












## Hedefler

Bu üniteyi çalıştıktan sonra;

-  PullDown (Aşağı Açılır) Menüleri öğrenmek,
-  MenuStrip Kontrolünü öğrenmek,
-  ToolStrip Kontrolünü öğrenmek,
-  ToolTip Kontrolünü öğrenmek,
-  StatusStrip Kontrolünü öğrenmek,
-  RichTextBox Kontrolünü öğrenmek,
-  ContextMenuStrip Kontrolünü öğrenmek,
-  NotifyIcon Kontrolünü öğrenmek,
-  Çoklu Formlarla çalışmayı öğrenmek,

### *İçindekiler*

- Menüler,
- Menü Elemanları,
- PullDown (Aşağı Açılır) Menüler,
- MenuStrip Kontrolü,
- ToolStrip Kontrolü,
- ToolTip Kontrolü,
- StatusStrip Kontrolü.
- RichTextBox Kontrolü,
- ContextMenuStrip Kontrolü,
- NotifyIcon Kontrolü,
- Çoklu Formlar,
- Çoklu Doküman Arayüzü (Multiple Dokument Interface (MDI))

## Menü Tasarımı

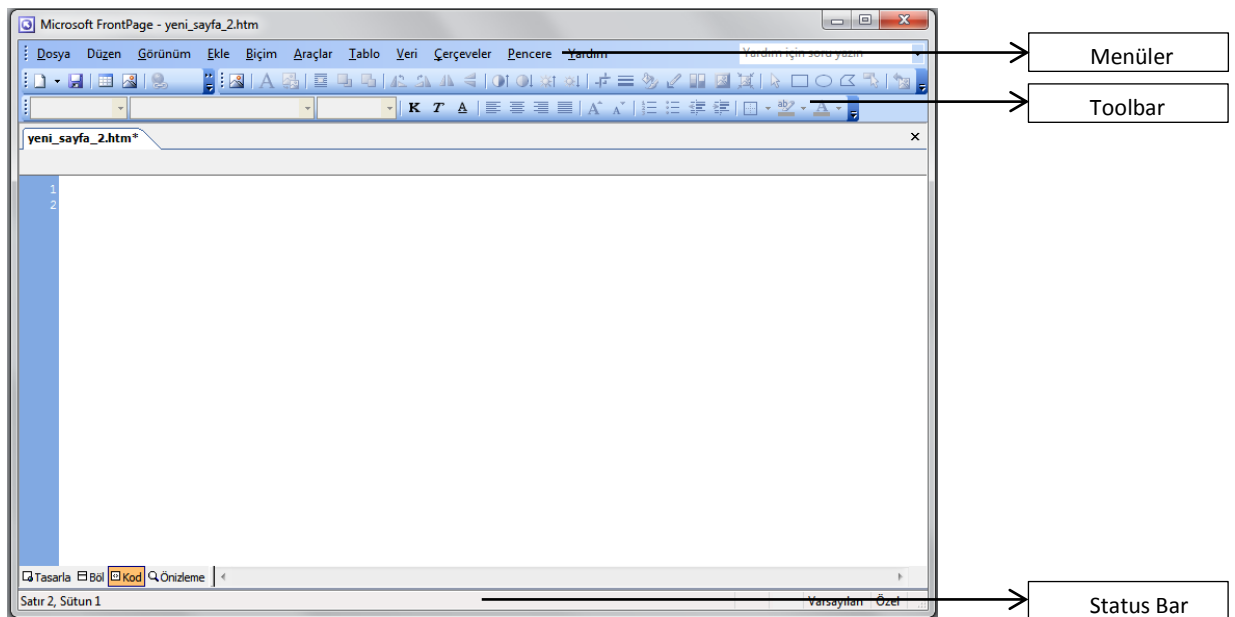
### Menüler

Şu ana kadar bir çok küçük uygulama geliştirdik. Fakat hiçbiri gerçek uygulamalar gibi görünmemektedir. Yani hepimizin kullandığı bir Word, Excel gibi uygulamalar değil. Peki ama neden? Geliştirilen uygulamaların hiçbiri menü çubuklarına sahip değildir.

Bu durumu düzeltmek kolaydır. Çünkü Visual Basic .NET'te menüler oluşturmak kolaydır. Menüler, şu ana kadar anlatılan nesneler gibi, "olay tabanlı" biçimde işlerler.

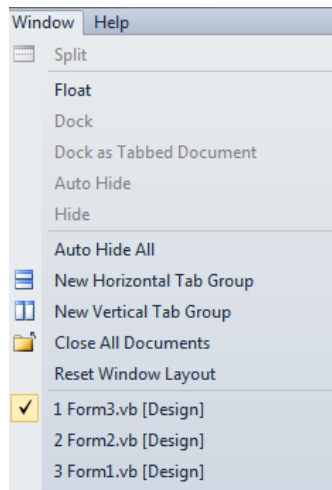
### Menü Elemanları

Günümüzde değişik amaçla geliştirilen uygulamaların birçoğunda, **PullDown** (Aşağı Açılır) menüler, **ToolBar** (Resimli Araç Çubukları) ve **StatusBar** (Durum Çubuğu) bulunmaktadır. Windows işletim sisteminin **WordPad** editör programı üzerinde bu elemanlar gösterilmiştir.



