

C# Advanced Meerdere vensters

Koen Bloemen

Sander De Puydt

Erdem Yaraci



**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be





Meerdere vensters

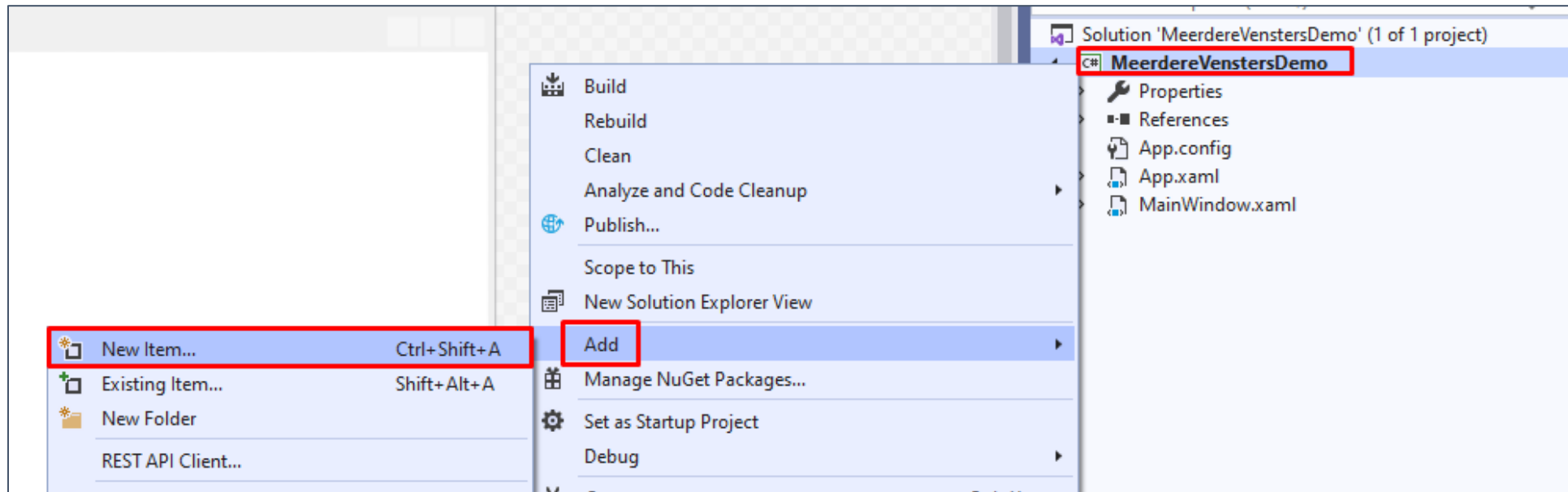
- Introductie
- Nieuwe Vensters
- Tabbladen
- Extra WPF besturingselementen
- Data communiceren

