

# Контейнеры компоновки



**Компоновка элементов управления** -  
это модель организации  
пользовательского интерфейса.

**Контейнер** – это элемент wprf, который  
может содержать другие элементы как  
дочерние

В **.NET 1.0** технология **Windows Forms** предоставляла весьма примитивную систему компоновки. Элементы управления были фиксированы на месте по жестко закодированным координатам.

Единственными средствами, которые позволяли элементам управления перемещаться и изменять свои размеры вместе с их контейнером, были:

- **привязка (anchoring)**
- **стыковка (docking)**

В **.NET 2.0** пробел в **Windows Forms** был восполнен, благодаря двум новым контейнерам компоновки:

- **FlowLayoutPanel**
- **TableLayoutPanel.**

**Элементы управления не должны иметь явно установленных размеров.** Вместо этого они растут, чтобы вместить их содержимое. Можно ограничить элементы управления приемлемыми размерами, устанавливая максимальное и минимальное их значение.

**Элементы не указывают свою позицию в экранных координатах.** Вместо этого они упорядочиваются своим контейнером на основе размера, порядка и (необязательно) другой информации, специфичной для контейнера компоновки. Если необходимо добавить пробел между элементами, то используется для этого свойство Margin.

**Контейнеры компоновки "разделяют" доступное пространство между своими дочерними элементами.** Они пытаются предоставить каждому элементу его предпочтительный размер (на основе его содержимого), если позволяет свободное пространство. Они могут также выделять дополнительное пространство одному или более дочерним элементам.

**Контейнеры компоновки могут быть вложенными.** Типичный пользовательский интерфейс начинается с **Grid** - наиболее развитого контейнера, и содержит другие контейнеры компоновки, которые организуют меньшие группы элементов.

## Процесс компоновки

Компоновка WPF происходит в два этапа:

- **этап измерения**
- **этап расстановки.**

На этапе измерения контейнер выполняет проход в цикле по дочерним элементам и опрашивает их **предпочтительные** размеры. На этапе расстановки контейнер помещает дочерние элементы **в соответствующие позиции**.

Все контейнеры компоновки WPF являются панелями, которые унаследованы от абстрактного класса **System.Windows.Controls**

|                    |  |
|--------------------|--|
| <b>Background</b>  | Получает или задает объект <b>Brush</b> , который используется для заполнения области между границами объекта <b>Panel</b> .   |
| <b>Children</b>    | Получает коллекцию <b>UIElementCollection</b> дочерних элементов этого объекта <b>Panel</b> .  |
| <b>IsItemsHost</b> | Получает или задает значение, которое указывает, что этот объект <b>Panel</b> является контейнером для элементов пользовательского интерфейса, созданных объектом <b>ItemsControl</b> (такие как узлы из <b>TreeView</b> или элементы списка <b>ListBox</b> ). |

|                    |   |
|--------------------|---|
| <b>StackPanel</b>  | Размещает элементы в горизонтальный или вертикальный стек. Этот контейнер компоновки обычно используется в небольших секциях крупного более сложного окна.  |
| <b>WrapPanel</b>   | Размещает элементы в сериях строк с переносом. В горизонтальной ориентации <b>WrapPanel</b> располагает элементы в строке слева направо, затем переходит к следующей строке. В вертикальной ориентации <b>WrapPanel</b> располагает элементы сверху вниз, используя дополнительные колонки для дополнения оставшихся элементов. |
| <b>DockPanel</b>   | Выравнивает элементы по краю контейнера.  |
| <b>Grid</b>        | Выстраивает элементы в строки и колонки невидимой таблицы. Это один из наиболее гибких и широко используемых контейнеров компоновки.  |
| <b>Canvas</b>      | Позволяет элементам позиционироваться абсолютно - по фиксированным координатам. Этот контейнер компоновки более всего похож на компоновщик <b>Windows Forms</b> , но не предусматривает средств привязки и стыковки. В результате это неподходящий выбор для окон переменного размера.  |
| <b>UniformGrid</b> | Помещает элементы в невидимую таблицу, устанавливая одинаковый размер для всех ячеек.   |

# Stack Panel



```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="300">
  <StackPanel>
    <Button>Button</Button>
    <Button>Welcome</Button>
    <Button>HelloWorld</Button>
    <Button>Button</Button>
    <Button>ButtonZLO</Button>
  </StackPanel>
</Window>
```



