

## Forum:[Hack Forums](#)

Thread: [Save INI files with ease.](#)

**Visual Freedom** - April 5, 2017, 2:15 am

Hey, here is a fixed INI lib code for your projects. Save INI files with ease + Unicode support unlike everything else you can find. Since there was nothing like this I release it here for you.

For your WinAPI module

Class code or in a helper module

If this is too complex for you, I recommend to learn the basics and to read the parameters. Have fun with this! ?

**Barret** - April 5, 2017, 3:09 am

Its an old method, goes back as far as 10+ years, i suppose its still valid. i can help you extend this library to a more advanced use if you want, you know where to find me (signature)

**Visual Freedom** - April 5, 2017, 3:11 am

I personally use it in most of my projects. The performance is quite good even for a few hundred settings to read or write.

It is still valid and supported on all systems. So a good reason to use this, especially because it is very simple.

Thanks for your offer, but if I would need otherwise I would already have done this.

**Barret** - April 5, 2017, 3:57 am

(Apr 5, 2017 03:11 AM)Visual Freedom Wrote: [\[CITE\]](#) I personally use it in most of my projects. The performance is quite good even for a few hundred settings to read or write.

It is still valid and supported on all systems. So a good reason to use this, especially because it is very simple.

Thanks for your offer, but if I would need otherwise I would already have done this.

Well alright, if you think u wanna up ur game check out

<https://github.com/TheBarret/Xconf>

gl ^^

**Visual Freedom** - April 5, 2017, 4:38 am

One question to your bigger project. Do you handle that language specific systems write numbers with "," or "."?"

**Visual Freedom** - April 8, 2017, 7:53 am

Anyone with other feedback? Especially beginners should benefit from this easy to use code.

**Visual Freedom** - April 16, 2017, 5:50 am

Are there any other people who would like to leave some feedback?

sh@rp - April 16, 2017, 9:39 am

Personally I used this : <https://pastebin.com/91jwdLbL>

Actually I prefer to use Xml serialization.

;)

Ninja - April 16, 2017, 11:33 am

Why do you keep bumping your thread? Are you so proud of your release that you want people to see it?

idb. - April 16, 2017, 12:07 pm

(Apr 8, 2017 07:53 AM)Visual Freedom Wrote: [\[CITE\]](#) Especially beginners should profit from this easy to use code.

Profit? Do you mean benefit?

Visual Freedom - April 16, 2017, 12:55 pm

Yes I mean benefit. I edited my previous post.

log dot zip - April 16, 2017, 1:03 pm

What you did is fill in the parameters to call a prewritten API. Why'd you bump your thread 3 times for that? Am I missing something?

gigajew - April 17, 2017, 9:13 am

All credits to OP, because I had previously not heard of this API method. Here's how I would do it:

```
Code using System;
using System.Runtime.InteropServices;
using System.ComponentModel;
using System.IO;
using System.Text;
```

```
namespace HackForums {
```

```
    public class ConfigFile {
```

```
        [DllImport("kernel32.dll", SetLastError = true)]
```

```
        public static extern IntPtr WritePrivateProfileString(string lpAppName, string lpKeyName, string lpString, string lpFileName);
```

```
        [DllImport("kernel32.dll", SetLastError = true)]
```

```
        public static extern IntPtr GetPrivateProfileString(string lpAppName, string lpKeyName, string lpDefault, StringBuilder lpReturnedString, uint nSize, string lpFileName);
```

```
        public string Filename { get; private set; }
```

```
        public string Name { get; private set; }
```

```
        public ConfigFile(string name, string filename) {
```

```
            this.Name = name;
```

```
            this.Filename = filename;
```

```
        }
```

```

public ConfigFile(Type t) {
    this.Name = t.Name;
    this.Filename = Path.GetFullPath(string.Format("{0}.ini", t.Name));
}

public string GetValue(string key, string @default) {
    StringBuilder sb = new StringBuilder(UInt16.MaxValue);
    IntPtr result = GetPrivateProfileString(this.Name, key, @default, sb, (uint)sb.Capacity,
this.Filename);
    if(result != IntPtr.Zero) {
        return sb.ToString();
    }else {
        throw new Win32Exception(Marshal.GetLastWin32Error());
    }
}

public bool SetValue(string key, string @value) {
    if(@value.Length > UInt16.MaxValue) {
        throw new Exception("Key value too long!");
    }
    IntPtr result = WritePrivateProfileString(this.Name, key, @value, this.Filename);
    if(result != IntPtr.Zero) {
        return true;
    }else {
        throw new Win32Exception(Marshal.GetLastWin32Error());
    }
}
}

```

```

public class Program {

    public static void Main(string[] argv) {

        /*
            Easy constructor
        */
        var config = new ConfigFile(typeof(Program));

        /*
            How to set a value
        */
        config.SetValue("Max turbo setting", "20"); //key name, value

        /*
            How to read a value
        */
        var turbo = config.GetValue("Max turbo setting", "540"); //key name, default value

        /*
            Print it out
        */
        Console.WriteLine("Turbo value = " + turbo);
    }
}

```

}

}