

应用栏 AppBar



大纲

1

AppBar简介

2

AppBar使用规范

3

AppBar编程

4

8.1新特性

AppBar用户体验

开始屏幕上的AppBar

演示

Headlines

The Pioneer Woman



A Sandwich a Day: Egg and Chorizo Torta from Tortas



Snapshot from Ecuador: Making Chicha at the Edge of the Amazon



Would You Hire This Man?



How to brew your own Munich Helles for Oktoberfest

Closet Cooking



Pasta with Whiskey, Wine, and Mushrooms



Chipotle Steak Salad



Spicy Beans



In the meantime

AppBar

- **功能：** 在需要时向用户提供各种操作接口
 - 导航
 - 触发命令
- **体验**
 - 一般默认隐藏(也可设置为始终可见)
 - 清扫屏幕上边缘或下边缘时显示
 - 被点击之后或失去焦点后隐藏
 - 也可通过编程方式控制其显示/隐藏



两种AppBar

- **底部AppBar**

- 底部AppBar通常用于触发命令
- 全局命令在右侧，上下文命令在左侧
- 当选中上下文命令的项目时，可以通过编程调用
- 如果有单个常用命令，可以将它放置在画布上
- 菜单弹出 可以用于显示相关的命令组

- **顶部AppBar**

- 顶部AppBar用于沉浸式视图中的导航（例如：返回按钮和标题）
- 在页面或视图件切换

首页

@我

评论

TopAppBar
顶部AppBar



微软中国Wi...

61 21

近期美国上市的粉色Lumia900搭配了一款相同色系、由Duality Cosmetics设计的粉色主题指甲油。该活动将在达拉斯、丹佛和洛杉矶三个城市展开，届时到场购买Lumia 900的用户还可享受美甲师的服务，Kimberly Schlegel Whitman就用新买的 Lumia 900和她刚做的粉色美甲合了个影，还是蛮搭的哦！[拍手]



微软中国Wi...

49 15

在2012.6.2及之后购买了预装Windows 7家庭普通版/家庭高级版/专业版/旗舰版的电脑请在2013.2.28前访问windowsupgradeof...

@微软中国：#新Win联播#【轻松升级成为Windows 8 pro！】在2012.6.2及之后购买了预装Windo...



微软中国Wi...

28 10

世界无限广阔，诱惑永无止境，然而，属于每一个人的现实可能性终究是有限的。不妨对一切可能性抱着开放的心态，因为那是人生魅力的源泉。但同时也要早一些在世界之海上...



今天 8:36 来自企业微博



微软中国Wi...

35 20

【创意视频分享】Nokia Lumia800 情场利器，Internet年代，交友也开始网络化了，片中的Cubie情侣以手机维系情感、分享生活，又用人脉功能...



固定到桌面



刷新



新增

BottomAppBar
底部AppBar

两种AppBar

演示

AppBar使用规范

AppBar最佳实践

- 一定要以同样的方式放置命令并按命令集对它们进行组织。
- 一定要将上下文命令放在AppBar上，在不更改视图的情况下选择某个项目时，以编程方式显示AppBar。
- 在显示上下文命令时，一定将AppBar的解除模式设置为粘滞。
- 当命令太多时，一定要使用菜单。
- 一定要考虑AppBar在贴靠视图和竖屏视图下的呈现。
- 一定将AppBar设计为水平滚动。
- 一定要使用命令、菜单和弹出窗口的默认样式。
- 使用底部的AppBar触发命令，使用顶部AppBar导航顶部进行导航。
- 不要将关键命令（用户完成任务必不可少的命令）放在AppBar上。
- 不要在AppBar中放置登录、注销或其他帐户管理命令。
- 不要将用于文本的剪贴板命令放在AppBar上。