## PullDown (Aşağı Açılır) Menüler

**Hızlı Erişim Tuşları :** Windows tabanlı uygulamalarının çoğunda, fareyi kullanmaksızın, bir menüden komutlar seçmek için klavye kullanılabilir. Bu klavye kısayolları, erişim tuşları olarak adlandırılırlar ve genellikle, **Alt** tuşunun ardından, menüde altı çizili olan karakterlere basmayı gerektirirler. Eğer çakışma yoksa erişim tuşu olarak menü öğesinin başlığının ilk karakteri kullanılır.



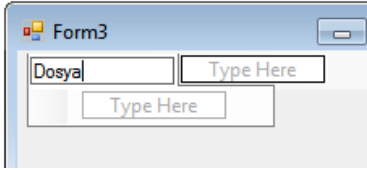
**Kısayol Tuşları :** Bu tuşlar genellikle **Ctrl+Bir harf** şeklinde kullanılır. Herhangi bir menü seçeneğini işleme hızlı bir şekilde sokmak için kullanılır. Örneğin **WordPad'de Cut** menü seçeneğinin kısayol tuşu **Ctrl+X** şeklindedir.

**Check Marks (Kontrol İşareti) :** Bazı menü seçenekleri bir araç çubuğunun gözükmüp gözükmemesini veya herhangi 2-3 değerden birisinin aktif olmasını sağlayabilir. Bu durumlarda menü seçeneğinin sol tarafına **Check Marks (✓)** işareti koyulur veya kaldırılır. Örneğin **Microsoft Visual Studio' da Window** menüsünde **Form3.vb[Design]** seçeneğinde bu işaret var (yani **Form3** gözükecek), fakat **Form2.vb** ve **Form1.vb** seçeneklerinde yok (bunlar gözükmeyecek).

## MenuStrip Kontrolü

**PullDown** (Aşağı açılır) menüleri oluşturmak için kullanılan bir kontroldür. Form üzerine eklenmeyip Visual Studio IDE penceresinin alt tarafındaki

panele eklenir. Visual Basic .NET' le menü oluşturma işi çok hızlı ve kolay yapılabilecek hale getirilmiştir.



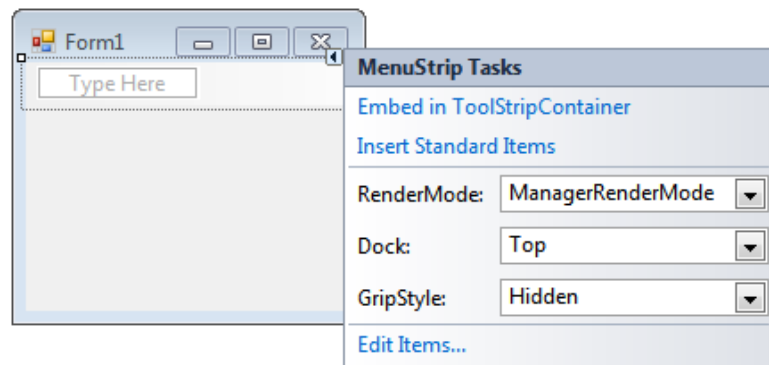
Form üzerine **MenuStrip** kontrolü yerleştirildikten sonra, **Pulldown** menü seçeneklerini düzenlemek çok kolaydır. Menü seçenekleri yazılırken, otomatik olarak olabilecek seçenekler sağda ve altta **Type Here**

şeklinde gözükür. Kullanıcı da geliştireceği uygulamaya göre menü seçeneklerini buraya yazar. Menü seçeneklerinin özellikleri, diğer kontrol özellikleri gibi **Properties** (Özellikler) penceresinden değiştirilebilir.

Properties (Özellikler)	
<b>Name</b>	<b>MenuStrip'</b> in ismi.
<b>Checked</b>	True olması durumunda menü seçeneğinde check işareti olur.
<b>Enabled</b>	True olması durumunda menü seçeneği seçilebilir. False durumunda menü seçeneği silik olarak gözükcektir ve seçilemeyecektir.
<b>Image</b>	Menü elemanı ile ilgili görüntülenecek olan resim dosyasını eklemek için kullanılır.
<b>ShortcutKeys</b>	Kısayol tuşunu belirlemek için kullanılan özelliktir. Eğer menü seçeneğine bir kısayol tuşu atanırsa, bu menü seçeneğinin sağ tarafında gözükcektir.
<b>ShowShortcutKeys</b>	Menü elemanları ile birlikte kısayol tuşlarının kullanılabilmesini sağlar.
<b>Text</b>	Menü seçeneğinin üzerinde gözükcek metin. Hızlı erişim tuşu (yani Alt + ...) ile kullanılacaksa, bu karakterin önüne & karakteri koyulmalıdır.
<b>ToolTipText</b>	Menü elemanı ile ilgili görüntülenecek açıklamalar bu özellik yardımıyla belirtilir.
<b>Visible</b>	Bu menü seçeneğinin gözüküp gözükmeyeceğini belirler.

## MenuStrip Tasks

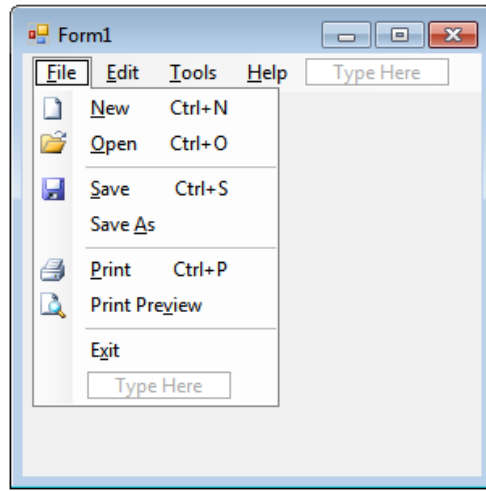
**MenuStrip** kontrolü seçili iken formun sağ üst köşesinde bulunan kulakçık tıklandığında **MenuStrip Tasks** menüsü görüntülenir.



