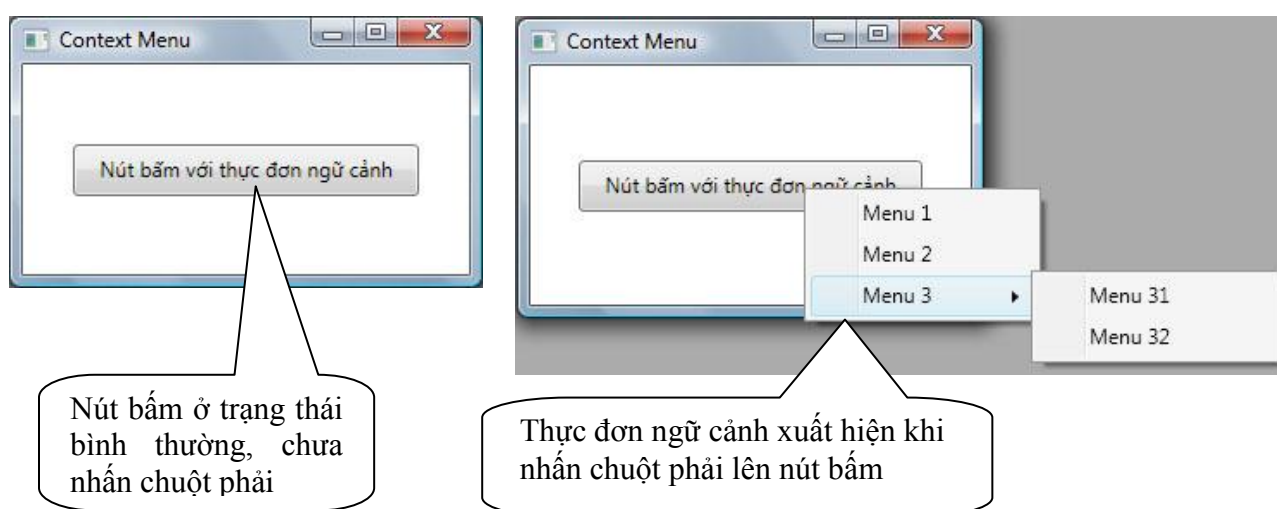


Bài 5

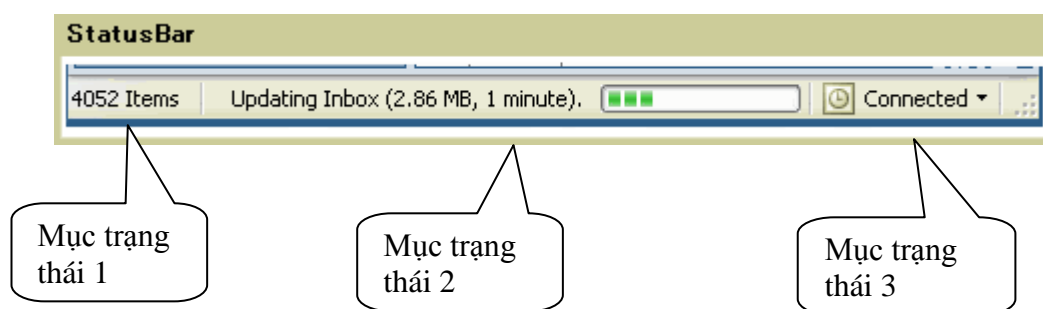
THỰC ĐƠN NGŨ CẢNH (CONTEXT MENU) VÀ THANH TRẠNG THÁI (STATUS BAR)

Thực đơn ngữ cảnh (Context Menu) là loại thực đơn gắn với một điều khiển cụ thể nào đó, chẳng hạn như một nút bấm hay một hộp soạn thảo,... Khi người dùng nhấn chuột phải vào điều khiển có gắn thực đơn ngữ cảnh thì thực đơn ngữ cảnh của điều khiển đó sẽ hiện ra và cho phép người dùng chọn công việc mong muốn từ thực đơn. Hình 5.1 minh họa thực đơn ngữ cảnh gắn với nút bấm.



Hình 5.1. Nút bấm với thực đơn ngữ cảnh (Context Menu)

Thanh trạng thái là thanh nằm ngang dưới đáy cửa sổ và hiển thị các thông tin về trạng thái hoạt động của ứng dụng. Một thanh trạng thái có thể có nhiều mục trạng thái (StatusBar Item) khác nhau, mỗi mục thể hiện một loại thông tin nào đó tới người dùng.



Hình 5.2. Thanh trạng thái gồm ba mục trạng thái khác nhau

Bài này giới thiệu các sử dụng thực đơn ngữ cảnh và thanh trạng thái bằng ngôn ngữ XAML.

