

Layout

Introduction

- ❖ Using the layout model, we can organize our content in a set of different layout containers. Each container implements its own layout logic.
- ❖ There are many containers we can choose from. We can even develop our own.

Introduction

- ❖ The Silverlight window can hold one single element. In order to fit in more than one element we will need to place them within the single main one.

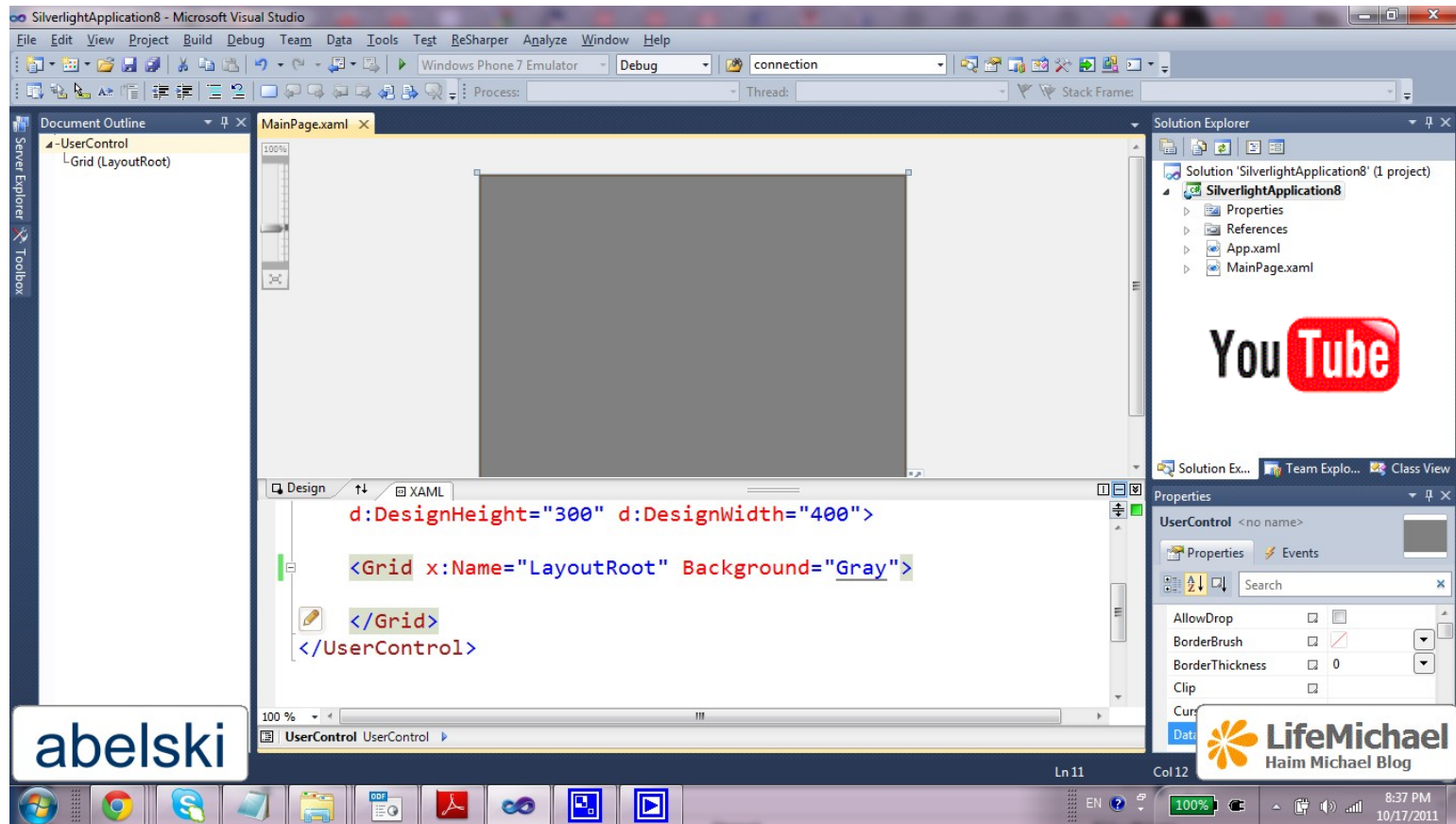
The Panel Class

- ❖ The Silverlight layout containers are panels derived from the `System.Windows.Controls.Panel` abstract class.
- ❖ Two important properties this class adds are `Background` and `Children`. The later is the collection of elements the container holds.

Background

- ❖ The `Background` property holds a `Brush` object. This way we can have sophisticated backgrounds. There is no need to limit ourselves with a solid simple background.