Bu menüde ;

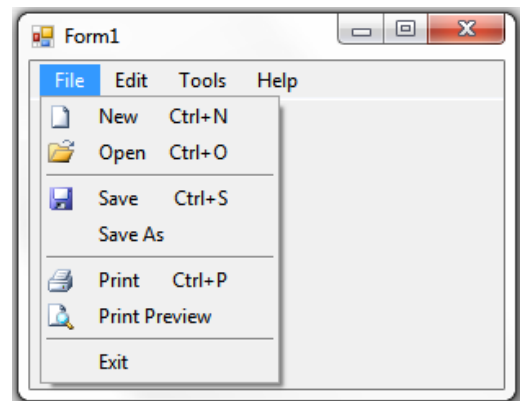
**Embed in ToolStripContainer** : Menüü *ToolStripContainer* içerisine yerleştirir.

**Insert Standart Items** : Alt menüleri ile birlikte (*File, Edit, Tools, Help*) hazır standart bir menü ekler.

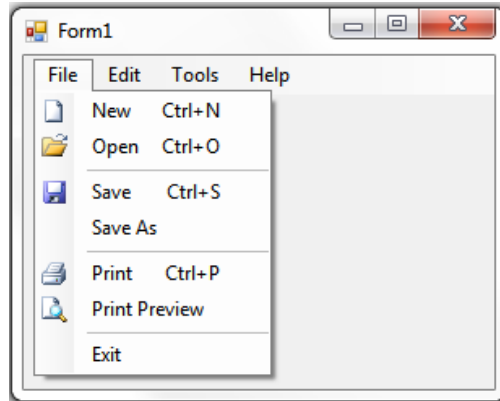


**Render Mode** : Menü çubuğunun görünümü ile ilgili ayarlar yapılır.

1. **System** : Menü çubuğunun Windows görünüm ayarları görüntülenmesini sağlar.

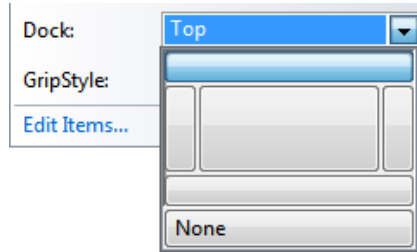


2. **Professional** : Menü çubuğunun Office uygulamaları görünümü şeklinde görüntülenmesini sağlar.

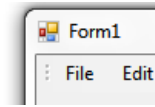


3. **ManagerRenderMode** : Özel menü stilleri tanımlamak için kullanılır.

**Dock** : Menü'nün form üzerindeki konumunu ayarlamak için kullanılır.



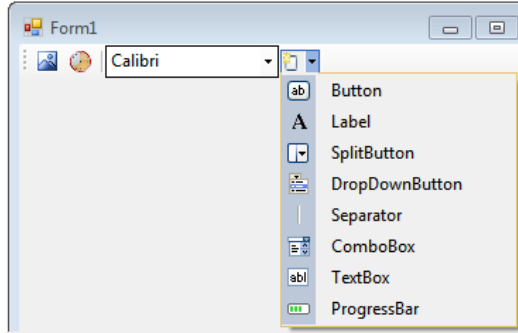
**GripStyle** : **Hidden** ve **Visible** olmak üzere iki farklı değer alabilir. **Hidden**, Menü çubuğunun kilitli olduğu anlamındadır. **Visible** ise kilitli olmadığı anlamındadır ve menü çubuğunun sol tarafında dikey olarak üç nokta gözükür.



## ToolStrip Kontrolü

Bugün kullanılan hemen hemen tüm Windows uygulamalarında **ToolBar** (Araç Çubuğu) vardır. Resimli butonlar, Açılır kutular şeklinde olan **ToolBar'** lar genellikle uygulamalarda çok kullanılan menü seçeneklerine kolay erişimi sağlarlar. Form üzerine eklenmeyip Visual Studio IDE penceresinin alt tarafındaki panele eklenir. Uygulamalarımıza **ToolBar** ekleyebilmek için **ToolStrip** seçeneği kullanılır.





Form üzerine **ToolStrip** kontrolü yerleştirildikten sonra, Form' un üzerinde çıkan **ToolBar** kısmında düğmeleri koyabilmemiz için açılır bir buton çıkacaktır. Bu açılır buton yardımı ile koymak istediğiniz araç çubuğu tipini seçebilirsiniz (**Button, Label, SplitButton** vs.). Araç çubuklarının özellikleri, diğer

kontrol özellikleri gibi **Properties** (Özellikler) penceresinden değiştirilebilir.

