How to drag a UserControl inside a Canvas

Asked **14 years, 6 months ago**   Modified **1 year, 5 months ago**   Viewed **31k times**

I have a Canvas in which user can add UserControl subclasses containing a form. User should be able to drag these UserControl around the Canvas.

8:25

104 ● 126

rted by: Highest score (default) ⇕

work the same.

he control:

wn);

);

useMove);

ender, MouseButtonEventArgs e)

der, MouseButtonEventArgs e)
{

```
        isDragging = false;
        var draggable = sender as UserControl;
        draggable.ReleaseMouseCapture();
    }

    private void Control_MouseMove(object sender, MouseEventArgs e)
    {
        var draggableControl = sender as UserControl;

        if (isDragging && draggableControl != null)
```

```
                    s.Parent as UIElement);

                    ansform as TranslateTransform;


                    ansform;


                    sition.X;
                    sition.Y;
```

ckpanel, or grid as well.

f you click anywhere in the control and ctly what you want.

e best way that I would implement this is alled DraggableControl that is built with DraggableControl.

this control. If you sort a column in the e updated the code so that isDragging is ous method to the LostMouseCapture

```
> { this.isDragging = false;
```

5:30   answered Sep 29, 2009 at 23:42

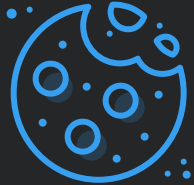   **Corey Sunwold**
● 32   **10.2k** ● 6 ● 52 ● 55

it worked. And more elegant than my tries. I

Well done! With some minor changes in mouse move handler it works perfectly. Thanks a lot! – loris
Sep 30, 2009 at 15:09

2   @loris: I am interested to hear what changes you needed to make and why you made them. I have used this exact code in a few projects already and it worked fine. Maybe there is something I overlooked.
– Corey Sunwold Sep 30, 2009 at 21:03

This really helped me. Thank you very much for posting this. – Ciel Feb 4, 2010 at 20:07

5   In `Control_MouseLeftButtonDown`, I found that I needed to replace `clickPosition =` ... `osition(this.Parent as UIElement);`.
... jump down and right (since it was getting a ... her than the hosting canvas). – Jeff May 2,

... cial element: memory of what the last ... se the mouse button, and then click that ... ol jumps back to its origin.

```
                        Object sender,


                    trol);
                    Element);


                    ject sender,



                    n as TranslateTransform);
```

```
    private void UserControl_MouseMove(Object sender, MouseEventArgs e)
    {
        var draggableControl = (sender as UserControl);
        if (isDragging && draggableControl != null)
        {
            var currentPosition = e.GetPosition(Parent as UIElement);
            var transform = (draggableControl.RenderTransform as
TranslateTransform);
            if (transform == null)
            {
                transform = new TranslateTransform();
                                            = transform;

                                    usePosition.X);
                                    usePosition.Y);
```

using them to augment the current
egate offset.

answered May 3, 2018 at 19:12

DonBoitnott
10.9k ● 6 ● 49 ● 69

vever, if the last IF condition in the
Feb 12, 2021 at 1:26

is one using the `MouseMove` event.

Code behind

```
void OnMouseMove(object sender, MouseEventArgs e)
{
  if (e.Source is Shape shape)
  {
    if (e.LeftButton == MouseButtonState.Pressed)
    {
      Point p = e.GetPosition(canvas);
      Canvas.SetLeft(shape, p.X - shape.ActualWidth / 2);
      Canvas.SetTop(shape, p.Y - shape.ActualHeight / 2);
      shape.CaptureMouse();
```

0:21   answered Oct 9, 2019 at 18:23

Themelis
4,128 ● 2 ● 25 ● 46

s are the best. This works a mint! Thanks :D

de. This works smoothly with mouse, however
d function for the click, this fails horribly (50%
023 at 21:26

p and MouseDown events and I
te as below :) I'm using Canvas.SetLeft()
need to store old position from

```
raggableControl != null)
```

17:58   answered Apr 30, 2015 at 17:50

Hawlett
896 ● 16 ● 25

ck to its original position!! You answered my
2015 at 10:53

ut the MouseDown event?

