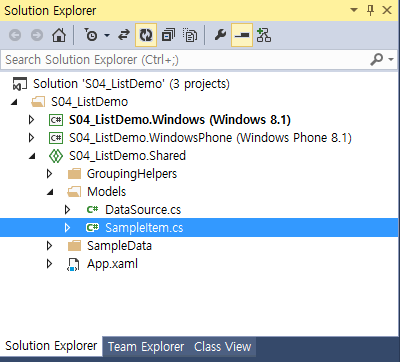
Universal App개발 실습(ListDemo)

이번 세션에서는 여러 리스트와 리스트 컨트롤에 사용되는 리스트 아이템 템플릿을 만드는법과 시맨틱줌에 대해 다룹니다.

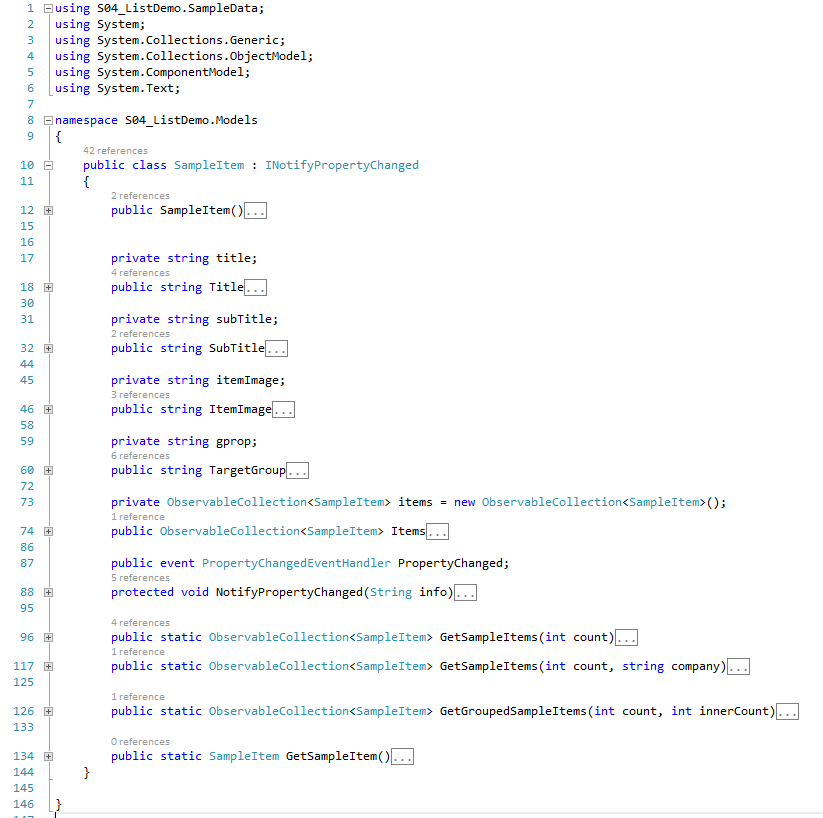
# List Item Template 만들기

### ItemTemplate 변경하기

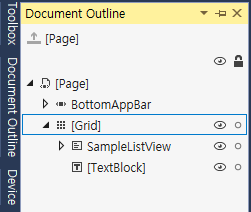
1. Session 4 폴더 > Start 폴더안의 S04\_ListDemo 솔루션 파일을 엽니다.
2. 아래 그림과 같이 솔루션 탐색기의 **S04\_ListDemo.Shared** 프로젝트를 확장 한 후 **SampleItem.cs** 파일을 엽니다.