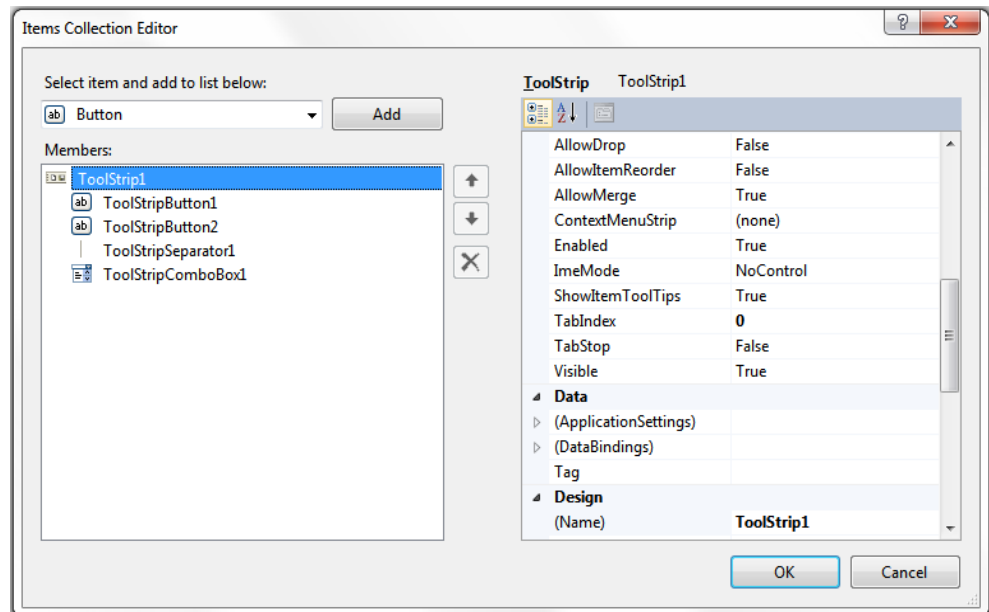
#### Properties (Özellikler)

<b>Name</b>	<b>ToolStrip</b> ' in ismi.
<b>Items</b>	Araç çubuğu' na düğmeleri eklemek, silmek ve özelliklerini değiştirmek için kullanılır.

#### Events (Olaylar)

<b>ButtonClick</b>	<b>ToolBar</b> butonlarından birine tıklanması olayı.
--------------------	---

**Items** özelliği, **ToolBar** üzerindeki her bir düğmeyi tanımlamak için kullanılır. **Items** özelliğinin yanındaki "..." düğmesi tıklandığında, **Items Collection Editor** diyalog kutusu gözükecektir. Bu editör yardımıyla, düğmeler eklenebilir, silinebilir, özellikleri değiştirilebilir. Bir düğme eklemek için yukarıdan düğme türü seçilip **Add** butonuna, silme için ise **"X"** butonuna tıklanır.





## ToolTip Kontrolü

**ToolTip** kontrolü ile uygulamadaki herhangi bir kontrole **ToolTip** (ipucu) ekleyebiliriz. Form üzerine eklenmeyip Visual Studio IDE penceresinin alt tarafındaki panele eklenir.



Properties (Özellikler)	
<b>Name</b>	<b>ToolTip'</b> in ismi.
<b>Active</b>	Kontrol üzerinde açıklamanın görüntülenip görüntülenmeyeceğini belirler. <b>True</b> görüntülenmesini, <b>False</b> görüntülenmemesini sağlar.
<b>AutomaticDelay</b>	İpucu bilgisinin gösterilme zaman uzunluğunu belirler.

Herhangi bir uygulamaya **ToolTip** kontrolü eklendikten sonra, otomatik olarak diğer kontrollerin özelliklerine de ToolTip özelliği eklenir. Buraya yazılacak ifade, o kontrolün **ToolTip** (ipucu) bilgisi olacaktır.

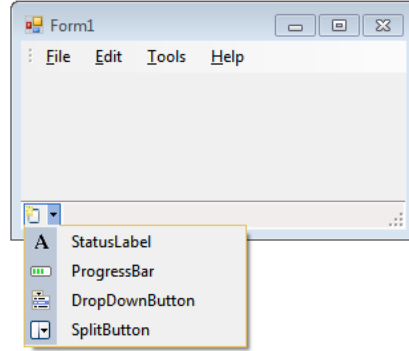
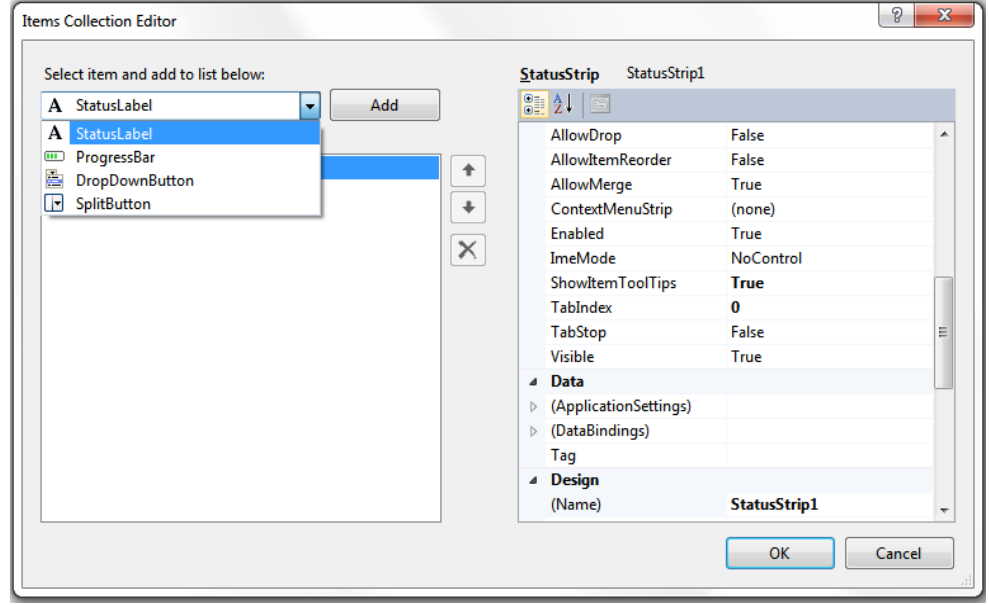
## StatusStrip Kontrolü

**StatusStrip** (Durum çubuğu) Formun altında yer alır ve o andaki işlemle ilgili olarak kullanıcıya bilgi vermek için kullanılır.