AppBar命令组织步骤

Step 1 : 组织命令



Step 2 : 创建命令集



Step 3 : 创建菜单



Step 4 : 将命令添加到AppBar



Step 5 : 为常用命令使用标准位置

Step 1 : 组织命令

第一步是确定所有应用命令，然后按照方案或位置组织它们。

- 什么命令应该显示在整个应用中？
- 什么命令应该只显示在某些页面上？
- 什么命令应该使用超级按钮或转至设置？

以下是一个地图相关应用常用的命令列表：



Step 2 : 创建命令集

接下来我们将相似的命令分组到命令集。AppBar将命令集显示为一个单元，各集之间有一个分隔符。

- 哪些命令在功能上相关？
- 哪些命令切换各种视图类型？
- 做出选择时应该显示哪些命令？

“查看”命令



View items
as list



View items
on a map



Filter by
distance



Filter by
rating



Filter by
store hours



Filter by
area



Sort by
match



Sort by
distance



Sort by
rating

“选择”命令



Show on
map



Bookmark
item



Redo search



Save map
view



New

Step 3 : 创建菜单

接下来，考虑你的命令集是否在一个命令菜单中更合适。

- AppBar是否太拥挤或存在太多命令而无法适应？
- 是否存在一个集可以从更长的标签或互动控件中获益？



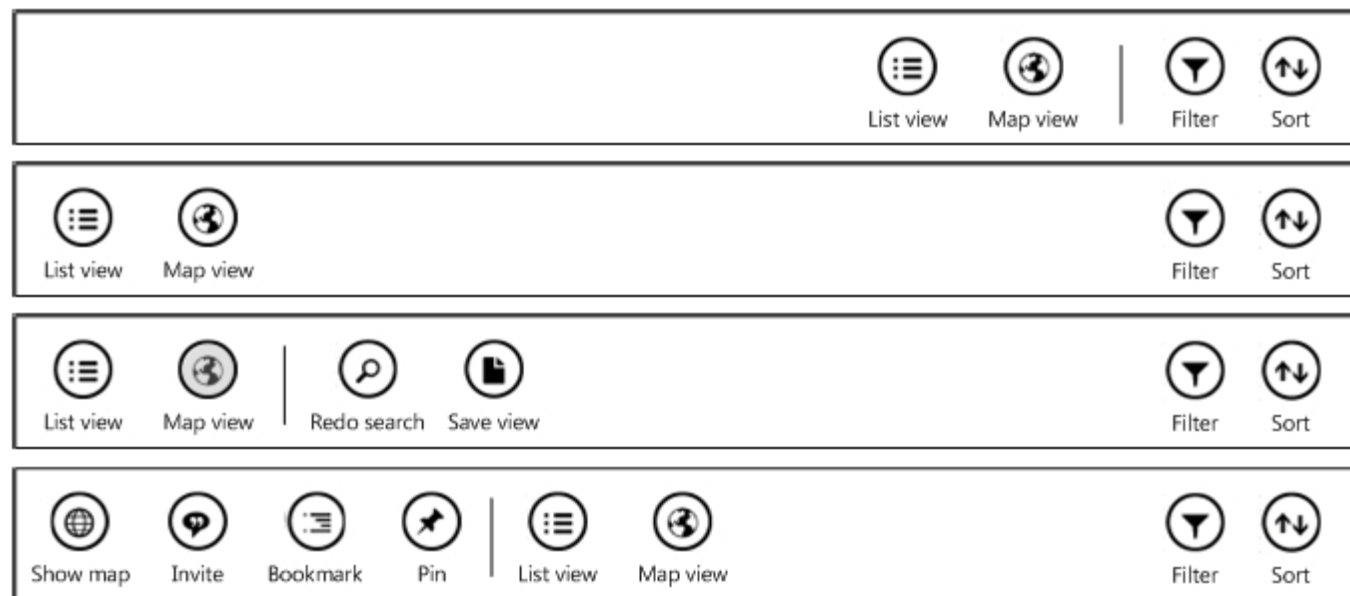
菜单允许你用更少的空间提供更多选项，并包含互动控件。
在此示例中，“排序”菜单弹出一个简单列表，以方便选择选项。
“筛选”菜单弹出一组控件，允许用户按照更复杂的条件筛选项目。

Step 4 : 将命令添加到AppBar

可通过多种方式将命令放在AppBar中，根据具体的环境，采用的方式可能有所不同。以下是应该尽可能遵守的命令放置规则。

- **可预测性**：尽可能在应用的所有视图上使用一致的互动和命令放置。
- **人体工程学**：考虑具体命令的放置能如何改进操作命令的速度或提高易用性。
- **美学**：限制命令的数量，避免AppBar看起来太复杂。选择易于理解或预测的图标。保持文本标签简短。