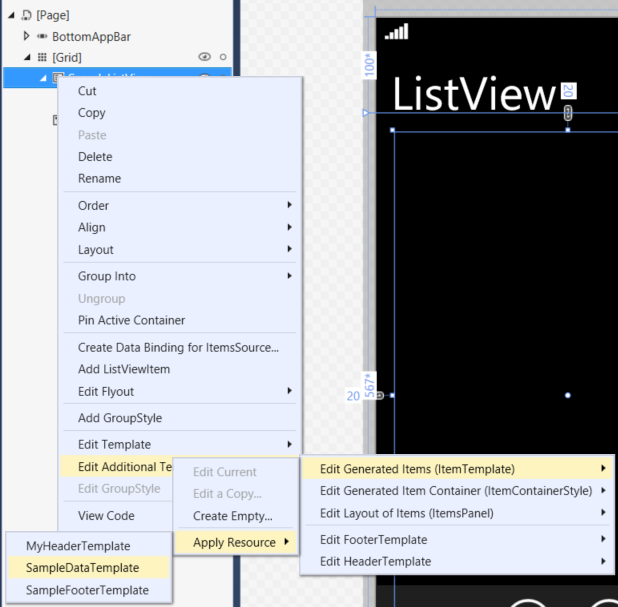
코드를 살펴보면 SampleItem 이라는 클래스가 있고, Title, SubTitle, ItemImage, TargetGroup 등의 속성을 가진 샘플 데이터를 제공하는 기능을 구현해 놓았습니다.



1. 시작 프로젝트를 Windows Phone 8.1 프로젝트로 변경한 후 실행합니다.
2. 실행된 앱의See ListView 버튼을 클릭하여 기본 렌더링을 확인합니다. 본 실습에서는 이를 미리 만들어둔 템플릿으로 변경하는 내용을 다룹니다.   
   
3. 애플리케이션을 종료합니다.
4. **S04\_ListDemo.Windows Phone** 프로젝트를 확장한 후 ListViewPage.xaml 파일을 더블클릭하여 엽니다.
5. 좌측의 Document Outline 메뉴를 선택하여 연 후 Grid 영역을 아래와 같이 확장합니다.



1. SampleListView 항목을 마우스 우클릭 한 후 “Edit Additional Templates” > “Edit Generated Items (ItemTemplate)” > “Apply Resource” > SampleDataTemplate 메뉴를 선택합니다.



1. 우측의 코드 편집기에서 ListView 항목을 찾아보면 ItemsTemplate 속성에 SampleDataTemplate이 아래와 같이 추가된 것을 보실 수가 있습니다.

<ListView x:Name="SampleListView" Grid.RowSpan="2" Grid.Row="1" Margin="20,20,20,0"

 ItemsSource="{Binding Source={StaticResource Items}}" ItemTemplate="{StaticResource SampleDataTemplate}"  />

1. 템플릿 코드를 한번 살펴봅시다. SampleListView를 우클릭해서, Edit Additional Template > Edit Generated Items(ItemTemplate) > Edit Current를 클릭합니다. 그러면 XAML 편집기 화면에서 아래와 같은 코드를 살펴볼 수 있습니다.



ListView에 보여지는 각 항목을 어떻게 보여줄 것인지를 정의하고 있습니다. 먼저 2개의 열을 가진 Grid가 있고, 그 안에 Image와 또 다른 하위 Grid가 있습니다. 하위 Grid에는 다시 2개의 열이 있어서 2개의 TextBlock이 있음을 확인할 수 있습니다. 결과적으로 아래와 같이 화면이 구성되어 보여집니다. 노란색 선으로 그리드가 어떻게 나누어지는지를 표시했습니다.



1. 다시 앱을 실행하여, See ListView 버튼을 눌러 ListView의 항목들을 확인합니다.
2. 하단의 앱 바에서 오른쪽의 버튼을 누르면 좌측에 체크박스가 나오고, 이 체크박스들을 이용해서 여러 개의 항목을 동시에 선택할 수 있는 것을 확인합니다.



