

عنوان: فشرده سازی فایل ها در .NET 4.5

نویسنده: مجتبی کاویانی

تاریخ: ۱۳:۵۰ ۱۳۹۱/۱۰/۰۶

آدرس: www.dotnettips.info

برچسب‌ها: Net Framework 4.5, Compression, Zip.

با اضافه شده فضای نام System.IO.Compression در .NET 4.5 دیگر بدون نیاز به کتابخانه‌های همچون [DotNetZip](#) به راحتی می‌توانید فایل‌های خود را فشرده یا باز کنید.

کلاس ZipFile

این کلاس امکان فشرده یا باز نمون فایل یا یک پوشه را در اختیارمان قرار میدهد. مثلا برای فشرده سازی یک پوشه از کد زیر استفاده می‌نمایید

```
string startPath = @"c:\example\start";
string zipPath = @"c:\example\result.zip";
ZipFile.CreateFromDirectory(startPath, zipPath);
```

برای بار نمون یک فایل فشرده هم از تابع ExtractToDirectory استفاده نمایید

```
string extractPath = @"c:\example\extract";
ZipFile.ExtractToDirectory(zipPath, extractPath);
```

کلاس ZipArchive

کلاس ZipFile و ZipArchive مکمل هم دیگر هستند و اکثرا با هم دیگر کاربرد دارد اما این کلاس برای دستکاری فایل Zip استفاده می‌شود مثلا برای ایجاد یک فایل zip و کنترل بیشتر بر روی آن در مثال زیر یک ابتدا یک فایل خالی ایجاد کرده ایم و با تابع CreateEntityFromFile فایل‌های مورد را با مسیر و نام آن و حتی کیفیت فشرده‌گی به آن اضافه نموده اید.

```
using (ZipArchive zipFile = ZipFile.Open(zipName, ZipArchiveMode.Create))
{
    zipFile.CreateEntryFromFile(@"C:\Temp\File1.txt", "File1.txt");
    zipFile.CreateEntryFromFile(@"C:\Temp\File2.txt", "File2.txt", CompressionLevel.Fastest);
}
```

اما اگر بخواهیم فایل Zip را در یک MemoryStream ایجاد کنیم کافیت از کد زیر استفاده نمایید

```
using (MemoryStream zipStream = new MemoryStream())
{
    using (ZipArchive zipFile = new ZipArchive(zipStream, ZipArchiveMode.Create))
    {
        zipFile.CreateEntryFromFile(filepath, filename);
    }
}
```

حال می‌خواهیم یک فایل Zip را باز کنیم. کلاس ZipArchiveEntry برای دسترسی به فایل‌های موجود در فایل Zip می‌باشد.

```
using (ZipArchive archive = ZipFile.OpenRead(zipName))
{
    foreach (ZipArchiveEntry file in archive.Entries)
    {
        Console.WriteLine("File Name: {0}", file.Name);
        Console.WriteLine("File Size: {0} bytes", file.Length);
        Console.WriteLine("Compression Ratio: {0}", ((double)file.CompressedLength /
file.Length).ToString("0.0%"));
        file.ExtractToFile(directoryPath);
    }
}
```

هرچند از دات نت 4 و نیم به بعد، الگوریتم Zip [به صورت توکار](#) پشتیبانی می‌شود، اما برای نگارش‌های پایین‌تر، کتابخانه [DotNetZip](#) جزو پرکاربردترین‌ها در این زمینه است.

از همین کتابخانه مدتی در یک سرور معمولی بدون مشکل استفاده کرده بودم تا اینکه پس از ارتقاء به سرور جدید با خراب بودن فایل‌های Zip حاصل مواجه شدم. پس از بررسی [مشخص شد](#) که این کتابخانه با CPUهای چند هسته‌ای مشکل دارد و باید این نوع پردازش موازی را در آن خاموش کرد:

```
using (var zf = new ZipFile())
{
    zf.UseUnicodeAsNecessary = true;
    zf.ParallelDeflateThreshold = -1;
    zf.Comment = "...";
    zf.AddFile(path, "programName");
    zf.Save(whereToSave);
}
```

همانطور که مشاهده می‌کنید، برای غیرفعال سازی پردازش موازی، فقط کافی است پارامتر ParallelDeflateThreshold با منهای یک مقدار دهی شود.

همچنین اگر نام فایل‌های شما فارسی است نیاز است UseUnicodeAsNecessary نیز به true تنظیم گردد.

در کل گاهی از اوقات همین نکات کوتاه می‌توانند از صدمات جدی در آینده جلوگیری کنند. خصوصا اگر از این محصول جهت تهیه پشتیبان استفاده شده باشد.

نظرات خوانندگان

نویسنده: رحمت اله رضایی
تاریخ: ۱۳۹۱/۱۲/۲۴ ۱۰:۴۶

موردی که برای من جای سوال دارد این است که چرا بعضی از امکاناتی که معمولا مورد نیاز برنامه نویس هاست، مثل همین zip، اینقدر دیر به صورت توکار در دات نت قرار میگیرد؟ چرا در نسخه‌های قبلی نبود؟ یا اصلا با توجه به فشردگی بالای [7zip](#) چرا به آن توجه نمیشود؟ الان روی اکثر کامپیوترها Winrar نصب است ولی کمتر کسی با 7zip آشناست در حالی که معمولا حجم فابل فشرده شده با آن نصف Winrar است.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۱۲/۲۴ ۱۰:۵۵

گاهی از اوقات مسایل مجوز استفاده از این الگوریتم‌ها مانع هستند تا مسایل فنی.