1. Xây dựng thực đơn ngữ cảnh

Tương tự như thực đơn (**Menu**) thông thường đã đề cập ở bài trước, thực đơn ngữ cảnh (**Context Menu**) gồm nhiều phần tử thực đơn được tổ chức dưới dạng phân cấp. Mỗi phần tử thực đơn có thể là **Command Menu Item** (được gắn trực tiếp với các bộ quản lý sự kiện - Event handler) hay **Popup Menu Item** (chứa các phần tử thực đơn cấp dưới). Tuy nhiên, thực đơn ngữ cảnh không nằm trên đỉnh cửa sổ, nó gắn với một điều khiển nào đó và xuất hiện khi người dùng nhấn chuột phải lên điều khiển tương ứng.

Hai đoạn code dưới đây minh họa dựng thực đơn ngữ cảnh minh họa ở hình 5.1 bằng mã lệnh XAML và bằng ngôn ngữ C# để tiện so sánh.

Đoạn mã trình xây dựng thực đơn ngữ cảnh bằng XAML:

```
<Grid>
    <Button Name="cmButton" Height="30" Width="200">
        Nút bấm với thực đơn ngữ cảnh
        <Button.ContextMenu>
            <ContextMenu Name="cm" >
                <MenuItem Header="Menu 1"/>
                <MenuItem Header="Menu 2"/>
                <MenuItem Header="Menu 3">
                    <MenuItem Header="Menu 31"/>
                    <MenuItem Header="Menu 32"/>
                </MenuItem>
            </ContextMenu>
        </Button.ContextMenu>
    </Grid>
```

```
</Button>

</Grid>
```

Đoạn mã trình xây dựng thực đơn ngữ cảnh bằng C#:

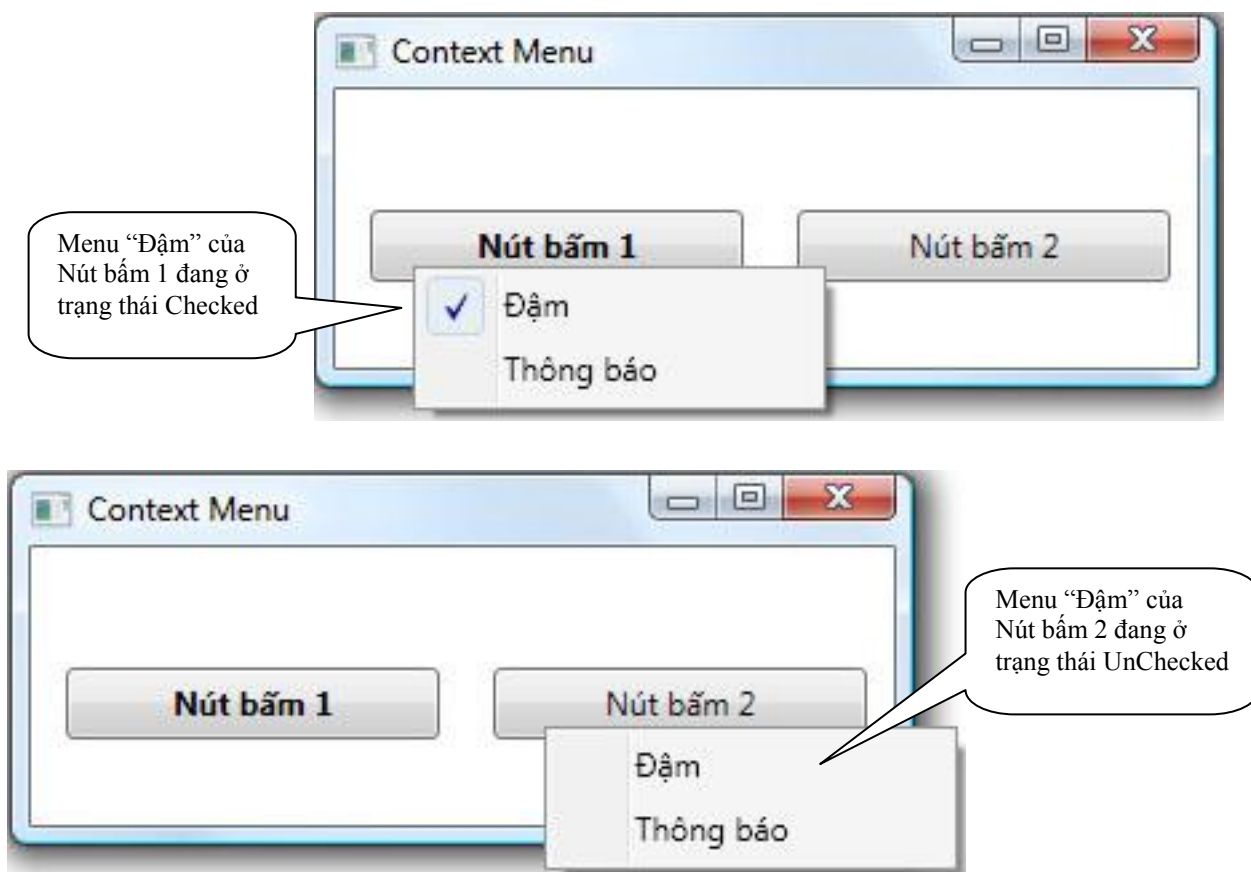
```
//Tạo nút bấm
btn = new Button();
btn.Content = "Nút bấm với thực đơn ngữ cảnh";
//Tạo thực đơn ngữ cảnh
contextmenu = new ContextMenu();
//Gán thực đơn ngữ cảnh cho nút bấm
btn.ContextMenu = contextmenu;
//Tạo các phần tử thực đơn cho thực đơn ngữ cảnh
mi = new MenuItem();
mi.Header = "Thực đơn ngữ cảnh";
mi1 = new MenuItem();
mi1.Header = "Menu 1";
mi.Items.Add(mi1);
mi2 = new MenuItem();
mi2.Header = "Menu 2";
mi.Items.Add(mi2);
mib3 = new MenuItem();
mib3.Header = "Menu 3";
mib.Items.Add(mib3);
mib31 = new MenuItem();
mib31.Header = "Menu 31";
mib3.Items.Add(mib31);
mib32 = new MenuItem();
mib32.Header = "Menu 32";
mib3.Items.Add(mib32);
//Đưa các phần tử thực đơn vào thực đơn ngữ cảnh
contextmenu.Items.Add(mi);
```

