A Little Program to fix one particular type of mojibake

devblogs.microsoft.com/oldnewthing/20161010-00

October 10, 2016

Raymond Chen

Has this ever happened to you? You're downloading your daughter's Chinese homework assignment, but the file name gets all up in your mojibake, and the results are nonsense.

Time to do some reverse-mojibake.

The first step in reversing mojibake is figuring out what wrong turn the encoding went through. I took an educated guess and assumed that the file name was encoded in UTF-8, which was then misinterpreted as ANSI. I suspect this type of error is pretty common, so it was my first stab.

To reverse it, therefore, we need to take the Unicode file name, convert it to ANSI bytes, then reinterpret those bytes as UTF-8. Let's try it:

```
using System.Text;

class Program
{
  static public void Main(string[] args)
  {
    foreach (var file in args)
    {
      var bytes = Encoding.Default.GetBytes(file);
      var s = Encoding.UTF8.GetString(bytes);
      System.IO.File.Move(file, s);
    }
  }
}
```

I'll take the file name on the command line, convert it via the default system code page into bytes, then take those bytes and convert them back into a string by reinterpret them as UTF-8. I then rename the file with the "fixed" name.

Fortunately, this worked. The file name got unscrambled.

U+00E5	U+00AE	U+00B6	U+00E5	U+00BA	U+00AD	U+00E8	U+0081	U+00
å	®	¶	å	º		è	ˆ	‒

Converted to bytes via code page 1252 Windows Western European Latin 1 (which is the default code page for the United States):

E5	AE	B6	E5	BA	AD	E8	81	AF	E7	B5	A1	E5	96	AE	2E	70	64

And then converted back to Unicode via UTF-8:

U+5BB6	U+5EAD	U+806F	U+7D61	U+55AE	U+002E	U+0070	U+0064	U+00
家	庭	聯	絡	單	.	p	d	f

Et voilà.

Raymond Chen

Follow