

Kapitel 6 Events i C#

Når vi snakker programmering er en Event når der aktiveres specielt indbygget kode. Der sker en hændelse (event). Man siger at eventet bliver udført. De events vi skal gennemgå i dette kapitel kaldes GUI Events (GUI står for Graphic User Interface). Det kan f.eks. være et klik med en mus, eller når en tekstboks forlades, højre klik og meget mere. Vi starter med at se på det event der sker når der klikkes på en knap.

Klik på en knap Event

Et klik event aktiveres når der klikkes på en knap. Undersøg standard koden for en knap nedenfor:

```
private void button1_Click(object sender, EventArgs e)
{
}
```

I mellem parenteserne har vi dette:

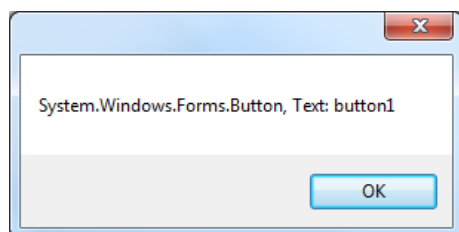
object sender, EventArgs e

Nøgleordet object refererer til objektet, som aktiverer eventet, en knap i dette tilfælde. Det placeres i en variabel der kaldes sender. Du kan selv teste det.

Start et nyt projekt. Tilføj en knap til din nye formular og dobbeltklik på den. Placer følgende kode mellem de krøllede parenteser:

MessageBox.Show(sender.ToString());

Vi bruger metoden ToString på variabelen sender. Kør dit program og klik på knappen. Du ser nu følgende:



Meddelelsen viser hvilket objekt, der var afsender af Eventen, såvel som der vises Text betingelsen på sender: knappen med Text "button1".

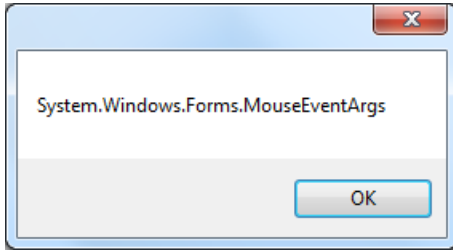
De andre argumenter i parenteserne er:

EventArgs e

EventArgs er en class. Det er en forkortelse for event arguments, og fortæller dig hvilke events, der blev startet. Bogstavet "e" definerer en variabel, der kan bruge denne class. Hvis du ændrer din kode til dette:

MessageBox.Show(e.ToString());

Vil meddelelsesboksen vise følgende:

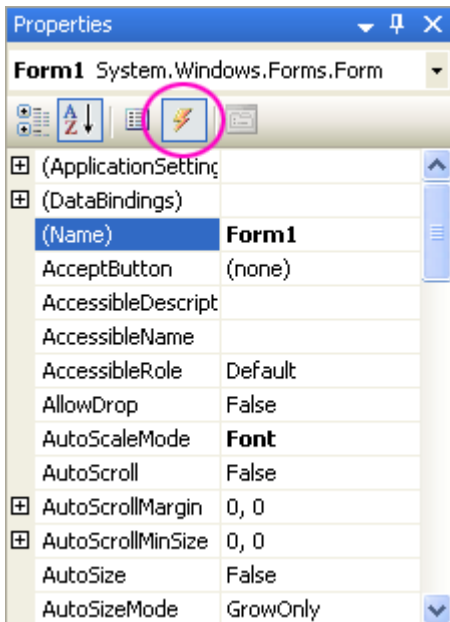


Når du klicker startes der et Mouse Event Argument. C# ved allerede på forhånd hvad, der skal ske med et Event af denne type, så du skal ikke selv skrive nogen speciel kode. Men hvad nu hvis du vil vide hvilken knap der blev klikket på, den højre eller venstre?

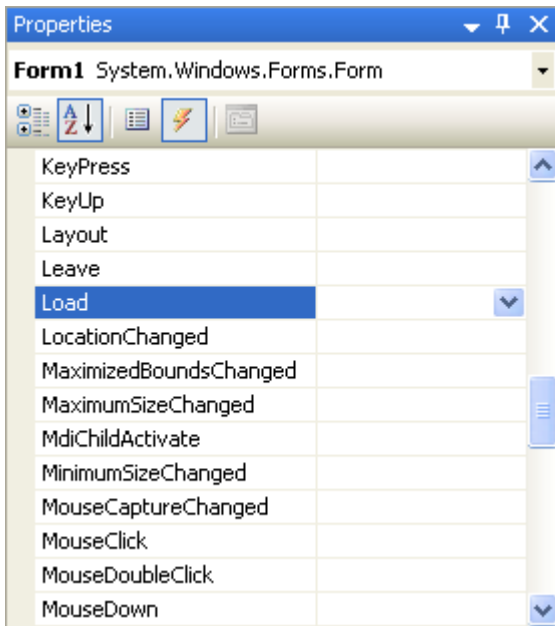
MouseDown Event i C#

Du kan se de events der er tilgængelige for et bestemt kontrolelement i Properties vinduet.

Når du er i Design View klicker du på din Form1. Når du skal se de events, der er tilgængelig for formularen selv klicker du på lynet i toppen af Properties området, som vist nedenfor:



Når du klicker på lynet, ser du en liste med events:



Du har allerede set Load event, så det kommer vi ikke ind på her. Men læg mærke til hvor mange events der er.

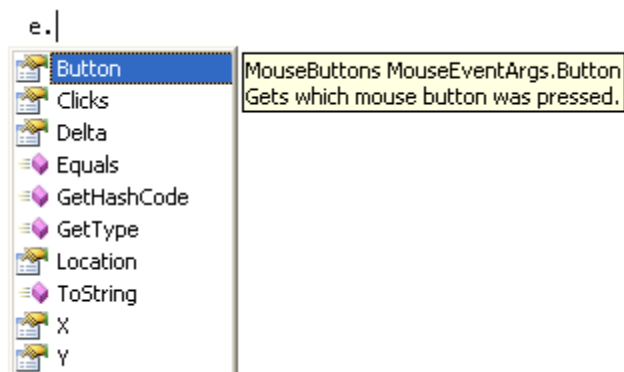
Find det event der hedder MouseDown. Dobbeltklik på ordet "MouseDown". Du ser nu en stump kode:

```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    |
}
```

I mellem parenteserne er der stadig et sender object. Men bemærk det nye argument:

MouseEventArgs e

Bogstavet "e" er et default variabelnavn. Variabeltypen tilhører MouseEventArgs. Du kan se hvad den gør ved at taste bogstavet "e", og derefter et punktum. Du ser nu IntelliSense listen:



Listen viser alle de betingelser og methods, der er tilgængelig for variabelen e. En af dem er Button betingelsen.

