

Forum:[Hack Forums](#)

Thread: [Tab not disposing](#)

Josh? - September 11, 2015, 6:24 am

Hi all,

I have a tab control which a user can dynamically add tabs and remove tabs.

The tabs contain controls that use up resources (specifically a connect to a SQL DB and numerous table adapters). Naturally, when a tab is opened, the processor and memory usage goes up. The issue is when the tab is closed, the usage doesn't drop again. Any idea what causes this?

Thanks, Josh

Odus - September 11, 2015, 6:44 am

First thing that comes to mind is to loop through and dispose all child controls on the tab page to be closed, then get rid of it.

aleeeeeeee - September 11, 2015, 7:28 am

or create control in runtime:

example

Code //tab control event

```
public override void OnControlAdded(args blalba)
```

```
{
```

```
    if (e.Control is TabPage)
```

```
    {
```

```
        var page = e.Control as tabpage;
```

```
        using (var button = new button() { //here properties like location and size})
```

```
        {
```

```
            button.Click += (sender,args) => //sql connection;
```

```
        page.Controls.Add(button);
```

```
        }
```

```
//using dispos object automatically
```

```
}
```

```
}
```

i haven't tried it on visual studio

Josh? - September 11, 2015, 7:32 am

(Sep 11, 2015 06:44 AM)Odus Wrote: [\[CITE\]](#) First thing that comes to mind is to loop through and dispose all child controls on the tab page to be closed, then get rid of it.

I do this:

Code Private Sub disposeControls(ByVal c As Control)

```
    If c.Controls.Count > 0 Then
```

```
        For Each ctrl As Control In c.Controls
```

```
            disposeControls(ctrl)
```

```
        Next
```

```
    End If
```

```
    c.Dispose()
```

End Sub

And when I want to remove a tab I use:

```
Code Dim tp As TabPage = 'Code to get the tab I want
disposeControls(tp)
TabControl1.TabPages.Remove(tp)
```

Odus - *September 11, 2015, 9:00 am*

Don't forget to call Dispose on the TabPage itself. Also for your control arg, you can use HasChildren property:

```
Code Private Sub DisposeControl(ctrl As Control)
    If ctrl.HasChildren Then
        For Each c As Control In ctrl.Controls
            DisposeControl(c)
        Next
    End If
```

```
        ctrl.Dispose()
End Sub
```

If this is definitely the source of the leak, are all your resources used by whatever a control calls are being disposed as well? I can't imagine controls being the only factor.