## Свойства компоновки

|                            |   |
|----------------------------|---|
| <b>HorizontalAlignment</b> | Определяет позиционирование дочернего элемента внутри контейнера компоновки, когда имеется дополнительное пространство по горизонтали. <b>Доступные значения: Center, Left, Right или Stretch</b>   |
| <b>VerticalAlignment</b>   | Определяет позиционирование дочернего элемента внутри контейнера компоновки, когда имеется дополнительное пространство по вертикали. <b>Доступные значения: Center, Top, Bottom или Stretch</b>   |
| <b>Margin</b>              | Добавляет некоторое пространство вокруг элемента. Свойство Margin — это экземпляр структуры <b>System.Windows.Thickness</b> , с отдельными компонентами для верхней, нижней, левой и правой граней  |
| <b>MinWidth MinHeight</b>  | Устанавливает минимальные размеры элемента. Если элемент слишком велик, чтобы поместиться в его контейнер компоновки, он будет усечен.  |
| <b>MaxWidth MaxHeight</b>  | Устанавливает максимальные размеры элемента. Если контейнер имеет свободное пространство, элемент не будет увеличен сверх указанных пределов, даже если свойства HorizontalAlignment и VerticalAlignment установлены в Stretch                                  |
| <b>Width и Height</b>      | Явно устанавливают размеры элемента. Эта установка переопределяет значение Stretch для свойств HorizontalAlignment и VerticalAlignment. Однако данный размер не будет установлен, если выходит за пределы, заданные в MinWidth, MinHeight, MaxWidth и MaxHeight |

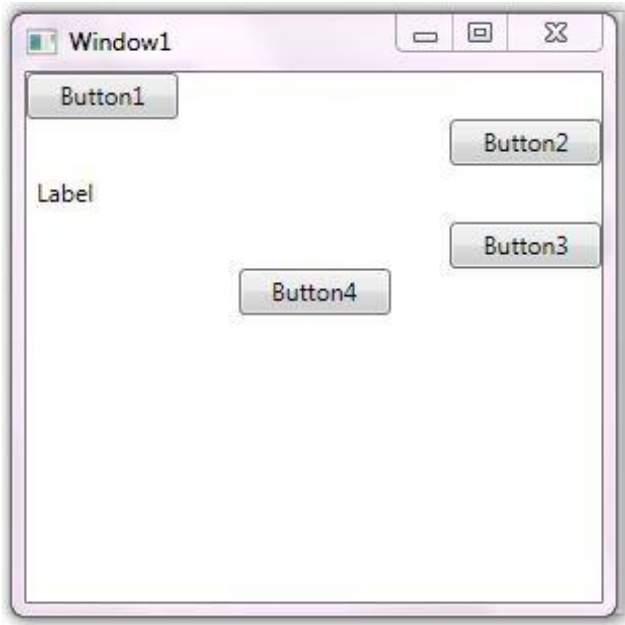
Свойства унаследованы от **FrameworkElement** и поддерживаются всеми графическими элементами (виджетами), которые можно использовать в окне WPF.

# Stack Panel



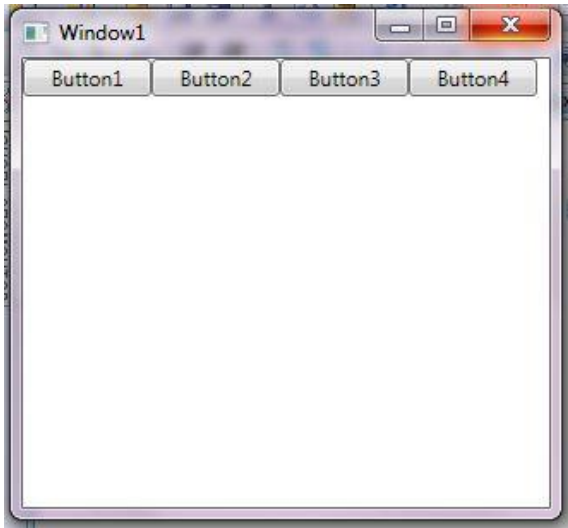
```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="300">
  <StackPanel>
    <Button Margin="2,2,2,2">Button</Button>
    <Button Margin="2,2,2,2">Welcome</Button>
    <Button Margin="2,2,2,2">HelloWorld</Button>
    <Button Margin="2,2,2,2">Button</Button>
    <Button Margin="2,2,2,2">ButtonZLO</Button>
  </StackPanel>
</Window>
```

