

کنترل‌های WPF در حالت پیش فرض و بدون اعمال قالب خاصی به آن‌ها عموماً خوب عمل می‌کنند. مشکل از جایی شروع می‌شود که قصد داشته باشیم حالت پیش فرض را اندکی تغییر دهیم و یا Visual tree این کنترل‌ها اندکی پیچیده شوند. برای نمونه مدل زیر را در نظر بگیرید:

```
using System;

namespace WpfLargeLists.Models
{
    public class User
    {
        public int Id { set; get; }
        public string FirstName { set; get; }
        public string LastName { set; get; }
        public string Address { set; get; }
        public DateTime DateOfBirth { set; get; }
    }
}
```

قصد داریم فقط 1000 رکورد ساده از این مدل را به یک ListView اعمال کنیم.

```
<ListView ItemsSource="{Binding UsersTab1}" Grid.Row="1" Margin="3">
    <ListView.View>
        <GridView>
            <GridViewColumn Header="Id" Width="50" DisplayMemberBinding="{Binding
Id, Mode=OneTime}" />
            <GridViewColumn Header="FirstName" Width="100"
DisplayMemberBinding="{Binding FirstName, Mode=OneTime}" />
            <GridViewColumn Header="LastName" Width="100"
DisplayMemberBinding="{Binding LastName, Mode=OneTime}" />
            <GridViewColumn Header="Address" Width="100"
DisplayMemberBinding="{Binding Address, Mode=OneTime}" />
            <GridViewColumn Header="DateOfBirth" Width="150"
DisplayMemberBinding="{Binding DateOfBirth, Mode=OneTime}" />
        </GridView>
    </ListView.View>
</ListView>
```

در اینجا UsersTab1، لیستی حاوی فقط 1000 رکورد از شیء User است. در حالت معمولی این لیست بدون مشکل بارگذاری می‌شود. اما با اعمال مثلاً قالب [MahApp.Metro](#)، بارگذاری همین لیست، حدود 5 ثانیه با CPU usage صد در صد طول می‌کشد. علت اینجا است که در این حالت WPF سعی می‌کند تا ابتدا در VisualTree، کل 1000 ردیف را کاملاً ایجاد کرده و سپس نمایش دهد.

راه حل توصیه شده برای بارگذاری تعداد بالایی رکورد در WPF : استفاده از UI Virtualization

UI Virtualization روشی است که در آن تنها المان‌هایی که توسط کاربر در حال مشاهده هستند، تولید و مدیریت خواهند شد. در این حالت اگر 1000 رکورد را به یک ListBox یا ListView ارسال کنید و کاربر بر اساس اندازه صفحه جاری خود تنها 10 رکورد را مشاهده می‌کند، WPF فقط 10 عنصر را در VisualTree مدیریت خواهد کرد. با اسکرول به سمت پایین، مواردی که دیگر نمایان نیستند dispose شده و مجموعه نمایان دیگری خلق خواهند شد. به این ترتیب می‌توان حجم بالایی از اطلاعات را در WPF با میزان مصرف پایین حافظه و همچنین مصرف CPU بسیار کم مدیریت کرد. این مجازی سازی در WPF به وسیله VirtualizingStackPanel در دسترس است.

برای اینکه WPF virtualization به درستی کار کند، نیاز است یک سری شرایط مقدماتی فراهم شوند:

- از کنترلی استفاده شود که از virtualization پشتیبانی می‌کند؛ مانند ListBox و ListView.
- ارتفاع کنترل لیستی باید دقیقاً مشخص باشد؛ یا درون یک ردیف از Grid ایی باشد که ارتفاع آن مشخص است. برای نمونه اگر ارتفاع ردیف Grid ایی که ListView را دربرگرفته است به * تنظیم شده، مشکلی نیست؛ اما اگر ارتفاع این ردیف به Auto تنظیم

شده، کنترل لیستی برای محاسبه vertical scroll bar خود دچار مشکل خواهد شد.

