

Forum:[Hack Forums](#)

Thread: [Split file](#)

Raymond - March 11, 2017, 5:07 am

Hello. Tell me how you can divide the file into parts. I think that you need to read all the bytes, divide them into parts and encrypt each part. Tell me how it can be done. If there is an example, I will be grateful. Thank you for the help.

severos - March 11, 2017, 6:20 am

- 1-open input file
- 2-read part
- 3-write to output file
- 4-if input file not done: goto 2

ShadowCloudGH - March 11, 2017, 6:32 am

You are essentially going to want to do something like this

(from <http://caioproiete.net/en/split-text-file-into-several-files-using-csharp/>)

Code using (var sourceFile = new StreamReader(sourceFileName))

```
{
    var fileCounter = 0;
    var destinationFile = new StreamWriter(
        string.Format(destinationFileName, fileCounter + 1));

    try
    {
        var lineCounter = 0;

        string line;
        while ((line = sourceFile.ReadLine()) != null)
        {
            // Did we reach the maximum number of lines for the file?
            if (lineCounter >= linesPerFile)
            {
                // Yep... Time to close this one and
                // switch to another file
                lineCounter = 0;
                fileCounter++;

                destinationFile.Dispose();
                destinationFile = new StreamWriter(
                    string.Format(destinationFileName, fileCounter + 1));
            }

            destinationFile.WriteLine(DoYourEncryptionHere(line));
            lineCounter++;
        }
    }
    finally
    {
```

```
destinationFile.Dispose();  
}  
}
```

You can either do it as per max lines like above or tweak this so it does equal parts, kind of like most decent compression utilities do it.

Raymond - March 11, 2017, 9:27 am

(Mar 11, 2017 08:04 AM)Piramyd Wrote: [\[CITE\]](#) I can already tell that you're trying to make a crypter just from reading your post. Anyways, I have my own crypter and I wrote code to do this. Essentially what you're going to want to do is:

- 1) ReadAllBytes of the payload
- 2) Declare an integer and a list. Set the integer to 0. The list will hold all the split segments of the file
- 3) Make a for loop, that loops the amount of times that you want the file to be split
- 4) Inside the loop, use LINQ's 'Skip' to skip over that integer we created in step 2. Why? Because that integer is going to hold the amount of bytes already split. Then, use LINQ's 'Take' to get the total amount of bytes in the payload(the length) divided by the amount of times you want the file split.
- 5) Add those bytes to the list we created. Also, add the count of the bytes to the integer we created in step 2. This will assure that we don't keep splitting the first part of the file repeatedly, since we're going to skip over the integer as described in the previous step
- 6) Just do a for each and encrypt all the files in the list.
Could you share an example? I have a problem with division into parts (

Zilog80 - March 11, 2017, 11:03 am

Step #1:

<https://www.nuget.org/packages/DotNetZip/>

Step #2:

PHP Code

```
<?php  
using (ZipFile zip = new ZipFile())  
{  
    zip.AddFile(put your file Path here);  
    zip.MaxOutputSegmentSize = 100*1024 ; // 100k segments  
    zip.CompressionLevel = Ionic.Zlib.CompressionLevel.None; // no compression  
    zip.Save("test.zip");  
}
```

Result:

```
test.zip  
test.z01  
test.z02  
test.z03  
...
```

Profit

Raymond - March 11, 2017, 11:18 am

(Mar 11, 2017 12:03 PM)Zilog80 Wrote: [\[CITE\]](#) Step #1:
<https://www.nuget.org/packages/DotNetZip/>

Step #2:

PHP Code

```
<?php
using (ZipFile zip = new ZipFile())
{
    zip.AddFile(put your file Path here);
    zip.MaxOutputSegmentSize = 100*1024 ; // 100k segments
    zip.CompressionLevel = Ionic.Zlib.CompressionLevel.None; // no compression
    zip.Save("test.zip");
}
```

Result:

test.zip
test.z01
test.z02
test.z03
...

Profit

I need to get byte for further encryption.

Zilog80 - March 11, 2017, 11:27 am

(Mar 11, 2017 12:18 PM)jepko Wrote: [\[CITE\]](#) I need to get byte for further encryption.

Post title: Split file.

1st sentence: Hello. Tell me how you can divide the file into parts.

The junk snippet I provided you on above reply is what post title said and what 1st sentence said.

Now, you are changing the premisses. Be more clear, my friend.

severos - March 11, 2017, 11:31 am

(Mar 11, 2017 12:18 PM)jepko Wrote: [\[CITE\]](#) I need to get byte for further encryption.

read the files again and encrypt them? or you can use a zip stream (single stream) but you'll have to manually split the encrypted stream.

Raymond - March 11, 2017, 11:33 am

(Mar 11, 2017 12:31 PM)severos Wrote: [\[CITE\]](#) read the files again and encrypt them? or you can use a zip stream (single stream) but you'll have to manually split the encrypted stream.

I think first of all to separate bute and then to encrypt them

Raymond - March 12, 2017, 4:09 am

(Mar 11, 2017 04:24 PM)Piramyd Wrote: [\[CITE\]](#) Show me what you've written so far. But for some reason when I collect a file it turns out smaller.

PHP Code

```

<?php
class SplitFile
{
    public IEnumerable<byte[]> Split(byte[] value, int bufferLength)
    {
        int countOfArray = value.Length / bufferLength;
        if (value.Length % bufferLength > 0)
            countOfArray++;
        for (int i = 0; i < countOfArray; i++)
        {
            yield return value.Skip(i * bufferLength).Take(bufferLength).ToArray();
        }
    }
}

public void FileSplit(string path, string outputPath, ref string rndName, int splitParts)
{
    string nameRes = String.Empty;
    byte[] tr = File.ReadAllBytes(path);
    if (tr.Length >= splitParts)
    {
        var temp = Split(tr, tr.Length / splitParts).ToArray();
        for (int i = 0; i < temp.Length; i++)
        {
            nameRes = RandomString(12);
            File.WriteAllBytes(outputPath + "\\" + nameRes, temp[i]);
            rndName += nameRes + "|";
        }
    }
    else
    {
        MessageBox.Show("Very small size file");
    }
}

private readonly Random _rng = new Random();
private const string _chars =
"ABCDIFGHEJKLOMPNRSTQWZadsadioaigysdLOMPNshdnuwqeuyigysdLOMPNRSTQWZadsa";

private string RandomString(int size)
{
    char[] buffer = new char[size];

    for (int i = 0; i < size; i++)
    {
        buffer[i] = _chars[_rng.Next(_chars.Length)];
    }
    return new string(buffer);
}
}

```

It turns out the code works. The problem is that when I collect back, the size is less and the file does not work.

Raymond - *March 12, 2017, 10:41 am*

The problem was that not all bytes were written down.) I understood. Thanks to all.