Write bytes to file

Asked 10 years, 1 month ago Active 2 years ago Viewed 277k times

91

I have a hexadecimal string (e.g `0CFE9E69271557822FE715A8B3E564BE`) and I want to write it to a file as bytes. For example,

```
Offset      0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
00000000   0C FE 9E 69 27 15 57 82  2F E7 15 A8 B3 E5 64 BE   .þži'.W,/ç.¨³åd¾
```

18

How can I accomplish this using .NET and C#?

c# string byte hex

Share Follow

edited Jun 18 '11 at 20:41
user195488

asked Jun 18 '11 at 16:08

John Doe
8,389 12 38 56

1 Possibly a duplicate of stackoverflow.com/questions/311165/... – Steven Mastandrea Jun 18 '11 at 16:10

1 @Steven: Only partial. Not the most important part. – John Doe Jun 19 '11 at 16:48

1 Possible duplicate of Can a Byte[] Array be written to a file in C#? (also maybe only a partial duplicate). – Jeff B Mar 31 '17 at 13:33

5 Answers

Active | Oldest | Votes

167

If I understand you correctly, this should do the trick. You'll need add `using System.IO` at the top of your file if you don't already have it.

```
public bool ByteArrayToFile(string fileName, byte[] byteArray)
{
    try
    {
        using (var fs = new FileStream(fileName, FileMode.Create, FileAccess.Write))
        {
            fs.Write(byteArray, 0, byteArray.Length);
            return true;
        }
    }
    catch (Exception ex)
    {
        Console.WriteLine("Exception caught in process: {0}", ex);
        return false;
    }
}
```

7,633 15 55 128 user195488

77

The simplest way would be to convert your hexadecimal string to a byte array and use the `File.WriteAllBytes` method.

Using the `StringToByteArray()` method from this question, you'd do something like this:

```
string hexString = "0CFE9E69271557822FE715A8B3E564BE";

File.WriteAllBytes("output.dat", StringToByteArray(hexString));
```

The `StringToByteArray` method is included below:

```
public static byte[] StringToByteArray(string hex) {
    return Enumerable.Range(0, hex.Length)
                     .Where(x => x % 2 == 0)
                     .Select(x => Convert.ToByte(hex.Substring(x, 2), 16))
                     .ToArray();
}
```

Share Follow

edited May 23 '17 at 11:47
Community ♦
1 1

answered Jun 18 '11 at 16:10
Donut
97.4k 18 124 140

Thx, this works fine. How can i append bytes to the same file ? (after the first 'string') – John Doe Jun 18 '11 at 16:23

1 @Robertico: You add a boolean value of true to the third parameter of WriteAllBytes. Have you discovered MSDN yet? This is the first google link when searching for WriteAllBytes append. – user195488 Jun 18 '11 at 20:43

1 I received an error adding the boolean value 'No overload for method 'WriteAllBytes' takes '3' arguments'. MSDN describes: 'However, if you are adding data to a file using a loop, a BinaryWriter object can provide better performance because you only have to open and close the file once.' I'm using a loop. I use the example from @0A0D and changed 'FileMode.Create' to 'FileMode.Append'. – John Doe Jun 19 '11 at 16:37

3

Try this:

```
private byte[] Hex2Bin(string hex)
{
 if ((hex == null) || (hex.Length < 1)) {
  return new byte[0];
 }
 int num = hex.Length / 2;
 byte[] buffer = new byte[num];
 num *= 2;
 for (int i = 0; i < num; i++) {
```

```
 return buffer;
}

private string Bin2Hex(byte[] binary)
{
 StringBuilder builder = new StringBuilder();
 foreach(byte num in binary) {
  if (num > 15) {
   builder.AppendFormat("{0:X}", num);
  } else {
   builder.AppendFormat("0{0:X}", num); /////// 大于 15 就多加个 0
  }
 }
 return builder.ToString();
}
```

Share Follow

edited May 29 '19 at 10:06

Prashant Pimpale
8,947 7 31 70

answered Jun 18 '11 at 16:14

xling
252 1 9

Thx, this also works fine. How can i append bytes to the same file ? (after the first 'string') – John Doe Jun 18 '11 at 16:37

2

You convert the hex string to a byte array.

```
public static byte[] StringToByteArray(string hex) {
return Enumerable.Range(0, hex.Length)
                 .Where(x => x % 2 == 0)
                 .Select(x => Convert.ToByte(hex.Substring(x, 2), 16))
                 .ToArray();
}
```

Credit: Jared Par

And then use WriteAllBytes to write to the file system.

Share Follow

edited May 23 '17 at 11:54

Community ♦
1 1

answered Jun 18 '11 at 16:13

Khepri
9,307 5 42 60

1 If you're referencing an existing Stack Overflow answer as the answer to this question, then it's a pretty safe bet that this is a duplicate question and should be flagged as such. – ChrisF ♦ Jun 18 '11 at 16:14

1 In this case it only answered part of his question, so I felt it didn't need to be marked as a dupe. He'd only get halfway there with that knowledge. – Khepri Jun 18 '11 at 16:16

0

This example reads 6 bytes into a byte array and writes it to another byte array. It does an XOR operation with the bytes so that the result written to the file is the same as the original starting values. The file is always 6 bytes in size, since it writes at position 0.

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main()
        {
        byte[] b1 = { 1, 2, 4, 8, 16, 32 };
        byte[] b2 = new byte[6];
        byte[] b3 = new byte[6];
        byte[] b4 = new byte[6];

        FileStream f1;
        f1 = new FileStream("test.txt", FileMode.Create, FileAccess.Write);

        // write the byte array into a new file
        f1.Write(b1, 0, 6);
        f1.Close();

        // read the byte array
        f1 = new FileStream("test.txt", FileMode.Open, FileAccess.Read);

        f1.Read(b2, 0, 6);
        f1.Close();

        // make changes to the byte array
        for (int i = 1; i < b2.Length; i++)
        {
            b2[i] = (byte)(b2[i] ^ (byte)10); //xor 10
        }

        f1 = new FileStream("test.txt", FileMode.Open, FileAccess.Write);
        // write the new byte array into the file
        f1.Write(b2, 0, 6);
        f1.Close();

        f1 = new FileStream("test.txt", FileMode.Open, FileAccess.Read);

        // read the byte array
        f1.Read(b3, 0, 6);
        f1.Close();

        // make changes to the byte array
        for (int i = 1; i < b3.Length; i++)
        {
            b4[i] = (byte)(b3[i] ^ (byte)10); //xor 10
        }

        f1 = new FileStream("test.txt", FileMode.Open, FileAccess.Write);

        // b4 will have the same values as b1
        f1.Write(b4, 0, 6);
        f1.Close();
        }
    }
}
```

Share Follow

edited Jul 15 '19 at 20:42

answered Jul 15 '19 at 15:40

live-love

35.4k 17 170 158