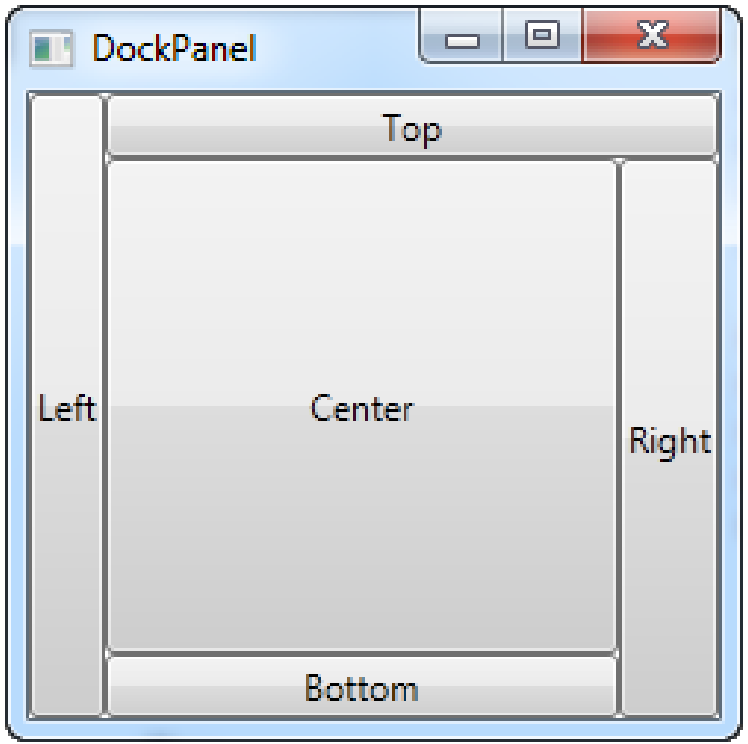
DockPanel
Menu
Tree View
StatusBar

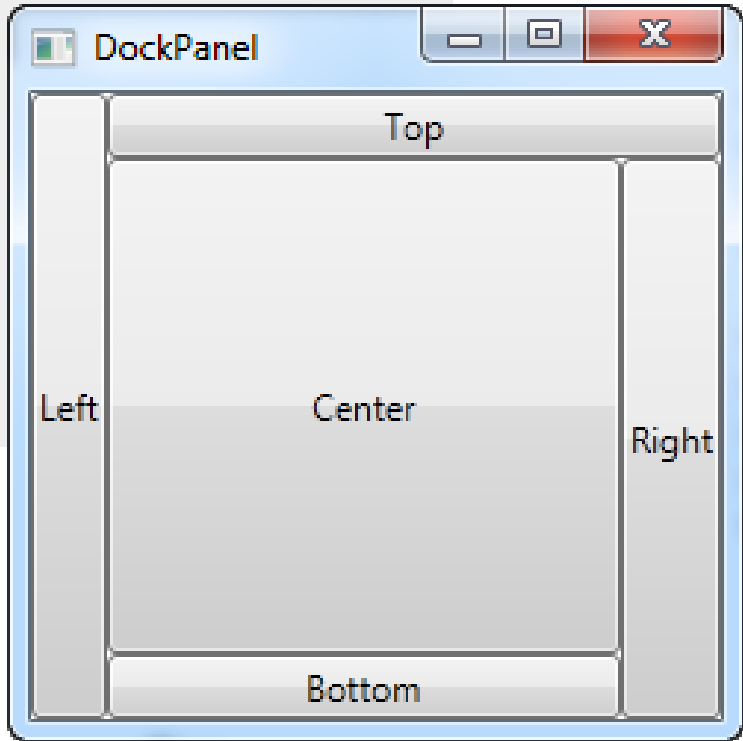
Dock

Il DockPanel permette di ancorare facilmente il contenuto ai quattro lati (sopra, sotto, sinistra e destra). Questo lo rende un'ottima scelta in molte situazioni, dove si vuole dividere la finestra in aree specifiche, specialmente perché di default, l'ultimo elemento all'interno del DockPanel, a meno che questa funzionalità non sia espressamente disabilitata, occuperà automaticamente tutto lo spazio rimanente (il centro).



Esempio

```
<Window x:Class="WpfTutorialSamples.Panels.DockPanel"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="DockPanel" Height="250" Width="250">
    <DockPanel>
        <Button DockPanel.Dock="Left">Left</Button>
        <Button DockPanel.Dock="Top">Top</Button>
        <Button DockPanel.Dock="Right">Right</Button>
        <Button DockPanel.Dock="Bottom">Bottom</Button>
        <Button>Center</Button>
    </DockPanel>
</Window>
```

