

RE: Install font from ClickOnce application

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework/2006-09/msg00549.html>

- *From:* Thomas Taylor <thomas@xxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 21 Sep 2006 07:17:03 -0700
-

Thanks for the reply, Jeffrey. I don't think I can get by with making it a private font since the application can write files for use by other applications (e.g., Excel), and the font will be need by the other applications. So, I really need to to a 'regular' install of the font in Windows.

I got the following prototype code working. Well, at least it appears to work on a clean Win XP machine; it still does not work on a Win 2003 machine (even under an admin account), but maybe that's a problem on that specific machine.

```
internal static void InstallFont()
{
    string fontsPath = GetFontsPath();
    string ttfFile = System.IO.Path.Combine(fontsPath, "MyFont.TTF");
    string fotFile = System.IO.Path.Combine(fontsPath, "MyFont.FOT");
    int ret;

    if (!System.IO.File.Exists(ttfFile))
    {
        //Write file from embedded resource
        System.IO.File.WriteAllBytes(ttfFile, Properties.Resources.MyFont);
        //Write resource file
        ret = CreateScalableFontResource(0, fotFile, ttfFile, String.Empty);
        //Add font resource
        ret = AddFontResource(fotFile);
        //Add registry entry so the font is also available next session
        Registry.SetValue(@"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows
NT\CurrentVersion\Fonts", "My Font (TrueType)", "MyFont.TTF",
Registry.ValueKind.String);
        //Broadcast to let all top-level windows know about change
        ret = SendMessage(HWND_BROADCAST, WM_FONTCHANGE, new IntPtr(0), new
IntPtr(0));
    }
}

// PInvoke to look up fonts path
[System.Runtime.InteropServices.DllImport("shfolder.dll", CharSet =
```

RE: Install font from ClickOnce application

```
System.Runtime.InteropServices.CharSet.Auto)]
private static extern int SHGetFolderPath(IntPtr hwndOwner, int nFolder,
IntPtr hToken, int dwFlags, StringBuilder lpszPath);
private const int CSIDL_FONTS = 0x0014;
private const int MAX_PATH = 260;
private static string GetFontsPath()
{
StringBuilder sb = new StringBuilder(MAX_PATH);
SHGetFolderPath(IntPtr.Zero, CSIDL_FONTS, IntPtr.Zero, 0, sb);
return sb.ToString();
}

// PInvoke to 'register' fonts and broadcast addition
[System.Runtime.InteropServices.DllImport("gdi32.dll")]
private static extern int AddFontResource(string lpszFilename);
[System.Runtime.InteropServices.DllImport("gdi32.dll")]
private static extern int CreateScalableFontResource(uint fdwHidden, string
lpszFontRes, string lpszFontFile, string lpszCurrentPath);
private static IntPtr HWND_BROADCAST = new IntPtr(0xffff);
private const uint WM_FONTCHANGE = 0x001D;
[System.Runtime.InteropServices.DllImport("user32.dll", CharSet =
System.Runtime.InteropServices.CharSet.Auto)]
private static extern int SendMessage(IntPtr hWnd, uint Msg, IntPtr wParam,
IntPtr lParam);
```

So, if anyone can point out something I'm doing wrong here, that would be great.

I'm also still thinking that ClickOnce should support font installs; this seems like a common task. Is there no way to get ClickOnce to do this for me?

.