# Stack Panel



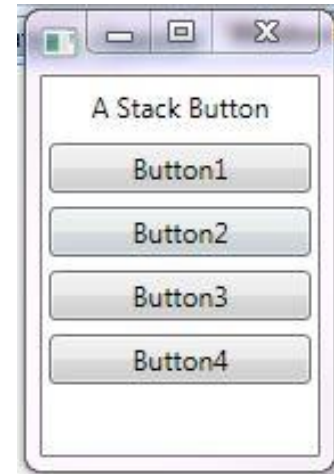
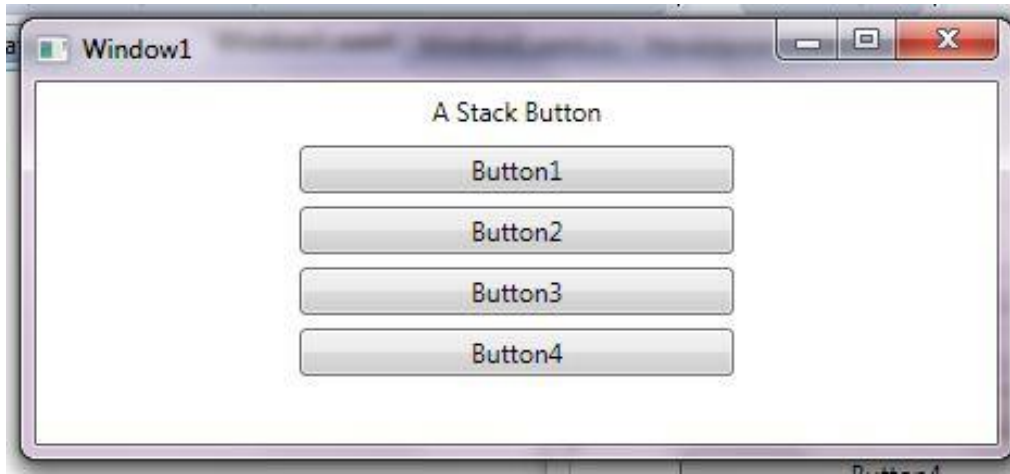
```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="300">
  <StackPanel>
    <Button Height="23" HorizontalAlignment="Left" Width="75">Button1</Button>
    <Button Height="23" HorizontalAlignment="Right" Width="75">Button2</Button>
    <Label Height="28" HorizontalAlignment="Left" Width="120">Label</Label>
    <Button Height="23" HorizontalAlignment="Right" Width="75">Button3</Button>
    <Button Height="23" HorizontalAlignment="Center" Width="75">Button4</Button>
  </StackPanel>
</Window>
```

# Stack Panel



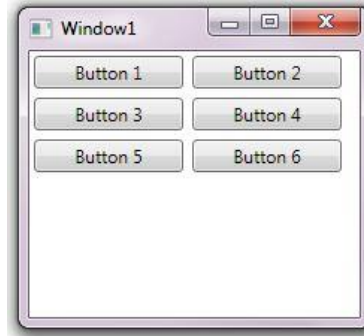
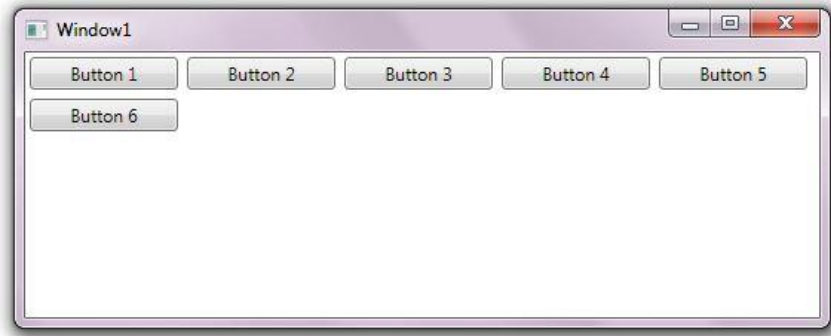
```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="323">
  <StackPanel Orientation="Horizontal">
    <Button VerticalAlignment="Top" Width="75">Button1</Button>
    <Button VerticalAlignment="Top" Width="75">Button2</Button>
    <Button VerticalAlignment="Top" Width="75">Button3</Button>
    <Button VerticalAlignment="Top" Width="75">Button4</Button>
  </StackPanel>
</Window>
```