Introductie

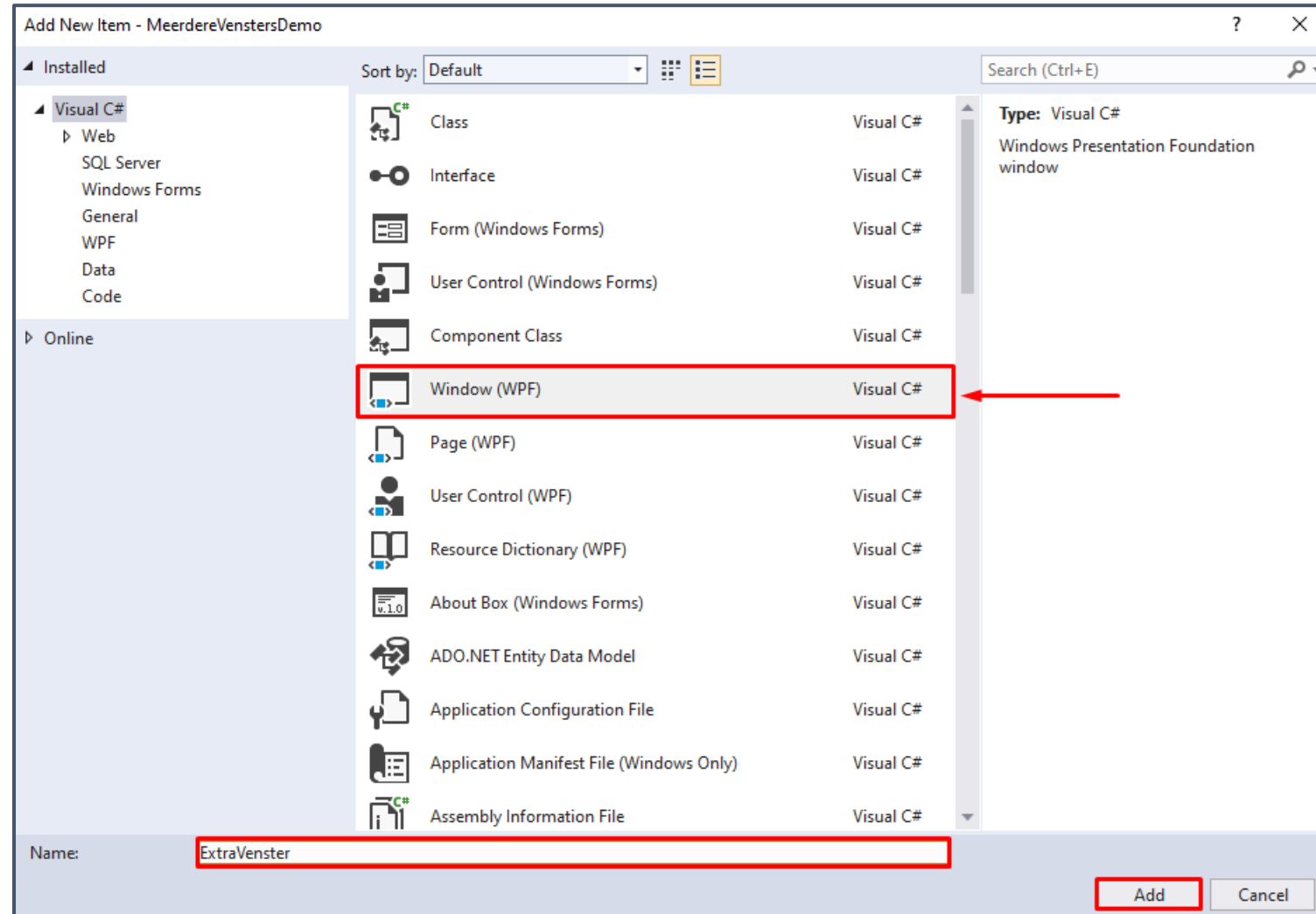
- Wanneer je de layout van de applicatie tijdens het uitvoeren van de applicatie compleet wilt veranderen of een extra formulier wilt openen of een zeer uitgebreide tooltip wilt geven, ... dan kun je dit oplossen met een **extra venster** of **tabbladen**.

Nieuwe Vensters

- Het toevoegen van een nieuw venster kan eenvoudig met de “Add new item”-wizard van visual studio.

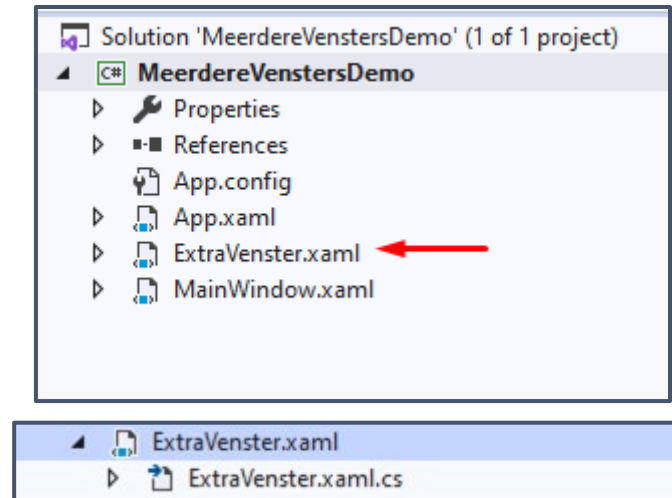


Nieuwe Vensters



Nieuwe Vensters

- Je kan het nieuwe venster terugvinden in de Solution Explorer.