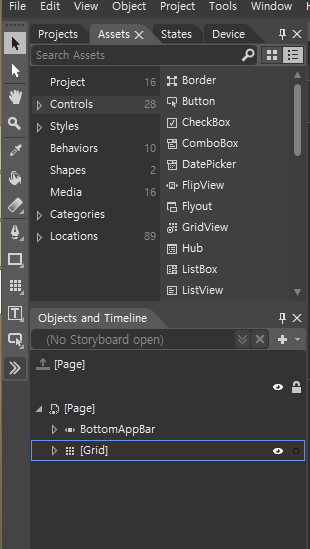
이 앱바의 버튼은 ListView의 SelectionMode 값을 아래와 같이 변경하는 코드가 적용되어 있어서 ListView의 동작이 달라지게 합니다.

SelectionMode = ListViewSelectionMode.Single 에서

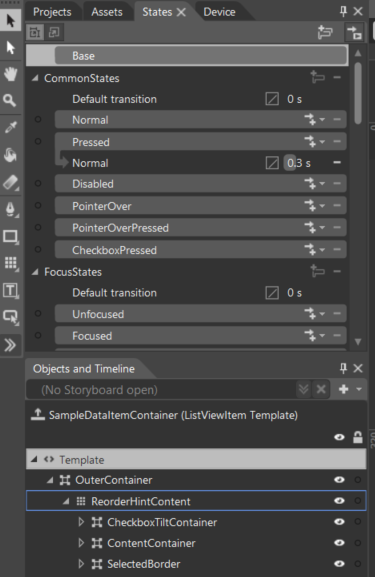
SelectionMode = ListViewSelectionMode.Multiple 로 변경

### ItemContainerStyle 을 변경하여 체크박스 색 바꾸기

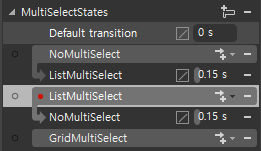
1. 솔루션 탐색기에서 **S04\_ListDemo.Windows Phone** 프로젝트에서 ListViewPage.xaml 파일을 마우스 우클릭 한 후 Open in Blend… 메뉴를 선택하여Blend에서 엽니다.
2. Blend의 좌측에 보면 Document Outline 메뉴와 유사한 Objects and Timeline 메뉴가 있습니다. (아래 그림 참조)

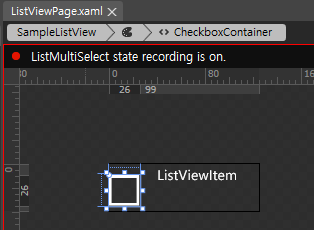


1. Grid를 확장한 후 SampleListView 항목을 마우스 우클릭 한 후 Edit Additional Templates… > Edit Generated Item Container (ItemContainerStyle) > Edit a Copy 메뉴를 선택합니다. 대화상자가 열리게 됩니다.
2. 대화상자에서 이름을 SampleDataItemContainer로 변경한 후 OK 버튼을 클릭합니다.
3. 좌측 상단의 “States” 탭을 선택합니다.

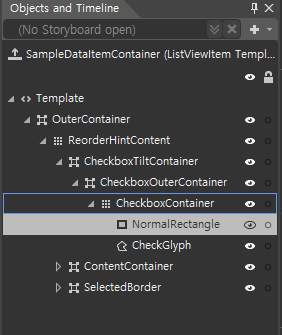


1. 스크롤을 내려 MultiSelectStates 섹션으로 이동한 후 ListMultiSelect 상태를 선택합니다. 우측 디자이너 화면에 체크박스가 나올 것입니다.