# Stack Panel



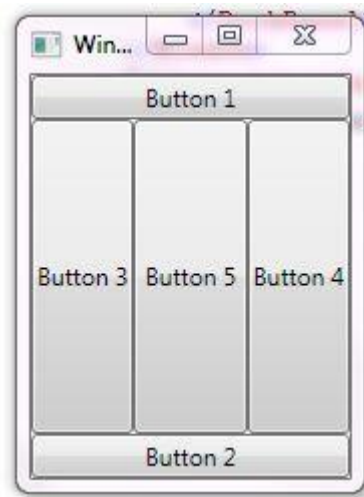
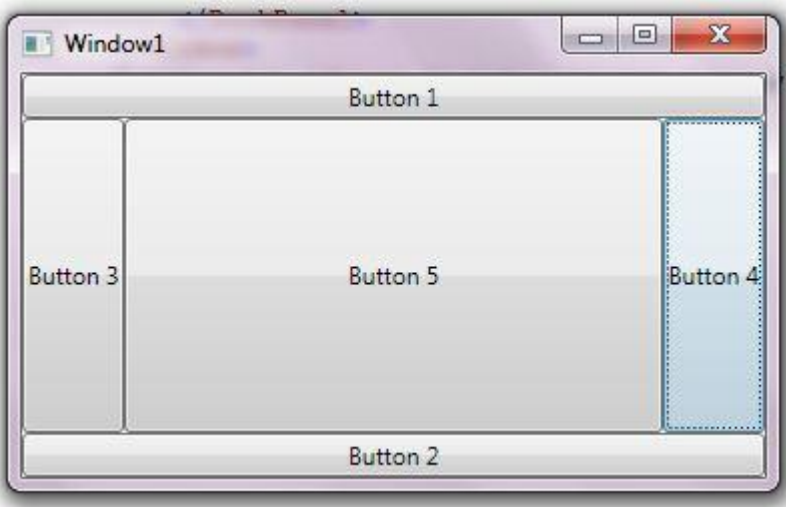
```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="323">
  <StackPanel Orientation="Vertical">
    <Label HorizontalAlignment="Center">A Stack Button</Label>
    <Button Margin="3" MinWidth="100" MaxWidth="200">Button1</Button>
    <Button Margin="3" MinWidth="100" MaxWidth="200">Button2</Button>
    <Button Margin="3" MinWidth="100" MaxWidth="200">Button3</Button>
    <Button Margin="3" MinWidth="100" MaxWidth="200">Button4</Button>
  </StackPanel>
</Window>
```

# Wrap Panel

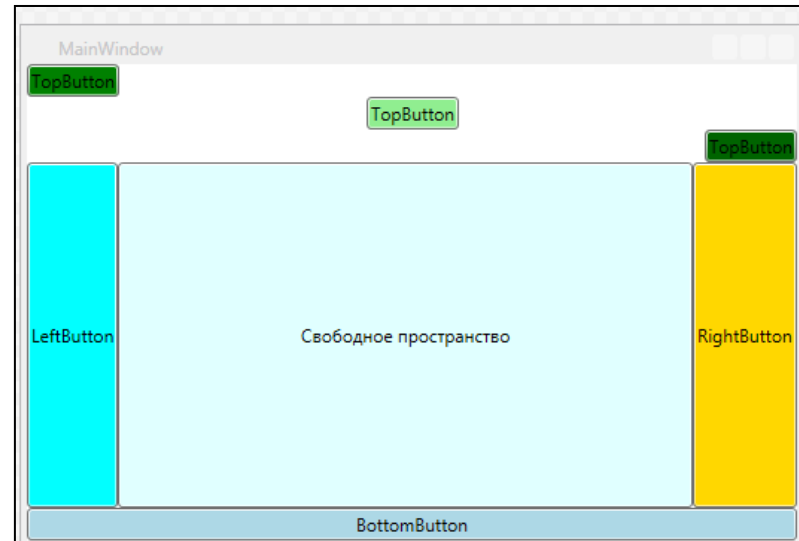
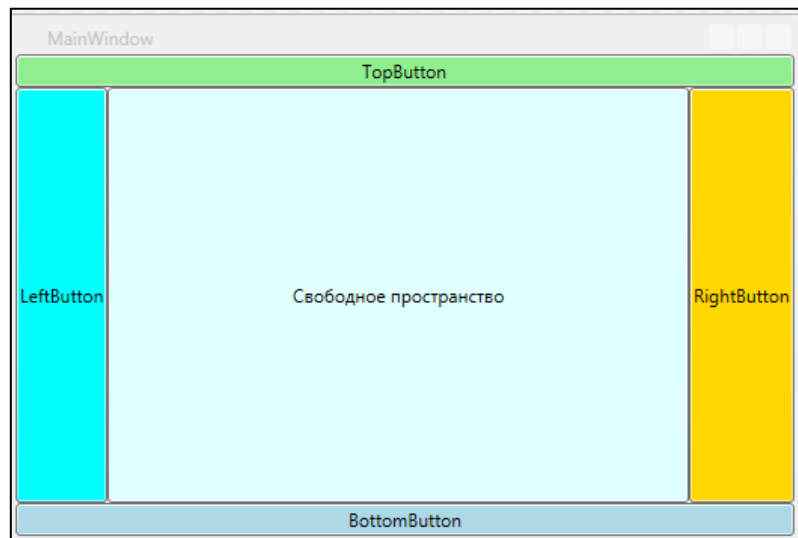


```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="323">
  <WrapPanel>
    <Button Margin="3" MinWidth="100">Button 1</Button>
    <Button Margin="3" MinWidth="100">Button 2</Button>
    <Button Margin="3" MinWidth="100">Button 3</Button>
    <Button Margin="3" MinWidth="100">Button 4</Button>
    <Button Margin="3" MinWidth="100">Button 5</Button>
    <Button Margin="3" MinWidth="100">Button 6</Button>
  </WrapPanel>
</Window>
```