Properties (Özellikler)	
<b>Dock</b>	Durum çubuğunun, formun neresinde yer alacağını belirlemek için kullanılır.
<b>Name</b>	<b>StatusStrip'</b> in ismi.
<b>Items</b>	Durum çubuğuna panel eklemek-çıkarmak için kullanılan <b>Items Collection Editor'</b> u çağırmak için kullanılır.

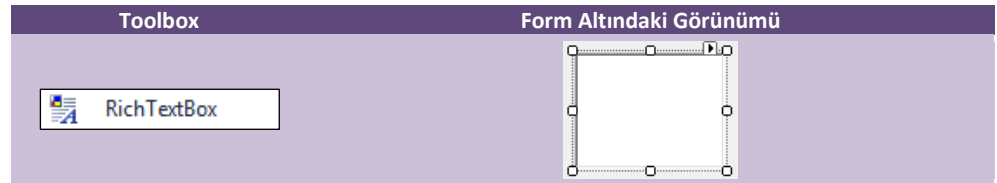
**Items** özelliği, Durum çubuğu üzerindeki panelleri tanımlamak için kullanılır. **Items** özelliğinin yanındaki **"..."** düğmesi tıklandığında, **Items Collection Editor** diyalog kutusu gözükecektir. Bu editör yardımıyla, paneller eklenebilir, silinebilir, özellikleri değiştirilebilir. Bir düğme eklemek için yukarıdan düğme türü seçilip **Add** butonuna, silme için ise **"X"** butonuna tıklanır.



**StatusStrip** üzerine; StatusLabel, ProgressBar, DropDownButton ve SplitButton eklenebilir.

## RichTextBox Kontrolü

**RichTextBox** kontrolü kullanıcıya bilgi girme, düzeltme, bilgiyi daha fazla formatlama özelliklerinin uygulanması imkanı sağlar. **TextBox** kontrolünün gelişmiş modelidir.



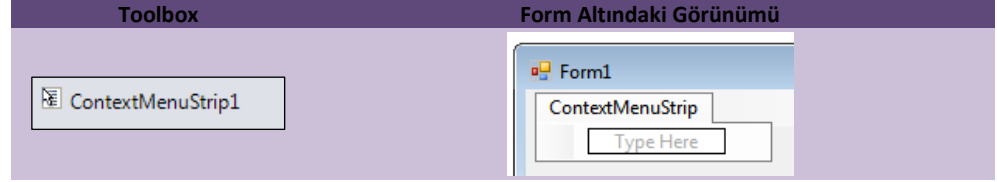
Properties (Özellikler)	
<b>Name</b>	<b>RichTextBox'</b> in ismi.
<b>BackColor</b>	Zemin Rengi
<b>BorderStyle</b>	<b>RichTextBox'</b> in <b>Border</b> (kenarlık) düzenlemesini yapar. 3 farklı seçenek vardır. <b>None</b> : Kenarlık eklenmez, <b>Fixed3D</b> : Sabit (boyutu değiştirilemeyen) 3 boyutlu kenarlık ekler, <b>FixedSingle</b> : Sabit tek çizgili kenarlık eklenir.
<b>Font</b>	Yazı tipi ayarları yapılır. Font diyalog kutusundan yararlanılır.
<b>ForeColor</b>	Yazı rengi
<b>HideSelection</b>	Seçilen bir yazının, başka bir forma veya diyalog kutusuna geçildiğinde, seçim işleminin kalıp kalmayacağını belirler.
<b>Lines</b>	Girilen bilginin saklandığı veya alındığı özelliktir.
<b>Location</b>	<b>RichTextBox'</b> in foruma göre konumu. <b>X;Y</b> olarak verilir. <b>X</b> ; kontrolün formun sol kenarına uzaklığı, <b>Y</b> ; kontrolün formun üst kenarına olan uzaklığıdır.
<b>MaxLength</b>	Kullanıcının bu <b>RichTextBox'</b> a gireceği maksimum karakter sayısı.
<b>MultiLine</b>	<b>True</b> olması <b>RichTextBox</b> içine birden fazla satırlık bilgi girilmesini sağlar.
<b>ReadOnly</b>	Bu özelliğin <b>True</b> olması durumunda, kullanıcı programın çalışması sırasındaki buradaki bilgiyi değiştiremez.
<b>ScrollBars</b>	<b>MultiLine</b> (çoklu satır) özelliğinin <b>True</b> olması durumunda işlerlik kazanan bir özelliktir. <b>RichTextBox</b> alanı içinde bilginin <b>Horizontal</b> (yatay), <b>Vertical</b> (düşey), <b>Both</b> (her ikisi) veya <b>None</b> (hiçbirisi) şeklinde kaydırılmasını sağlar.
<b>Size</b>	<b>RichTextBox'</b> in boyutları. <b>Width</b> (genişlik); <b>Height</b> (yükseklik).
<b>Text</b>	<b>RichTextBox'</b> ta görüntülenecek bilgi.

