

در WPF و Silverlight می‌توان با استفاده از مقید سازی ( [DataBinding](#) ) کنترل‌ها را به منبع‌های داده متصل کرد. این منابع به چند شیوه مختلف مانند استفاده مستقیم از خصوصیت [Source](#) قابل دسترسی هستند. یکی از این روش‌ها، ارث بری از [DataContext](#) نزدیک‌ترین والد است.

همانطور که گفته شد DataContext هر کنترل، توسط تمامی فرزندان آن قابل دسترسی است. اما در بعضی مواقع، زمانی که کنترل فرزند، بخشی از [visual](#) یا [logical tree](#) نباشند، دسترسی به DataContext وجود ندارد.

برای مثال زمانی که نیاز است خصوصیت ItemsSource مربوط به یک به لیستی خارج از ItemsSource کنترل DataGrid DataGridTemplateColumn مثلاً به لیستی درون ViewModel مربوط به Window در مثال زیر مقید شود، به صورت معمول باید به این صورت عمل کرد:

: ViewModel

```
public List<People> ComboBoxDataSource{get; set;}
```

XAML :

```
<Window x:Class="WpfApplication1.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow"
        x:Name="this">
    <Grid>
        <DataGrid ItemsSource="{Binding DataCollection}">
            <DataGrid.Columns>
                <DataGridComboBoxColumn ItemsSource="{Binding DataContext.ComboBoxDataSource,
ElementName=this}"/>
            </DataGrid.Columns>
        </DataGrid>
    </Grid>
</Window>
```

با اینکه همه چیز درست به نظر می‌رسد اما در عمل هیچ اتصالی صورت نمی‌گیرد و در پنجره Output ویژوال استادیو خطای زیر مشاهده می‌شود:

```
System.Windows.Data Error: 2 : Cannot find governing FrameworkElement or FrameworkContentElement for
target element.
BindingExpression:Path=ComboBoxDataSource; DataItem=null;
target element is 'DataGridComboBoxColumn' (HashCode=17334644); target property is 'ItemsSource' (type
'IEnumerable')
```

این خطا مشخص می‌کند که WPF نمیتواند تشخیص بدهد که کدام FrameworkElement قرار است از DataContext استفاده کند؛ چرا که همانطور که قبلاً عنوان شد DataGridTemplateColumn بخشی از [visual](#) یا [logical tree](#) نیست.

برای مشکل فوق در صورتیکه خصوصیت مورد نظر، یک خصوصیت از فرزندان کنترل باشد، از طریق استایل‌ها می‌توان مشکل را حل کرد. برای مثال به جای DataSource مربوط به DataGridComboBoxColumn می‌توان خصوصیت DataSource کنترل ComboBox درون آن را تنظیم کرد.

```
<DataGridComboBoxColumn DisplayMemberPath="FirstName">
    <DataGridComboBoxColumn.EditingElementStyle>
        <Style TargetType="ComboBox">
            <Setter Property="ItemsSource" Value="{Binding DataContext.ComboBoxDataSource ,
ElementName=this}"/>
        </Style>
    </DataGridComboBoxColumn.EditingElementStyle>
</DataGridComboBoxColumn>
```

اما در صورتیکه نیاز باشد یک خصوصیت از خود DataGridComboBoxColumn مانند Visibility مقید سازی شود، روش بالا کارساز نخواهد بود. برای حل مشکل فوق می‌توان از کلاس‌های Freezable استفاده کرد؛ چرا که این کلاس‌ها می‌توانند از DataContext ارث بری کنند حتی زمانی که بخشی از [visual](#) یا [logical tree](#) نباشند. برای این کار می‌توان کلاس زیر را ایجاد کرد:

```
public class DataBindingHelper : Freezable
{
    protected override Freezable CreateInstanceCore()
    {
        return new DataBindingHelper();
    }
    public object Data
    {
        get { return (object)GetValue(DataProperty); }
        set { SetValue(DataProperty, value); }
    }

    public static readonly DependencyProperty DataProperty =
        DependencyProperty.Register("Data", typeof(object), typeof(DataBindingHelper), new
        UIPropertyMetadata(null));
}
```

و یک نمونه از آن را در Resource های DataGrid ساخت:

```
<DataGrid.Resources>
    <local:DataBindingHelper x:Key="bindingHelper"Data="{Binding}"/>
</DataGrid.Resources>
```

و هنگام مقید سازی خصوصیت Visibility مربوط به DataGridComboBoxColumn، از نمونه ساخته شده به عنوان Source استفاده نمود.

```
<DataGridComboBoxColumn Visibility="{Binding Data.IsVisible,Converter={StaticResource
visibilityConverter}},Source={StaticResource bindingHelper}"/>
```