Thực đơn ngữ cảnh có thể là loại thực đơn ngữ cảnh riêng biệt, gắn với một điều cụ thể, hoặc có thể là loại thực đơn ngữ cảnh chia sẻ cho nhiều điều khiển dùng chung (**Shared ContextMenu**).

1.1 Xây dựng thực đơn ngữ cảnh riêng biệt

Thực đơn ngữ cảnh riêng biệt là thực đơn ngữ cảnh gắn với một điều khiển cụ thể, các trạng thái của menu này chỉ dành riêng cho điều khiển chứa nó sử dụng. Mã lệnh tạo thực đơn ngữ cảnh loại này

đặt trực tiếp bên trong cặp thẻ của điều khiển chứa nó (như minh họa ở đoạn mã XAML trên). Xem ví dụ minh họa ở hình 5.3, minh họa hai nút bấm, mỗi nút có một thực đơn ngữ cảnh riêng. Khi chọn Menu “Đậm” của nút nào thì nội dung của nút đó hiển thị dạng chữ đậm và đồng thời Menu tương ứng cũng ở trạng thái Checked và ngược lại. Trạng thái Checked của mục “Đậm” của menu ngữ cảnh thuộc nút bấm 1 không ảnh hưởng tới thực đơn của nút bấm 2.



Hình 5.3. Ví dụ về thực đơn ngữ cảnh riêng biệt của từng điều khiển khác nhau

Mã lệnh XAML của ví dụ trên như sau.

Đoạn mã trình Menu tạo thực đơn ngữ cảnh bằng XAML:

```
<Window x:Class="ContextMenuApp1.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Context Menu" Height="142" Width="339">
    <Grid>
        <Button Name="cmButton1" Margin="13,44.5,0,32"
HorizontalAlignment="Left" Width="140">
            Nút bấm 1
            <Button.ContextMenu>
                <ContextMenu Name="cm1" >
```

```

        <MenuItem Header="Đậm" ToolTip="Chữ đậm"
IsCheckable="True"
        Checked="Bold_Checked1"
        Unchecked="Bold_Unchecked1"/>
        <MenuItem Header="Thông báo" Click="Message1"/>
    </ContextMenu>
</Button.ContextMenu>
</Button>

    <Button HorizontalAlignment="Right" Margin="0,44.5,10,32"
Name="cmButton2" Width="140">
        Nút bấm 2
        <Button.ContextMenu>
            <ContextMenu Name="cm2">
                <MenuItem Header="Đậm" IsCheckable="True" ToolTip="Chữ
đậm"
                Checked="Bold_Checked2"
Unchecked="Bold_Unchecked2" />
                <MenuItem Header="Thông báo" Click="Message2"/>
            </ContextMenu>
        </Button.ContextMenu>
    </Button>
</Grid>
</Window>

```

Trong đoạn mã trên, ta có hai nút bấm với nhãn là “Nút bấm 1” và “Nút bấm 2”, mỗi nút bấm có một thực đơn ngữ cảnh riêng.

