

Re: searching for the highest index within a directory

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2005-03/1032.html>

From: Nick Malik [Microsoft] (nickmalik_at_hotmail.nospam.com)

Date: 03/03/05

Date: Thu, 3 Mar 2005 07:35:16 -0800

Hi Ada,

(I used to write code in the Ada programming language... ;-)

The reason that I asked about your intent for incrementing file names: if you just needed a unique file name once, that you were going to delete when you are done, and you didn't want to overwrite another application using their own unique file name on the same computer, you could have used `Path.GetTempFileName()` which will return a unique name in the users "Temporary Files" folder. Sounds like that isn't what you are doing.

>
> *Nick, to answer your question....*
> *it's a small exercise for myself. :-)*
> *an application for this type of naming is a sequential graphic animation*
> *files.*
> *i'm doing this to avoid overwriting the existing file and resume sequence*
> *from the highest index file.*
>
> *can you elaborate why*
> *testFile_4*
> *comes after*
> *testFile_34?*

Because "testFile_34" is a string, not a number. If a string cannot be converted to a number, there is no way to compare them as numbers. We compare strings lexically (alphabetical order... kinda... see below). That means we look at the first character in each string. If one is less than the other (occurs earlier in the character table), then we stop because we have found the "earlier" string. If the characters in that position are the same, we move to the next position and compare.

In this case, the 10th character of testFile_4 is '4' while the 10th character of testFile_34 is '3'. Since '3' comes before '4' in the character table, then "testFile_34" comes before "testFile_4". That is why

microsoft.public.dotnet.languages.csharp: Re: searching for the highest index within a directory

I suggested that you would want to embed leading zeros in your number field... In the character table, '0' comes before the rest of the digits, so your alphabetical sort will be the same as a numeric sort (until you run out of digits).

Important Note: in the character tables, All of the upper case characters come before the first lower case character. Therefore, if you use a "case sensitive sort" (the default), then "ZestFile" will come before "testFile" even though in a dictionary, it wouldn't work that way. That's because the capital 'Z' occurs before the lowercase 't'.

If you want to see what the character tables look like, visit

<http://www.unicode.org/charts/>

Note that we are most familiar with the "Basic Latin" character set. Another note: read the charts one column at a time (top to bottom, left to right). I have no idea why they did that :-(.

There is a way around this, of course. You can tell the system to ignore the case of the letters when sorting your strings.

I've included a snippet of text from an article on DevX.com:

<snippet>

The .NET Framework defines several comparer classes for you to use. One of them, the CaseInsensitiveComparer class, allows you to sort strings by ignoring their casing. So in this case, the .NET framework ignores the String class' CompareTo method, and instead uses the rules defined in the CaseInsensitiveComparer class. The following code illustrates this feature:

```
Dim aryLastNames() As String = {"sMiTh, ZULU", "smith, john&", "SMITH, TerrY"}
```

```
Array.Sort(aryLastNames, New CaseInsensitiveComparer)
```

```
' aryLastNames order:
```

```
,
```

```
' smith, john
```

```
' SMITH, TerrY
```

```
' sMiTh, ZULU
```

</snippet>You can find the full article at:

<http://www.devx.com/dotnet/Article/21089I> hope this helps,---

--- Nick Malik [Microsoft]

MCSD, CFPS, Certified Scrummaster

<http://blogs.msdn.com/nickmalik>

Disclaimer: Opinions expressed in this forum are my own, and not representative of my employer.

I do not answer questions on behalf of my employer. I'm just a programmer helping programmers.

--

"Ada" <Ada@discussions.microsoft.com> wrote in message

Re: searching for the highest index within a directory

microsoft.public.dotnet.languages.csharp: Re: searching for the highest index within a directory

news:F94A4607-88C9-40D6-A17F-746ABB63BE70@microsoft.com...

>