

Re: Data Records from Flat File (COBOL Style)

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-03/0506.html>

From: Bruce Wood (brucewood_at_canada.com)

Date: 03/01/05

Date: 1 Mar 2005 12:00:17 -0800

There is a speed penalty.

The COBOL PIC clause absolutely does not have an internal length. It's just a sequence of bytes, that the `_compiler_` knows happens to be 10 long, or whatever. It does `_not_` contain any attached "true length".

That is, if you declare

```
ABC PIC X(10)
```

in COBOL, there is `_no way_` to indicate that you stored only 5 characters in there.

A call to `Substring(5,10)`, or something like that, copies the ten characters to another place in memory and adds a length (in this case, 10) to form a string object. Picking the fields out of a larger string one-by-one involves copying each field in memory and constructing a string for it, which costs time and memory. My point was that it doesn't cost much in the grand scheme of things.

In more detail, here is what happens in COBOL and C# in situations like this one.

COBOL:

```
ABC PIC X(10).
```

Any reference to "ABC" simply points to the start of the ten characters. The compiler may or may not generate instructions to make sure that you don't run off the end of the 10 characters, depending on compiler switches you specify. You can manipulate the characters in place, without copying them anywhere.

C#

```
string customerNumber = lineString.Substring(5, 10);
```

The Substring method copies ten characters from the lineString and uses them to construct a string elsewhere in memory, with a length of 10 and the characters copied from the lineString. A reference to that string is then stored in customerNumber. Yes, a more intensive operation than COBOL (which didn't need to move anything anywhere). If you're processing a million rows you will notice a difference, but as with many such applications most of your time will be spent doing I/O, so even if the in-memory processing is 10x slower, it still won't make all that much real-time difference.