Thực đơn ngữ cảnh của nút bấm được bắt đầu bằng `<Button.ContextMenu>` và kết thúc bằng `</Button.ContextMenu>`. Trong cặp thẻ này là cặp thẻ `<ContextMenu>` và `</ContextMenu>`. Trong cặp thẻ `<ContextMenu>` và `</ContextMenu>` chứa các thẻ `<MenuItem>` định nghĩa các mục của thực đơn.

Các mục thực đơn của thực đơn ngữ cảnh hoạt động tương tự như thư đơn thông thường được đề cập ở bài trước.

Dưới đây là đoạn mã trình C# khai báo các hàm xử lý sự kiện khi nhấn vào các mục thực đơn trên.

```

namespace ContextMenuApp1
{
    /// <summary>

```

```

    /// Interaction logic for Window1.xaml
    /// </summary>
    public partial class Window1 : Window
    {
        public Window1()
        {
            InitializeComponent();
        }

        private void Bold_Checked1(object sender, RoutedEventArgs e)
        {
            cmButton1.FontWeight = FontWeights.Bold;
        }

        private void Bold_Unchecked1(object sender, RoutedEventArgs e)
        {
            cmButton1.FontWeight = FontWeights.Normal;
        }

        private void Message1(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Bạn chọn thực đơn nút bấm 1");
        }

        private void Bold_Checked2(object sender, RoutedEventArgs e)
        {
            cmButton2.FontWeight = FontWeights.Bold;
        }

        private void Bold_Unchecked2(object sender, RoutedEventArgs e)
        {
            cmButton2.FontWeight = FontWeights.Normal;
        }

        private void Message2(object sender, RoutedEventArgs e)
        {
            MessageBox.Show("Bạn chọn thực đơn nút bấm 2");
        }
    }
}

```

1.2 Thực đơn ngữ cảnh chia sẻ (Shared Context Menu)

Thực đơn ngữ cảnh chia sẻ là loại thực đơn ngữ cảnh có thể gắn với nhiều điều khiển khác nhau. Khi một mục trên thực đơn ngữ cảnh của một điều khiển được Checked thì tất cả các điều khiển khác cũng chia sẻ trạng thái này.

Ví dụ sau đây minh họa bốn điều khiển gồm hai Button và hai CheckBox cùng chia sẻ chung một thực đơn ngữ cảnh, khi Check vào mục đầu tiên của thực đơn ngữ cảnh trên một Button hay một CheckBox thì mục tương ứng của các thực đơn ngữ cảnh trên các Button hay CheckBox khác cũng có trạng thái Check tương ứng.



Hình 5.4. Ví dụ về thực đơn ngữ cảnh chia sẻ

Đoạn mã sau minh họa mã lệnh XAML của ví dụ này.

```
<Window
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    SizeToContent="WidthAndHeight" Title="Vi du Shared Context Menu">
    <Window.Resources>
        <ContextMenu x:Key="ContextMenuChiase" x:Shared="True">
            <MenuItem Header="Đây là mục thực đơn có trạng thái"
                IsCheckable="True" />
            <Separator/>
            <MenuItem Header="Đây là mục thực đơn thông thường" />
        </ContextMenu>
    </Window.Resources>
    <StackPanel Margin="5">
```

```

        <TextBlock TextWrapping="WrapWithOverflow" Width="400"
FontSize="12">
            <Run FontSize="24">Shared ContextMenu </Run>
            <LineBreak/>
            Có bốn điều khiển chia sẻ thực đơn ngữ cảnh.
            x:Shared được gán giá trị <Bold>True</Bold>, để cho phép các
            điều khiển chia sẻ ContextMenu. Bạn có thể thử bằng cách
            check vào mục thực đơn đầu tiên của một điều khiển
            sau đó mở thực đơn ngữ cảnh ở các điều khiển khác để xem
            trạng thái của mục đầu tiên của mỗi thực đơn ngữ cảnh.
        </TextBlock>
        <Button Margin="0,5,0,0" Background="LightBlue"
            Content="Nút bấm này có một ContextMenu"
            ContextMenu="{DynamicResource ContextMenuChiase}" />
        <Button Background="Pink"
            Content="Nút bấm này sử dụng ContextMenu tương tự"
            ContextMenu="{DynamicResource ContextMenuChiase}" />
        <CheckBox BorderBrush="Red"
            Content="Check Box sử dụng ContextMenu tương tự"
            ContextMenu="{DynamicResource ContextMenuChiase}" />
        <CheckBox BorderBrush="Green"
            Content="Check Box sử dụng ContextMenu tương tự"
            ContextMenu="{DynamicResource ContextMenuChiase}" />
    </StackPanel>
</Window>