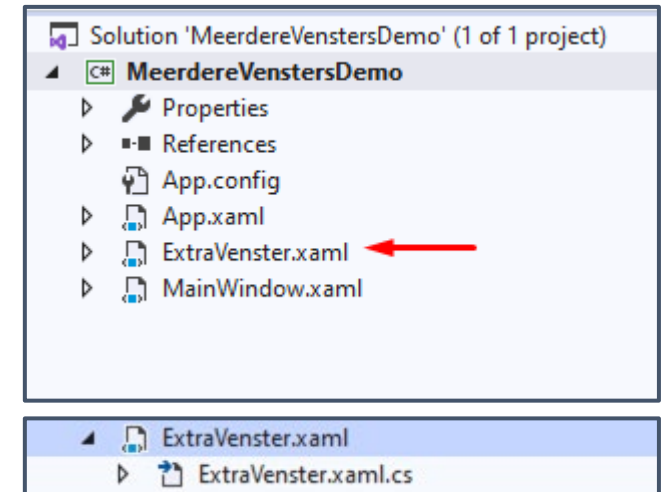


Nieuwe Vensters

- Je kan het nieuwe venster als volgt oproepen in MainWindow:

```
public partial class MainWindow : Window
{
    public MainWindow() {
        InitializeComponent();
    }

    private void OpenVenster_Click(object sender,
        RoutedEventArgs e) {
        ExtraVenster venster = new ExtraVenster();
        venster.Show();
        // of
        // venster.ShowDialog();
    }
}
```



Nieuwe Vensters

- Zodra een nieuw venster wordt aangemaakt is het aanwezig, maar nog niet zichtbaar.
- Je kan Vensters Tonen en Verbergen
 - Show()
 - Hide()

```
ExtraVenster venster = new ExtraVenster();  
// Het venster is nu aangemaakt, maar nog niet zichtbaar
```


Tabbladen

- Wanneer je geen gebruik wil maken van een extra venster, dan kan je ook tabbladen gebruiken voor een meer uitgebreide layout.
- **TabControl**
 - **TabItem**
 - Header
 - Content

```
<TabControl HorizontalAlignment="Stretch"  
VerticalAlignment="Stretch">  
  <TabItem Header="TabItem1">  
    <Grid Background="#FFE5E5E5">>  
      <TextBlock Text="Dit is het eerste tabblad. Alle controls worden  
        binnen het Grid ontworpen." />  
    </Grid>  
  </TabItem> ...
```

Extra WPF besturingselementen

- Statusbalk:
 - Weergeeft de status van het programma
 - Vaak in een DockPanel
 - **StatusBar**

```
Ready C#  vrijdag 24 januari 2020 11:42:51
```

```
<StatusBar DockPanel.Dock="Bottom" Height="30"  
VerticalAlignment="Top" >  
    <TextBlock Text=" Ready" FontStyle="Italic" FontSize="16"/>  
    <Image Source="Afbeeldingen/CSharpLogo.png"  
    Margin="10,0,0,0"/>  
    <Image Source="Afbeeldingen/FileFromWeb_6281.png"  
    Margin="5,0,0,0"/>  
    <TextBlock x:Name="TextBlockDatumTijd" Text="Datum en tijd"  
    Margin="400,0,0,0"/>  
</StatusBar>
```

Extra WPF besturingselementen

- PasswordBox
 - Verbergt de tekst die je er in typt
 - PasswordChar: karakter dat weergegeven wordt tijdens de invoer
 - MaxLength: max. lengte dat ingevoerd kan worden
 - Password: de tekst die ingevoerd is

```
<PasswordBox Name="PasswordBoxLogin"  
              MaxLength="4" PasswordChar="*" />
```

Data Communiceren

- Hoe krijg je gegevens van één venster naar het volgende?
 - Via een public value
 - Via de constructor
 - Via een static class

Data Communiseren

- Via een public value:
 - Aanpassing in MainWindow:

```
public partial class MainWindow : Window
{
    public static string IedereenKanHierAan;

    public MainWindow()
    {
        InitializeComponent();
        IedereenKanHierAan =
            "Tekst van een public value van MainWindow";
    }

    ...
}
```

Data Communiseren

- Via een public value:
 - Aanpassing in ExtraVenster:

```
public partial class ExtraVenster : Window
{
    public ExtraVenster()
    {
        InitializeComponent();
        TekstVeldVoorPublicValue.Text =

        MainWindow.IedereenKanHierAan;
        // Een public value is toegankelijk ookal staat deze in
        // een andere klasse

    }
    ...
}
```

Data Communiseren

- Via de constructor:
 - Een constructor maakt een instantie van een klasse:
 - Aanpassing in ExtraVenster

```
public partial class ExtraVenster : Window
{
    public ExtraVenster(string tekstVanMain)
    {
        InitializeComponent();
        TekstVeldVoorConstructor.Text = tekstVanMain;
    }
    ...
}
```

