

Forum:[Hack Forums](#)

Thread: [Smooth Vertical Scroll On FlowLayoutPanel \[C#\]](#)

FryPan - July 3, 2018, 4:21 pm

How would I do this? I've tried searching for ways to do this with no working methods.

Thanks! ?

itcorp - July 4, 2018, 12:45 pm

(Jul 3, 2018 04:21 PM)FryPan Wrote: [\[CITE\]](#) How would I do this? I've tried searching for ways to do this with no working methods.

Thanks! ?

Why not use wpf,in wpf WrapPanel has similar behaviour as WinForms FlowLayoutPanel. Here is one solution that might help you in wpf. You need to put the panel in ScrollViewer

```
Code <ScrollViewer x:Name="hfSV" HorizontalScrollBarVisibility="Disabled"
VerticalScrollBarVisibility="Hidden">
    <WrapPanel x:Name="pnlHF"/>
</ScrollViewer>
up scroll
Code var scrollViwer = POSUIHelper.GetScrollViewer(hfSV) as ScrollViewer;
    if (scrollViwer != null)
    {
        scrollViwer.ScrollToVerticalOffset(scrollViwer.VerticalOffset - 5);
    }
down scroll
Code var scrollViwer = POSUIHelper.GetScrollViewer(hfSV) as ScrollViewer;
    if (scrollViwer != null)
    {
        scrollViwer.ScrollToVerticalOffset(scrollViwer.VerticalOffset + 5);
    }
Code public static class POSUIHelper
{
    public static DependencyObject GetScrollViewer(DependencyObject o)
    {
        if (o is ScrollViewer)
        { return o; }

        for (int i = 0; i < VisualTreeHelper.GetChildrenCount(o); i++)
        {
            var child = VisualTreeHelper.GetChild(o, i);

            var result = GetScrollViewer(child);
            if (result == null)
            {
                continue;
            }
            else
```

```

    {
        return result;
    }
}
return null;
}

```

public static T FindChild<T>(DependencyObject parent, string childName) where T : DependencyObject

```

{
    if (parent == null) return null;

    T foundChild = null;

    int childrenCount = VisualTreeHelper.GetChildrenCount(parent);
    for (int i = 0; i < childrenCount; i++)
    {
        var child = VisualTreeHelper.GetChild(parent, i);
        T childType = child as T;
        if (childType == null)
        {
            foundChild = FindChild<T>(child, childName);

            if (foundChild != null) break;
        }
        else if (!string.IsNullOrEmpty(childName))
        {
            var frameworkElement = child as FrameworkElement;
            if (frameworkElement != null && frameworkElement.Name == childName)
            {
                foundChild = (T)child;
                break;
            }
        }
        else
        {
            foundChild = (T)child;
            break;
        }
    }

    return foundChild;
}

```

```

public static T FindVisualParent<T>(DependencyObject child) where T : DependencyObject
{
    DependencyObject parentObject = VisualTreeHelper.GetParent(child);
    if (parentObject == null) return null;

    T parent = parentObject as T;
    if (parent != null)
    {
        return parent;
    }
}

```

```

    }
    else
    {
        return FindVisualParent<T>(parentObject);
    }
}

```

}And u can and ScrollChangedEventArgs to track VerticalOffset changes,and you can use this solution for horizontal scrolling.

FryPan - July 4, 2018, 3:05 pm

(Jul 4, 2018 12:45 PM)itcorp Wrote: [CITE](#) Why not use wpf,in wpf WrapPanel has similar behaviour as WinForms FlowLayoutPanel.

Here is one solution that might help you in wpf.You need to put the panel in ScrollViewer

Code <ScrollViewer x:Name="hfSV" HorizontalScrollBarVisibility="Disabled"
VerticalScrollBarVisibility="Hidden">

<WrapPanel x:Name="pnlHF"/>

</ScrollViewer>

up scroll

Code var scrollViwer = POSUIHelper.GetScrollViewer(hfSV) as ScrollViewer;

if (scrollViwer != null)

{

scrollViwer.ScrollToVerticalOffset(scrollViwer.VerticalOffset - 5);

}

down scroll

Code var scrollViwer = POSUIHelper.GetScrollViewer(hfSV) as ScrollViewer;

if (scrollViwer != null)

{

scrollViwer.ScrollToVerticalOffset(scrollViwer.VerticalOffset + 5);

}

Code public static class POSUIHelper

{

public static DependencyObject GetScrollViewer(DependencyObject o)

{

if (o is ScrollViewer)

{ return o; }

for (int i = 0; i < VisualTreeHelper.GetChildrenCount(o); i++)

{

var child = VisualTreeHelper.GetChild(o, i);

var result = GetScrollViewer(child);

if (result == null)

{

continue;

}

else

{

return result;

}

}

return null;

}

```
public static T FindChild<T>(DependencyObject parent, string childName) where T :  
DependencyObject
```

```
{  
    if (parent == null) return null;  
  
    T foundChild = null;  
  
    int childrenCount = VisualTreeHelper.GetChildrenCount(parent);  
    for (int i = 0; i < childrenCount; i++)  
    {  
        var child = VisualTreeHelper.GetChild(parent, i);  
        T childType = child as T;  
        if (childType == null)  
        {  
            foundChild = FindChild<T>(child, childName);  
  
            if (foundChild != null) break;  
        }  
        else if (!string.IsNullOrEmpty(childName))  
        {  
            var frameworkElement = child as FrameworkElement;  
            if (frameworkElement != null && frameworkElement.Name == childName)  
            {  
                foundChild = (T)child;  
                break;  
            }  
        }  
        else  
        {  
            foundChild = (T)child;  
            break;  
        }  
    }  
  
    return foundChild;  
}
```

```
public static T FindVisualParent<T>(DependencyObject child) where T : DependencyObject  
{  
    DependencyObject parentObject = VisualTreeHelper.GetParent(child);  
    if (parentObject == null) return null;  
  
    T parent = parentObject as T;  
    if (parent != null)  
    {  
        return parent;  
    }  
    else  
    {  
        return FindVisualParent<T>(parentObject);  
    }  
}
```

}And u can and ScrollChangedEventArgs to track VerticalOffset changes,and you can use this solution for horizontal scrolling.

Found it really hard to get in to WPF, but will definitely give it another go soon.