```

Khác với thực đơn ngữ cảnh thông thường, vị trí câu lệnh tạo thực đơn ngữ cảnh được không nằm giữa cặp thẻ của các điều khiển chứa nó mà được khai báo dưới dạng tài nguyên chung của Window.

```

<Window.Resources>
    <ContextMenu x:Key="ContextMenuChiase" x:Shared="True">
        <MenuItem Header="Đây là mục thực đơn có trạng thái"
IsCheckable="True" />
        <Separator/>
        <MenuItem Header="Đây là mục thực đơn thông thường" />
    </ContextMenu>
</Window.Resources>

```

Thuộc tính `x:Key` dùng để khai báo tên của ContextMenu, sẽ được dùng để gán cho các điều khiển muốn sử dụng ContextMenu này. Các điều khiển sẽ gán ContextMenu nhờ thuộc tính `ContextMenu="{DynamicResource ContextMenuChiase}"`.


```

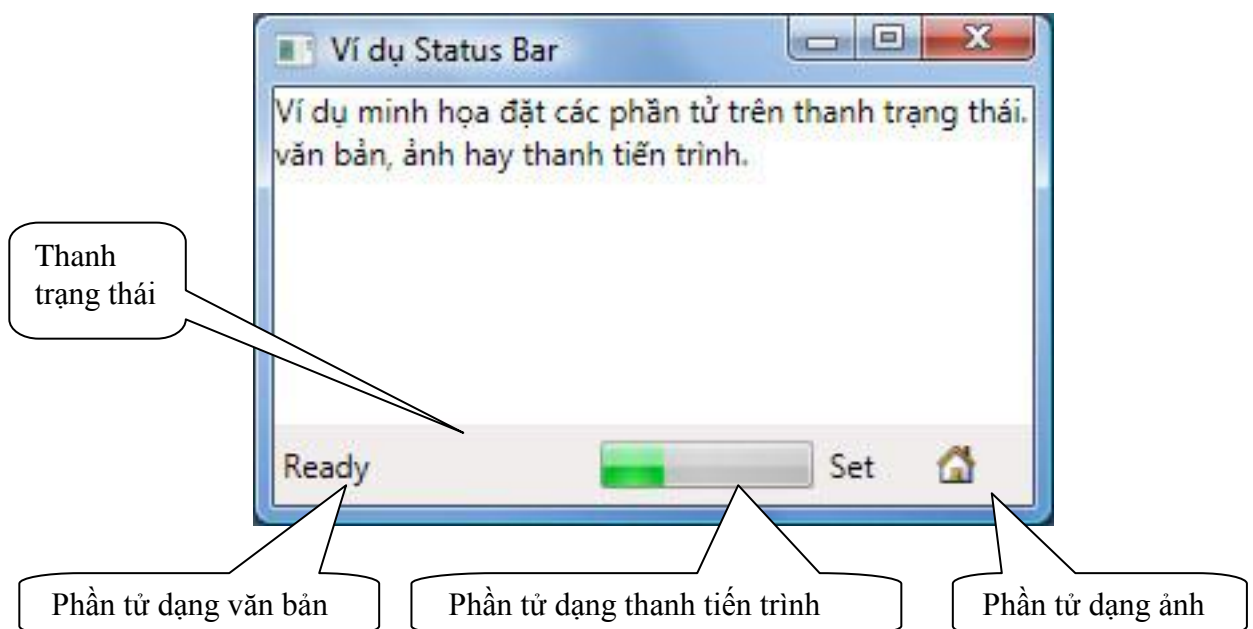
<Button Margin="0,5,0,0" Background="LightBlue"
    Content="Nút bấm này có một ContextMenu"
    ContextMenu="{DynamicResource ContextMenuChiase}" />
<Button Background="Pink"
    Content="Nút bấm này sử dụng ContextMenu tương tự"
    ContextMenu="{DynamicResource ContextMenuChiase}" />
<CheckBox BorderBrush="Red"
    Content="Check Box sử dụng ContextMenu tương tự"
    ContextMenu="{DynamicResource ContextMenuChiase}" />
<CheckBox BorderBrush="Green"
    Content="Check Box sử dụng ContextMenu tương tự"
    ContextMenu="{DynamicResource ContextMenuChiase}" />

```

Chú ý, phải đặt giá trị cho `x:Shared="True"` thì ContextMenu này mới có thể được chia sẻ cho các điều khiển. Nếu đặt là `x:Shared="False"` thì các điều khiển vẫn sử dụng được MenuContext này, nhưng mỗi điều khiển có một thể hiện riêng của ContextMenu (không chia sẻ chung).