- کنترل مورد استفاده نباید در یک کنترل ScrollViewer محصور شود؛ در غیر اینصورت virtualization رخ نخواهد داد. علاوه بر آن در خود کنترل باید خاصیت ScrollViewer.HorizontalScrollBarVisibility نیز به Disabled تنظیم گردد.
- در کنترل در حال استفاده، ScrollViewer.CanContentScroll باید به true تنظیم شود.

مورد مشخص بودن ارتفاع بسیار مهم است. برای نمونه در برنامه‌ای پس از فعال سازی مجازی سازی، کنترل لیستی کلاً از کار افتاد و حرکت scroll bar آن سبب بروز CPU Usage بالایی می‌شد. این مشکل با تنظیم ارتفاع آن به شکل زیر برطرف شد:

```
Height="{Binding Path=RowDefinitions[1].ActualHeight, RelativeSource={RelativeSource AncestorType=Grid}}"
```

در این تنظیم، ارتفاع کنترل، به ارتفاع سطر دوم گرید دربرگیرنده ListView متصل شده است.

- پس از اعمال موارد یاد شده، باید VirtualizingStackPanel کنترل را فعال کرد. ابتدا دو خاصیت زیر باید مقدار دهی شوند:

```
VirtualizingStackPanel.IsVirtualizing="True"  
VirtualizingStackPanel.VirtualizationMode="Recycling"
```

سپس ItemsPanelTemplate کنترل باید به صورت یک VirtualizingStackPanel مقدار دهی شود. برای مثال اگر از ListBox استفاده می‌کنید، تنظیمات آن به نحو زیر است:

```
<ListBox.ItemsPanel>  
  <ItemsPanelTemplate>  
    <VirtualizingStackPanel IsVirtualizing="True" VirtualizationMode="Recycling" />  
  </ItemsPanelTemplate>  
</ListBox.ItemsPanel>
```

و اگر از ListView استفاده می‌شود، تنظیمات آن مشابه ListBox است:

```
<ListView.ItemsPanel>  
  <ItemsPanelTemplate>  
    <VirtualizingStackPanel  
      IsVirtualizing="True"  
      VirtualizationMode="Recycling" />  
  </ItemsPanelTemplate>  
</ListView.ItemsPanel>
```