Vi kan bruge en if sætning når vi skal undersøge hvilken museknap der blev klikket på. Tilføj dette til din kode:

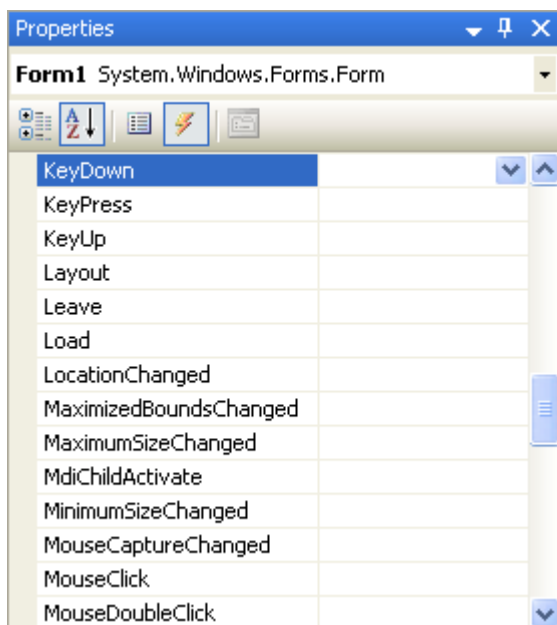
```
private void Form1_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        MessageBox.Show("Left button clicked");
    }
    else if (e.Button == MouseButtons.Right)
    {
        MessageBox.Show("Right button clicked");
    }
}
```

Kør dit program og klik med begge museknapper i din formular. Du ser en meddelelsesboks.

KeyDown Event i C#

KeyDown event fungerer på samme måde som MouseDown event – du får adgang til EventArgs og kan skrive kode for at undersøge hvilken tast, der blev klikket på.

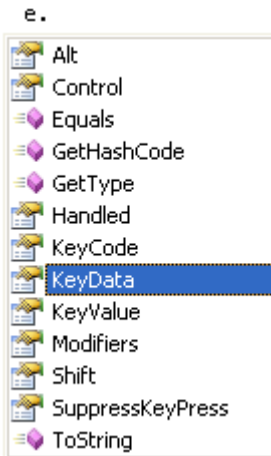
Skift til Designvisning og find det event der hedder KeyDown for din formular:



Dobbeltklik på navnet for at se den tilhørende kode. Du ser nu følgende:

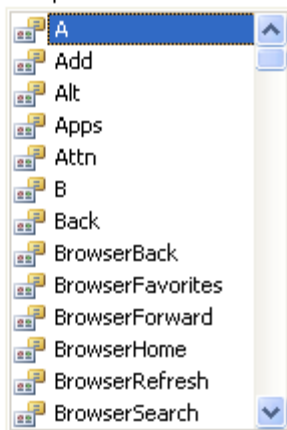
```
private void Form1_KeyDown(object sender, KeyEventArgs e)
{
    |
}
```

Denne gang ser du **KeyEventArgs** i stedet for MouseEventArgs. Når du taster "e" ser du IntelliSense med en liste af muligheder:



Den du skal bruge når du skal undersøge hvilken tast, der er trykket på er betingelsen **KeyData**. Efter det dobbelte lighedstegn vælger du i listen Key. Tast derefter punktum og du ser følgende:

```
e.KeyData == Keys.
```



På listen ser du alle bogstaver, tegn og taster på dit tastatur.

Pak det hele ind i en IF sætning og du har følgende:

```
if (e.KeyData == Keys.A)
{
    MessageBox.Show("Letter A Pressed");
}
```

Kør dit program og test det. Du ser nu en meddelelsesboks når du taster bogstavet "a" på dit tastatur.

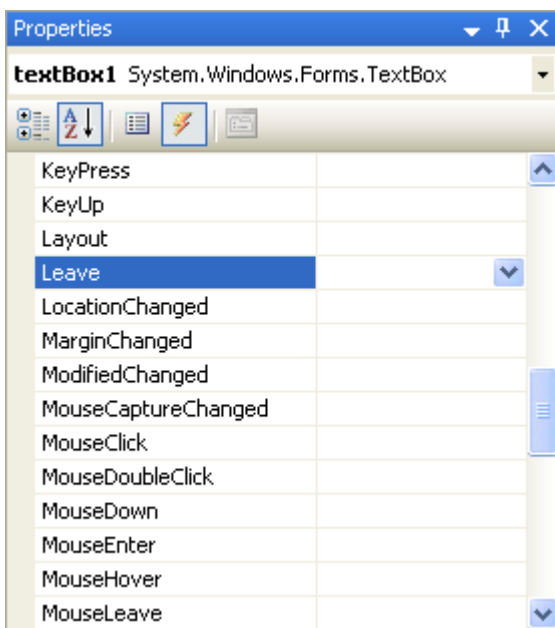
Leave Event i C#

En meget brugbar event du kan bruge i forbindelse med dine tekstbokse er Leave event. Det giver dig mulighed for at validere en tekstboks, når brugeren prøve at forlade den. Du kan for eksempel undersøge om tekstboksen er tom. Hvis tekstboksen skal udfyldes kan du få brugeren til at blive i tekstboksen. Lad os prøve det.

Tilføj to tekstbokse til din formular. Find betingelse TabIndex betingelse for textBox1 og sæt den til 0. Sæt TabIndex for textboks2 til 1.

Betingelsen TabIndex referere til hvilket kontrolelement der vil blive valgt når Tab tasten trykkes. I en formular som brugeren skal udfylde er du interesseret i at cursoren springer til den næste tekstboks. Du vil ikke have at cursoren f.eks. hopper fra textboks1 til textboks8 – det gør ikke noget at der er en pæn rækkefølge i kontrolelementerne. TabIndex giver dig mulighed for at sætte Tab rækkefølgen for dine kontrolelementer i din formular.

Marker TextBox1 og klik derefter på lynen i Properties vinduet for at se en liste over events for din tekstboks. Du ser nu følgende:



Dobbeltklik på Leave eventet for at se koden. Indtast nu følgende kode:

```
private void textBox1_Leave(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("You can't leave this box blank");

        textBox1.Focus();
    }
}
```

Det koden gør er at den undersøger om tekstboksen er tom. Men bemærk at den undersøger det når brugeren prøver at forlade tekstboksen. Bemærk også hvordan brugeren sendes tilbage til tekstboks:

textBox1.Focus();

Focus method kan bruges til at tvinge valget af et kontrolelement. Når du arbejder med tekstbokse betyder det at cursoren til stå og blinke i tekstboksen.