2. Xây dựng và sử dụng thanh trạng thái (StatusBar)

Thanh công trạng thái là thanh nằm ngang, bên dưới đáy cửa sổ, gồm nhiều phần tử nhằm thể hiện thông tin về các trạng thái hoạt động của ứng dụng. Mỗi phần tử có thể là một văn bản, một biểu tượng hay một thanh tiến trình (Minh họa hình 5.5).

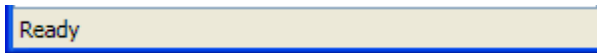


Hình 5.5 .Thanh trạng thái

2.1. Thanh trạng thái với các phần tử văn bản

Để bắt đầu tìm hiểu từng bước xây dựng thanh trạng thái, ta tìm hiểu mã lệnh XAML tạo thanh trạng thái đơn giản với các phần tử dạng văn bản.

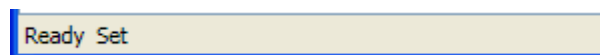
Thanh trạng thái đơn giản, chỉ gồm một phần tử dạng văn bản.



Thanh trạng thái được xây dựng bằng đoạn mã XAML sau:

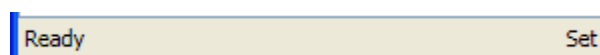
```
< StatusBar Grid.Row="1">
    < StatusBarItem>
        < TextBlock> Ready </TextBlock>
    </ StatusBarItem>
</ StatusBar>
```

Thanh trạng thái gồm nhiều hơn một phần tử. Ví dụ, thanh trạng thái gồm hai mục Ready và Set



```
< StatusBar Grid.Row="1">
    < StatusBarItem>
        < TextBlock> Ready </TextBlock>
    </ StatusBarItem>
    < StatusBarItem>
        < TextBlock> Set </TextBlock>
    </ StatusBarItem>
</ StatusBar>
```

Ở ví dụ trên phần tử thứ hai nằm ngay bên cạnh phần tử thứ nhất, nếu muốn phần tử thứ nhất có rộng lớn hơn và phần tử thứ hai được căn bên phải cửa sổ cho thuận tiện (như hình dưới) ta phải sử dụng kỹ thuật tạo layout (bố cục) trong thanh trạng thái để phân chia thanh trạng thái thành các vùng mong muốn và đặt các phần tử cần thiết vào các vùng tương ứng



```

<StatusBar>
    <StatusBar.ItemsPanel>
        <ItemsPanelTemplate>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="*" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="Auto" />
                </Grid.ColumnDefinitions>
            </Grid>
        </ItemsPanelTemplate>
    </StatusBar.ItemsPanel>
    <StatusBarItem Grid.Column="0">
        <TextBlock>Ready</TextBlock>
    </StatusBarItem>
    <StatusBarItem Grid.Column="1">
        <TextBlock>Set</TextBlock>
    </StatusBarItem>
</StatusBar>

```

Ở đoạn mã XAML trên ta sử dụng thẻ `<StatusBar.ItemsPanel>` và `</ItemsPanelTemplate>` để định nghĩa bố cục của các phần tử trong thanh trạng thái. Trong đó là cặp thẻ `<ItemsPanelTemplate>` và `</ItemsPanelTemplate>`. Kế tiếp, sử dụng thẻ `<Grid>` để định nghĩa các vùng hiển thị kiểu ô lưới. Trong ví dụ này, lưới bao gồm một dòng và hai cột. Cột thứ nhất có độ dài tự co dãn để chiếm toàn bộ không gian trống nhờ thuộc tính `Width="*" />`. Cột thứ hai có độ dài bằng độ dài nội dung mà nó chứa nhờ thuộc tính `Width="Auto" />`.

Phần tử thứ nhất được gắn với cột thứ đầu tiên của lưới (`Grid.Column="0"`).

```

<StatusBarItem Grid.Column="0">
    <TextBlock>Ready</TextBlock>
</StatusBarItem>

```

Cột thứ hai được gắn với cột thứ hai của lưới (`Grid.Column="1"`).