Background



Border

- ❖ The `Border` property describes the border. This class is responsible for adding a background or a border for the nested component it refers.
- ❖ The properties this type defines include: `Background`, `BorderBrush`, `BorderThickness`, `CornerRadius` **and** `Padding`.

Border

The screenshot displays the Microsoft Visual Studio IDE for a Silverlight application. The main design view shows a grid layout with two rows and two columns. The top-left cell contains a grey rectangle labeled "Top Left", and the bottom-left cell contains a grey rectangle labeled "Bottom Left". The right column contains a YouTube logo. A blue border is applied to the top-left cell. The XAML code in the bottom pane is as follows:

```
<RowDefinition></RowDefinition>
<RowDefinition></RowDefinition>
</Grid.RowDefinitions>
<Border BorderBrush="Aqua" BorderThickness="5">
  <Button Margin="8" Grid.Row="1" Content="Top Left"></Button>
</Border>
<Button Margin="8" Grid.Row="2" Content="Bottom Left"></Button>
</Grid>
```

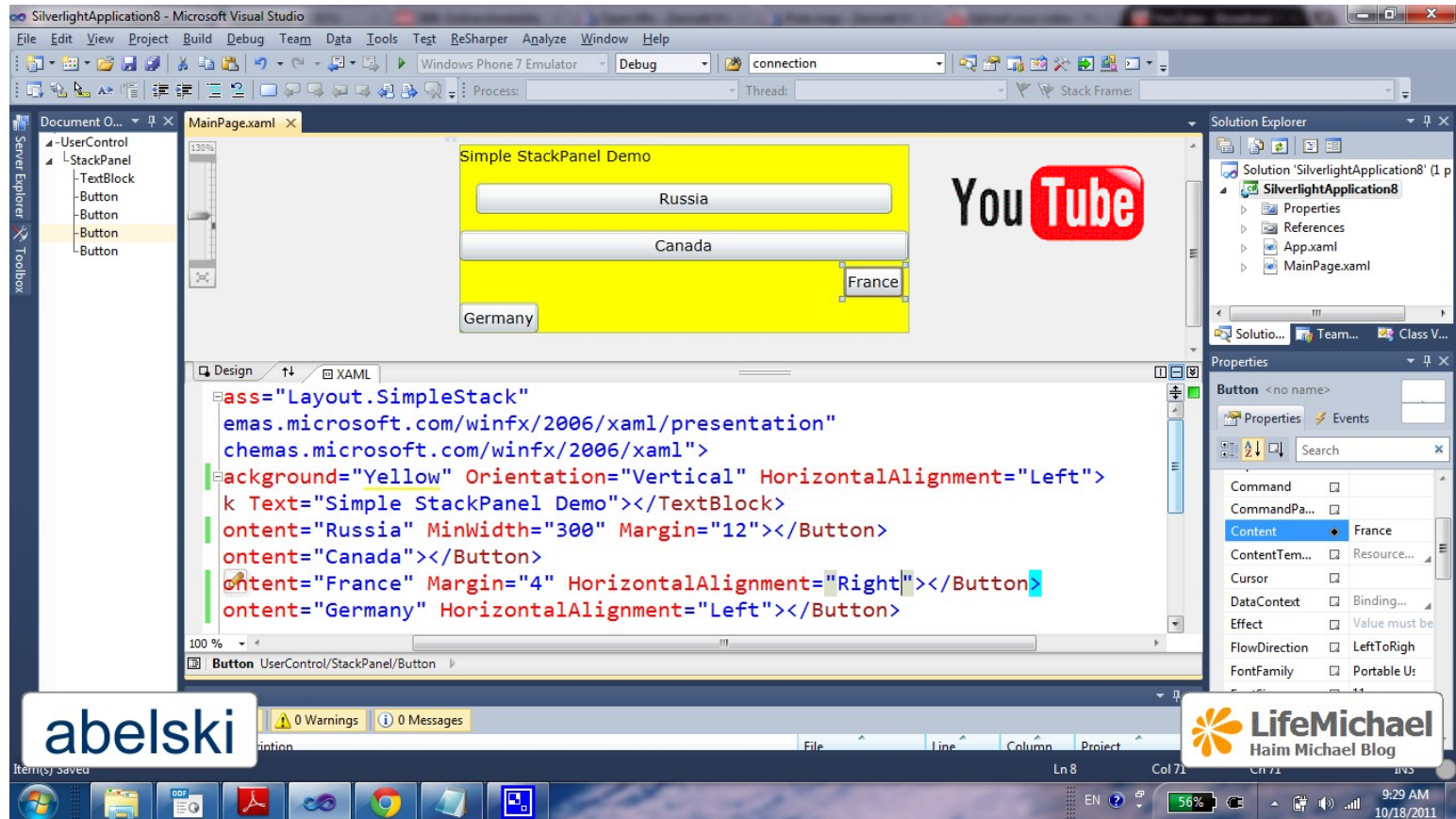
The Solution Explorer on the right shows the project structure for "SilverlightApplication8", including "App.xaml" and "MainPage.xaml". The Properties window shows the "Grid" control with various properties like "AllowDrop", "Background", "Clip", "Cursor", "DataContent", "Effect", "FlowDirection", "Grid.Column", and "Grid.Column...".

