

WINDOWSPROGRAMMERING LEKTION 1

Mahmud Al Hakim
mahmud@alhakim.se
www.alhakim.se

1

MÅL

Kursens övergripande mål

Efter avslutad kurs ska den studerande kunna tillverka grafiska applikationer i C# WPF/UWP, t. ex. enklare spel och användargränssnitt med XAML.

Kursens innehåll

Kursen kommer att behandla följande centrala områden:

- WPF/UWP
- Användargränssnitt med XAML
- Media
- Layout och kontrollor
- Databindning
- Dokumentmallar och stilmallar

2

BETYGSKRITERIER

För Godkänt (G) krävs att

Studenten ska självständigt kunna utveckla **enklare applikationer** med hjälp av grafiska användargränssnitt i XAML.

För Väl Godkänd (VG) krävs att

Studenten ska självständigt kunna utveckla mer avancerade applikationer med grafiska användargränssnitt i XAML och tillhörande **databaskopplingar**.

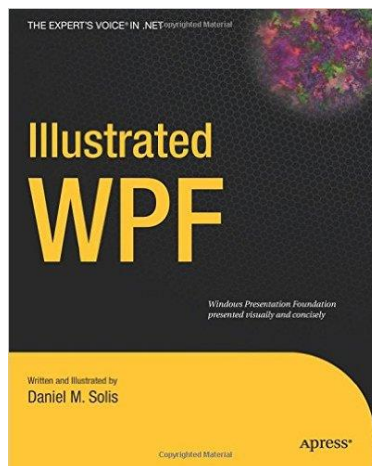
3

REKOMMENDERAD REFERENSLITTERATUR

Titel: Illustrated WPF

Författare: Daniel M. Solis

ISBN: 9781430219101



4

← → ↻ www.wpf-tutorial.com



TIPS

HTTP://WWW.WPF-TUTORIAL.COM

Home Contact Us Table of contents

The complete WPF tutorial

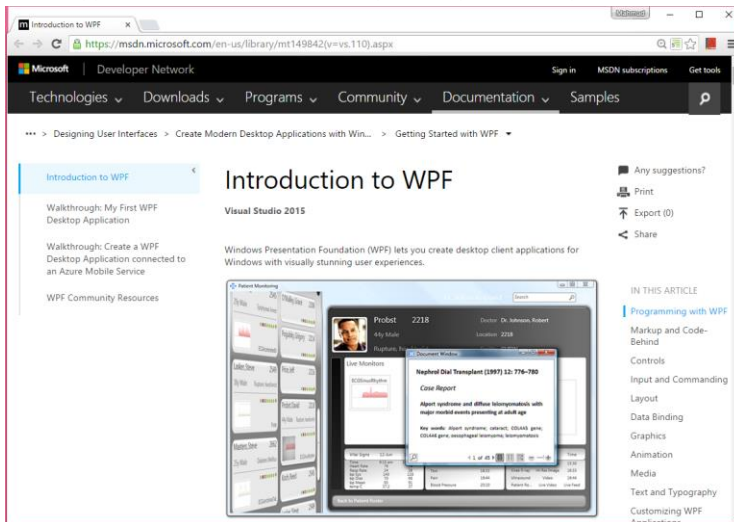
Welcome to this **WPF tutorial**, currently consisting of **103 articles**, where you'll learn to make your own applications using the WPF UI framework. If you're brand new to WPF, then we recommend that you start from the first chapter and then read your way through all of it.

Have a look at the Table of contents to the right, where all the chapters are listed and be sure to come back regularly, as we will keep adding new chapters to it. We hope that this tutorial will get you started properly on WPF.

5

INTRODUCTION TO WPF (VISUAL STUDIO 2015) PÅ MSDN

[HTTPS://MSDN.MICROSOFT.COM/EN-US/LIBRARY/MT149842\(V=VS.110\).ASPX](https://msdn.microsoft.com/en-us/library/mt149842(v=vs.110).aspx)



Introduction to WPF

Visual Studio 2015

Windows Presentation Foundation (WPF) lets you create desktop client applications for Windows with visually stunning user experiences.

IN THIS ARTICLE

- Programming with WPF
- Markup and Code-Behind
- Controls
- Input and Commanding
- Layout
- Data Binding
- Graphics
- Animation
- Media
- Text and Typography
- Customizing WPF Applications

6

VAD ÄR WPF?