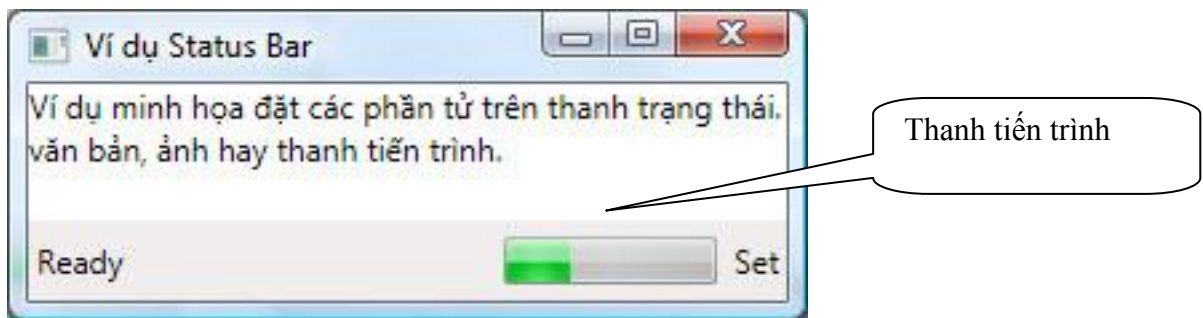
```

<StatusBarItem Grid.Column="1">
    <TextBlock>Set</TextBlock>
</StatusBarItem>

```

2.2. Thanh trạng thái với phần tử là thanh tiến trình (Progress Bar)

Đôi khi, trên thanh trạng thái ta muốn thể hiện trạng thái thực hiện của một công việc nào đó, ta có thể đưa thanh tiến trình vào một phần tử trên thanh trạng thái (minh họa hình 5.6).



Hình 5.6. Thanh trạng thái với phần tử là thanh tiến trình

Mã lệnh minh họa thanh trạng thái trên như sau:

```
<StatusBar>
  <StatusBar.ItemsPanel>
    <ItemsPanelTemplate>
      <Grid>
        <Grid.RowDefinitions>
          <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
          <ColumnDefinition Width="*" />
          <ColumnDefinition Width="Auto" />
          <ColumnDefinition Width="Auto" />
        </Grid.ColumnDefinitions>
      </Grid>
    </ItemsPanelTemplate>
  </StatusBar.ItemsPanel>
  <StatusBarItem Grid.Column="0">
    <TextBlock>Ready</TextBlock>
  </StatusBarItem>
  <StatusBarItem Grid.Column="1">
    <ProgressBar Value="30" Width="80" Height="18" />
  </StatusBarItem>
  <StatusBarItem Grid.Column="2">
```

```

        <TextBlock>Set</TextBlock>
    </StatusBarItem>

</StatusBar>

```

Ở ví dụ trên, ta đặt thanh trạng thái vào mục thứ hai của thanh trạng thái.

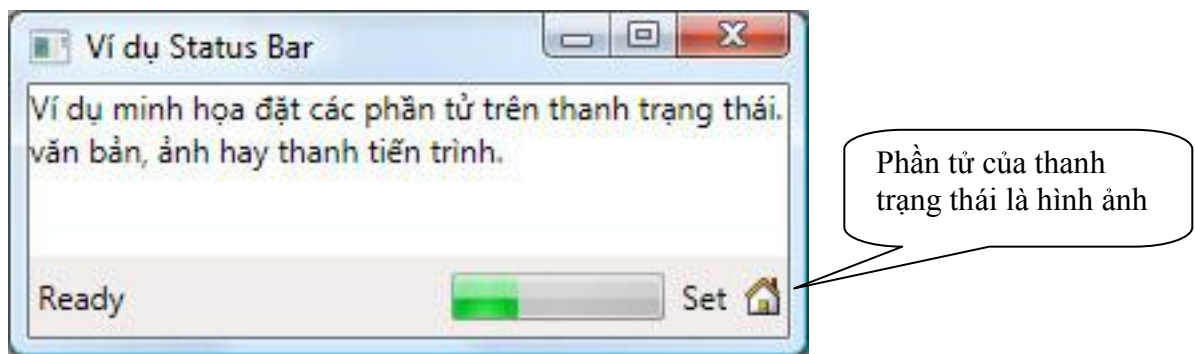
```

<StatusBarItem Grid.Column="1">
    <ProgressBar Value="30" Width="80" Height="18"/>
</StatusBarItem>

```

2.3. Thanh trạng thái với phần tử là hình ảnh

Hình 57. Minh họa phần tử thanh trạng thái là hình ảnh biểu tượng.