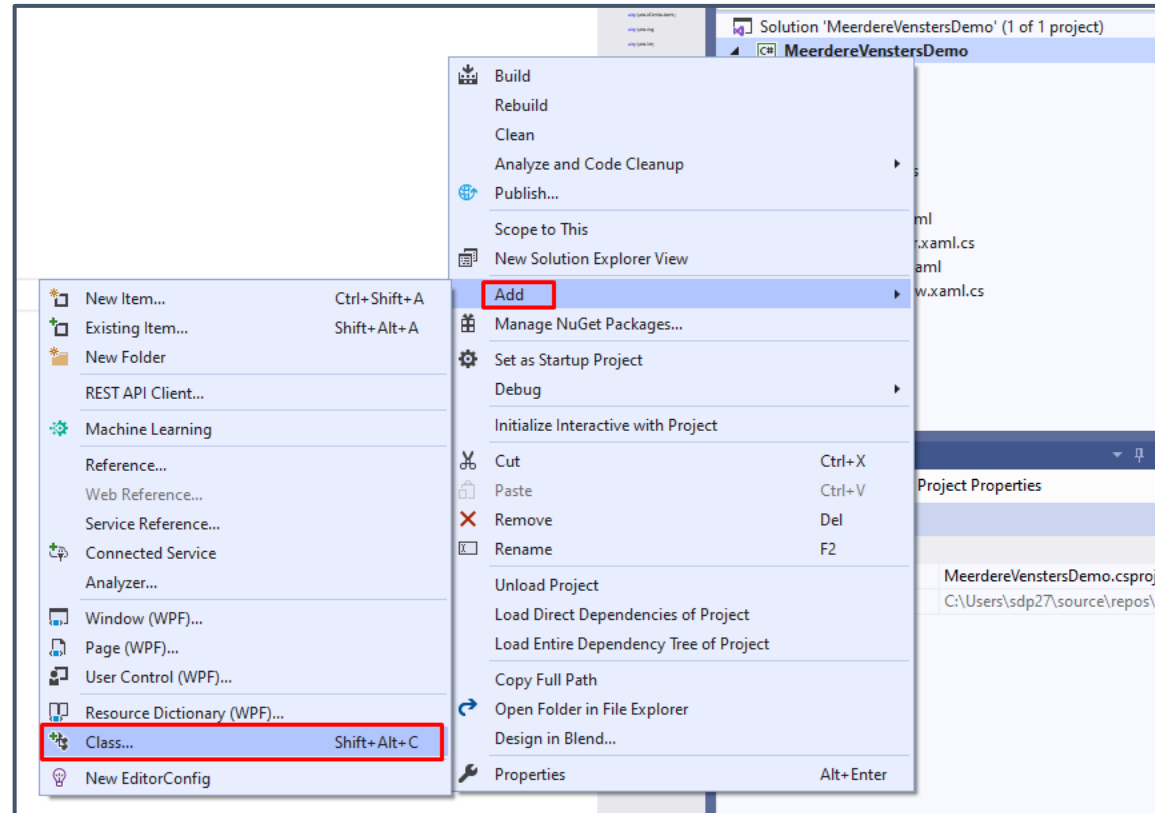

Data Communieren

- Via de constructor:
 - Aanpassing in MainWindow

```
...
private void OpenVenster_Click(object sender, RoutedEventArgs e)
{
    ExtraVenster venster =
        new ExtraVenster("Tekst via Constructor");
    venster.Show();
}
...
```