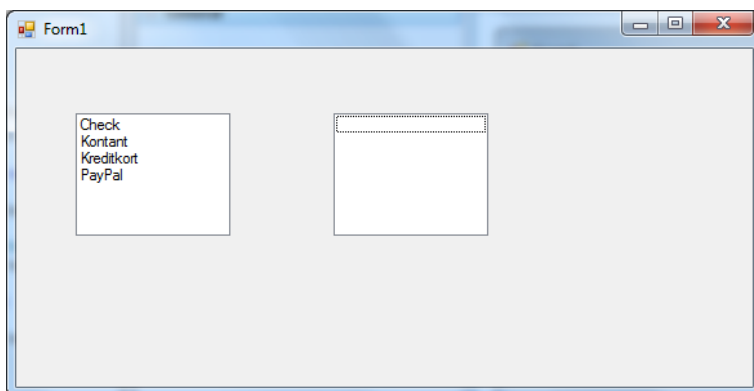
Du kan f.eks. også konvertere tekst til storebogstaver, når brugere taster Tab for at komme væk fra tekstboksen. Her er noget kode der konvertere til storebogstaver:

```
private void textBox1_Leave(object sender, EventArgs e)
{
    if (textBox1.Text == "")
    {
        MessageBox.Show("You can't leave this box blank");
        textBox1.Focus();
    }
    else
    {
        textBox1.Text = textBox1.Text.ToUpper();
    }
}
```

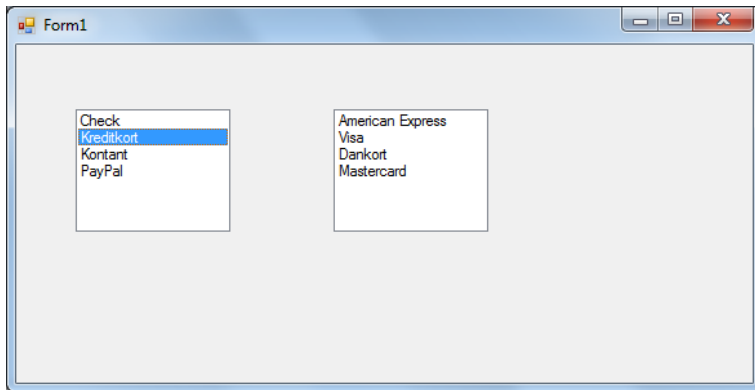
I det næste afsnit skal vi se på Events til ListBoxes og ComboBoxes.

ListBox og ComboBox events i C#

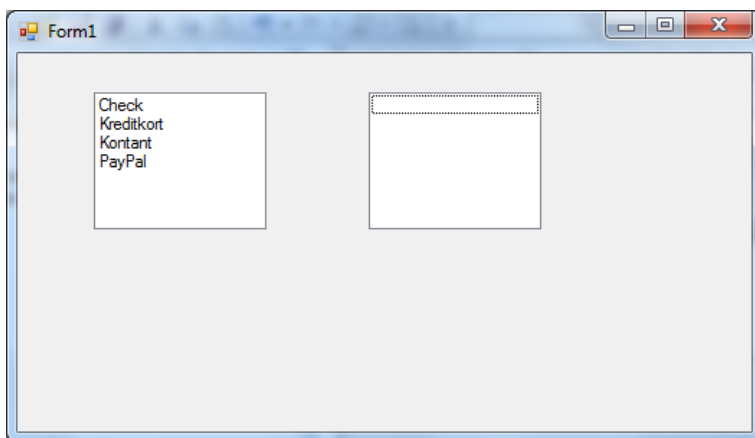
En brugbar event til en ListBox og en ComboBox er **SelectedIndexChanged**. Du kan bruge denne event til at finde ud af hvilket element der er valgt i listen eller comboboksen. Eventet vil køre igen når du vælger et nyt element. Hvorfor er det så brugbart? Vi kan se på følgende eksempel. Du har to styks ListBox eller Combox. Den anden boks vil blive udfyldt med de elementer, der findes på basis af det valg, der er foretaget i den første boks. Se på billedet nedenfor:



Vi har to ListBoxe i vores formular. Den første viser forskellige betalingsmuligheder og den anden er tom. Når kreditkort vælges i den første ListBox, bemærk så hvad der sker i den anden ListBox:

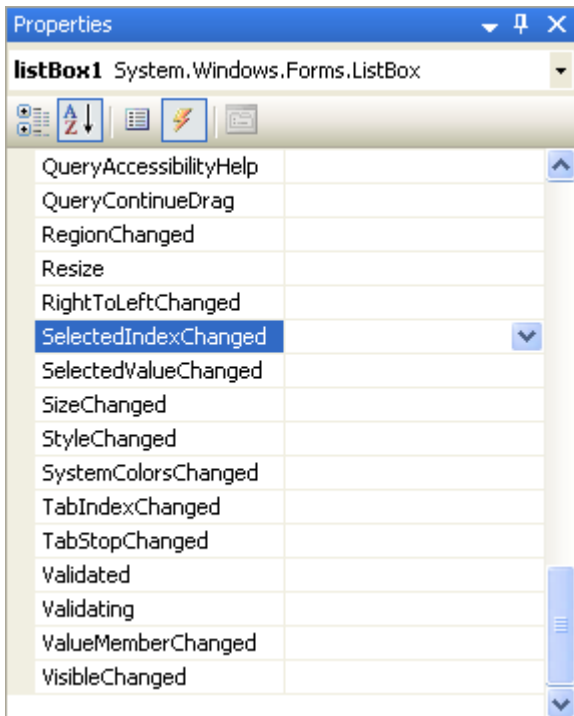


Den anden ListBox viser nu de typer af kreditkort der accepteres. Klik på Check og du får nu en anden række muligheder:



Alt dette sker med det event der hedder SelectedIndexChanged.