Hình 5.7. Thanh trạng thái với phần tử là hình ảnh

Mã lệnh minh họa thanh trạng thái trên như sau:

```

<StatusBar>
    <StatusBar.ItemsPanel>
        <ItemsPanelTemplate>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="*" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="Auto" />
                    <ColumnDefinition Width="Auto" />
                    <ColumnDefinition Width="20" />
                </Grid.ColumnDefinitions>
            </Grid>
        </ItemsPanelTemplate>
    </StatusBar.ItemsPanel>

```

```

        </Grid>
    </ItemsPanelTemplate>
</StatusBar.ItemsPanel>
<StatusBarItem Grid.Column="0">
    <TextBlock>Ready</TextBlock>
</StatusBarItem>
<StatusBarItem Grid.Column="1">
    <ProgressBar Value="30" Width="80" Height="18"/>
</StatusBarItem>
<StatusBarItem Grid.Column="2">
    <TextBlock>Set</TextBlock>
</StatusBarItem>
<StatusBarItem Grid.Column="3">
    <Image Source="Resources/Home.png" Width="16" Height="16"/>
</StatusBarItem>
</StatusBar>

```

Ở ví dụ trên, ta đặt thanh trạng thái vào mục thứ tư của thanh trạng thái.

```

<StatusBarItem Grid.Column="3">
    <Image Source="Resources/Home.png" Width="16" Height="16"/>
</StatusBarItem>

```

2.4. Mã lệnh tổng hợp của thanh trạng thái

```

<Window x:Class="StatusBarApp1.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Title="Ví dụ Status Bar" Height="300" Width="300">

    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="*" />
            <RowDefinition Height="30" />
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="0" >
            <TextBlock FontSize="12" TextWrapping="WrapWithOverflow"
Width="400">

                Ví dụ minh họa đặt các phần tử trên thanh trạng thái.
                Mỗi phần tử có thể là văn bản, ảnh hay thanh tiến trình.
            </TextBlock>

```

```

</StackPanel>
<StatusBar Grid.Row="1">
    <StatusBar.ItemsPanel>
        <ItemsPanelTemplate>
            <Grid>
                <Grid.RowDefinitions>
                    <RowDefinition Height="*" />
                </Grid.RowDefinitions>
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="*" />
                    <ColumnDefinition Width="Auto" />
                    <ColumnDefinition Width="Auto" />
                    <ColumnDefinition Width="20" />
                </Grid.ColumnDefinitions>
            </Grid>
        </ItemsPanelTemplate>
    </StatusBar.ItemsPanel>
    <StatusBarItem Grid.Column="0">
        <TextBlock>Ready</TextBlock>
    </StatusBarItem>
    <StatusBarItem Grid.Column="1">
        <ProgressBar Value="30" Width="80" Height="18" />
    </StatusBarItem>
    <StatusBarItem Grid.Column="2">
        <TextBlock>Set</TextBlock>
    </StatusBarItem>
    <StatusBarItem Grid.Column="3">
        <Image Source="Resources/Home.png" Width="16"
Height="16" />
    </StatusBarItem>
</StatusBar>
</Grid>
</Window>

```

3. Câu hỏi ôn tập

1. *Context Menu* xuất hiện khi nhấn chuột nào lên điều khiển?
 - A. Chuột trái

- B. Chuột phải

Câu trả lời: B

2. **Context Menu có thể chứa cả Command Menu Item và Popup Menu Item?**

- A. Đúng
B. Sai

Câu trả lời: A

3. **Câu lệnh XAML định nghĩa thực đơn ngữ cảnh phải nằm trong cặp thẻ định nghĩa điều khiển chứa thực đơn?**

- A. Đúng
B. Sai

Câu trả lời: B

4. **Thực đơn ngữ cảnh có thể chia sẻ cho nhiều điều khiển dùng chung?**

- A. Đúng
B. Sai

Câu trả lời: A

5. **Thanh trạng thái chỉ được phép chứa một phần tử trạng thái:**

- A. Đúng
B. Sai

Câu trả lời: B

6. **Thanh trạng thái có thể chứa những các phần tử thuộc loại nào?**

- A. Văn bản.
B. Hình ảnh.
C. Các điều khiển khác như Button, ProgressBar,..
D. Cả ba loại trên.

Câu trả lời: D

4. Tài liệu tham khảo

1. Windows Presentation Foundation, URL: <http://msdn.microsoft.com/en-us/library/ms754130.aspx>.

2. MSDN – Context Menu, URL: <http://msdn.microsoft.com/en-us/library/ms743670.aspx>.
3. MSDN – Context Menu sample, URL: [http://msdn.microsoft.com/en-us/library/ms771470\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms771470(VS.85).aspx) .
4. MSDN – StatusBar, URL: <http://msdn.microsoft.com/en-us/library/ms752075.aspx> .
5. MSDN – StatusBar sample, URL: <http://msdn.microsoft.com/en-us/library/ms771396.aspx>.
6. The Perfect WPF StatusBar, URL: <http://kentb.blogspot.com/2007/10/perfect-wpf-statusbar.html>.