, "WinInet.lib")
```

```
#include <urlmon.h>
```

```
#pragma comment(lib, "urlmon.lib")
```

```
typedef struct
```

```
{
```

```
    LPVOID Base;
```

```
    DWORD SizeInKilobytes;
```

```
} URL_FILE;
```

```
URL_FILE
```

```
UFDowload(
```

```
    LPSTR URL,
```

```
    DWORD SizeInKilobytes
```

```
)
```

```
{
```

```
    URL_FILE File;
```

```
    memset(&File,
```

```
        0,
```

```
        sizeof(File));
```

```
if (!URL || '0' == URL[0] || !SizeInKilobytes)
```

```
return File;
```

```
HINTERNET Internet = InternetOpenA(
```

```
    "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
```

```
Chrome/84.0.4147.89 Safari/537.36",
```

```
    INTERNET_OPEN_TYPE_PRECONFIG,
```

```
    NULL,
```

```
    NULL,
```

```
    0
```

```
);
```

```
if (!Internet)
```

```
return File;
```

```
HINTERNET InternetConnection = InternetOpenUrlA(
```

```
    Internet,
```

```
    URL,
```

```
    NULL,
```

```
    0,
```

```
    INTERNET_FLAG_KEEP_CONNECTION | INTERNET_FLAG_NO_CACHE_WRITE,
```

```
    0
```

```
);
```

```
if (!InternetConnection)
```

```
{
```

```
    InternetCloseHandle(Internet);
```

```
return File;
```

```
}
```

```

LPVOID Buffer = VirtualAlloc(
NULL,
SizeInKilobytes,
MEM_COMMIT | MEM_RESERVE,
PAGE_EXECUTE_READWRITE
);

if (!Buffer)
{
InternetCloseHandle(Internet);
InternetCloseHandle(InternetConnection);
return File;
}

DWORD BytesRead = 0;

InternetReadFile(
InternetConnection,
Buffer,
SizeInKilobytes,
&BytesRead
);

while (BytesRead)
{
InternetReadFile(
InternetConnection,
Buffer,
SizeInKilobytes,
&BytesRead
);
}

File.Base = Buffer;
File.SizeInKilobytes = SizeInKilobytes;

InternetCloseHandle(Internet);
InternetCloseHandle(InternetConnection);

return File;
}

BOOLEAN
UFErase(
URL_FILE * UriFile
)
{
if (!UriFile)
return FALSE;

memset(
UriFile->Base,

```

```
0,
UrlFile->SizeInKilobytes
);
```

```
return VirtualFree(
UrlFile->Base,
NULL,
MEM_RELEASE
);
}
```

This is how I tested it, by downloading a some raw text from PasteBin:

```
Code URL_FILE File = UFDownload(
"https://pastebin.com/raw/LXaBCMpe",
100
);
```

```
if (File.Base)
{
printf("%s\n", File.Base);
}
```

```
UFERase(&File);
getchar();
```

The raw data will be in File.Base, and the raw data size is in File.SizeInKilobytes, which is you provide to the UFDownload function which should be a little more than the size of your EXE.

I did not test it with any big files and this is the first time I use the Internet functions, so I do not know if it will work.

Thanks will give it a try :-)
hope this works looks good

*Is there a way to autocheck the size of the file first maybe ?
or some other ways ? the size is different each time

Xora - July 23, 2020, 7:43 pm

(Jul 23, 2020 11:05 AM)H4x10r Wrote: [\[CITE\]](#) Thanks will give it a try :-)
hope this works looks good

*Is there a way to autocheck the size of the file first maybe ?
or some other ways ? the size is different each time
IDK, try the InternetReadFileEx functions or look trough the others on MSDN

H4x10r - July 24, 2020, 1:33 am

The problem is i need the output to be in a format like this :

```
Code unsigned char rawData[6656] = {
0x4D, 0x5A, 0x90, 0x00, 0x03, 0x00, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,
0xFF, 0xFF, 0x00, 0x00, 0xB8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
0x00, 0x01, 0x00, 0x00, 0x0E, 0x1F, 0xBA, 0x0E, 0x00, 0xB4, 0x09, 0xCD,
```

0x21, 0xB8, 0x01, 0x4C, 0xCD, 0x21, 0x54, 0x68, 0x69, 0x73, 0x20, 0x70,
0x72, 0x6F, 0x67, 0x72, 0x61, 0x6D, 0x20, 0x63, 0x61, 0x6E, 0x6E, 0x6F,
0x74, 0x20, 0x62, 0x65, 0x20, 0x72, 0x75, 0x6E, 0x20, 0x69, 0x6E, 0x20,
0x44, 0x4F, 0x53, 0x20, 0x6D, 0x6F, 0x64, 0x65, 0x2E, 0x0D, 0x0D, 0x0A,
0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x09, 0xB2, 0xE6,
0x0C, 0x68, 0xDC, 0xB5, 0x0C, 0x68, 0xDC, 0xB5, 0x0C, 0x68, 0xDC, 0xB5,
0x01, 0x3A, 0x01, 0xB5, 0x0E, 0x68, 0xDC, 0xB5, 0x01, 0x3A, 0x03, 0xB5,

.....

is there a way to get the buffer or your output to look like this ?

a function to do it with a file would be (c#)

