

Forum: [Hack Forums](#)

Thread: [Compress a Directory in net 4.0? No DLL's?](#)

Pim Fortuyn - December 11, 2016, 11:38 am

How would i compress a directory into a .zip or .gz without any external dll's?

I can do it with with dotnetzip but then you have an external dll which i would like to avoid.

I know in net 4.5 you can use this:

[https://msdn.microsoft.com/en-us/library/system.io.compression.zipfile\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.io.compression.zipfile(v=vs.110).aspx)

But I need a solution for 4.0

Zilog80 - December 11, 2016, 11:52 am

DotNetZip:

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

+

Costura Fody:

<https://www.nuget.org/packages/Costura.Fody/>

=

Standalone .EXE file (even in 4.0)

webdev77 - December 11, 2016, 11:55 am

You could implement LZMA compression via 7-zip's SDK for C#. <http://7-zip.org/sdk.html>

Pim Fortuyn - December 11, 2016, 12:05 pm

(Dec 11, 2016 12:52 PM)Zilog80 Wrote: [\[CITE\]](#) DotNetZip:

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

+

Costura Fody:

<https://www.nuget.org/packages/Costura.Fody/>

=

Standalone .EXE file (even in 4.0)

Doesn't seem to work when building the solution using xbuild (linux)

mrkoolaid - December 11, 2016, 12:08 pm

.NET 4 you probably have to use gzipstream if you dont want third party libs

Pim Fortuyn - December 11, 2016, 12:14 pm

(Dec 11, 2016 01:08 PM)mrkoolaid Wrote: [\[CITE\]](#) .NET 4 you probably have to use gzipstream if you dont want third party libs

It can only compress single files. I would like to compress a folder/directory.

(Dec 11, 2016 12:55 PM)webdev77 Wrote: [\[CITE\]](#) You could implement LZMA compression via 7-zip's SDK for C#. <http://7-zip.org/sdk.html>

Are there any easier alternatives? seems quite complicated ?

Actually I found this which seems to work nicely:

```
Code public static void zipFolder(string inputFolder, string outputZip)
{
    using (Package Zip = Package.Open(outputZip, FileMode.Create))
    {
        foreach (string F in Directory.GetFiles(inputFolder, "*", SearchOption.AllDirectories))
        {
            string F1 = F.Replace(' ', '_');
            Uri URI = new Uri(F1.Substring(inputFolder.Length), UriKind.Relative);
            PackagePart ZipPart = Zip.CreatePart(PackUriHelper.CreatePartUri(URI),
MediaTypeNameNames.Application.Zip, CompressionOption.Maximum);
            byte[] B = File.ReadAllBytes(F);
            ZipPart.GetStream().Write(B, 0, B.Length);
        }
    }
}
```