# Dock Panel



```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="323">
  <DockPanel>
    <Button DockPanel.Dock="Top" >Button 1</Button>
    <Button DockPanel.Dock="Bottom">Button 2</Button>
    <Button DockPanel.Dock="Left">Button 3</Button>
    <Button DockPanel.Dock="Right">Button 4</Button>
    <Button >Button 5</Button>
  </DockPanel>
</Window>
```



### <DockPanel>

```
<Button DockPanel.Dock="Top" Background="LightGreen">TopButton</Button>
<Button DockPanel.Dock="Bottom" Background="LightBlue">BottomButton</Button>
<Button DockPanel.Dock="Left" Background="Aqua">LeftButton</Button>
<Button DockPanel.Dock="Right" Background="Gold">RightButton</Button>
<Button Background="LightCyan">Свободное пространство</Button>
```

### </DockPanel>

### <DockPanel>

```
<Button DockPanel.Dock="Top" Background="Green" HorizontalAlignment="Left">TopButton</Button>
<Button DockPanel.Dock="Top" Background="LightGreen" HorizontalAlignment="Center">TopButton</Button>
<Button DockPanel.Dock="Top" Background="DarkGreen" HorizontalAlignment="Right">TopButton</Button>
<Button DockPanel.Dock="Bottom" Background="LightBlue">BottomButton</Button>
<Button DockPanel.Dock="Left" Background="Aqua">LeftButton</Button>
<Button DockPanel.Dock="Right" Background="Gold">RightButton</Button>
<Button Background="LightCyan">Свободное пространство</Button>
```

### </DockPanel>



# Grid



```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="323">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition/>
      <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
    </Grid.RowDefinitions>

    <Button Margin="3" Grid.Column="0" Grid.Row="0">Button 1</Button>
    <Button Margin="3" Grid.Column="1" Grid.Row="0">Button 2</Button>
    <Button Margin="3" Grid.Column="0" Grid.Row="1">Button 3</Button>
    <Button Margin="3" Grid.Column="1" Grid.Row="1">Button 4</Button>
    <Button Margin="3" Grid.Column="0" Grid.Row="2">Button 5</Button>
    <Button Margin="3" Grid.Column="1" Grid.Row="2">Button 6</Button>
  </Grid>
</Window>
```



# Grid

## Статический размер:

```
<Grid.ColumnDefinitions>
  <ColumnDefinition Width="70"></ColumnDefinition>
  <ColumnDefinition></ColumnDefinition>
  <ColumnDefinition Width="70"></ColumnDefinition>
</Grid.ColumnDefinitions>
```



