

Forum:[Hack Forums](#)

Thread: [Add properties/variable to a class at runtime?](#)

Ajumma - April 2, 2019, 6:38 pm

So say I have the following class I created...

```
Code Public Class House
    Public Property Door As String
    Public Property Roof As String
    Public Property Window As String
End Class
```

What if I wanted to add another property to that Class, like Window2 during runtime, is it possible?

Something like, House.AddProperty("Window2", String)

gigajew - April 2, 2019, 7:56 pm

You'd be better off just making like a Dictionary<string, string> and adding like this myProperties.Add("window", "whatever value"); instead of trying to edit it at runtime.

Ajumma - April 2, 2019, 9:31 pm

(Apr 2, 2019 07:56 PM)gigajew Wrote: [\[CITE\]](#) You'd be better off just making like a Dictionary<string, string> and adding like this myProperties.Add("window", "whatever value"); instead of trying to edit it at runtime.

This is much better! Thank you

itcorp - April 3, 2019, 4:58 am

(Apr 2, 2019 09:31 PM)Ajumma Wrote: [\[CITE\]](#) This is much better! Thank youYou can also achieve this by using custom dynamic types in certain scenarios

<https://docs.microsoft.com/en-us/dotnet/api/system.dynamic.dynamicobject?redirectedfrom=MSDN&view=netframework-4.7.2>

https://docs.microsoft.com/en-us/dotnet/api/system.dynamic.dynamicobject.trygetmember?redirectedfrom=MSDN&view=netframework-4.7.2#System_Dynamic_DynamicObject_TryGetMember_System_Dynamic_GetMemberBinder_System_Object

Or you can use 3rd party pacages

<https://www.nuget.org/packages/Westwind.Utilities/>

Code public class House:Westwind.Utilities.Expando

```
{
    public string Door { get; set; }
    public string Roof { get; set; }
    public string Window { get; set; }

    public House():base()
```

```

    {
    }

    public House(object instance):base(instance)
    {
    }
}
Code class Program
{
    static void Main(string[] args)
    {
        dynamic myHouse = new ExpandoObject();

        myHouse.Door = "1";
        myHouse.Window = "white trim 49x49";

        Console.WriteLine(myHouse.Window);

        var house = new House();

        house.Door = "1";
        house.Roof = "big red";
        house.Window = "white trim 48x48";
        //One way to add
        house["Window1"] = "white trim 49x49";
        //Second way to add
        dynamic add = house;
        add.Windows2 = "white trim 50x50";

        Console.WriteLine(house.Door);
        Console.WriteLine(house.Roof);
        Console.WriteLine(house.Window);
        Console.WriteLine(house["Window1"]);
        Console.WriteLine(add.Windows2);
        Console.ReadLine();

    }
}

```

Ajumma - April 3, 2019, 7:11 am

(Apr 3, 2019 04:58 AM)itcorp Wrote: [\[CITE\]](#) You can also achieve this by using custom dynamic types in certain scenarios

<https://docs.microsoft.com/en-us/dotnet/api/system.dynamic.dynamicobject?redirectedfrom=MSDN&view=netframework-4.7.2>

https://docs.microsoft.com/en-us/dotnet/api/system.dynamic.dynamicobject.trygetmember?redirectedfrom=MSDN&view=netframework-4.7.2#System_Dynamic_DynamicObject_TryGetMember_System_Dynamic_GetMemberBinder_System_Object

Or you can use 3rd party packages

<https://www.nuget.org/packages/Westwind.Utilities/>

Code public class House:Westwind.Utilities.Expando

```

{
    public string Door { get; set; }
    public string Roof { get; set; }
    public string Window { get; set; }

    public House():base()
    {
    }

    public House(object instance):base(instance)
    {
    }
}
Code class Program
{
    static void Main(string[] args)
    {
        dynamic myHouse = new ExpandoObject();

        myHouse.Door = "1";
        myHouse.Window = "white trim 49x49";

        Console.WriteLine(myHouse.Window);

        var house = new House();

        house.Door = "1";
        house.Roof = "big red";
        house.Window = "white trim 48x48";
        //One way to add
        house["Window1"] = "white trim 49x49";
        //Second way to add
        dynamic add = house;
        add.Windows2 = "white trim 50x50";

        Console.WriteLine(house.Door);
        Console.WriteLine(house.Roof);
        Console.WriteLine(house.Window);
        Console.WriteLine(house["Window1"]);
        Console.WriteLine(add.Windows2);
        Console.ReadLine();

    }
}

```

Thanks for the suggestion, I'll save it for future reference but for now, I'd like to use as little 3rd party tools as possible if I can. I forget its there sometimes xD

itcorp - April 3, 2019, 9:51 am

(Apr 3, 2019 07:11 AM)Ajumma Wrote: [CITE](#) Thanks for the suggestion, I'll save it for future reference but for now, I'd like to use as little 3rd party tools as possible if I can. I forget its there sometimes xD

You're welcome, but I think using dynamic is a solution for you. All the best