WPF står för "Windows Presentation Foundation".

WPF är ett ramverk för utveckling av bättre **UI (User Interface)** för desktop-applikationer.

Huvudsyftet med WPF är att separera användargränssnitt (design) från logiken (koden).

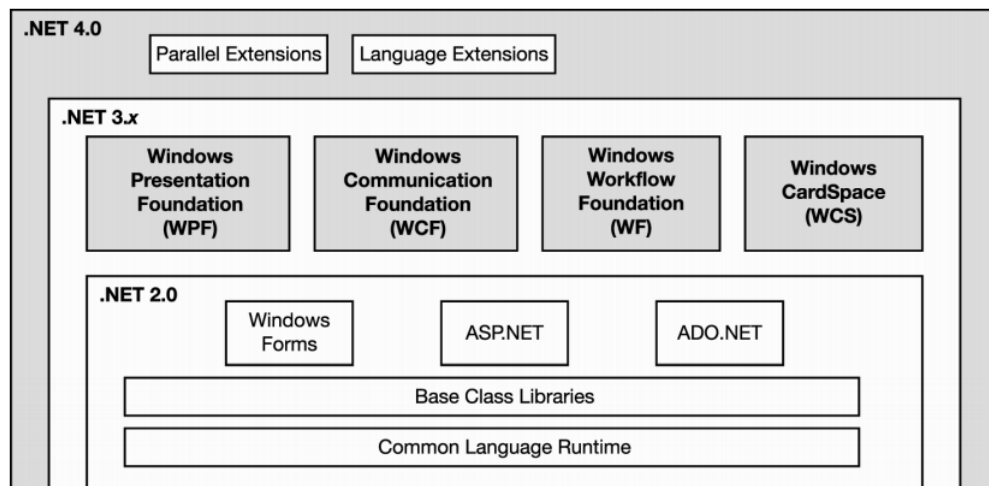
En designer ska kunna arbeta med utseendet, medan en utvecklare arbetar parallellt med logiken bakom.

WPF använder **XAML** för att skapa användargränssnittet.

Som programspråk används oftast **C#** men du kan använda valfri .Net-språk t.ex. Visual Basic.

7

WPF FINNS SEDAN .NET 3.X



Bildkälla: : Illustrated WPF, sidan 14

8

VAD ÄR XAML?

XAML står för "Extensible Application Markup Language".

XAML baseras på XML-syntax.

XAML används i bl.a. WPF och UWP för att beskriva och definiera användargränssnittet (UI) t.ex. fönster, dialogrutor, figurer, grafik, animeringar m.m.

9

VAD ÄR UWP?

UWP står för "Universal Windows Platform".

UWP är plattformshomogen (enhetlig) applikationsarkitektur som skapats av Microsoft och introducerades i Windows 10 .

Syftet med denna mjukvaruplattform är att bidra till att utveckla "universella applikationer" som körs på Windows 10.

UWP är en del av Windows 10 så, UWP-program körs inte på tidigare Windows-versioner.

Källa: https://en.wikipedia.org/wiki/Universal_Windows_Platform

10

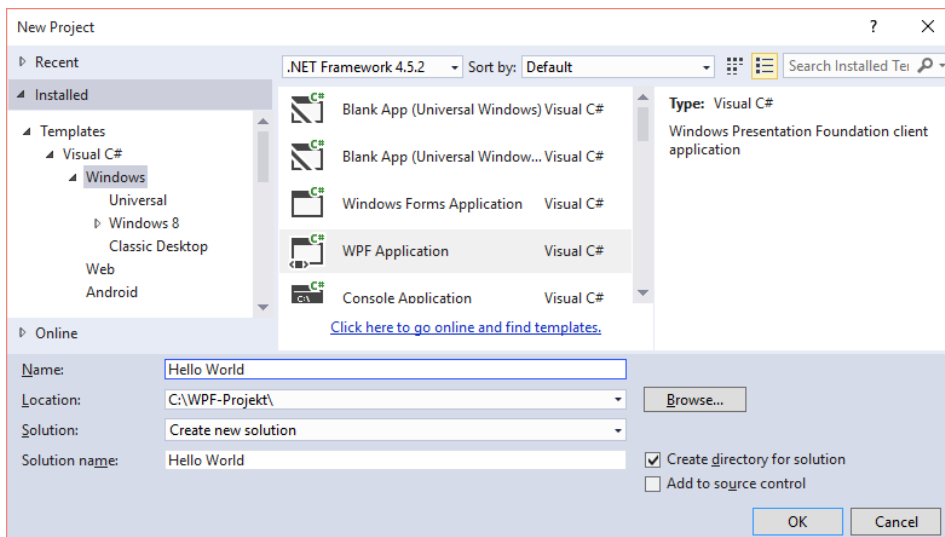
UNIVERSAL WINDOWS PLATFORM



Källa: <https://msdn.microsoft.com/windows/uwp/get-started/universal-application-platform-guide>