Watermarks are present at the bottom of the screenshot: "abelski" on the left and "LifeMichael Haim Michael Blog" on the right.

The StackPanel Layout

- ❖ This layout stacks its children in a single row or a single column.

The StackPanel Layout



The screenshot displays the Microsoft Visual Studio interface for a Silverlight application named 'SilverlightApplication8'. The main window shows a design view of a 'Simple StackPanel Demo' with a yellow background. The demo contains four buttons stacked vertically: 'Russia', 'Canada', 'France', and 'Germany'. A 'YouTube' logo is also visible on the right side of the design view.

The XAML code in the Design view is as follows:

```
<StackPanel xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
            background="Yellow" Orientation="Vertical" HorizontalAlignment="Left">
    <TextBlock Text="Simple StackPanel Demo" />
    <Button Content="Russia" MinWidth="300" Margin="12" />
    <Button Content="Canada" />
    <Button Content="France" Margin="4" HorizontalAlignment="Right" />
    <Button Content="Germany" HorizontalAlignment="Left" />
</StackPanel>
```

The Properties window on the right shows the selected 'Button' element with its 'Content' property set to 'France'. The Solution Explorer on the right shows the project structure, including 'MainPage.xaml'.

Watermarks for 'abelski' and 'LifeMichael Haim Michael Blog' are visible at the bottom of the screenshot.

The WrapPanel & DockPanel Layout

- ❖ In order to use these two layouts we first need to add a reference to the assembly file that includes their definition. Its name is `System.Windows.Controls.Toolkit.dll`.
- ❖ This assembly is available as part of the Silverlight toolkit. You can download it at <http://silverlight.codeplex.com>.
- ❖ Once we install the Silverlight tool kit we should map to the namespace so it will be available.



The WrapPanel & DockPanel Layout

- ❖ The `WrapPanel` lays out the controls in the available space, one line (or column) at a time. The default orientation is horizontal. The controls are arranged left to right and then on the subsequent rows.
- ❖ The `DockPanel` stretches the controls against the edge we set.

The Grid Layout

- ❖ This is the default layout. When creating a new XAML document the visual studio automatically adds the `Grid` tags as the first-level container, nested within the `UserControl` root element.
- ❖ The Grid layout places its component into invisible grid of rows and columns.

The Grid Layout

- ❖ Although each cell can include more than one element as multiple elements will overlap with each other, we will usually place one element only.

The Grid Layout

The screenshot displays the Visual Studio IDE with the following components:

- Document Window:** Shows the XAML code for `MainPage.xaml`. The code defines a `Grid` with two rows and one column. The first row contains a `Image` (YouTube logo) and the second row contains a `TextBlock` with the text "Simple Demo for Nested Layouts".
- Server Explorer:** Shows the project structure for `SilverlightApplication11`, including `UserControl`, `Grid`, `RowDefinitions`, `RowDefinition`, `ColumnDefinition`, `TextBlock`, and `StackPanel`.
- Solution Explorer:** Shows the project structure for `SilverlightApplication11`, including `Properties`, `References`, `App.xaml`, and `MainPage.xaml`.
- Properties Window:** Shows the properties for the selected `RowDefinition`, including `Height`, `MaxHeight`, and `MinHeight`.
- Error List:** Shows 0 Errors, 0 Warnings, and 0 Messages.
- Taskbar:** Shows the Windows taskbar with various application icons and the system tray.

Watermarks for **abelski** and **LifeMichael Haim Michael Blog** are visible in the bottom right corner of the screenshot.

The Canvas Layout

- ❖ Using `Canvas` we can specify the exact position of each control.

The Canvas Layout

The screenshot displays the Microsoft Visual Studio IDE for a Silverlight application named "SilverlightApplication11". The main window shows the XAML code for "MainPage.xaml" in Design view. The code defines a UserControl with a Canvas background and two buttons. The design view shows a white canvas with a YouTube logo. The Solution Explorer on the right shows the project structure, and the Properties window shows the properties of the selected Button. The Error List at the bottom shows 0 errors, 0 warnings, and 0 messages.