I down voted because you didn't provide the code. I was having the same problem as Duncan, and apparently you found the solution.... you just didn't show it. – Aug 4, 2015 at 17:25

Updated **minimal solution** code by Themelis with preserving grab position:

```
private Point translation;
private bool isDragging;
```

```
                                               sed)
```

```
                           vas.GetLeft(shape), p.Y -
```

```
                                      on.X);
                                      n.Y);
```

15:50   answered Oct 28, 2022 at 15:49

ironrock
1 ● 1

up with this:

```
                                 serControl
```

```
                                wn);
```

```
                           );
                           L_MouseMove);
```
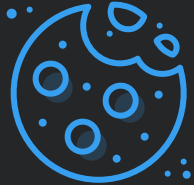
```
    private void Control_MouseLeftButtonDown(object sender,
MouseButtonEventArgs e)
    {
        _isDragging = true;
        _mouseLocationWithinMe = e.GetPosition(this);

        CaptureMouse();
    }

    private void Control_MouseLeftButtonUp(object sender, MouseButtonEventArgs
```

```
                                            MouseEventArgs e)
```

```
                               (Parent as UIElement);
```

```
                               t.X -
```

```
                               .Y -
```

Hawlett. It works ONLY when the parent
with some limits to keep the user from

answered Feb 15, 2016 at 1:33

Wally Hynds
7

e in the defined environment. – Wally Hynds

```
entHandler(BtTable_Click));
vnEvent, new
vn));
Event, new
));
```

```
newBtn.AddHandler(Button.PreviewMouseMoveEvent, new
MouseEventHandler(BtTable_MouseMove));
```

Button Move

```
private object movingObject;
private double firstXPos, firstYPos;
private int ButtonSize = 50;
```

ender, MouseButtonEventArgs e)

ize;

op of the others

der, MouseButtonEventArgs e)

op of the others

seEventArgs e)

&& sender == movingObject)

- firstXPos -

.ActualWidth -

.ActualWidth -

```
        else if (newLeft < canvas.Margin.Left)
            newLeft = canvas.Margin.Left;

        newBtn.SetValue(Canvas.LeftProperty, newLeft);

        //Vertical
        double newTop = e.GetPosition(canvas).Y - firstYPos -
canvas.Margin.Top;
        // newTop inside canvas bottom-border?
        // -- Bottom --
        if (newTop > canvas.Margin.Top + canvas.ActualHeight -
```

answered Sep 15, 2017 at 15:10

user8447700

d added all the code in user control itself
t as par your need.

```
t sender, MouseButtonEventArgs

                                Control).X;
                                Control).Y;

                                Move;

}
```

```
        }

    private void MouseMove(object sender, MouseEventArgs e)
    {
        /*
         * In this event, at first we check the mouse left button state. If it
is pressed and
         * event sender object is similar with our moving object, we can move
our control with
         * some effects.
         */
```

```
                                   Object as FrameworkElement));
                                   sed)



                          ld not go outside the canvas.
                          ect as
                          FirstXPos > 0) &&
                          Parent as FrameworkElement).X
                          as



                          GetPosition((MovingObject as
                          FirstXPos);



                          ld not go outside the canvas.
                          ect as
                          FirstYPos > 0) &&
                          Parent as FrameworkElement).Y
                          as



                          GetPosition((MovingObject as
                          FirstYPos);



                          nUp_1(object sender,
```

on through which you want to drag the

```csharp
public sealed partial class DragUserControl : UserControl
    {
        MovingObject;
        double FirstXPos, FirstYPos;

        public DragUserControl()
        {
            InitializeComponent();
        }
```

```
                                            sender,

                      object as Control).Position.X;
                      object as Control).Position.Y;

                      nterMoved;

                      ender, PointerRoutedEventArgs

                      oint(canvas).Position;
                      oint((MovingObject as

                          as
                      ter.IsInContact ==true)

                      ld not go outside the canvas
                      gObject as
                      tion.X - FirstXPos > 0) &&
                      ent).Parent as
                      ActualWidth - (MovingObject

                      GetCurrentPoint((MovingObject
                      Position.X - FirstXPos);

                      ld not go outside the canvas
                      gObject as
                      tion.Y - FirstYPos > 0) &&
                      ent).Parent as
                      ActualHeight - (MovingObject

                      GetCurrentPoint((MovingObject
                      Position.Y - FirstYPos);
```

```
                    }
                }
            }

            private void Ellipse_PointerReleased(object sender,
PointerRoutedEventArgs e)
            {
                MovingObject = null;
            }
        }
}
```

h which you want to drag the control.

We Care About Your Privacy

We and our **4** partners store and/or access information on a device, such as unique IDs in cookies to process personal data. You may accept or manage your choices by clicking below, including your right to object where legitimate interest is used, or at any time in the privacy policy page. These choices will be signaled to our partners and will not affect browsing data.

Cookie Policy.

We and our partners perform the following based on your settings:

Use precise geolocation data. Actively scan device characteristics for identification. Understand audiences through statistics or combinations of data from different sources. Store and/or access information on a device. Develop and improve services. Create profiles to personalise content. Measure content performance. Use limited data to select content. Measure advertising performance. Use limited data to select advertising. Create profiles for personalised advertising. Use profiles to select personalised advertising. Use profiles to select personalised content.