CodePlex Archive will be shut down after July 1st, 2021.

CodePlex was Microsoft's free, open source project hosting site, which ran from 2006 through 2017. The site has been in archive mode for 3 years. We now encourage customers to use [Github](#) for their open source project hosting needs.

CodePlex will continue as an archive until next July (2021), at which point it will be shut down. Until then, you can browse published projects, documentation, issues, and discussions which were posted before the site went into archive mode three years ago.

For questions or comments please contact [CodePlex Archive support](#).

CodePlex Archive *Open Source Project Archive*

longpathsupport

LongPathSupport

download archive

This class library project enables .NET applications to work with very long paths > 256 characters. It's used in my project
http://robopowercopy.codeplex.com

"LongPathSupport": Operate with files and directories with "long paths" in .NET

In my project RoboPowerCopy (http://robopowercopy.codeplex.com) I've implemented a basic set of classes and methods for use with files and directories with long paths. NTFS supports paths with up to 32000 characters. But .NET only supports 256 characters. So System.IO.FileInfo and System.IO.DirectoryInfo will not work with such a file:

```
c:\users\ikarstein\appdata\local\11111222223333334444455555566666777778888899999900000\111112222233333444
44555555666667777788888899999900000\
11111222223333334444455555566666777778888899999900000\11111222223333334444455555566666777778888899999900000\11111
77888889999900000\11111222223333334444455555566666777778888899999900000\1111122222333333444445555566666777778888
44444555555666667777788888899999900000\11111222223333334444455555566666777778888899999900000\11111222223333344444!
00\11111222223333334444455555566666777778888899999900000\11111222223333334444455555566666777778888899999900000\hel
```

This path has 655 characters. No way in .NET without using the Win32 API.

I've wrapped the API calls into a set of <u>basic</u> .NET classes to provide a <u>basic</u> support for files and directors with long paths...

Here it is!

I recommand you to use only the "Source" version, not binary. Please use it carefully!

Please feel free to contribute the project or post you experiences on my blog or on Codeplex!

For the project I've read/used some other source in the Internet. Especially this three I'd like to refer to:

- Dave Midgley talking about "Reparse Points in Vista": http://www.codeproject.com/KB/vista/ReparsePointID.aspx
- "NTFS Hard Links, Directory Junctions, and Windows Shortcuts": http://www.flexhex.com/docs/articles/hard-links.phtml

This is the corresponding article on my blog:

http://ikarstein.wordpress.com/2011/08/22/longpathsupport-operate-with-files-and-directories-with-long-paths-in-net-codeplex-project/