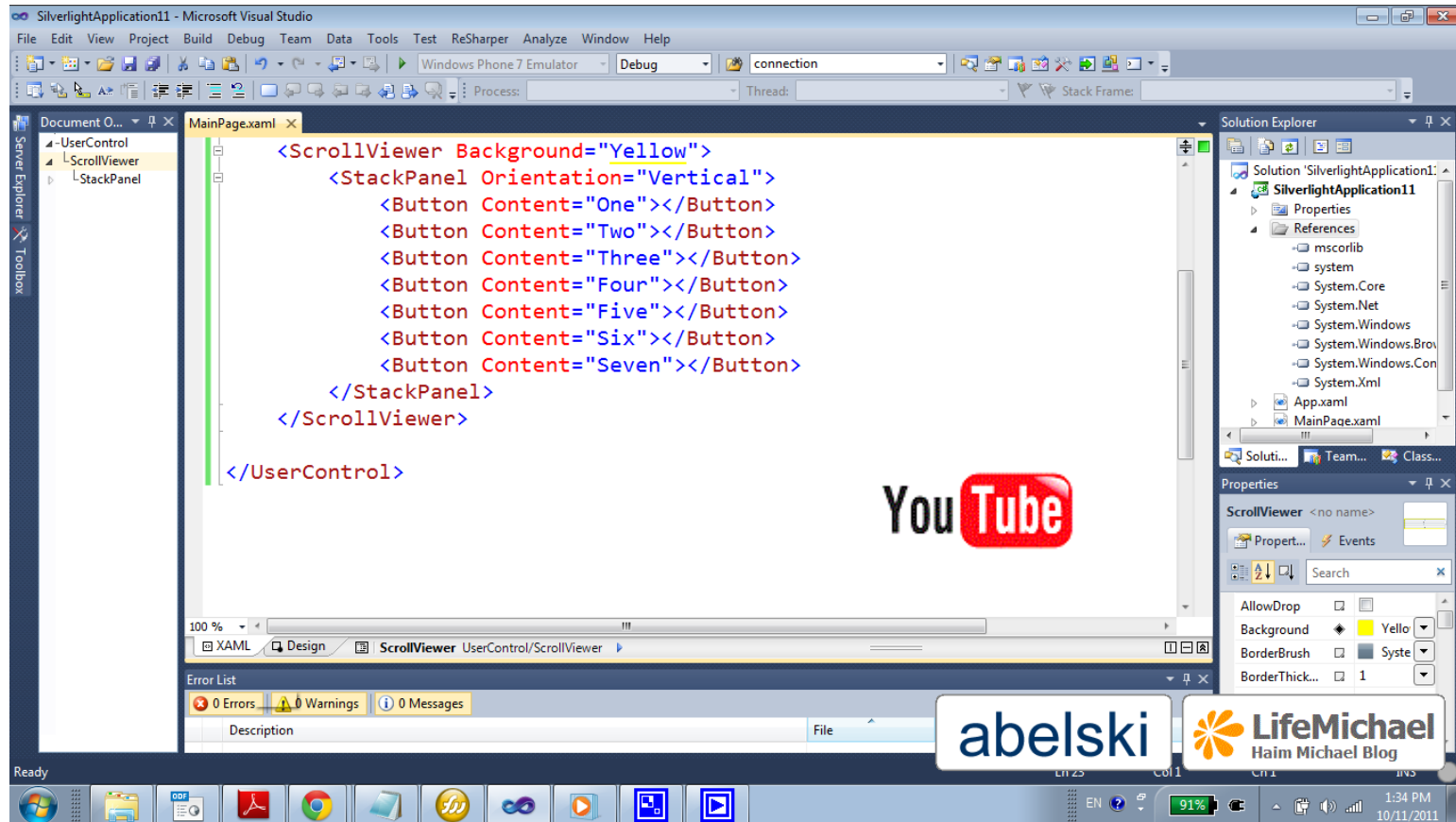
```
<UserControl x:Class="SilverlightApplication11.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  d:DesignHeight="300" d:DesignWidth="400">
  <Canvas Background="White">
    <Button Canvas.Left="12" Canvas.Top="52" Content="(12,52)"></Button>
    <Button Canvas.Left="140" Canvas.Top="60" Content="(140,60)"></Button>
  </Canvas>
</UserControl>
```

abelski LifeMichael Haim Michael Blog

Scrolling

- ❖ Placing the layout container within a `ScrollView` we will get scrollbars for scrolling over the layout.