Methods (Metodlar)	
<b>AppendText</b>	<b>RichTextBox'</b> a bilgi eklemek için kullanılır.
<b>Clear</b>	<b>RichTextBox'</b> taki tüm bilgiyi siler.
<b>Copy</b>	<b>RichTextBox'</b> taki bilgiyi kopyalar.
<b>Cut</b>	<b>RichTextBox'</b> taki bilgiyi keser.
<b>Focus</b>	Kursörü belirtilen <b>RichTextBox'</b> a konumlandırır.
<b>LoadFile</b>	Bir dosyanın içeriğini kontrole yükler.
<b>Paste</b>	Clipboard' daki bilgiyi yapıştırır.
<b>SaveFile</b>	Kontrol içindeki bilgiyi tüm format özellikleri ile bir dosyaya kaydeder.
<b>Undo</b>	Son düzenleme işlemini geri alır.

Events (Olaylar)	
<b>Click</b>	Kullanıcı tarafından <b>RichTextBox'</b> in tıklanması
<b>GotFocus</b>	Farenin, <b>RichTextBox'</b> a konumlanması.
<b>KeyDown</b>	Klavye'den bir tuşa basılması. Basılan tuşun KeyCode değeri belirlenir.
<b>KeyPress</b>	Klavye'den bir tuşa basılması. Basılan tuşun ASCII değeri belirlenir.
<b>LostFocus</b>	Kullanıcının <b>RichTextBox'</b> ı terketmesi.
<b>TextChanged</b>	Yazı üzerinde değişiklik olması.

## ContextMenuStrip Kontrolü

Herhangi bir kontrol üzerinde iken, farenin sağ tuşuna tıklandığında, **PullDown** (Aşağı açılır) menüleri oluşturmak için kullanılan bir kontroldür. **ContextMenuStrip** kontrolü belirli bir kontrole bağlandığında, o kontrole sağ tıklanması halinde geçerli olmaktadır.



Form üzerine **ContextMenuStrip** kontrolü yerleştirildikten sonra, **PullDown** menü seçenekleri, **MainMenuStrip** kontrolündeki gibi yapılıdır. Menü seçenekleri yazılırken, otomatik olarak olabilecek seçenekler sağda ve altta **Type Here** şeklinde gözükür. Kullanıcı da geliştireceği uygulamaya göre menü seçeneklerini buraya yazar. Menü seçeneklerinin özellikleri, diğer kontrol özellikleri gibi **Properties** (Özellikler) penceresinden değiştirilebilir.

Properties (Özellikler)	
<b>Name</b>	<b>ContextMenuStrip</b> ' in ismi.
<b>Checked</b>	<b>True</b> olması durumunda menü seçeneğinde check işareti olur
<b>Enabled</b>	<b>True</b> olması durumunda menü seçeneği seçilebilir. <b>False</b> durumunda menü seçeneği silik olarak gözükecektir ve seçilemeyecektir.
<b>Items</b>	Açılır Menü' ye seçenekleri eklemek, silmek ve özelliklerini değiştirmek için kullanılır.
<b>ShortcutKeys</b>	Kısayol tuşunu belirlemek için kullanılan bir özelliktir. Eğer menü seçeneğine bir kısayol tuşu atanırsa, bu menü seçeneğinin sağ tarafında gözükecektir.
<b>Text</b>	Menü seçeneğinin üzerinde gözükecek metin. Hızlı erişim tuşu (yani Alt + ... ) ile kullanılacaksa, bu karakterin önüne & karakteri koyulmalıdır. Örneğin &Dosya gibi. D harfi altı çizili gözükecektir. Alt+D tuşu ile bu menüye klavyeden erişilebilecektir.
<b>Visible</b>	Bu menü seçeneğinin gözükp gözükmeyeceğini belirler.

## NotifyIcon Kontrolü

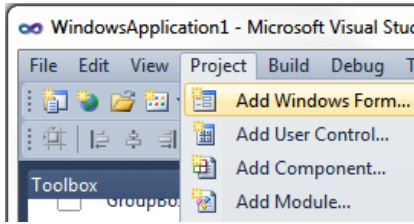
Windows işletim' sisteminde ekranın sağ alt köşesinde program-ikonları yer alır. Mesela **TR** veya **EN** yazan ikon, veya virüs programı kullanılıyorsa, virüs programı ikonu veya **SQL Server** ikonu gibi. Burada yer alan ikonların gösterdiği programlar aslında bilgisayarın hafızasında çalışıyorlar, fakat biz göremiyoruz. İkonların üzerine tıkladığımızda, programlar karşımıza çıkar. İşte **NotifyIcon** kontrolü, bizim geliştirdiğimiz programları da benzer şekilde ikon haline getirip ekranın sağ alt köşesine yerleştirebilir ve istediğimizde bu ikonu tıklayarak programı ekrana getirebiliriz.



Properties (Özellikler)	
<b>Name</b>	<i>NotifyIcon</i> un ismi.
<b>ContextMenu</b>	Mutlaka bir <b>ContextMenuStrip</b> kontrolü daha önceden yerleştirilmiş olmalıdır. Bu menünün seçenekleri, sağ alt köşede ikonun üstündeyken farenin sağ tuşuna basınca çıkacaktır. Seçilen menü seçeneğine göre işlem yapılacaktır.
<b>Icon</b>	Sağ alt köşede görülecek ikon. Bir ikon dosyası buraya eklenmelidir.
<b>Text</b>	Farenin ikon üstünde dolaşırken göstereceği <b>ToolTip</b> (ipucu) bilgisi.
<b>Visible</b>	<b>True</b> değeri programımızı sağ alt köşede ikon haline getiriyor.

