Is a Collection of Collections possible and/or the best way? C# .Net 3.5

Asked 16 years, 1 month ago Modified 5 years, 4 months ago

Viewed 9k times

5

I have created a class for a dashboard item which will hold information such as placement on the dashboard, description, etc. I am currently using a pair of Collections to hold those dashboard items contained in the "library" and those items showing on the dashboard itself. I have been asked to make this dashboard multi-tab, and my first inclination was to make a new Collection for each tab. For this I would want some type of array or collection which could have many of these dashboard item collections added to it as more tabs are added to the dashboard.

Is this possible, and if so, could I get a little code snip for the declaration of such a collection? I have considered using a single collection with a variable to show which tab the item will be shown in... However, the display and routines to manage dashboard item movement between screen and library currently need those individual collections.

Edit: Thank you for your answers. While I do find them all interesting I believe I am going to go with James solution and will be marking it as the accepted answer.

c# .net collections

Share

Improve this question

Follow

edited Oct 27, 2008 at 15:00

asked Oct 27, 2008 at 14:14

Totty

916 ● 2 ● 10 ● 19

6 Answers

Sorted by: Highest score (default)

8

```
List< List<Placement>> ListofListOfPlacements = new Li

List<Placement> dashboard1 = new List<Placement>();
List<Placement> dashboard2 = new List<Placement>();
List<Placement> dashboard3 = new List<Placement>();
List<Placement> dashboard4 = new List<Placement>();

ListofListOfPlacements.Add(dashboard1);
ListofListOfPlacements.Add(dashboard2);
ListofListOfPlacements.Add(dashboard3);
ListofListOfPlacements.Add(dashboard4);
```

Share Improve this answer Follow

answered Oct 27, 2008 at 14:21

James Curran
103k ● 37 ● 185 ● 262

2

Since you are talking about tabs, it sounds to me like you want something closer to a dictionary keyed on the tab name, with a set of items per tab. .NET 3.5 added the `ILookup<,>` interface:

```
    ILookup<string, Foo> items = null; //TODO
    foreach (Foo foo in items["SomeTab"])
    {
        Console.WriteLine(foo.Bar);
    }
```

Note that the MS implementation is immutable - you can't edit it after creation; however, I wrote an `EditableLookup<,>` in MiscUtil that allows you to work more effectively (just like a regular .NET collection):

```
        var items = new EditableLookup<string, Foo>();
        items.Add("SomeTab", new Foo { Bar = "abc" });
        items.Add("AnotherTab", new Foo { Bar = "def"
        items.Add("SomeTab", new Foo { Bar = "ghi" });
        foreach (Foo foo in items["SomeTab"])
        { // prints "abc" and "ghi"
            Console.WriteLine(foo.Bar);
        }
```

Without `EditableLookup<,>`, you need to build the lookup via the `Enumerable.ToLookup` extension method.

If any part of this sounds like an option, I can add more detail...

Share Improve this answer Follow

answered Oct 27, 2008 at 14:41

Marc Gravell
1.1m ● 273 ● 2.6k ● 3k

1

Why not create a class that contains your second collection and any of the previous information, and just have a collection of these items?

Share Improve this answer Follow

answered Oct 27, 2008 at 14:17

Charles Graham
24.8k ● 14 ● 47 ● 56

I agree. At some point you're probably going to need more informaction for each "tab" rather than what objects are on it. – Kibbee Oct 27, 2008 at 14:24

1

why not put your collection of dash board items in a tabpage class?

like so:

```
public class DashboardTabPage : TabPage
{
    public List<DashboardItem> DashboardItems { get; s

    public DashboardTabPage() : this (new List<Dashboa
    {

    }

    public DashboardTabPage(List<DashboardItem> items)
Thing")
    {
        DashboardItems = items;
    }

}
```

public class DashboardItem { }

then you can do this:

```
c.TabPages.Add(new DashboardTabPage());
```

Share Improve this answer Follow

edited Oct 27, 2008 at 14:40

answered Oct 27, 2008 at 14:35

Hath

12.8k ● 7 ● 38 ● 38

0

I think you should go with composition, as below

Per user Dashboard-home(having multiple tabs) object containing list of dashboard objects containing list of dashboard item objects having various operations on them defined. Again the dashboard item can be a usercontrol having all possible events defined which are handled either there in dashboard item object and/or left to do for dashboard UI object that'll work with individual dashboard objects.

I guess this would be better manageable, if you are going to follow SOA and reusable component based architecture.

Share Improve this answer Follow

edited Mar 16, 2009 at 12:12

answered Mar 16, 2009 at 12:07

Kunal S
97 ● 5 ● 11

0

You can also use a tuple like this: `List<(string Father, IEnumerable<string> Children)>`.

Example:

```
var list = new List<(string Parent, IEnumerable<string
{
   ("Bob", new[] { "Charlie", "Marie" }),
   ("Robert", new[] { "John", "Geoff", "Oliver" }),
};
```

Share Improve this answer

Follow

answered Aug 8, 2019 at 9:54

Shimmy Weitzhandler

105k ● 126 ● 434 ● 639