Vi skal se hvordan der virker. Start derfor et nyt projekt og tilføj to Listboxe til din formular. Klik på den først og i betingelsen Item tilføjer du følgende elementer: Check, Kontant, Kreditkort og PayPal. Klik nu på lynet for at se de events der høre til en listbox:



Dobbeltklik på eventet SelectedIndexChanged for at se koden:

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    |
}
```

For at få fat i elementet i en ListBox eller ComboBox, der er valgt kan du bruge betingelsen SelectedIndex. Det første element er 0, det næste er 1 osv. Tilføj derfor denne if sætning til din kode:

```
if (listBox1.SelectedIndex == 1)
{
    loadListBox();
}
```

Vi undersøger her om det valgte index har værdien 1. Hvis det er opfyldt kalder vi en method – loadListBox. Tilføj denne method til din kode:

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (listBox1.SelectedIndex == 1)
    {
        loadListBox();
    }
}
private void loadListBox()
{
    listBox2.Items.Clear();

    listBox2.Items.Add("American Express");
    listBox2.Items.Add("Visa");
    listBox2.Items.Add("Dankort");
    listBox2.Items.Add("Mastercard");
}
```

Du har allerede set på ListBoxe tidligere så vi vil ikke gå i dybden med dem her. Kør dit program og test det. Det skulle gerne virke som beskrevet.

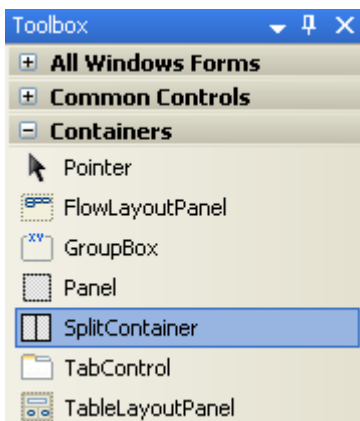
Øvelse

Tilføj yderligere en method, der indlæser elementer når betalingsbetingelsen Check vælges. Hent den nye method fra eventet SelectedIndexChanged når element 0 er valgt i den første Listbox. Når dit program kører vil betalingsbetingelserne for Check vises i den anden Listbox, i stedet for Kreditkort muligheden.

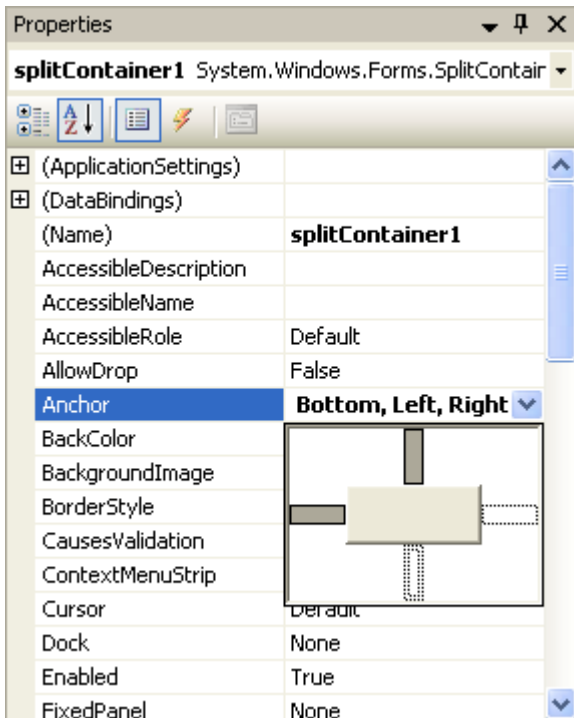
Byg din egen Web Browser i C#

Vi skal til slut se på hvordan du bygger din egen browser i C#. Vi vil gøre brug af kontrolelementer: en SplitContainer, en TreeView og selvfølgelig en WebBrowser. Det er overraskende nemt!

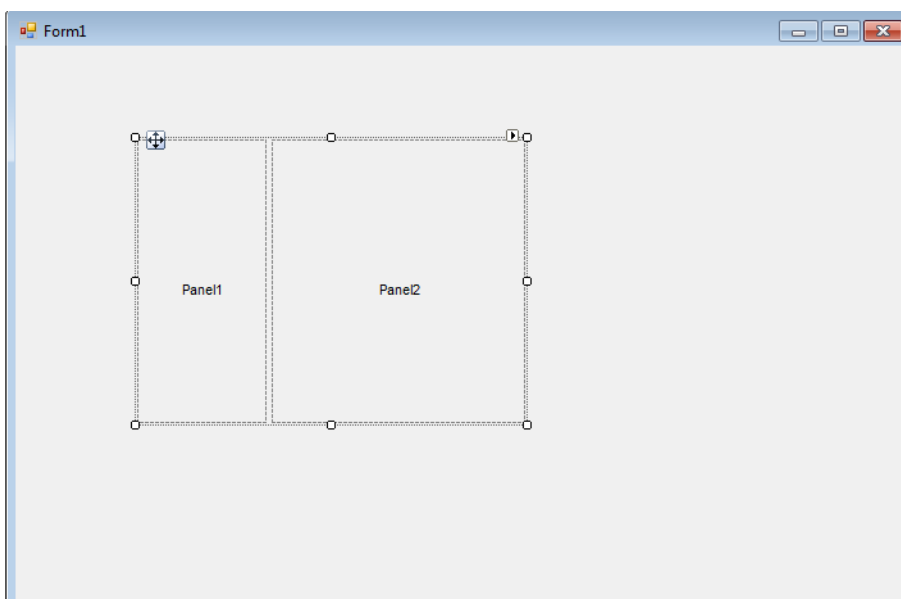
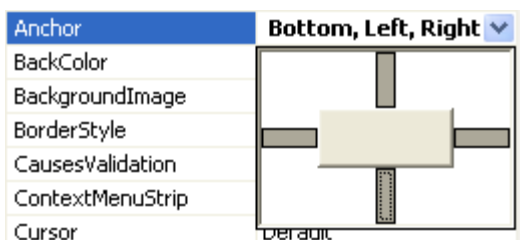
Start et nyt projekt. Marker din formular og gør den tilpas stor. Udvid din Toolbox og find SplitContainer. En SplitContainer gør det muligt for dig at have et kontrolelement i den ene side og et andet kontrolelement i den anden side. Vi vil indsætte kontrolelementet i venstre side og browseren i højre side. SplitContainer ser ud på følgende måde:



Dobbeltklik for at tilføje en SplitContainer til din formular. Som default fylder den hele din formular og hæfter sig fast i øverste venstre hjørne. Hvis du vil ændre på dette skal du finde Anchor betingelsen for din SplitContainer. Klik på listepilen for at se følgende:



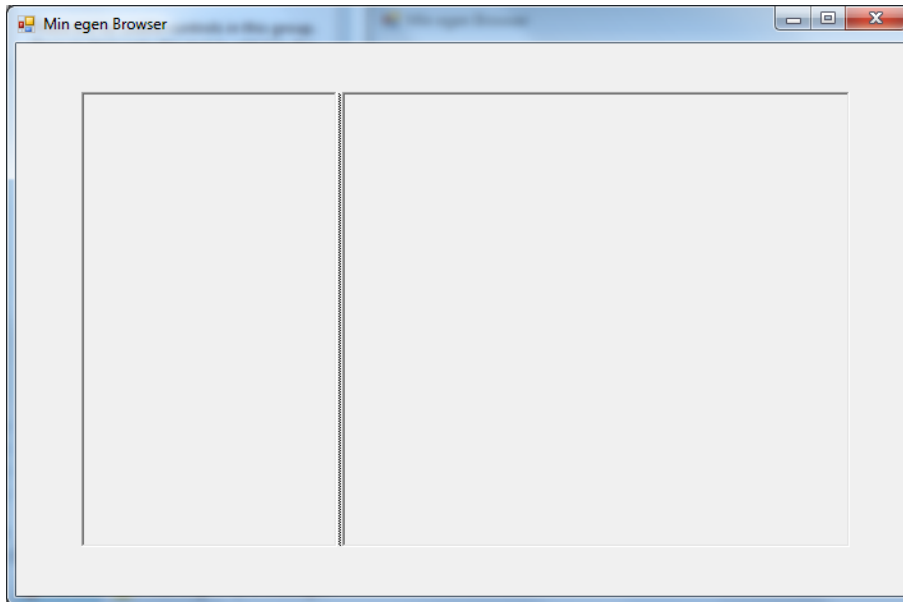
Top, Left er indikeret ved de grå firkanter. Klik på de to små hvide rektangler for at gøre dem grå:



Som du kan se er der et Panel1 og et Panel2. Vi vil indsætte kontrolelementet TreeView i Panel1 og kontrolelementet WebBrowser i Panel2. (En lille bemærkning omkring din SplitContainer er at den kan

være noget drilsk når du vil markere selve din SplitContainer, i stedet for et af din Panels. Hvis du klikker i det tynde område mellem de Panels markerer du hele elementet.)

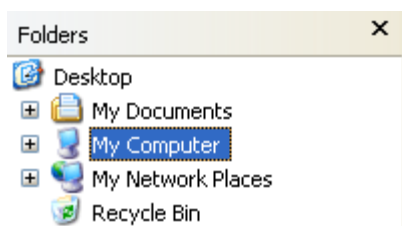
Inden vi går videre til elementet TreeView, skal du finde betingelsen BorderStyle og ændre den vil enten Fixed Single eller Fixed 3D. Hvis du kører dit program vil din formular nu se sådan ud:



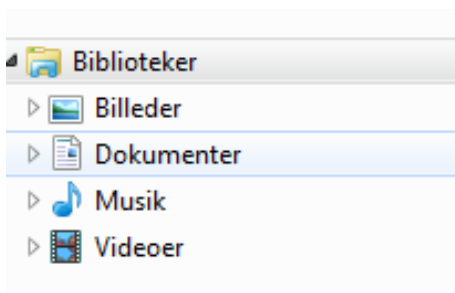
Du kan ændre på størrelsen af de to paneler.

Kontrollementet TreeView i C#

Et andet brugbart kontrollement du kan tilføje din formular er kontrollementet TreeView. Du har set det adskillige gange her er du det fra XP:

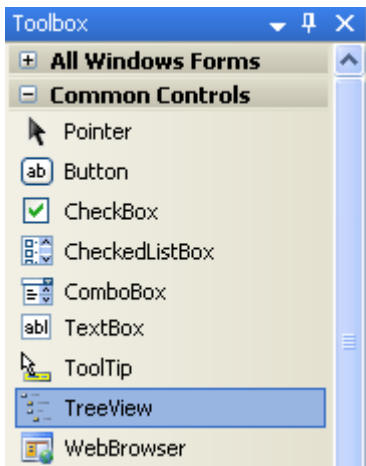


I Vista og Windows 7 er plus symbolet nu en pil:

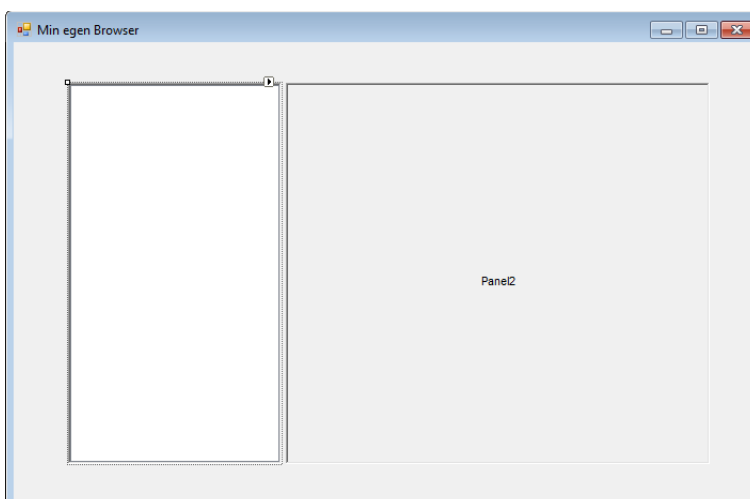


Kontrollementet TreeViews har pile eller plus symboler du kan klikke på. Når du gør det udvider du et element (kaldet en Node). Den Node der er valgt i billedet ovenfor er Dokumenter.

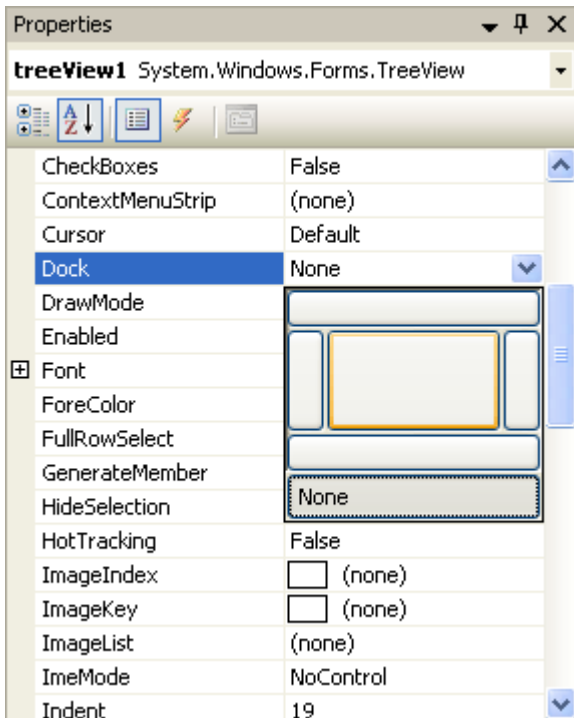
Når du skal tilføje kontrollementet TreeViews skal du i din Toolbox finde det i Common Controls:



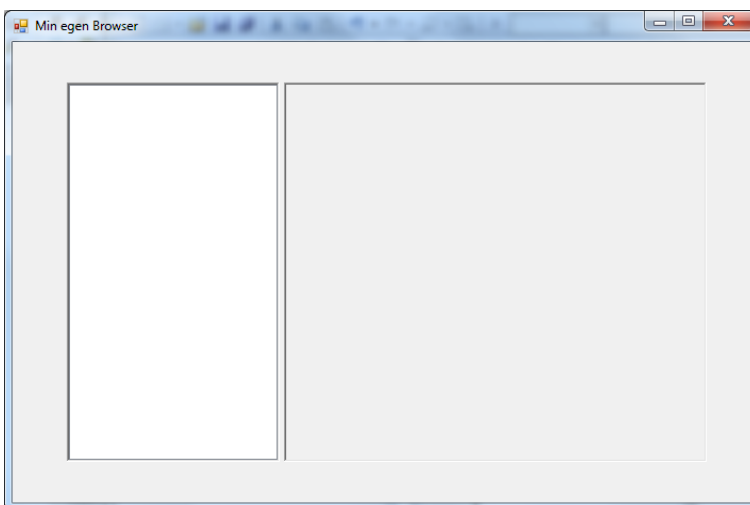
Træk det ud i din formular og placer den i Panel1. Tilpas størrelsen så den fylder hele Panel1. Din formular ser nu sådan ud:



Lig nu kan du ikke flytte din TreeView. Når du gør det vil den flyde ud i det andet panel og udenfor selve containeren. For at undgå dette markerer du kontrollementet TreeView. Find nu betingelsen Dock. Klik på listepilen for at se følgende:



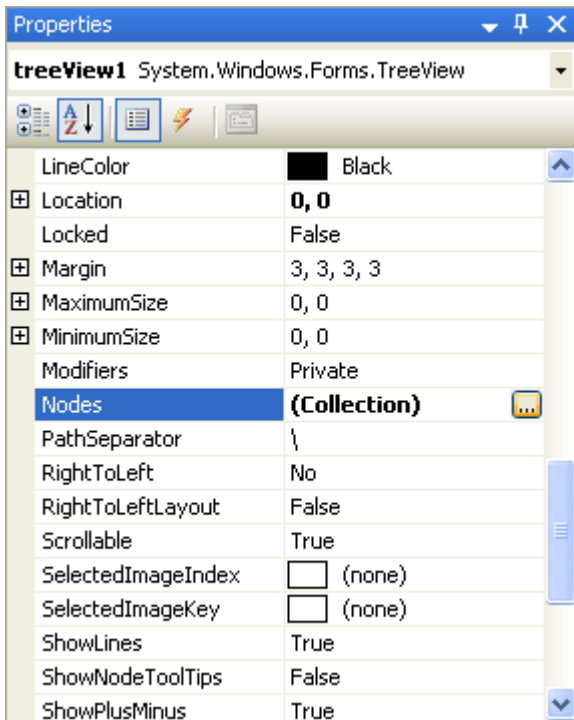
Klik på det grå rektangel i midten af listen. Det er en mulighed der hedder Fill. Den vil simpelthen udfylde hele Panel1. Din formular ser nu sådan ud når du kører den:



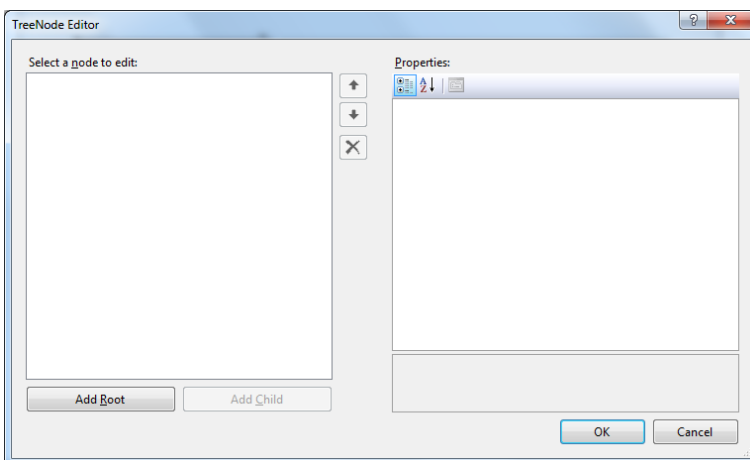
Vi har ikke tilføjet nodes til vores TreeViews endnu, så lad os gøre det.

Tilføj Nodes til et TreeView element i C#

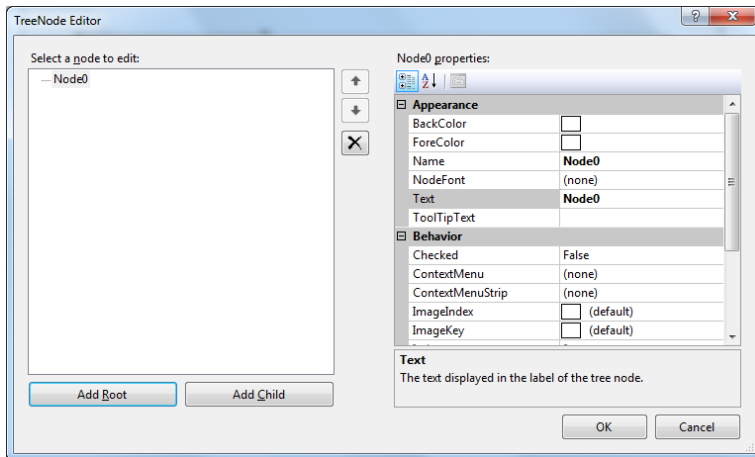
Når du skal tilføje Nodes til dit TreeView, klikker du på TreeView for at markere det. Find derefter betingelsen Nodes i dit Properties vindue:



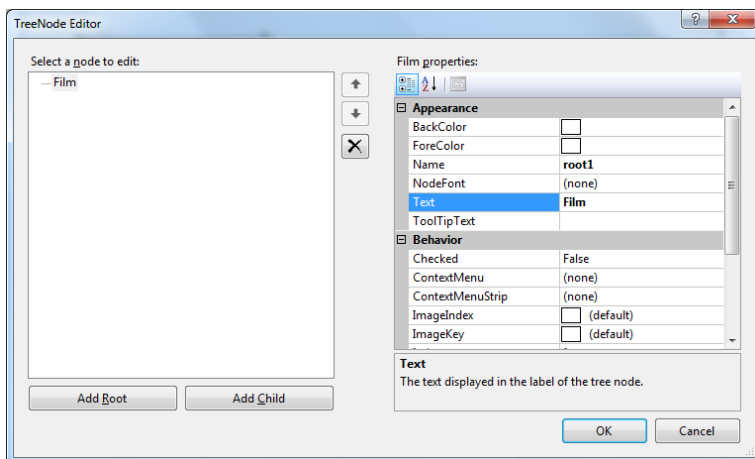
Klik på knappen med de tre små prikker, lige ved siden af Collection. Du ser nu dialogboksen `TreeNode` Editor:



Der findes to typer af Nodes du kan tilføje – en Root Node og en Child Node. Klik på knappen **Add Root**:



Find betingelsen Name i højre side og skift den til root1. Find Text betingelsen og skift den fra Node0 til Film. Det bliver hovedtitlen i vores første Root Note. Din TreeNode Editor ser nu sådan ud:



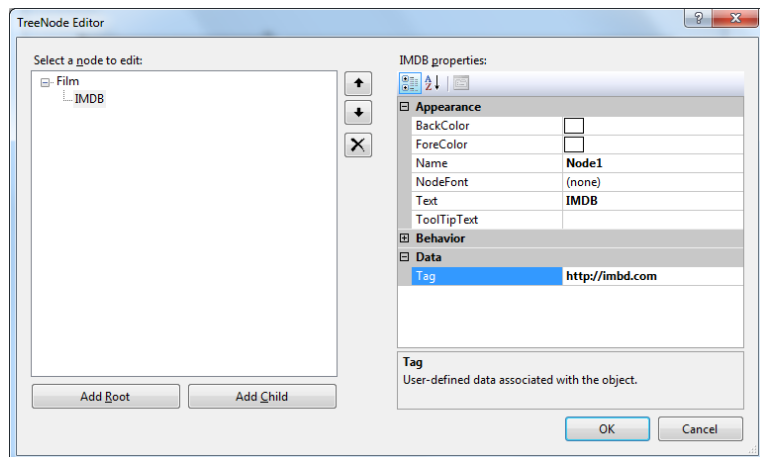
Når din Film Node er markeret klikker du på knappen Add Child. Det placerer en Node under Film, som er din rod Node.

Det vi er i gang med nu er at oprette et sæt af elementer til vores Tree. Hvert element skal associeres med en hjemmeside. Når du klikke på en Child Node til venstre vil den tilhørende hjemmeside vises i browseren i højre side.

Mens dit Child Node er markeret kan du se at betingelsen Name som standard er Node1. Det kan lade stå. Find betingelsen Text og ændre det til IMDB. Find betingelsen Tag og skriv følgende adresse:

<http://imdb.com>

Din Editor ser nu sådan ud:



Klik tilbage på Film. Klik på knappen Add Child igen. Du ser en ny Node under IMDB. Angiv følgende betingelser:

Name: Node2

Tag: http://dfi.dk

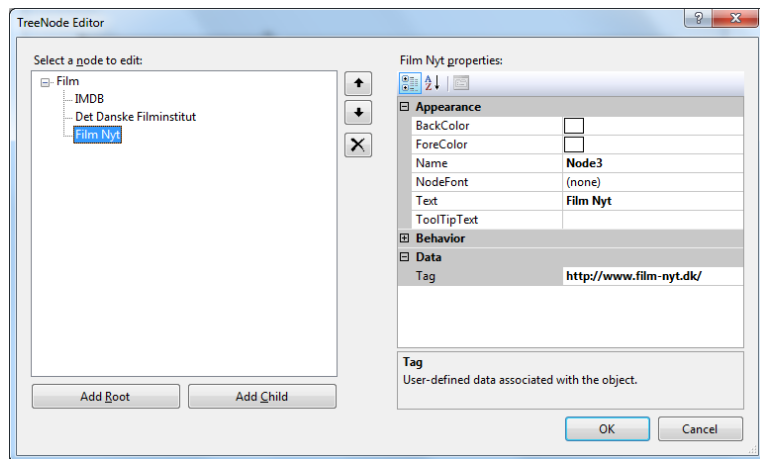
Text: Det Danske Filminstitut

Den anden hjemmeside tilføjes. Klik tilbage på Film igen. Klik på knappen Add Child for at tilføje en ny Node under Film. Angiv følgende betingelser:

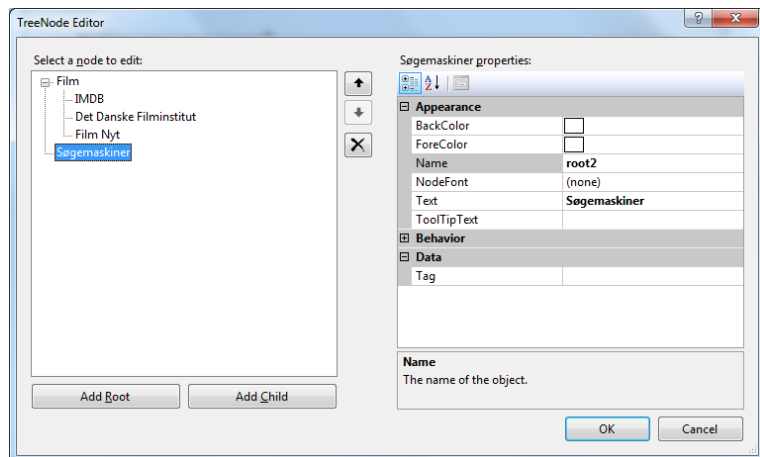
Name: Node3

Tag: http://www.film-nyt.dk/

Text: Film nyt



Vi kan også tilføje endnu en Root Node. Klik på knappen Add Root. Ændre Text betingelsen for den nye Root Node til Søgmaskiner. Ændre Name betingelsen til root2. Din Editor ser nu sådan ud:



Brug den samme teknik som ovenfor til at tilføje en række Child Nodes under Søgemaskiner. Angiv følgende betingelser:

