

## Re: Form.Owner property on Windows NT iiii HELP !!!!

**Source:**

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.windowsforms/2004-03/0196.html>

---

**From:** Jerry (*tolega\_at\_hotmail.com*)

**Date:** 03/03/04

Date: 3 Mar 2004 06:53:05 -0800

I can add some information.

The strange behaviour is related to the tooltip property, but it persists in other situations in which there are not tooltips defined.

I don't understand how a so basic feature is not working properly on NT.

Please we do need help, as we have done a lot of work that wont work on NT.

As a bad solution we are calling to the SetParent Win32 Function. This way all child windows remain over the main window, but remaining inside the bounds of this window (not desirable).

What we only want is the right behaviour of the Form.Owner property on NT.

In my previous post I enclosed a sample code thas shows the problem.

Thanks in advance,

César

tolega@hotmail.com (Jerry) wrote in message

news:<6c3bdfe5.0403021011.b936e27@posting.google.com>...

> *We have been developing a Winforms app on Windows 2000. Our target*

> *systems comprise Windows NT. When we installed the app in this OS,*

> *¡Surprise!, a very strange windowing behaviour arised.*

>

> *Our app is made up of a main form, and several child forms with the*

> *"owner" property set to the main form. It happens that, when we show*

> *the child windows, they don't stay over the main form (on Windows 2000*

> *it works properly).*

>

> After lot of tests, isolating the code, we have noticed that the  
> problem disappears if we don't assign tooltips to the controls of the  
> child forms, ¿?!

>

> I enclose a sample code that shows this behaviour. It's made of three  
> forms: FrmMain (parent), in which constructor we create the child  
> forms and assign them the owner property. Form1, simple form with a  
> tooltip set on the form itself, and Form2, a simple form without  
> controls.

>

> Run the app. Press button1 and button2 to show the child forms. Click  
> on Form2 to focus, then click on FrmMain: Form2 will then go under  
> FrmMain, and this SHOULD NOT HAPPEN, it should remain unfocused over  
> FrmMain.

>

> The target OS meets the specified requirements: Windows NT SP 6a, IE  
> 5.5 SP1, Framework 1.1 (download date 3/2/2004).

>

> I think this is a very important bug, that should have been detected  
> before by Microsoft.

>

> Can anybody please help us??

>

> Thanks in advance,

>

> César Arriaga

>

>

> Sample code:

>

```
> public class FrmMain : System.Windows.Forms.Form  
> {  
>     private Form1 mFrm1;  
>     private Form2 mFrm2;  
>     private System.Windows.Forms.Button button1;  
>     private System.Windows.Forms.Button button2;  
>     private System.ComponentModel.Container components = null;  
>  
>     public FrmMain()  
>     {  
>         InitializeComponent();  
>         mFrm1 = new Form1();  
>         mFrm2 = new Form2();  
>         mFrm1.Owner = this;  
>         mFrm2.Owner = this;  
>     }  
>     protected override void Dispose( bool disposing )  
>     {  
>         if( disposing )  
>         {  
>             if( components != null )
```

```
> {
> components.Dispose();
> }
> }
> base.Dispose( disposing );
> }
> private void InitializeComponent()
> {
> this.button1 = new System.Windows.Forms.Button();
> this.button2 = new System.Windows.Forms.Button();
> this.SuspendLayout();
> this.button1.Location = new System.Drawing.Point(416, 24);
> this.button1.Name = "button1";
> this.button1.Size = new System.Drawing.Size(88, 32);
> this.button1.TabIndex = 0;
> this.button1.Text = "button1";
> this.button1.Click += new System.EventHandler(this.button1_Click);
> this.button2.Location = new System.Drawing.Point(416, 72);
> this.button2.Name = "button2";
> this.button2.Size = new System.Drawing.Size(88, 32);
> this.button2.TabIndex = 1;
> this.button2.Text = "button2";
> this.button2.Click += new System.EventHandler(this.button2_Click);
> this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
> this.ClientSize = new System.Drawing.Size(528, 309);
> this.Controls.Add(this.button3);
> this.Controls.Add(this.button2);
> this.Controls.Add(this.button1);
> this.Name = "FrmMain";
> this.Text = "Main Form";
> this.ResumeLayout(false);
> }
> [STAThread]
> static void Main()
> {
> Application.Run(new FrmMain());
> }
> private void button1_Click(object sender, System.EventArgs e)
> {
> mFrm1.Show();
> }
> private void button2_Click(object sender, System.EventArgs e)
> {
> mFrm2.Show();
> }
> }
> public class Form2 : System.Windows.Forms.Form
> {
> private System.ComponentModel.IContainer components;
> public Form2()
> {
```

```
> InitializeComponent();
> }
> protected override void Dispose( bool disposing )
> {
> if( disposing )
> {
> if(components != null)
> {
> components.Dispose();
> }
> }
> base.Dispose( disposing );
> }
> private void InitializeComponent()
> {
> this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
> this.ClientSize = new System.Drawing.Size(336, 117);
> this.Name = "Form2";
> this.Text = "Form2";
>
> }
> }
> public class Form1 : System.Windows.Forms.Form
> {
> private System.Windows.Forms.ToolTip toolTip1;
> private System.ComponentModel.IContainer components;
> public Form1()
> {
> InitializeComponent();
> }
> protected override void Dispose( bool disposing )
> {
> if( disposing )
> {
> if(components != null)
> {
> components.Dispose();
> }
> }
> base.Dispose( disposing );
> }
> private void InitializeComponent()
> {
> this.components = new System.ComponentModel.Container();
> this.toolTip1 = new System.Windows.Forms.ToolTip(this.components);
> this.AutoScaleBaseSize = new System.Drawing.Size(5, 13);
> this.ClientSize = new System.Drawing.Size(320, 109);
> this.Name = "Form1";
> this.Text = "Form1";
> this.toolTip1.SetToolTip(this, "Tooltip sample");
> }
> }
```

microsoft.public.dotnet.framework.windowsforms: Re: Form.Owner property on Windows NT iiii HELP !!!!

> }