1. 将永久性命令放在右侧
2. 使用边缘
3. 显示/隐藏已禁用的命令
4. 插入选择命令



Step 5 : 为常用命令使用标准位置

一些命令是常用的，会在许多应用中出现。要建立一致性和培养用户自信心，在决定将命令放在AppBar的何处时应该遵循以下指南。

“选择” 命令



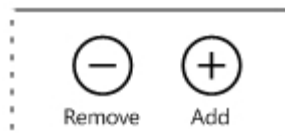
“新建项” 命令



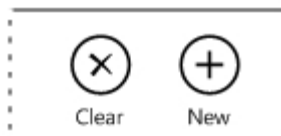
“删除” 命令



“移除” 命令



“清除” 命令



AppBar编程

AppBar一般使用步骤

Step 1 : 在XAML中定义AppBar



Step 2 : 在XAML中或C#代码中注册相应的事件



Step 3 : 在C#代码中添加相应的事件处理逻辑

定义AppBar

```
<AppBar Padding="10,0,10,0">
  <Grid>
    <Button Click="GoHome" HorizontalAlignment="Left"
      IsEnabled="{Binding Frame.CanGoBack, ElementName=pageRoot}"
      Style="{StaticResource HomeAppBarButtonStyle}"/>
    <StackPanel Orientation="Horizontal"
      HorizontalAlignment="Right">
      <Button IsEnabled="{Binding CanFlipPrevious}"
        Click="PreviousButton_Click"
        Style="{StaticResource PreviousAppBarButtonStyle}"/>
      <Button IsEnabled="{Binding CanFlipNext}"
        Click="NextButton_Click"
        Style="{StaticResource NextAppBarButtonStyle}"/>
    </StackPanel>
  </Grid>
</AppBar>
```


添加AppBar

```
<Page.TopAppBar>
  <AppBar Padding="10,0,10,0">
    <Grid>
      <Button Click="GoHome" HorizontalAlignment="Left"
        IsEnabled="{Binding Frame.CanGoBack, ElementName=pageRoot}"
        Style="{StaticResource HomeAppBarButtonStyle}"/>
      <StackPanel Orientation="Horizontal" HorizontalAlignment="Right">
        <Button IsEnabled="{Binding CanFlipPrevious}"
          Click="PreviousButton_Click"
          Style="{StaticResource PreviousAppBarButtonStyle}"/>
        <Button IsEnabled="{Binding CanFlipNext}"
          Click="NextButton_Click"
          Style="{StaticResource NextAppBarButtonStyle}"/>
      </StackPanel>
    </Grid>
  </AppBar>
</Page.TopAppBar>
```

添加BottomAppBar

```
<Page ...>  
  <Page.BottomAppBar>  
    <AppBar .../>  
  </Page.BottomAppBar>  
</Page>
```

AppBar主要属性及事件

```
<Page.BottomAppBar>
    <AppBar x:Name="bottomAppBar"
        Opened="AppBar_Opened" Closed="AppBar_Closed">
        <Button Click="Refresh_Click" HorizontalAlignment="Right"
            Style="{StaticResource RefreshAppBarButtonStyle}"/>
    </AppBar>
</Page.BottomAppBar>
```

```
private void AppBar_Opened(object sender, object e)
{
    //处理逻辑
}
```

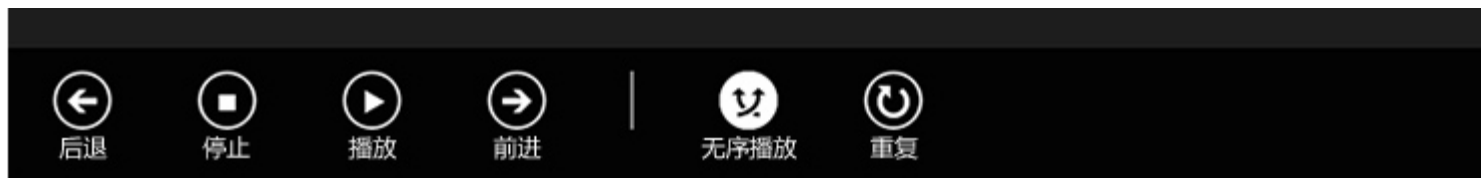
```
private void AppBar_Closed(object sender, object e)
{
    //处理逻辑
}
```

```
private void Refresh_Click(object sender, RoutedEventArgs e)
{
    contentView.Navigate(new Uri("http://www.microsoft.com"));
    bottomAppBar.IsOpen = false;
}
```