Name: Node5

Tag: <http://google.dk>

Text: Google

Name: Node6

Tag: <http://dk.msn.com>

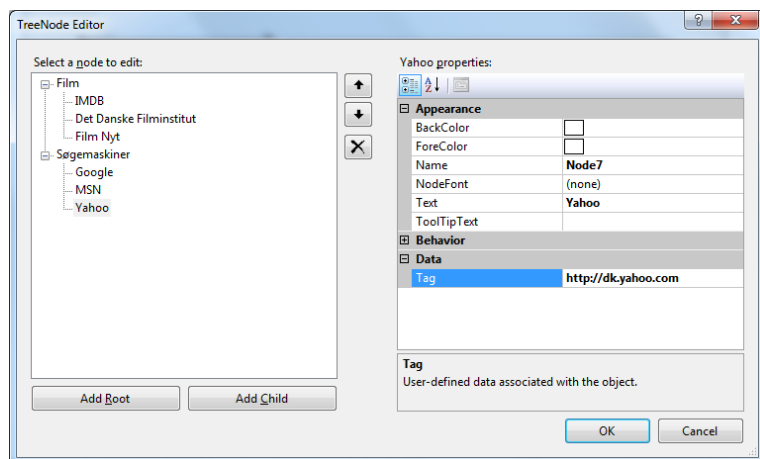
Text: MSN

Name: Node7

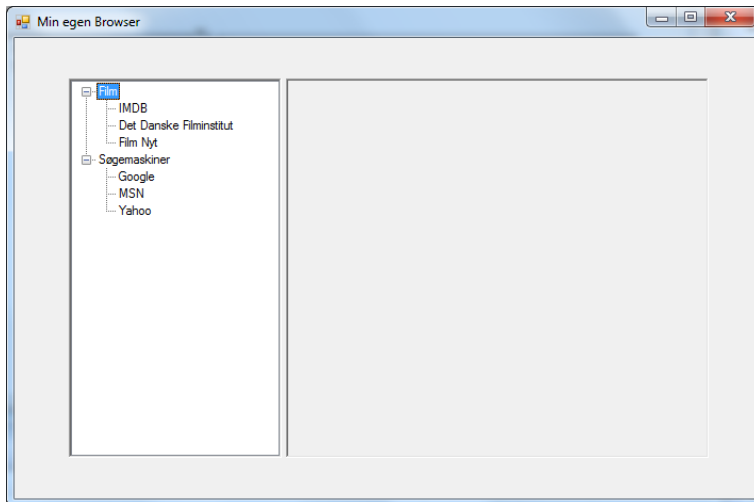
Tag: <http://dk.yahoo.com>

Text: Yahoo

Din Editor ser nu sådan ud:



Klik OK for at vende tilbage til din formular. Kør din formular for at se hvordan den ser ud. Den ligner nu dette:

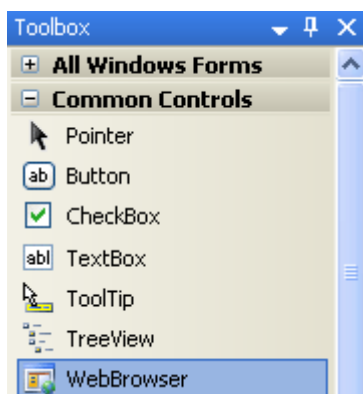


Klik på plus/minus symbolerne for at udvide og lukke dit træ. Luk formularen og vend tilbage til C#.

Dine Nodes foretager sig dog ikke noget endnu. Det får vi dem dog til inden længe. Først skal vi tilføje en Web Browser.

Tilføj en WebBrowser til din Formular i C#

Det er ganske nemt at tilføje en Web Browser. Find kontrolelementet WebBrowser i din Toolbox under Common Controls:



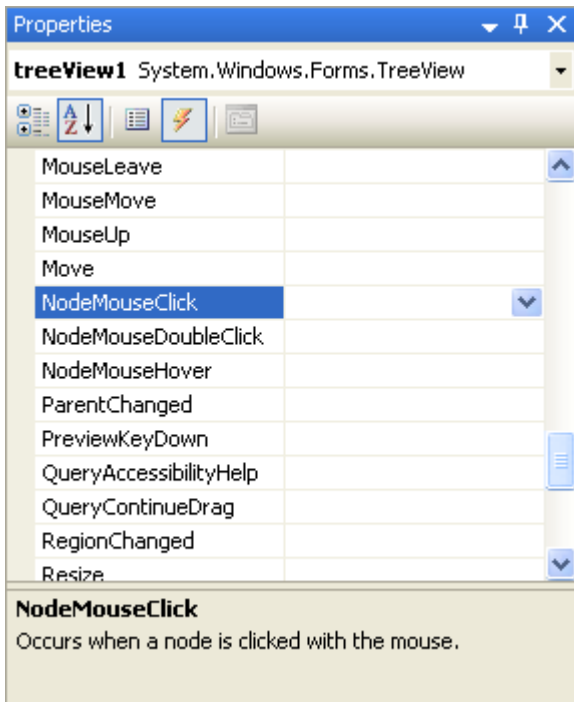
Træk en WebBrowser ud i det andet panel i din SplitContainer. Den udfylder som default Panel2. Skulle det ikke ske, kan du bruge Dock betingelsen og sætte den til Fill.

Det meste af det du har brug for at gøre med en WebBrowser sker gennem dets Methods, snare end gennem dets betingelser. Der er derfor ikke meget mere vi skal gøre med dette kontrolelement!

Koden til kontrolelementet TreeView

Det vi gerne vil have vores WebBrowser til at gøre er at hente den hjemmeside frem, der bliver klikket på i TreeView. Det kan vi bruge et event til, der hedder NodeMouseClicked.

I Design View klikker du på kontrolelementet TreeView. Klik derefter på lynet for at se de tilhørende events:



Dobbeltklik på NodeMouseClick for at se den tilhørende kode:

```
private void treeView1_NodeMouseClick(object sender, TreeNodeMouseClickEventArgs e)
{
    |
}
```

Tilføj følgende kode mellem parenteserne:

```
TreeNode nName = e.Node;
string webPage = "";

webPage = (string)nName.Tag;

if (webPage != "")
{
    webBrowser1.Navigate(webPage);
}
```

Koden opretter en ny TreeNode og tilgår Tag betingelsen. Det er her vi opbevare vores web adresser. Din WebBrowser har en Navigate method, der bruges til at finde hjemmesiden frem. Vi overfører simpelthen web adressen til Tag betingelsen.

Koden kaldes hver gang der klikkes på en Node i træet og dermed startes eventet NodeMouseClick. Kør dit program og test det. Du har en simpel browser.

Øvelse

Tilføj en tekstboks og en knap til din formular. Skriv kode således at når der klikkes på knappen vil browseren hente den hjemmeside ind du har indtastet i din tekstboks.