

# How do you place a file in recycle bin instead of delete?

Asked 16 years, 4 months ago   Modified 11 years, 7 months ago

Viewed 12k times



Programmatic solution of course...

29

c#

.net

c++

windows

io



Share



Improve this question



Follow

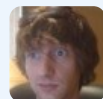
edited Apr 28, 2013 at 3:27



Patrick D'Souza

3,553 ● 2 ● 24 ● 39

asked Aug 20, 2008 at 8:43



Brian Leahy

35.4k ● 12 ● 46 ● 60

- 
- 1   Possible duplicate of [Send a File to the Recycle Bin](#)  
– [MicroVirus](#) Jun 21, 2016 at 10:56
- 

3 Answers

Sorted by:

Highest score (default)



<http://www.daveamenta.com/2008-05/c-delete-a-file-to-the-recycle-bin/>

38

From above:



```
using Microsoft.VisualBasic;
```

```
string path = @"c:\myfile.txt";  
FileIO.FileSystem.DeleteDirectory(path,  
    FileIO.UIOption.OnlyErrorDialogs,  
    RecycleOption.SendToRecycleBin);
```

Share Improve this answer

edited Aug 20, 2008 at 8:51

Follow



[Jeff Atwood](#)

63.9k ● 48 ● 151 ● 153

answered Aug 20, 2008 at 8:48



[TK.](#)

47.8k ● 47 ● 121 ● 148

---

I'd use DeleteFile instead of DeleteDirectory to be more clear.

– [scobi](#) Dec 19, 2008 at 3:59

---

8 +1 for thinking outside the box and referencing a disliked namespace rather than resorting to ugly unmanaged code.

– [BenAlabaster](#) Dec 22, 2008 at 15:26

---

4 What do these FileIO classes and methods have to do with Visual Basic? Putting them in Microsoft.VisualBasic makes absolutely no sense to me. I must be missing something.

– [I. J. Kennedy](#) Feb 21, 2011 at 20:20

---

1 @I. J. Kennedy If I had to guess it might have been that the Visual basic team implanted the feature and rather than muck around with it the .NET team decided to leave the functionality where it was. – [TK.](#) Feb 21, 2011 at 20:29

---

4 any solution without using "Microsoft.VisualBasic" ?

– [Kiquenet](#) Apr 6, 2011 at 7:56

---



You need to delve into unmanaged code. Here's a static class that I've been using:

18



```
public static class Recycle
{
    private const int FO_DELETE = 3;
    private const int FOF_ALLOWUNDO = 0x40;
    private const int FOF_NOCONFIRMATION = 0x0010;

    [StructLayout(LayoutKind.Sequential, CharSet = CharSet.Auto)]
    public struct SHFILEOPSTRUCT
    {
        public IntPtr hwnd;
        [MarshalAs(UnmanagedType.U4)]
        public int wFunc;
        public string pFrom;
        public string pTo;
        public short fFlags;
        [MarshalAs(UnmanagedType.Bool)]
        public bool fAnyOperationsAborted;
        public IntPtr hNameMappings;
        public string lpzProgressTitle;
    }

    [DllImport("shell32.dll", CharSet = CharSet.Auto)]
    static extern int SHFileOperation(ref SHFILEOPSTRUCT fileop);

    public static void DeleteFileOperation(string filePath)
    {
        SHFILEOPSTRUCT fileop = new SHFILEOPSTRUCT();
        fileop.wFunc = FO_DELETE;
        fileop.pFrom = filePath + "\\*.*";
        fileop.fFlags = FOF_ALLOWUNDO | FOF_NOCONFIRMATION;

        SHFileOperation(ref fileop);
    }
}
```

Addendum:

- Tsk tsk @ Jeff for "using Microsoft.VisualBasic" in C# code.
- Tsk tsk @ MS for putting all the goodies in VisualBasic namespace.

Share Improve this answer

edited Aug 20, 2008 at 8:57

Follow

answered Aug 20, 2008 at 8:47



Ishmaeel

14.4k ● 9 ● 73 ● 84

---

12 I guess this is one of the areas where VB is just better than C#...besides, it's not "not C#" just because it's in the VB namespace - an object is an object, not to use it just because it's in a namespace you don't like the name of is a little ridiculous don't you think?... – BenAlabaster Dec 22, 2008 at 15:18

---

10 ...if the namespace was Microsoft.UsefulUtilities you wouldn't have an reservations about using them, so what's the difference? – BenAlabaster Dec 22, 2008 at 15:18

---

3 It's not about the namespace of course, but that you have to link in VB libraries to use that namespace. Should be obvious. – Mike Lischke Apr 28, 2013 at 9:36

---



13



The best way I have found is to use the VB function

```
FileSystem.DeleteFile .
```

```
Microsoft.VisualBasic.FileIO.FileSystem.DeleteFile(fil  
Microsoft.VisualBasic.FileIO.UIOption.OnlyErrorDia  
Microsoft.VisualBasic.FileIO.RecycleOption.SendToR
```

It requires adding `Microsoft.VisualBasic` as a reference, but this is part of the .NET framework and so isn't an extra dependency.

Alternate solutions require a P/Invoke to [SHFileOperation](#), as well as defining all the various structures/constants. Including `Microsoft.VisualBasic` is much neater by comparison.

Share Improve this answer

edited Aug 22, 2008 at 7:00

Follow

answered Aug 20, 2008 at 8:50



Zooba

11.4k ● 3 ● 38 ● 40