

Pivot kontrolü Panorama gibi Windows Phone üzerinde "Tab" yapısına alternatif bir yöntem olarak önerilir ve aynı şekilde Panorama gibi navigasyon amaçlı olarak kullanılır. Aralarında sadece görünüm farkı vardır. Panorama'da olduğu gibi birçok sayfayı *PivotItem*'lar vasıtasıyla oluşturabilir ya da görüntüleyebilirsiniz.

Pivot kontrolünü birçok farklı amaç için kullanabilirsiniz. Örneğin Ajanda, Görev Listesi veya E-posta gibi listeleme amacı olan birçok yere uygundur. Basitçe <phone:PivotItem> içerisine bir liste yerleştirerek gerçekleştirebilirsiniz.

Bunun yanı sıra çalışma anında kodlu işlemler yapabilirsiniz. Örneğin SelectionChanged event handler'ını kullanarak sayfalar arası geçiş gerçekleştirirken otomatik kayıt, liste yükleme vb. işlemleri arka planla yapma olanağına sahip olursunuz.

```
<object SelectionChanged="SelectionChangedEventHandler" .../>
```

Farklı bir örnekte ise uygulama ilk defa çalıştırılıyorsa çalışma anında Pivot kontrolündeki PivotItem'ları kullanıcının yetkilendirme sistemindeki rolüne bağlı olarak yükleyebilirsiniz.

Örnek Kod – Pivot Demosu

```
<phone:Pivot HorizontalAlignment="Left" Height="748" Margin="10,10,0,0" Title="Örnek"
VerticalAlignment="Top" Width="460">
```

```
<phone:PivotItem CacheMode="{x:Null}" Header="Sayfa 1">
<Grid/>
</phone:PivotItem>
```

```
<phone:PivotItem CacheMode="{x:Null}" Header="Sayfa 2">
<Grid/>
</phone:PivotItem>
</phone:Pivot>
```

Uygulamayı çalıştırınca, aşağıdaki görüntü elde edilir:



```
List<PivotItem> listPivotItems = null;
private void BindPivotData()
{
    listPivotItems = new List<PivotItem>() { /*liste oluşturulur*/ };
    foreach (PivotItem item in listPivotItems)
    {
        //Pivota eklenecek itemlarda sayfa değişim anında bir önceki örnekteki işlemleri uygulamak
        //handlerını bu satırdaki gibi tanımlayabiliriz.
        item.Selected += new PivotItem.SelectedEventHandler(item_Selected);
        myPivot.Items.Add(item);
    }
}
```

yaparsanız yeni sayfaya listeler gelecektir.