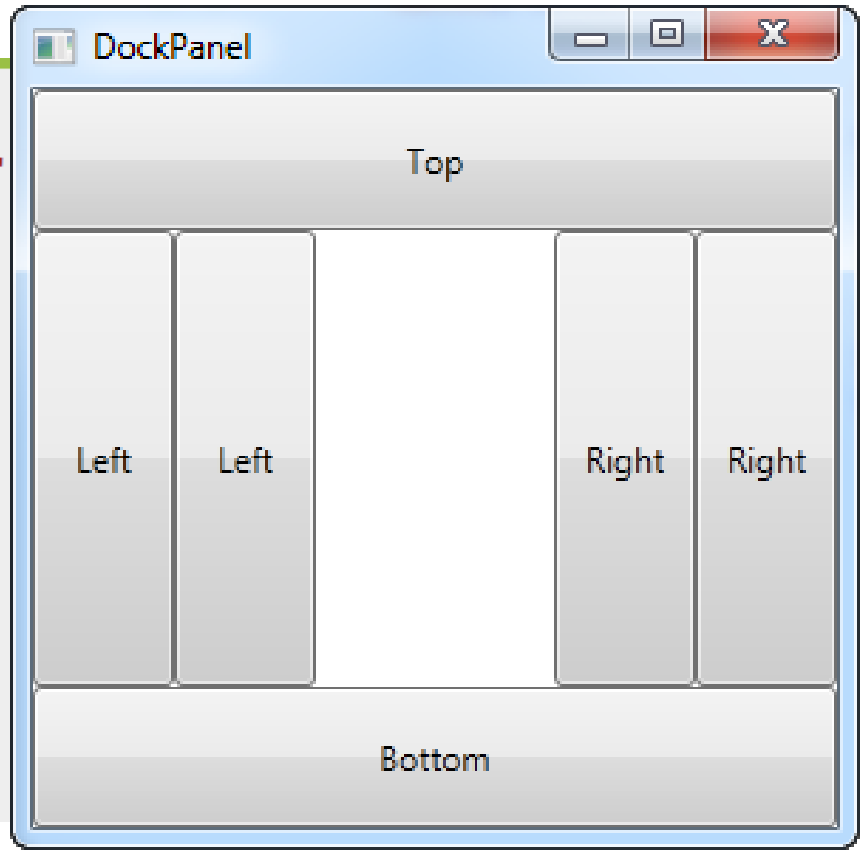


LastChildFill

Il comportamento predefinito è che l'ultimo figlio del DockPanel occupi il resto dello spazio, ma questo può essere disabilitato utilizzando LastChildFill.

Ecco un esempio in cui lo disabilitiamo e allo stesso tempo ancoriamo più di un controllo sullo stesso lato.

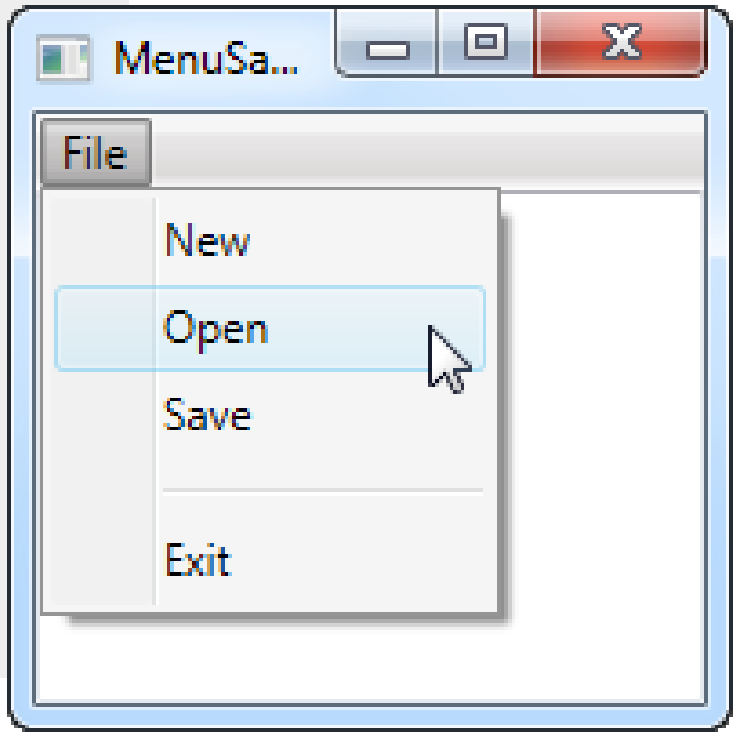
```
<Window x:Class="WpfTutorialSamples.Panels.DockPanel"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="DockPanel" Height="300" Width="300">
    <DockPanel LastChildFill="False">
        <Button DockPanel.Dock="Top" Height="50">Top</Button>
        <Button DockPanel.Dock="Bottom" Height="50">Bottom</Button>
        <Button DockPanel.Dock="Left" Width="50">Left</Button>
        <Button DockPanel.Dock="Left" Width="50">Left</Button>
        <Button DockPanel.Dock="Right" Width="50">Right</Button>
        <Button DockPanel.Dock="Right" Width="50">Right</Button>
    </DockPanel>
</Window>
```