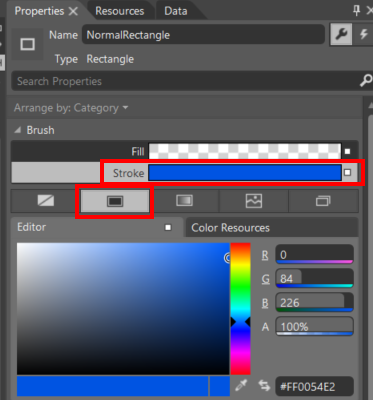




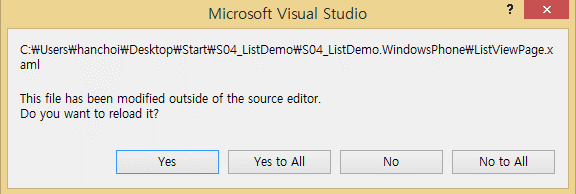
1. States 탭에서 다시 최상단의 Base 상태를 선택합니다.
2. 아래 그림과 같이 Objects and Timeline 에서 ReorderHintContent를 계속 확장하여 NormalRectangle 아이템을 선택합니다.



1. 우측의 Properties 메뉴에서 아래 그림처럼 Stroke를 선택한 후 Solid Color Brush를 선택합니다. 색상을 파란색으로 변경합니다.



1. 저장한 후 Blend를 종료합니다.
2. Visual Studio로 돌아오면 아래와 같이 파일이 변경되었음을 알리고 다시금 로드할지를 물어보는 대화상자가 열립니다. Yes to All을 선택합니다.



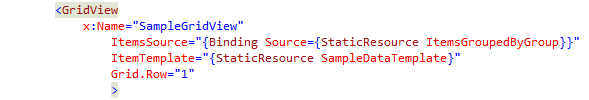
1. 다시 애플리케이션을 실행해서 ListView를 확인해 보면 아래 그림과 같이 체크박스의 색이 변경되었음을 볼 수 있습니다. 애플리케이션을 종료합니다.



# 시맨틱줌 JumpList 만들기

시맨틱줌 컨트롤을 이용하여 JumpList를 만들어 봅니다.

1. **S04\_ListDemo.Windows Phone** 프로젝트를 확장한 후 GroupedGridView.xaml 파일을 더블클릭하여 엽니다. Grouped GridView는 미리 구현되어 있습니다. 코드 편집기에서는 69번째 줄부터 입니다.

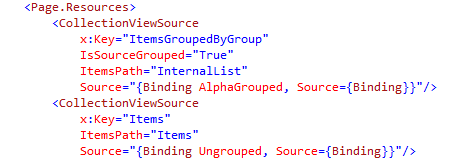


코드를 간단히 살펴보면 ItemsSource 속성에 ItemsGroupedByBroup이라는 그룹으로 묶여진 항목 데이터가 바인딩되어 있음을 확인할 수 있습니다.

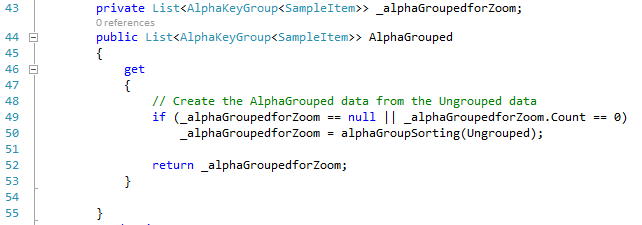


그룹 헤더의 스타일을 정의한 GroupStyle.HeaderTemplate에는 TextBlock의 Text 속성에 {Binding Key}가 설정되어 있고, 앱을 실행하였을 때 각 그룹의 Key 값을 표시합니다.

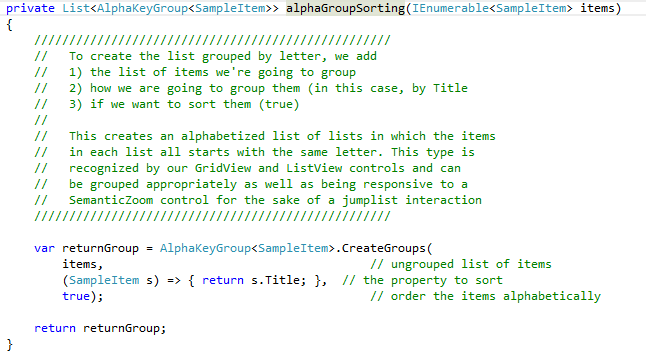
1. 코드의 맨 위로 스크롤해서 아래와 같이 2개의 CollectionViewSource 리소스를 찾아봅니다. 간단히 비교하면 하나는 그룹으로 묶여진 것이고 나머지 하나는 그냥 단순한 리스트입니다. 각각 AlphaGrouped와 Ungrouped로 바인딩되어 있습니다.