## Автоматический размер::

```
<Grid.ColumnDefinitions>
  <ColumnDefinition Width="auto"></ColumnDefinition>
  <ColumnDefinition></ColumnDefinition>
  <ColumnDefinition Width="auto"></ColumnDefinition>
</Grid.ColumnDefinitions>
```



## Пропорциональный размер:

```
<Grid.ColumnDefinitions>
  <ColumnDefinition Width="*"></ColumnDefinition>
  <ColumnDefinition Width="3*"></ColumnDefinition>
  <ColumnDefinition Width="*"></ColumnDefinition>
</Grid.ColumnDefinitions>
```



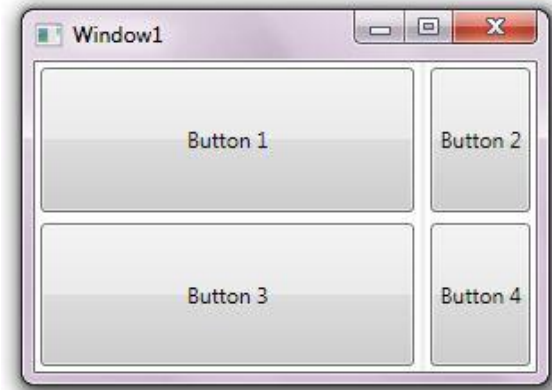
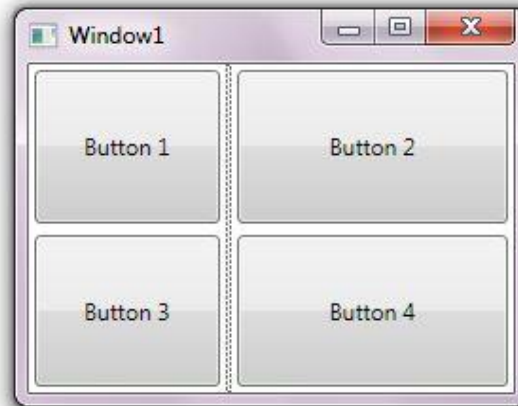
# Grid

```
<Window x:Class="WpfApplication2.Window1"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Window1" Height="300" Width="323">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="*"></ColumnDefinition>
      <ColumnDefinition Width="*"></ColumnDefinition>
      <ColumnDefinition Width="*"></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
      <RowDefinition Height="*"></RowDefinition>
      <RowDefinition Height="*"></RowDefinition>
      <RowDefinition Height="*"></RowDefinition>
    </Grid.RowDefinitions>

    <Button Margin="3" Grid.Column="0" Grid.Row="0">Button 1</Button>
    <Button Margin="3" Grid.Column="1" Grid.Row="0">Button 2</Button>
    <Button Margin="3" Grid.Column="2" Grid.Row="0">Button 3</Button>
    <Button Margin="3" Grid.Column="0" Grid.Row="1">Button 4</Button>
    <Button Margin="3" Grid.Column="1" Grid.Row="1" Grid.ColumnSpan="2">Button
5</Button>
    <Button Margin="3" Grid.Column="0" Grid.Row="2">Button 7</Button>
    <Button Margin="3" Grid.Column="1" Grid.Row="2">Button 8</Button>
    <Button Margin="3" Grid.Column="2" Grid.Row="2">Button 9</Button>
  </Grid>
</Window>
```



# Grid



**<Grid>**

**<Grid.ColumnDefinitions>**

**<ColumnDefinition Width="\*"></ColumnDefinition>**

**<ColumnDefinition Width="auto"></ColumnDefinition>**

**<ColumnDefinition Width="\*"></ColumnDefinition>**

**</Grid.ColumnDefinitions>**

**<Grid.RowDefinitions>**

**<RowDefinition Height="\*"></RowDefinition>**

**<RowDefinition Height="\*"></RowDefinition>**

**</Grid.RowDefinitions>**

**<Button Margin="3" Grid.Column="0" Grid.Row="0">Button 1</Button>**

**<Button Margin="3" Grid.Column="2" Grid.Row="0">Button 2</Button>**

**<Button Margin="3" Grid.Column="0" Grid.Row="1">Button 3</Button>**

**<Button Margin="3" Grid.Column="2" Grid.Row="1">Button 4</Button>**

**<GridSplitter Grid.Row="0" Grid.Column="1" Grid.RowSpan="2"**  
**Width="3" HorizontalAlignment="Center">**

**</GridSplitter>**

**</Grid>**