Menu

Esempio

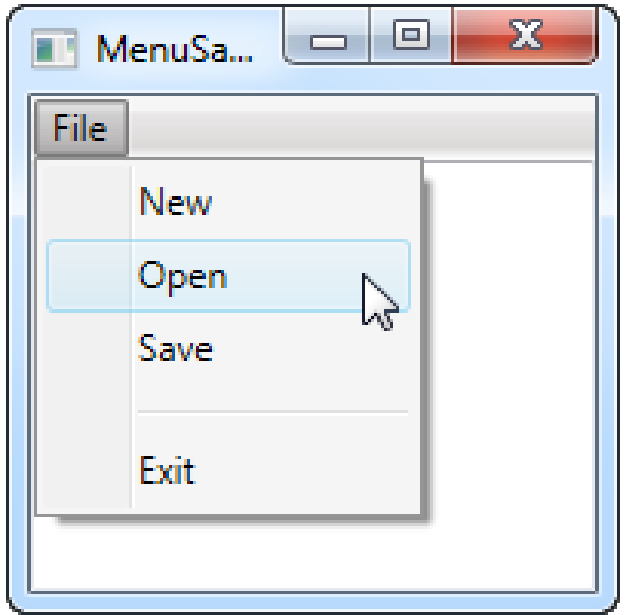
```
<Window x:Class="WpfTutorialSamples.Common_interface_controls.MenuSample"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MenuSample" Height="200" Width="200">
    <DockPanel>
        <Menu DockPanel.Dock="Top">
            <MenuItem Header="_File">
                <MenuItem Header="_New" />
                <MenuItem Header="_Open" />
                <MenuItem Header="_Save" />
                <Separator />
                <MenuItem Header="_Exit" />
            </MenuItem>
        </Menu>
        <TextBox AcceptsReturn="True" />
    </DockPanel>
</Window>
```



Menu

Uno degli elementi più comuni di un'applicazione Windows è il menu, a volte indicato come menu principale perché di solito ne esiste soltanto uno nell'applicazione.

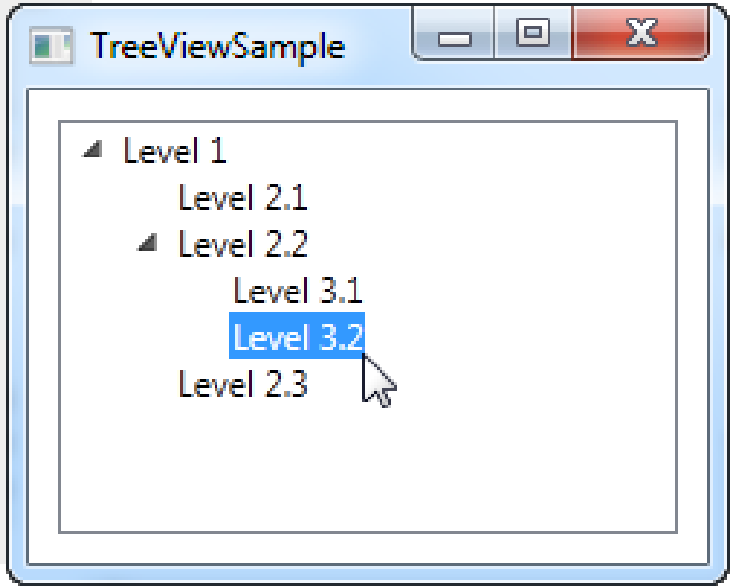
Il menu è un elemento pratico perché offre molte opzioni, utilizzando pochissimo spazio.



Tree View

TreeView può essere utilizzato in maniera molto semplice, aggiungendovi oggetti TreeViewItem, sia da Code-behind che semplicemente dichiarandoli direttamente nel vostro XAML.

```
<Grid Margin="10">
    <TreeView>
        <TreeViewItem Header="Level 1" IsExpanded="True">
            <TreeViewItem Header="Level 2.1" />
            <TreeViewItem Header="Level 2.2" IsExpanded="True">
                <TreeViewItem Header="Level 3.1" />
                <TreeViewItem Header="Level 3.2" />
            </TreeViewItem>
            <TreeViewItem Header="Level 2.3" />
        </TreeViewItem>
    </TreeView>
</Grid>
```



StatusBar

Con la parte superiore della finestra dell'applicazione solitamente occupata dal menu principale e/o dalle barre degli strumenti, descritte nei capitoli precedenti, la parte inferiore della finestra è solitamente la sede della barra di stato. La barra di stato viene utilizzata per mostrare varie informazioni sullo stato corrente dell'applicazione, come la posizione del cursore, il conteggio delle parole, l'avanzamento delle attività e così via. Fortunatamente per noi, WPF è dotato di un bel controllo StatusBar, che semplifica l'aggiunta della funzionalità della barra di stato alle tue applicazioni.

```
<StatusBar DockPanel.Dock="Bottom">
    <StatusBarItem>
        <Label Content="Sono una barra di stato..."/>
    </StatusBarItem>
</StatusBar>
```