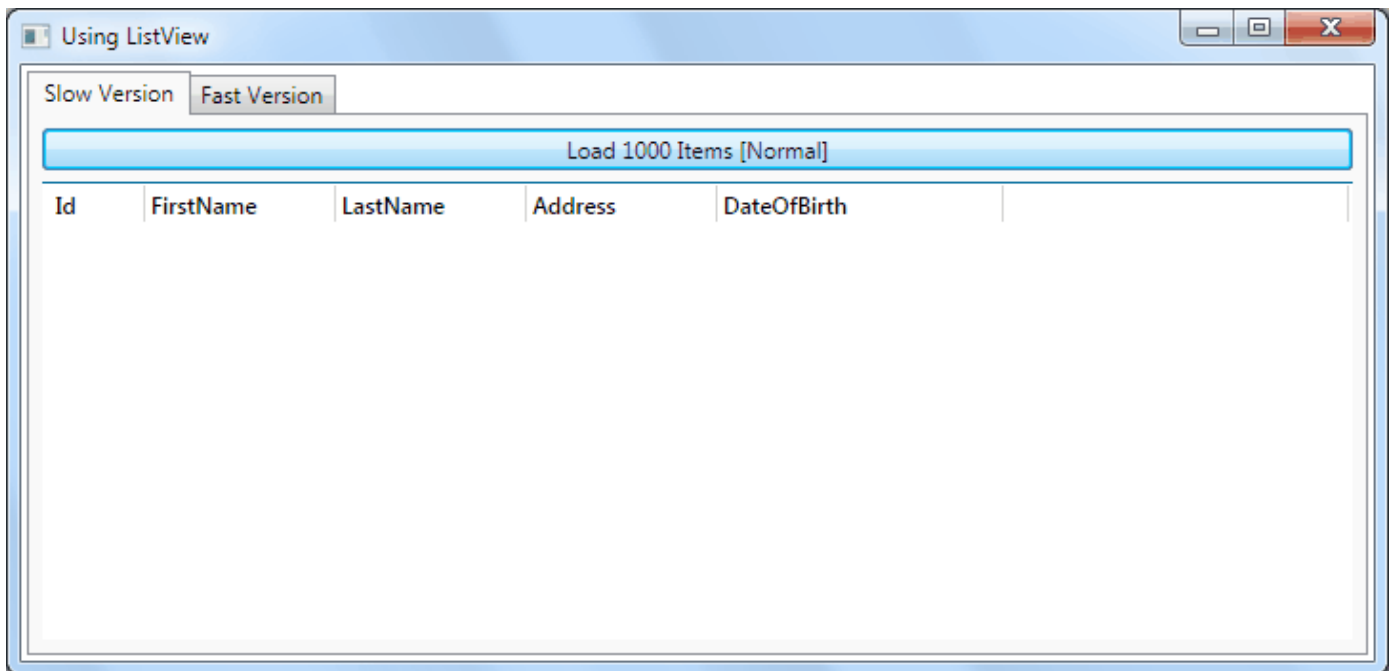
با این توضیحات ابتدای بحث به شکل زیر تغییر خواهد یافت تا مجازی سازی آن فعال گردد:

```
<ListView ItemsSource="{Binding UsersTab2}" Grid.Row="1" Margin="3"  
  ScrollViewer.HorizontalScrollBarVisibility="Disabled"  
  ScrollViewer.CanContentScroll="True"  
  VirtualizingStackPanel.IsVirtualizing="True"  
  VirtualizingStackPanel.VirtualizationMode="Recycling">  
  <ListView.ItemsPanel>  
    <ItemsPanelTemplate>  
      <VirtualizingStackPanel  
        IsVirtualizing="True"  
        VirtualizationMode="Recycling" />  
    </ItemsPanelTemplate>  
  </ListView.ItemsPanel>  
  <ListView.View>  
    <GridView>  
      <GridViewColumn Header="Id" Width="50" DisplayMemberBinding="{Binding  
Id, Mode=OneTime}" />  
      <GridViewColumn Header="FirstName" Width="100"  
DisplayMemberBinding="{Binding FirstName, Mode=OneTime}" />  
      <GridViewColumn Header="LastName" Width="100"  
DisplayMemberBinding="{Binding LastName, Mode=OneTime}" />  
      <GridViewColumn Header="Address" Width="100"  
DisplayMemberBinding="{Binding Address, Mode=OneTime}" />  
      <GridViewColumn Header="DateOfBirth" Width="150"
```

```
DisplayMemberBinding="{Binding DateOfBirth, Mode=OneTime}" />
    </GridView>
    </ListView.View>
</ListView>
```

کدهای کامل مثال فوق را از اینجا می‌توانید دریافت کنید: [WpfLargeLists.zip](#)

در این مثال دو برگه را ملاحظه می‌کنید. برگه اول حالت normal ابتدای بحث است و برگه دوم پیاده‌سازی UI Virtualization را انجام داده است.



نظرات خوانندگان

نویسنده:

سیما

تاریخ:

۱۸:۵۳ ۱۳۹۳/۰۳/۲۳

با سلام،

میخواستم بدونم امکانش هست از WrapPanel هم بعنوان یک ItemsPanelTemplate با رعایت Virtualization استفاده کرد؟

نویسنده:

وحید نصیری

تاریخ:

۱۹:۲۲ ۱۳۹۳/۰۳/۲۳

Virtualizing WrapPanel : [^](#) و [^](#)