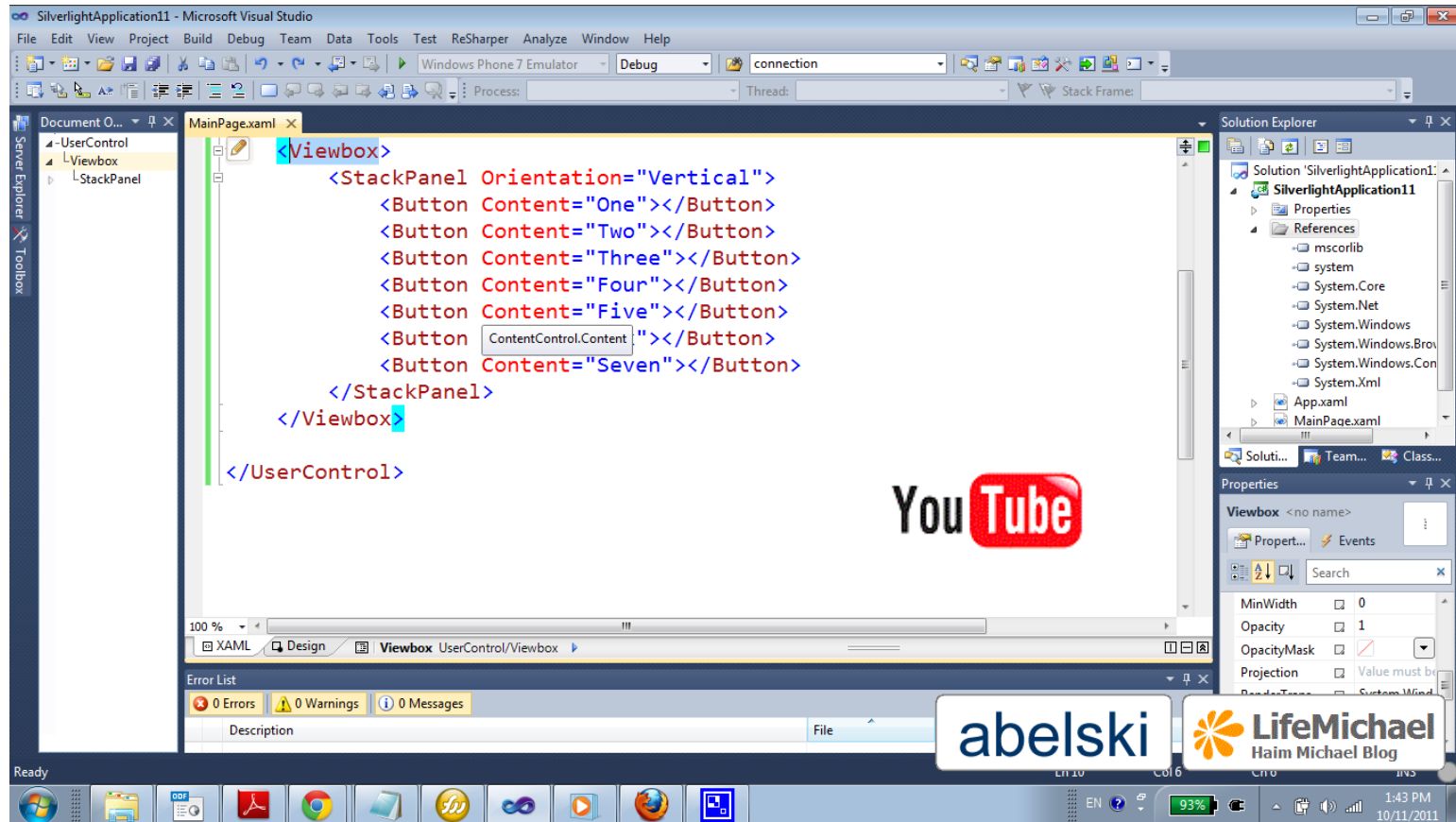
Scrolling



Scaling

- ❖ Placing the layout container within a `ViewBox` will allow the user to scale the view he gets.

Scaling



Full Screen Mode

- ❖ Silverlight applications have the capability to enter into a full-screen mode. They can do so in a response for a user event only.

```
private void Bt_Click(object sender, RoutedEventArgs e)
{
    Application.Current.Host.Content.IsFullScreen = true;
}
```

Full Screen Mode

- ❖ While in full screen mode the keyboard access is limited and it is possible to respond the following keys only: Tab, Enter, Home, End, Page Up, Page Down, spacebar and the arrow keys.