## Çoklu Formlar

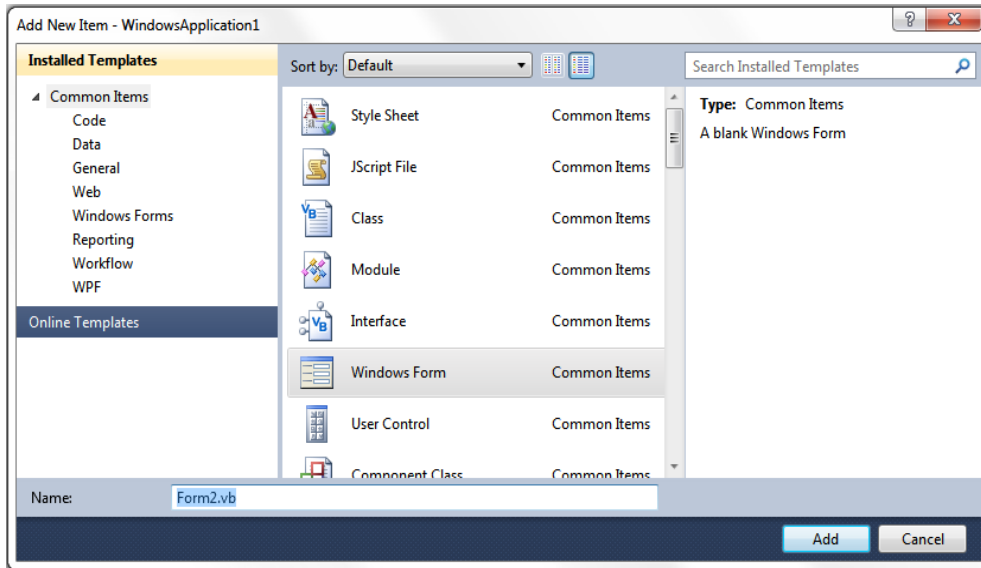
Şu ana yaptığımız uygulamalarda hep tek form kullandık. Fakat çoğu Windows uygulamasında olduğu gibi, bizim de geliştireceğimiz uygulamalarda ikinci, üçüncü, .... formlara ihtiyacımız olacaktır.



Yeni bir formu uygulamamıza nasıl ekleriz?

**Project** menüsünde **Add Windows Form** seçilir. Yeni formun ismini isteyen bir diyalog kutusu ekranda görülecektir. **Name** (isim) yazıldıktan sonra, **Open** butonuna basılarak yeni formun uygulamaya eklenmesi sağlanmış olur. Yeni

eklenen form, hemen **Solution Explorer** penceresinde listelenecektir.

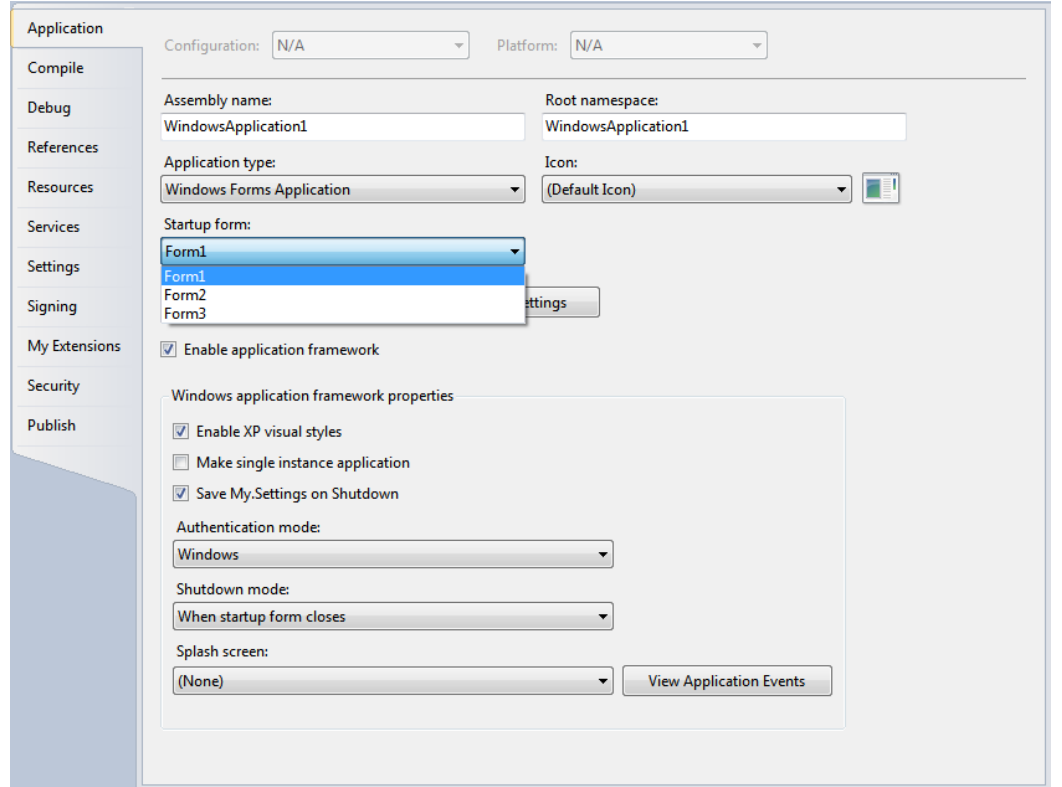
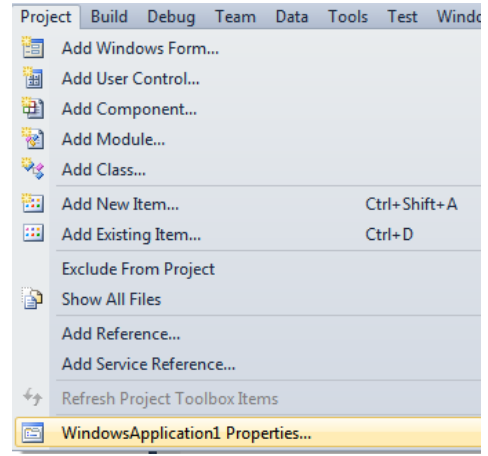