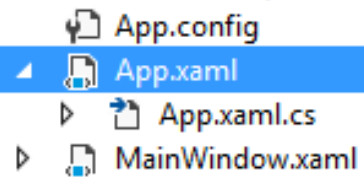
11

WPF APPLICATION – HELLO WORLD



12

FILE APP.XAML

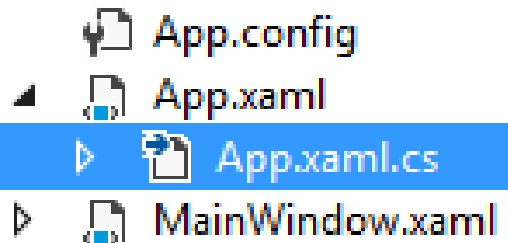


```
<Application x:Class="WPF_Hello_World.App"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="clr-namespace:WPF_Hello_World"
  StartupUri="MainWindow.xaml">
  <Application.Resources>

  </Application.Resources>
</Application>
```

13

FILE APP.XAML.CS



```
using System.Windows;
namespace WPF_Hello_World
{
  /// <summary>
  /// Interaction logic for App.xaml
  /// </summary>
  public partial class App : Application
  {
  }
}
```

14

FILEN APP.XAML.CS SKAPA ETT FÖNSTER DIREKT VIA "CODE BEHIND"


```
using System.Windows;
namespace WPF_Hello_World {
public partial class App : Application {
private void Application_Startup(object sender, StartupEventArgs e) {
    Window f = new Window();
    f.Title = "Hello World";
    f.Show();
}
}
}
```

15

FILEN APP.XAML ÄNDRA STARTUPURI TILL STARTUP

```
<Application x:Class="WPF_Hello_World.App"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="clr-namespace:WPF_Hello_World"
Startup="Application_Startup" >
<Application.Resources>

</Application.Resources>
</Application>
```



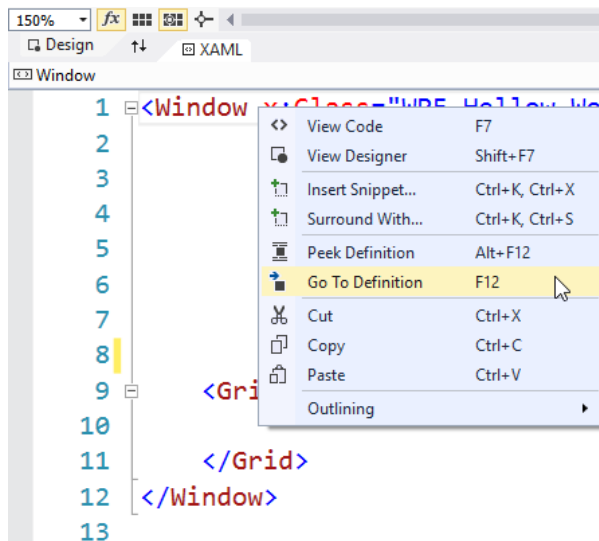
16

FILEN MAINWINDOW.XAML

```
<Window x:Class="WPF_Hello_World.MainWindow"
        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"
        xmlns:local="clr-namespace:WPF_Hello_World"
        mc:Ignorable="d"
        Title="Hello World" Height="350" Width="525">
    <Grid>
</Grid>
</Window>
```