Windows 8.1 AppBar新特性

AppBar新特性

- AppBarButton
- AppBarToggleButton
- AppBarSeparator



- IsCompact



创建AppBar按钮

```
<!-- App bar button with symbol icon. -->
<AppBarButton Icon="Like" Label="SymbolIcon" Click="AppBarButton_Click"/>

<!-- App bar button with bitmap icon. -->
<AppBarButton Label="BitmapIcon" Click="AppBarButton_Click">
  <AppBarButton.Icon>
    <BitmapIcon UriSource="ms-appx:///Assets/globe.png"/>
  </AppBarButton.Icon>
</AppBarButton>

<AppBarSeparator />

<!-- App bar toggle button with font icon. -->
<AppBarToggleButton Label="FontIcon" Click="AppBarButton_Click">
  <AppBarToggleButton.Icon>
    <FontIcon FontFamily="Candara" Glyph="&#x03A3;"/>
  </AppBarToggleButton.Icon>
</AppBarToggleButton>

<!-- App bar toggle button with path icon. -->
<AppBarToggleButton Label="PathIcon" Click="AppBarButton_Click">
  <AppBarToggleButton.Icon>
    <PathIcon Data="F1 M 20,20L 24,10L 24,24L 5,24"/>
  </AppBarToggleButton.Icon>
</AppBarToggleButton>
```



AppBarButton - BitmapIcon

```
<AppBarButton Label="Favorite">  
  <AppBarButton.Icon>  
    <BitmapIcon UriSource="assets/foo.png" />  
  </AppBarButton.Icon>  
</AppBarButton>
```

AppBarButton – PathIcon

```
<AppBarButton Label="Favorite">  
  <AppBarButton.Icon>  
    <PathIcon Data="..." />  
  </AppBarButton.Icon>  
</AppBarButton>
```

AppBarButton – FontIcon

```
<AppBarButton Label="Favorite">  
  <AppBarButton.Icon>  
    <FontIcon FontFamily="assets/myfont.ttf#Font Name"  
      Glyph="H" />  
  </AppBarButton.Icon>  
</AppBarButton>
```

AppBarButton – SymbolIcon

```
<AppBarButton Icon="Favorite" Label="Add to Favorites" />
```

// below is the verbose method for symbol

```
<AppBarButton Label="Favorite">  
    <AppBarButton.Icon>  
        <SymbolIcon Symbol="Favorite" />  
    </AppBarButton.Icon>  
</AppBarButton>
```

CommandBar

```
<Page.BottomAppBar>  
  <CommandBar>  
    <CommandBar.PrimaryCommands>  
      <!-- app bar buttons here -->  
    </CommandBar.PrimaryCommands>  
    <CommandBar.SecondaryCommands>  
      <!-- app bar buttons here -->  
    </CommandBar.SecondaryCommands>  
  </CommandBar>  
</Page.BottomAppBar>
```

CommandBar

```
<Page.BottomAppBar>
```

```
  <CommandBar>
```

```
    <AppBarButton Icon="Back" Label="Back" Click="AppBarButton_Click"/>
```

```
    <AppBarButton Icon="Stop" Label="Stop" Click="AppBarButton_Click"/>
```

```
    <AppBarButton Icon="Play" Label="Play" Click="AppBarButton_Click"/>
```

```
    <AppBarButton Icon="Forward" Label="Forward" Click="AppBarButton_Click"/>
```

```
    <AppBarSeparator/>
```

```
    <AppBarToggleButton Icon="Shuffle" Label="Shuffle" Click="AppBarButton_Click"/>
```

```
    <AppBarToggleButton Icon="RepeatAll" Label="Repeat" Click="AppBarButton_Click"/>
```

```
  <CommandBar.SecondaryCommands>
```

```
    <AppBarButton Icon="Like" Label="Like" Click="AppBarButton_Click"/>
```

```
    <AppBarButton Icon="Dislike" Label="Dislike" Click="AppBarButton_Click"/>
```

```
  </CommandBar.SecondaryCommands>
```

```
</CommandBar>
```

```
</Page.BottomAppBar>
```



The image features the Microsoft logo in a white, italicized, sans-serif font, centered on a solid blue background. The background is decorated with several white-outlined squares of various sizes, some of which are slightly offset or overlapping, creating a geometric pattern.

Microsoft®