1. S04\_ListDemo.Shared 프로젝트의 Models 폴더에서 DataSource.cs파일을 열고, AlphaGrouped 속성이 구현된50번째 줄을 살펴봅니다. Ungrouped 단순 리스트에서 파생된 것입니다.



1. 50번째 줄의 alphaGroupSorting() 메서드를 클릭하고 F12를 눌러서 정의된 부분으로 이동합니다.



AlphaKeyGroup<object>.CreateGroups 메서드로 정렬을 하는 것을 확인할 수 있습니다.

1. 애플리케이션을 실행한 후 See GroupedGridView 버튼을 클릭합니다.



1. 알파벳 순으로 그룹이 만들어져 있고 헤더가 클릭에 반응은 하지만 리스트로 이동하는 기능은 구현되어 있지 않아 결과 반응은 일어나지 않습니다. 애플리케이션을 종료합니다.
2. GroupedGridView.xaml 파일을 열고 아래의 코드를 68번째 줄의 <GridView> 바로 위에 복사하여 넣습니다.

<SemanticZoom

Grid.Row="1"

Margin="20,20,20,0" >

<SemanticZoom.ZoomedInView>

<GridView />

</SemanticZoom.ZoomedInView>

<SemanticZoom.ZoomedOutView>

<GridView />

</SemanticZoom.ZoomedOutView>

</SemanticZoom>

위 코드에서 보시다시피 시맨틱줌은 2개의 GridView를 포함하고 있습니다. ZoomedInView와 ZoonedOutView는 각각 시맨틱줌의 줌인, 줌아웃 상태를 나타내는데, 같은 데이터를 사용하는 GridView 컨트롤이나 ListView 컨트롤을 포함해야 합니다.

1. 아래 그림처럼 ZoomedInView의 GridView를 지우고 기존의 GridView로 대체합니다.

<SemanticZoom.ZoomedInView>

//기존 GridView

</SemanticZoom.ZoomedInView>

1. 기존 GridView에서 Grid.Row="1" 속성을 지웁니다.
2. 8번과 유사하게 ZoomedOutView 의 GridView를 지우고 아래 코드를 복사하여 넣습니다.

<GridView

ItemsSource="{Binding Source={StaticResource ItemsGroupedByGroup}, Path=CollectionGroups}"

Background="#AA000000"

>

<GridView.ItemTemplate>

<DataTemplate>

<Border Padding="5">

<Border Background="{Binding Converter={StaticResource BackgroundConverter}}"

Width="82" Height="82"

HorizontalAlignment="Left">

<TextBlock Text="{Binding Group.Key}"

Foreground="{Binding Converter={StaticResource ForegroundConverter}}"

FontSize="48"

Padding="6"

HorizontalAlignment="Left"

VerticalAlignment="Center"/>

</Border>

</Border>

</DataTemplate>

</GridView.ItemTemplate>

</GridView>

기존 GridView과 동일하게 ItemsSource가 적용되어 있지만, 마지막에 Path=CollectionGrous가 추가되어 있습니다. 또한, ItemTemplate의 TextBlock에는 기존에 {Binding Key}로 되어 있던 것이 {Binding Group.Key}로 변경되어 있습니다.

1. 애플리케이션을 실행합니다. 다시 한번 See GroupedGridView 버튼을 클릭합니다.
2. 파란색 헤더를 클릭하였을 때 JumpList가 나오는지 확인합니다.