17

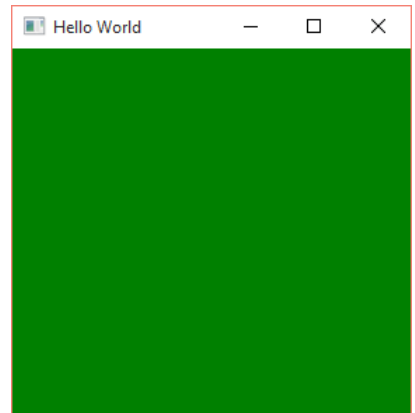
KLASSEN WINDOW



18

KLASSEN WINDOW – NÅGRA EGENSKAPER

```
<Window x:Class="WPF_Hello_World.MainWindow"
  xmlns ...
  Title="Hello World"
  Height="300"
  Width="300"
  Background="Green"
  WindowStartupLocation="CenterScreen"
  ResizeMode="NoResize"
>
</Window>
```



19

KONTROLLER (CONTROLS)

I WPF, används ordet "**Control**" som paraplybegrepp för ett antal klasser som har ett användargränssnitt (UI) och implementerar vissa beteenden (händelser) t.ex.

- TextBlock
- Button
- Grid
- Image
- Calendar
- Menu

Tips

[https://msdn.microsoft.com/en-us/library/mt149842\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/mt149842(v=vs.110).aspx)

20

TEXTBLOCK

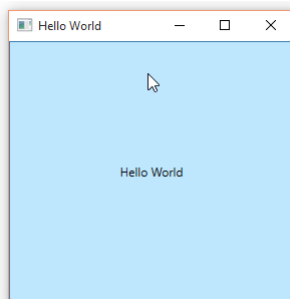
```
<TextBlock  
HorizontalAlignment="Center"  
VerticalAlignment="Center"  
FontSize="72">  
    Hej WPF!  
</TextBlock>
```



21

KNAPPAR

```
<Grid>  
    <Button Name="knapp">Hello World</Button>  
</Grid>
```



22

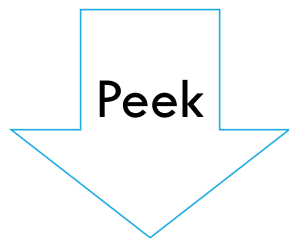
EN HÄNDELSEHANTERARE I CODE-BEHIND

```
public partial class MainWindow : Window {
    public MainWindow()
    {
        InitializeComponent();
    }
    private void knapp_Click(object sender, RoutedEventArgs e) {
        MessageBox.Show("Hello, WPF!");
    }
}
```

23

SKAPA EN HÄNDELSE I XAML

```
<Button Name="knapp" Click="knapp_Click">
    Hello World
</Button>
```



```
146 // Occurs when a System.Windows.Controls.Button is clicked.
147 [Category("Behavior")]
148 public event RoutedEventHandler Click;
```

24

BUTTON — FLERA EGENSKAPER

```
<Button Name="knapp"  
    Click="knapp_Click"  
    Width="150" Height="50"  
    FontFamily="Verdana" FontSize="18" FontWeight="Bold"  
    BorderThickness="5" BorderBrush="AliceBlue"  
    Margin="195,160,0,0"  
    HorizontalAlignment="Left" VerticalAlignment="Top"  
    ToolTip="Klicka här!" >  
    Hello World  
</Button>
```

25

ANVÄND ACCESS KEYS (MNEMONICS)

```
<Button Name="knapp" Click="knapp_Click"  
    HorizontalAlignment="Center"  
    VerticalAlignment="Center"  
    FontSize="50"  
    Foreground="DarkRed">  
    _Klicka här!  
</Button>
```



Klicka här!



Håll ner ALT+K

26

LAYOUT

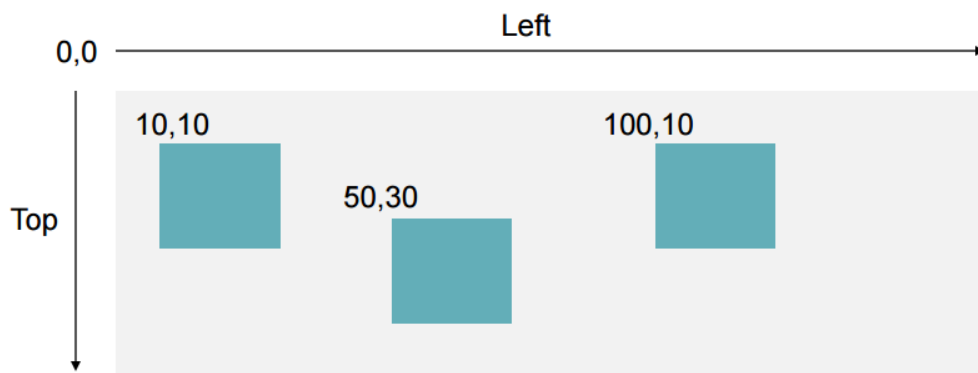
När du skapar ett användargränssnitt behöver du ordna kontroller efter position och storlek.