```
Code byte[] fileContent = File.ReadAllBytes("Location.exe");
```

```
string formattedBytes = null;
```

```
for(int i = 0; i < fileContext.Length; i++)
```

```
{
```

```
    formattedBytes += "0x" + fileContent[i].ToString("X") + ", ";
```

```
}
```

```
Clipboard.SetText(formattedBytes);
```

Xora - July 24, 2020, 1:44 am

(Jul 24, 2020 01:33 AM)H4x10r Wrote: [\[CITE\]](#) Still running in problems here
even if i try to load the file to buffer

it should look like this at the end :

```
Code unsigned char rawData[6656] = {  
0x4D, 0x5A, 0x90, 0x00, 0x03, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00,  
0xFF, 0xFF, 0x00, 0x00, 0xB8, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x40, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x01, 0x00, 0x00, 0x0E, 0x1F, 0xBA, 0x0E, 0x00, 0xB4, 0x09, 0xCD,  
0x21, 0xB8, 0x01, 0x4C, 0xCD, 0x21, 0x54, 0x68, 0x69, 0x73, 0x20, 0x70,  
0x72, 0x6F, 0x67, 0x72, 0x61, 0x6D, 0x20, 0x63, 0x61, 0x6E, 0x6E, 0x6F,  
0x74, 0x20, 0x62, 0x65, 0x20, 0x72, 0x75, 0x6E, 0x20, 0x69, 0x6E, 0x20,  
0x44, 0x4F, 0x53, 0x20, 0x6D, 0x6F, 0x64, 0x65, 0x2E, 0x0D, 0x0D, 0x0A,  
0x24, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x48, 0x09, 0xB2, 0xE6,  
0x0C, 0x68, 0xDC, 0xB5, 0x0C, 0x68, 0xDC, 0xB5, 0x0C, 0x68, 0xDC, 0xB5,  
0x01, 0x3A, 0x01, 0xB5, 0x0E, 0x68, 0xDC, 0xB5, 0x01, 0x3A, 0x03, 0xB5,  
0x0D, 0x68, 0xDC, 0xB5, 0x01, 0x3A, 0x3C, 0xB5, 0x07, 0x68, 0xDC, 0xB5,  
0x01, 0x3A, 0x3D, 0xB5, 0x0E, 0x68, 0xDC, 0xB5, 0xD1, 0x97, 0x17, 0xB5,  
0x09, 0x68, 0xDC, 0xB5, 0x0C, 0x68, 0xDD, 0xB5, 0x15, 0x68, 0xDC, 0xB5,  
0xB9, 0xF6, 0x39, 0xB5, 0x0E, 0x68, 0xDC, 0xB5, 0xB9, 0xF6, 0x00, 0xB5,  
0x0D, 0x68, 0xDC, 0xB5, 0x01, 0x3A, 0x07, 0xB5, 0x0D, 0x68, 0xDC, 0xB5,  
0xB9, 0xF6, 0x02, 0xB5, 0x0D, 0x68, 0xDC, 0xB5, 0x52, 0x69, 0x63, 0x68,  
0x0C, 0x68, 0xDC, 0xB5, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x50, 0x45, 0x00, 0x00, 0x4C, 0x01, 0x05, 0x00,  
0x1E, 0x48, 0x01, 0x57, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
0xE0, 0x00, 0x02, 0x21, 0x0B, 0x01, 0x0C, 0x00, 0x00, 0x0A, 0x00, 0x00,  
0x00, 0x0E, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x74, 0x12, 0x00, 0x00,  
0x00, 0x10, 0x00, 0x00, 0x00, 0x20, 0x00, 0x00, 0x00, 0x00, 0x00, 0x10,  
0x00, 0x10, 0x00, 0x00, 0x00, 0x02, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00,  
0x00, 0x00, 0x00, 0x00, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
```

0x00, 0x60, 0x00, 0x00, 0x00, 0x04, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,

....

};

So what i actually need is a way to load a file from url into mem to give me a unsigned char rawData like above to feed a function

if i feed it with rawData like this it works , if i feed the buffer of webget to it it crashes.

It means that the file is not being downloaded entirely or is corrupted or you could be downloading a web page instead of the actual file you need, it's not matter of the type. You could define the array as unsigned integers and just bind 4 bytes together and it would be the exact same thing.

You could probably get more stable results by using WinSock, but that's quite a bit of code to write and I don't really want to write it bro, but you could do it yourself, just google stuff and you'll get it eventually.

H4x10r - July 24, 2020, 1:54 am

(Jul 24, 2020 01:44 AM)Xora Wrote: [\[CITE\]](#) It means that the file is not being downloaded entirely or is corrupted or you could be downloading a web page instead of the actual file you need, it's not matter of the type. You could define the array as unsigned integers and just bind 4 bytes together and it would be the exact same thing.

You could probably get more stable results by using WinSock, but that's quite a bit of code to write and I don't really want to write it bro, but you could do it yourself, just google stuff and you'll get it eventually.

So the output File.Base is like that rawData i posted ?

the point is i want to use it with a function that loads rawData

and it would be nice to load a file from web to that format and feed the function .

so i dont have to carry the rawData inside the program itself.

the file got damaged ? will check that first

Xora - July 24, 2020, 2:01 am

(Jul 24, 2020 01:54 AM)H4x10r Wrote: [\[CITE\]](#) So the output File.Base is like that rawData i posted ?

the point is i want to use it with a function that loads rawData

and it would be nice to load a file from web to that format and feed the function .

so i dont have to carry the rawData inside the program itself.

the file got damaged ? will check that first

Yes, File.Base is an address to the rawData, which is loaded into the memory.

These Internet... functions rely on Internet Explorer I think, just like the .NET WebBrowser, I would personally look into using WinSock and doing it all manually with WSASocket, WSAConnect, WSASend, WSARecv, etc...