

جهت « [بهبود کارایی کنترل‌های لیستی WPF در حین بارگذاری تعداد زیادی از رکوردها](#) » توصیه شده‌است که مجازی سازی UI فعال گردد. به این ترتیب بجای تولید یکباره‌ی برای مثال 1000 ردیف، تنها 10 ردیفی که نمایان هستند تولید می‌شوند. بنابراین مصرف حافظه و سرعت برنامه به نحو قابل ملاحظه‌ای افزایش خواهد یافت. اما ... این مجازی سازی، اسکرول مطلوبی ندارد و بریده بریده به نظر می‌رسد.

خاصیت‌های جدید VirtualizingPanel در دات نت 4.5

تمام کنترل‌های مشتق شده‌ی از ListBox مانند ListView و امثال آن، در WPF 4.5 امکان تنظیم یک چنین خواصی را یافته‌اند:

```
<ListBox ItemsSource="{Binding Persons}"
    VirtualizingPanel.IsVirtualizing="True"
    VirtualizingPanel.ScrollUnit="Pixel"
    VirtualizingPanel.IsVirtualizingWhenGrouping="True"
    VirtualizingPanel.CacheLength="100"
    VirtualizingPanel.CacheLengthUnit="Pixel"/>
```

در WPF 4.5 پس از نمایش تعداد ردیف‌های قابل ملاحظه‌ی توسط کاربر، سیستم کش خودکاری با حق تقدم پایین فعال شده و آیتم‌هایی را که هنوز نمایان نیستند، ایجاد و کش می‌کند. به این ترتیب کاربر با اسکرول به سمت پایین یا بالا، کندی رندر یا بریده بریده به نظر رسیدن اسکرول را حس نخواهد کرد.

در اینجا می‌توان دو خاصیت CacheLength و CacheLengthUnit را مقدار دهی کرد. مقدار پیش فرض CacheLength مساوی 1.1 است. CacheLengthUnit می‌تواند یکی از مقادیر Item، Pixel یا Page را بپذیرد. هر Page در اینجا بر اساس اندازه‌ی viewport یا قسمت نمایان لیست، تعریف می‌شود. مقدار پیش فرض آن نیز Item است. همچنین در اینجا می‌توان ScrollUnit را نیز تنظیم کرد که مقادیر Item یا Pixel را می‌پذیرد. حالت پیش فرض آن Item است؛ به این معنا که اگر ردیفی کاملاً در viewport جای نگرفت، نمایش داده نمی‌شود. این مورد را با تنظیم مقدار ScrollUnit به Pixel می‌توان بهبود بخشید.

IsVirtualizingWhenGrouping سبب فعال سازی مجازی سازی حتی در حالت گروه بندی اطلاعات می‌گردد. به صورت پیش فرض اگر گروه بندی فعال شود، دیگر مجازی سازی رخ نخواهد داد.

فعال سازی قابلیت‌های دات نت 4.5 در برنامه‌های WPF 4

اگر برنامه‌ی WPF 4 خود را فعلاً قصد ندارید به دات نت 4.5 ارتقاء دهید، با توجه به اینکه اگر کاربر دات نت 4.5 را نصب کرده باشد، برنامه‌ی شما به صورت خودکار همانند یک برنامه‌ی WPF 4.5 رفتار می‌کند (دات نت 4.5 جایگزین دات نت 4 می‌شود)، می‌توان با اندکی Reflection این قابلیت‌ها را در صورت وجود، فعال کرد:

```
using System;
using System.Reflection;
using System.Windows;
using System.Windows.Controls;

namespace DNTProfiler.Common.Behaviors
{
    /// <summary>
    /// Smooth scrolling VirtualizingStackPanels, without sacrificing virtualization.
    /// </summary>
    public static class PixelBasedScrollingBehavior
    {
        private static readonly MethodInfo _setScrollUnit = typeof(VirtualizingPanel)
            .GetMethod("SetScrollUnit", BindingFlags.Public | BindingFlags.Static);

        private static readonly MethodInfo _setCacheLengthUnit = typeof(VirtualizingPanel)
            .GetMethod("SetCacheLengthUnit", BindingFlags.Public | BindingFlags.Static);
    }
}
```

```

private static readonly MethodInfo _setCacheLength = typeof(VirtualizingPanel)
    .GetMethod("SetCacheLength", BindingFlags.Public | BindingFlags.Static);

public static bool GetIsEnabled(DependencyObject obj)
{
    return (bool)obj.GetValue(IsEnabledProperty);
}

public static void SetIsEnabled(DependencyObject obj, bool value)
{
    obj.SetValue(IsEnabledProperty, value);
}

public static readonly DependencyProperty IsEnabledProperty =
    DependencyProperty.RegisterAttached("IsEnabled", typeof(bool),
    typeof(PixelBasedScrollingBehavior), new UIPropertyMetadata(false, handleIsEnabledChanged));

private static void handleIsEnabledChanged(DependencyObject obj,
    DependencyPropertyChangedEventArgs e)
{
    var listView = obj as ListView;
    if (listView == null)
    {
        throw new InvalidOperationException("This behavior can only be attached to a
        ListView.");
    }

    if (_setScrollUnit != null)
    {
        // It's .NET 4.5
        _setScrollUnit.Invoke(listView, new object[] { listView, /*Pixel*/ 0 });
    }

    if (_setCacheLengthUnit != null)
    {
        // It's .NET 4.5
        _setCacheLengthUnit.Invoke(listView, new object[] { listView, /*Pixel*/ 0 });
    }

    if (_setCacheLength != null)
    {
        // It's .NET 4.5
        var type = Type.GetType("System.Windows.Controls.VirtualizationCacheLength,
        PresentationFramework, Version=4.0.0.0, Culture=neutral, PublicKeyToken=31bf3856ad364e35");
        if (type == null) return;
        var instance = Activator.CreateInstance(type, 100.0);

        _setCacheLength.Invoke(listView, new[] { listView, instance });
    }
}
}
}

```

در اینجا با تعریف یک Attached property جدید، خواصی مانند CacheLength و ScrollUnit، CacheLengthUnit توسط Reflection یافت خواهند شد. اگر مقدار آن‌ها نال نباشند، یعنی برنامه‌ی دات نت 4 در سیستمی که بر روی آن دات نت 4.5 نصب است، در حال اجرا می‌باشد. بنابراین در این حالت می‌توان اسکرول مبتنی بر Pixel را فعال کرد تا برنامه اسکرول روان‌تری را پیدا کند.

برای استفاده از آن خواهیم داشت:

```

<ListView ItemsSource="{Binding}"
    behaviors:PixelBasedScrollingBehavior.IsEnabled="True">

```