XAML innehåller flera layoutkontroller t.ex.

- **Grid:** rutnätslayout med rader och kolumner.
- **Canvas:** valfri "absolut" placering av objekt.
- **StackPanel:** Staplar automatiskt objekten lodrätt eller vågrätt.
- **DockPanel:** Objekt justeras till panelens kanter.
- **WrapPanel:** Objekt anpassas till fönstret horisontellt eller vertikalt.

27

CANVAS



28

GRAFIK

WPF har stöd till ett antal grafikkontroller, både 2D och 3D, samt även effekter och animeringar.

Exempel på enkla figurer som ärver från klassen **Shape**

- Line
- Rectangle
- Ellipse
- Polygon

Layoutkontrollen **Canvas** är lämpligast att använda för att rita figurer. Objekten kan placeras i valfri position.

29

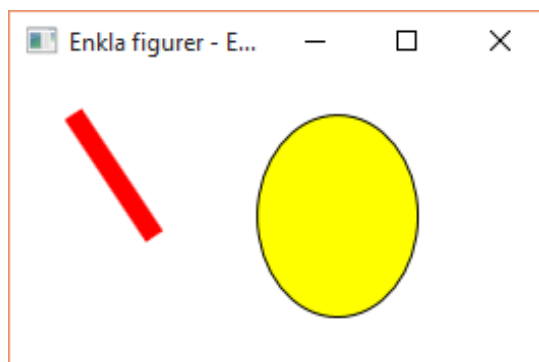
ENKLA FIGURER

```
<Canvas Height="300" Width="300">
```

```
<Line X1="10" Y1="10" X2="50" Y2="70"  
      Stroke="Red" StrokeThickness="10" />
```

```
<Ellipse  
      Fill="Yellow" Height="100" Width="80"  
      StrokeThickness="1" Stroke="Black"  
      Canvas.Left="100" Canvas.Top="10" />
```

```
</Canvas>
```

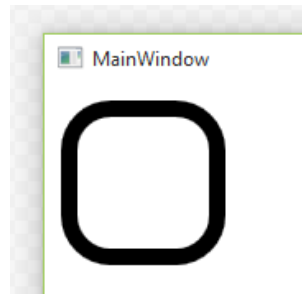


Canvas.Left och
Canvas.Top
Kallas
"Attached properties"

30

BORDER

```
<Canvas>  
    <Border  
        BorderBrush="Black"  
        Width="100"  
        Height="100"  
        BorderThickness="10"  
        CornerRadius="25"  
        Canvas.Left="10"  
        Canvas.Top="10">  
    </Border>  
</Canvas>
```



31

KAPSLADE ELEMENT

```
<Grid>  
    <Button Width="200" Height="100">  
        <Ellipse Fill="Yellow" Height="90" Width="190" />  
    </Button>  
</Grid>
```



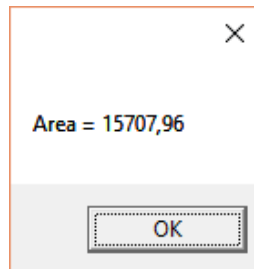
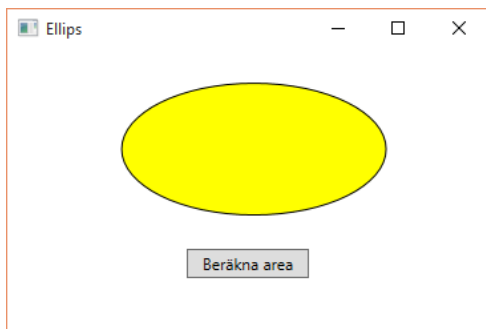
32

ÖVNINGAR

Rita en valfri ellips.

Skapa en knapp som beräknar ellipsens area.

Visa resultatet i en dialogruta med 2 decimaler.