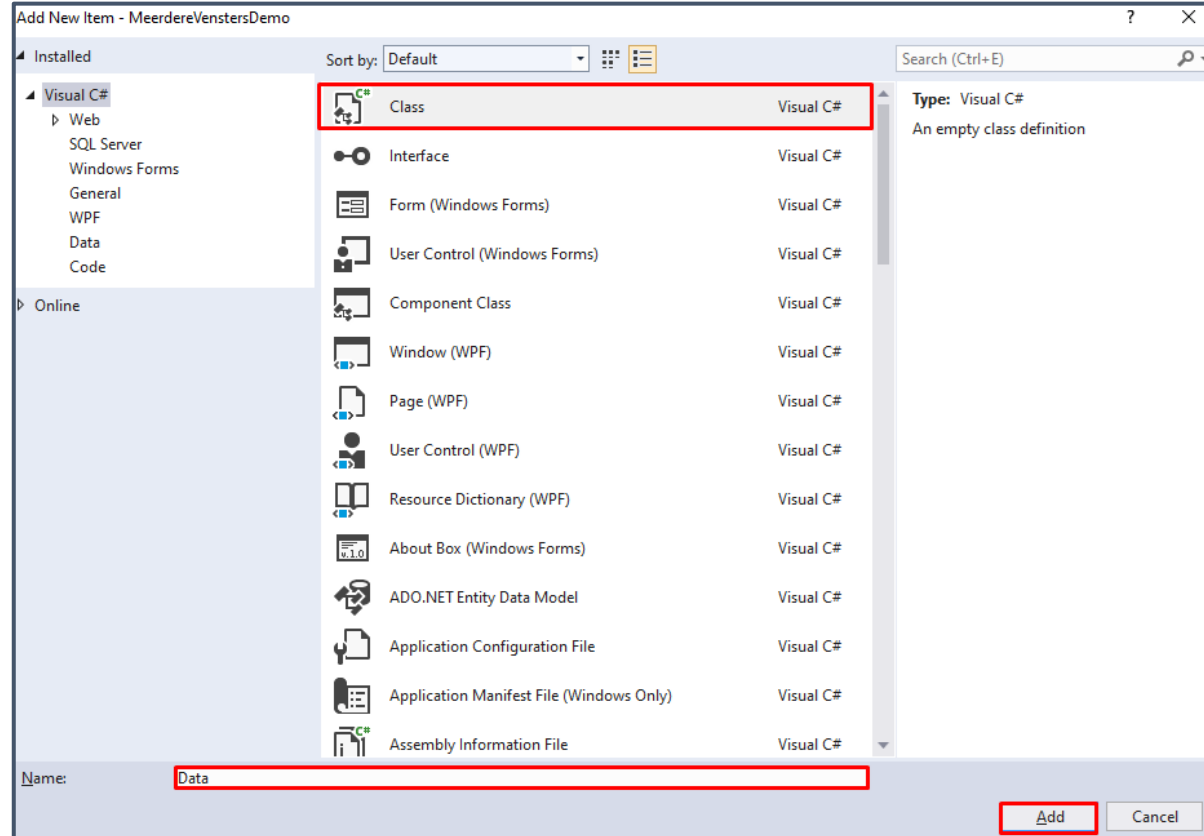
Data Communiseren

- Via een static class:
 - Aanmaken van een static class



Data Communiseren

- Via een static class:
 - Aanmaken van een static class



Data Communiceren

- Via een static class:
 - Aanpassingen in static class

```
...  
    public static class Data  
    {  
        public static string Tekst = "tekst van Data";  
    }  
...
```

Data Communiseren

- Via een static class:
 - Aanpassing in MainWindow

```
...  
    public partial class MainWindow : Window  
    {  
        public MainWindow()  
        {  
            InitializeComponent();  
            Data.Tekst = "Tekst van Main voor Static Class";  
        }  
    }  
...
```

Data Communiseren

- Via een static class:
 - Aanpassing in ExtraVenster

```
...  
    public ExtraVenster(string tekstVanMain)  
    {  
        InitializeComponent();  
        TekstVeldVoorStaticClass.Text = Data.Tekst;  
    }  
...
```

Data Communiceren

- Opdracht:

Maak een applicatie met een tekstveld en een knop.

- Als gebruiker kan ik op de knop klikken om een tweede venster te openen waarin de tekst van het tekstveld getoond wordt.
- Uitdaging: hoe zorg ik er voor dat de tekst update in het nieuwe venster wanneer deze wordt aangepast?

Assembly information

- In het bestand AssemblyInfo.cs (Project/Properties) instellingen wijzigen
- Via Assembly class settings uitlezen:

```
Assembly assembly = Assembly.GetExecutingAssembly();  
FileVersionInfo fvi = FileVersionInfo.GetVersionInfo(assembly.Location);
```

```
TextBlockVersion.Text += $"{fvi.ProductVersion}";  
TextBlockProductname.Text += $"{fvi.ProductName}";  
TextBlockCompanyname.Text += $"{fvi.CompanyName}";  
TextBlockCopyright.Text += $"{fvi.LegalCopyright}";  
TextBlockDescription.Text += $"{fvi.Comments}";
```