Üçüncü, dördüncü, kısacası ne kadar form gerekiyorsa, benzer şekilde ekleyebiliriz.

Şu ana kadar tek form üzerinde yaptığımız gibi, bu formlar üzerinde kullanıcı arabirimi tasarlanır, kontrollere özellikleri atanır ve kontrollere kod eklenir.

Herhangi bir form uygulamadan kaldırılmak istenirse, **Solution Explorer** penceresinde ilgili formun üzerinde farenin sağ tuşuna basılır. Çıkan menüde **Delete** seçilerek **Form** uygulamadan kaldırılır.

Çoklu formun olduğu uygulamalarda en önemli problem, uygulamanın hangi formla başlayacağıdır. Bunu belirlemek için, **Project** menüsünden **Properties** seçeneği kullanılır. Aşağıdaki diyalog kutusu ekranda gözükecektir. Burada **Startup object** açılan menüde, hangi formla uygulamaya başlayacaksa, o seçilir.



Uygulama çalıştığında seçilen formla uygulama başlar. Peki diğer formları uygulamaya nasıl getireceğiz? Veya formları uygulamada nasıl kapatacağız?

Bir formu uygulamada göstermek için 2 yol vardır :

1. **Form\_ismi.ShowDialog()** : Form bu şekilde uygulamada gösterildiğinde, form açıkken diğer formlara erişmek mümkün değildir. Yani bu diyalog kutusuna benzer.
2. **Form\_ismi.Show()** : Form bu şekilde uygulamada gösterildiğinde, form açıkken diğer formlara erişmek mümkündür. Formlar tıklanarak aktif hale getirilebilir.

Bir formu kaldırmak için de 2 yol vardır:

1. **Me.Close()** : Form bu şekilde kapatılır. Bu formu hafızadan da siler.
2. **Form\_ismi.Hide()** : Form bu şekilde gizlenir. Ama hafızadan silinmez.

## Çoklu Doküman Arayüzü (Multiple Document Interface (MDI))

Bir önceki bölümde anlattığımızdan farklı olarak, Visual Basic.NET bize çoklu form sistemi sunmaktadır. Bu **Multiple Document Interface (MDI)** (Çoklu Doküman Arayüzü) olarak bilinir. Bu sistemde bütün formlar bir tek formun içinde kontrol edilir. Çoklu formlara en iyi bilinen örnek olarak Word, Excel programları gösterilebilir.

Bir **MDI** uygulaması, kullanıcıya aynı anda birçok formu gösterme imkanı vermektedir. **MDI** uygulaması bir **MDI** formu ve bunun içinde çok sayıda alt formdan oluşmaktadır. **MDI** formuna **Parent** (ebeveyn) form, içindeki formlara ise **Child** (çocuk) form denir.

**MDI** formların üzerinde buton gibi kontroller yer alamaz. Bu formlar genellikle menüler aracılığıyla yönetilirler. Bir **MDI** uygulaması oluşturmanın 2 aşaması vardır:

1. Önce **MDI** formu oluşturulur. Bu nasıl yapılır? Uygulama başladığında otomatik olarak Form1 oluşturulur. Bu formun **IsMdiContainer** özelliği **True** yapılırsa, artık bu form MDI formu olmuştur.
2. Child formlar, önceki bölümde anlatıldığı gibi, form ekleme şeklinde yapılır. **Project** menüsünde **Add Windows Form** seçilir. Yeni formun ismini isteyen bir diyalog kutusu ekranda gözükecektir. **Name** (isim) yazıldıktan sonra, **Open** butonuna basılarak yeni formun uygulamaya eklenmesi sağlanmış olur. Yeni eklenen **child** form, hemen **Solution Explorer** penceresinde listelenecektir.

**MDI parent** ve **child** formların özellikleri şöyle sıralanabilir:

- a) Programın çalışması sırasında **Child** formlar, **Parent** formunun çalışma alanını kullanırlar.
- b) **Parent** form simge durumuna getirildiğinde görev çubuğunda görülür. **Child** formlar simge durumuna getirildiğinde ise **Parent** form simge durumuna getirildiğinde görev çubuğunda görülür. **Child** formlar simge durumuna getirildiğinde ise **Parent** form içinde görüntülenir.
- c) Bir **Child** form ekranı kapladığında, başlığı **Parent** formunun başlığı olur.
- d) Aktif **Child** formunun menüleri, **Parent** formunun menü çubuğunda görüntülenir.
- e) Çalışma anında yeni **Child** formlar oluşturulabilir.