33

```
<Grid>
```

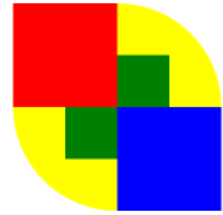
```
  <Canvas width="300" Margin="21,10,21,187">
    <Ellipse Name="ellipse"
      Fill="Yellow" Height="100" width="200"
      StrokeThickness="1" Stroke="Black"
      Canvas.Left="54" Canvas.Top="15"/>
  </Canvas>
  <Button Name="area" Margin="135,150,136,41"
    Click="area_Click">Beräkna area</Button>
```

```
</Grid>
```

```
private void area_Click(object sender, RoutedEventArgs e){
  MessageBox.Show("Area = " +
    Math.Round(Math.PI * ellipse.Height/2 * ellipse.Width/2, 2));
}
```

34

Z-INDEX



<Canvas>

```
<Ellipse Panel.ZIndex="2" Fill="Yellow" Canvas.Left="25" Canvas.Top="25"
Width="100" Height="100" />
```

```
<Rectangle Panel.ZIndex="3" Fill="RED" Canvas.Left="25" Canvas.Top="25"
Width="50" Height="50" />
```

```
<Rectangle Panel.ZIndex="2" Fill="GREEN" Canvas.Left="50" Canvas.Top="50"
Width="50" Height="50" />
```

```
<Rectangle Panel.ZIndex="4" Fill="Blue" Canvas.Left="75" Canvas.Top="75"
Width="50" Height="50" />
```

</Canvas>

35

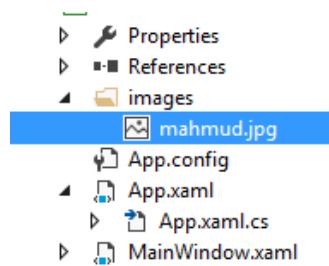
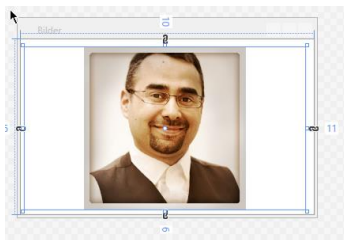
ÖVNINGAR

1. Rita en linje i en Canvas från punkt (10,20) till punkt (100,20).
2. Rita en kvadrat (100x100). Ändra bakgrundsfärgen till svart.
3. Rita en cirkel med diameter 100.

36

MEDIA — VISA EN BILD

```
<Grid>  
<Image Source="images/mahmud.jpg" />  
</Grid>
```



37

IMAGE — EGENSKAPEN STRETCH

```
<Image Source="images/mahmud.jpg" Stretch="none" />
```



```
<Image Source="images/mahmud.jpg" Stretch="Fill" />
```

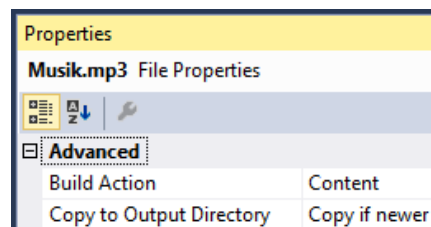
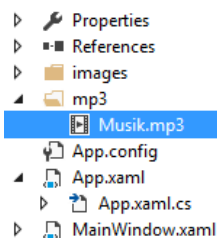


38

MEDIA — SPELA UPP MP3

```
<Grid>
<MediaElement
Name="musik"
Source="mp3/Musik.mp3" LoadedBehavior="Manual" />
<Button Name="spela" Margin="125,155,146,36"
Click="spela_Click">Spela</Button>
</Grid>
```

```
private void spela_Click(object sender, RoutedEventArgs e) {
    musik.Play();
}
```

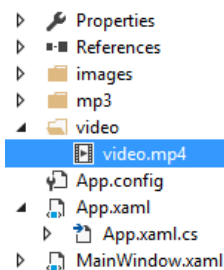


39

MEDIA — SPELA UPP VIDEO

```
<Grid>
<MediaElement
Name="video"
Source="video\video.mp4" LoadedBehavior="Manual" />
<Button Name="spela" Width="100" Height="50" Click="spela_Click"
HorizontalAlignment="Right" VerticalAlignment="Bottom" Opacity="0.5">Spela
</Button>
</Grid>
```

```
private void spela_Click(object sender, RoutedEventArgs e) {
    video.Play();
}
```



Tips! www.sample-videos.com

40

ÖVNINGAR

1. Visa en valfri bild i en knapp.
2. Skapa en enkel WPF-applikationer som spelar upp MP3-filer.