

برخی اوقات نیاز است در یک فرم ویندوزی، کنترل‌های آن‌را در حال اجرا با استفاده از ماوس جابجا کنیم و یا اندازه‌ی آن‌ها را تغییر بدیم.

در وب راهکارهای مختلفی برای این کار ارائه شده، ولی این راه‌ها معمولا یا فقط برای تغییر مکان و یا فقط برای تغییر اندازه کنترل‌ها ارائه شده‌اند. من [یکی از مقالات](#) کد پروجکت را که به جابجا کردن کنترل‌ها پرداخته بود، توسعه دادم که امکان تغییر اندازه هم به آن اضافه شود. مقاله‌ی من (به زبان انگلیسی) در [اینجا](#) قرار دارد.

چون از کلاس و متدهای استاتیک استفاده کردم، روش استفاده از این کلاس ساده بوده و افزودن قابلیت تغییر اندازه و جابجایی زمان اجرا با ماوس برای هر کنترل فقط با یک خط کد قابل انجام است:

```
ControlMoverOrResizer.Init(button1);
```

نحوه‌ی استفاده از کلاس:

برای فعال کردن قابلیت تغییر اندازه و جابجایی یک کنترل در حال اجرای برنامه با موس ما باید متد Init از کلاس MoveAndResizeControls را فراخوانی کنیم و کنترل را به عنوان پارامتر به آن بفرستیم.

```
ControlMoverOrResizer.Init(button1);
```

اگر که ما بخواهیم به همراه تغییر کنترل، خواص container آن را هم تغییر دهیم، باید کنترل container را به عنوان پارامتر دوم به متد مذکور ارسال کنیم.

```
ControlMoverOrResizer.Init(button2,panel1);
```

برخی اوقات ممکن است که ما فقط بخواهیم که یا کنترل‌ها را جابجا کنیم و یا اندازه‌ی آن‌ها را تغییر دهیم؛ در این مواقع ما باید خاصیت WorkType کلاس MoveAndResizeControls را تغییر دهیم به یکی از مقادیر ذیل تغییر دهیم.

```
internal enum MoveOrResize
{
    Move,
    Resize,
    MoveAndResize
}
```

مثالی از نحوه‌ی کار با کلاس :

```
using System;
using System.Windows.Forms;
using ControlManager;
namespace MoveAndResizeControls
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            ControlMoverOrResizer.Init(button1);
            ControlMoverOrResizer.Init(groupBox1);
            ControlMoverOrResizer.Init(textBox1);
            ControlMoverOrResizer.Init(button2,panel1);
            comboBox1.SelectedIndex = 0;
        }
    }
}
```

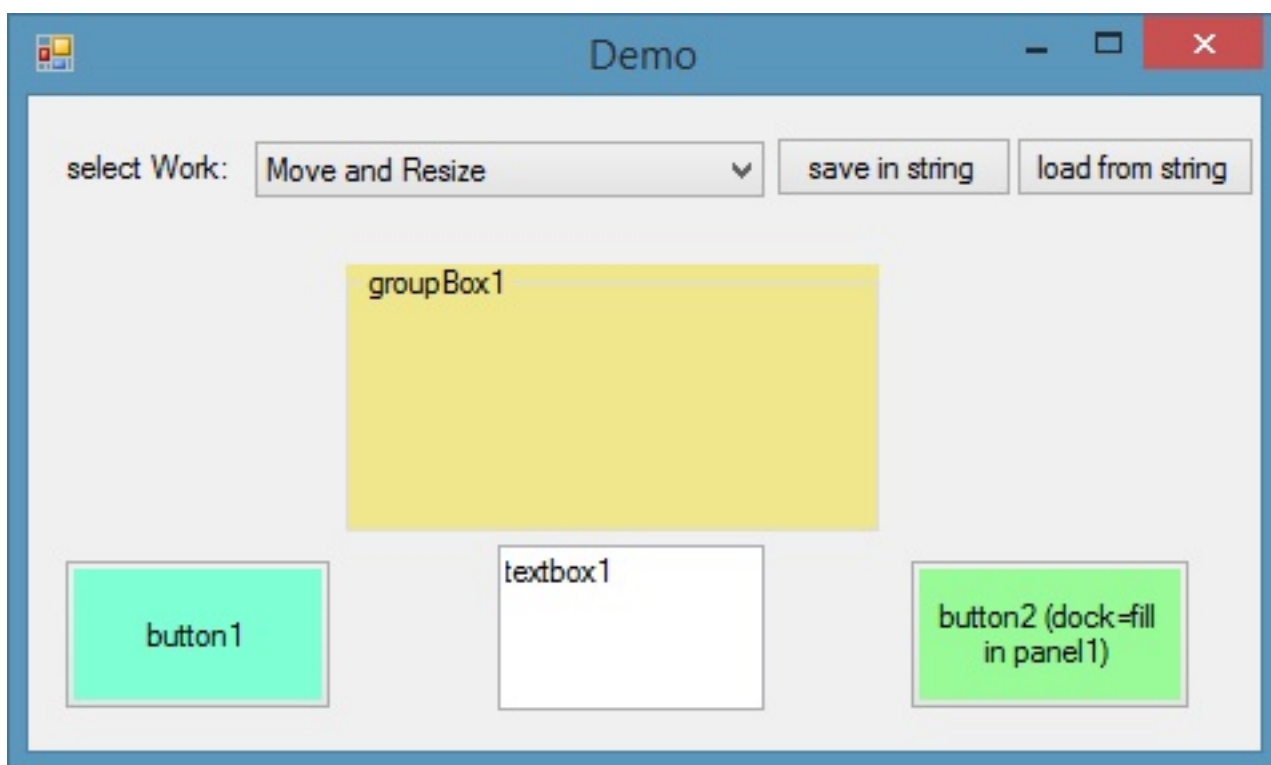
```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (comboBox1.SelectedIndex)
    {
        case 0:
            ControlMoverOrResizer.WorkType=ControlMoverOrResizer.MoveOrResize.MoveAndResize;
            break;
        case 1:
            ControlMoverOrResizer.WorkType = ControlMoverOrResizer.MoveOrResize.Move;
            break;
        case 2:
            ControlMoverOrResizer.WorkType = ControlMoverOrResizer.MoveOrResize.Resize;
            break;
    }
}
}
```

نکته: بعد از انجام تغییرات، جهت ذخیره وضعیت کنترل‌ها و بازیابی مجدد آنها می‌توان از متدهای زیر استفاده کرد:

GetSizeAndPositionOfControlsToString ، SetSizeAndPositionOfControlsFromString

[دانلود سورس](#)

شکل حالت اولیه:



شکل حالت نتیجه:

