Cannot declare instance members in a static class in C#

Asked 9 years, 3 months ago   Active 2 years, 10 months ago   Viewed 93k times

▲

37

▼

★

2

I have a `public static class` and I am trying to access `appSettings` from my app.config file in C# and I get the error described in the title.

```
public static class employee
{
    NameValueCollection appSetting = ConfigurationManager.AppSettings;
}
```

How do I get this to work?

`c#`

edited Oct 18 '16 at 20:21   asked Jun 17 '10 at 16:13

CDspace   acadia
2,414   8   28   35   **921**   8   31   56

---

2   make NameValueCollection static.. – Srinivas Reddy Thatiparthy Jun 17 '10 at 16:14

---

7   Do you understand what static means? From your question, I don't think you do. You should perhaps re-familiarize yourself with it's definition, especially
      with regards to c# classes. See: msdn.microsoft.com/en-us/library/98f28cdx(VS.71).aspx – i_am_jorf Jun 17 '10 at 16:15

---

5   This has got to be my first time seeing a static `employee` class. – BoltClock ♦ Jun 17 '10 at 16:16

---

1   @BoltClock has a very good point; are you sure you want your employee class to be static? You almost certainly don't want that behaviour. You'd
      probably be better off removing the static constraint from the class and the members. – John Weldon Jun 17 '10 at 16:18

---

11   maybe it's a one man band? – Pharabus Jun 17 '10 at 16:19

---

7 Answers

80

```
static NameValueCollection appSetting = ConfigurationManager.AppSettings;
```

Are you sure you want your employee class to be static? You almost certainly don't want that behaviour. You'd probably be better off removing the static constraint from the class and the members.

edited Oct 18 '16 at 20:24　　　answered Jun 17 '10 at 16:14

Robert Harvey　　　　　John Weldon
153k 38 295 435　　　**32.8k** 9 81 120

1　is there any reason why we have to explicitly declare static class members as static .. if they have to be static why isn't each member assumed to be static, saves precious typing time. – Sherlock Mar 18 '13 at 20:29

2　good question @Sherlock, to (approximately) quote Eric Lippert; it's probably because someone would have had to have prioritized and implemented that behaviour... :) – John Weldon Mar 18 '13 at 21:14

　more probably because if you take a look at JUST the member, there is a fair chance you will be puzzled for a long time why your members behaviour is like a static member, but it is not declared statis. Forcing us to type "static" saves a future contributor some trouble. – Hacky Oct 29 '16 at 23:36 ✏

It says what it means:

6

make your class non-static:

```
public class employee
{
    NameValueCollection appSetting = ConfigurationManager.AppSettings;
}
```

or the member static:

```
public static class employee
{
    static NameValueCollection appSetting = ConfigurationManager.AppSettings;
}
```

answered Jun 17 '10 at 16:15

1   "It says what it means" is not helpful, especially for people new to this language. Nowhere in the error message does it explain WHY an instance member can't be declared in a static class. It simply says it cannot be done with no suggestions on how to solve the error. – dsanchez Jul 5 at 14:42

Beside the first sentence "it says what it means" there are two solutions for the OPs problem in my answer. And *why* it is this way is not clear - see the comments below the accepted answer. – tanascius Jul 8 at 9:09

▲

5

▼

It is not legal to declare an instance member in a static class. Static class's cannot be instantiated hence it makes no sense to have an instance members (they'd never be accessible).

answered Jun 17 '10 at 16:15

JaredPar
606k   127   1111
1369

▲

3

▼

I know this post is old but...

I was able to do this, my problem was that I forgot to make my property static.

```
public static class MyStaticClass
{
    private static NonStaticObject _myObject = new NonStaticObject();

    //property
    public static NonStaticObject MyObject
    {
        get { return _myObject; }
        set { _myObject = value; }
    }
}
```

answered Dec 7 '16 at 18:16

JustSomeGuy
31   1

1

```
public static class employee
{
    static NameValueCollection appSetting = ConfigurationManager.AppSettings;
}
```

As John Weldon said all members must be static in a static class. Try

1

```
public static class employee
{
    static NameValueCollection appSetting = ConfigurationManager.AppSettings;

}
```

1

```
public static class Employee
{
    public static string SomeSetting
    {
        get
        {
            return ConfigurationManager.AppSettings["SomeSetting"];
        }
    }
}
```

Declare the property as static, as well. Also, Don't bother storing a private reference to ConfigurationManager.AppSettings. ConfigurationManager is already a static class.

```csharp
public static class Employee
{
    private static NameValueCollection _appSettings=ConfigurationManager.AppSettings;

    public static NameValueCollection AppSettings { get { return _appSettings; } }

}
```

It's good form to always give an explicit access specifier (private, public, etc) even though the default is private.

answered Jun 17 '10 at 16:16

3Dave
21.9k 14 78 142