

T.C.
MİLLİ EĞİTİM BAKANLIĞI



MEGEP

(MESLEKİ EĞİTİM VE ÖĞRETİM SİSTEMİNİN
GÜÇLENDİRİLMESİ PROJESİ)

BİLİŞİM TEKNOLOJİLERİ

GÖRSEL PROGRAMLAMADA ARA YÜZ

ANKARA 2007

Milli Eğitim Bakanlığı tarafından geliştirilen modüller;

- Talim ve Terbiye Kurulu Başkanlığının 02.06.2006 tarih ve 269 sayılı Kararı ile onaylanan, Mesleki ve Teknik Eğitim Okul ve Kurumlarında kademeli olarak yaygınlaştırılan 42 alan ve 192 dala ait çerçeve öğretim programlarında amaçlanan mesleki yeterlikleri kazandırmaya yönelik geliştirilmiş öğretim materyalleridir (Ders Notlarıdır).
- Modüller, bireylere mesleki yeterlik kazandırmak ve bireysel öğrenmeye rehberlik etmek amacıyla öğrenme materyali olarak hazırlanmış, denenmek ve geliştirilmek üzere Mesleki ve Teknik Eğitim Okul ve Kurumlarında uygulanmaya başlanmıştır.
- Modüller teknolojik gelişmelere paralel olarak, amaçlanan yeterliği kazandırmak koşulu ile eğitim öğretim sırasında geliştirilebilir ve yapılması önerilen değişiklikler Bakanlıkta ilgili birime bildirilir.
- Örgün ve yaygın eğitim kurumları, işletmeler ve kendi kendine mesleki yeterlik kazanmak isteyen bireyler modüllere internet üzerinden ulaşılabilirler.
- Basılmış modüller, eğitim kurumlarında öğrencilere ücretsiz olarak dağıtılır.
- Modüller hiçbir şekilde ticari amaçla kullanılamaz ve ücret karşılığında satılamaz.

İÇİNDEKİLER

AÇIKLAMALAR	iii
GİRİŞ	1
ÖĞRENME FAALİYETİ-1	3
1. MDI VE SDI UYGULAMALAR	3
1.1. “Show, Hide, BringToFront, SendToBack” Metotları	3
1.1.1. Show()	3
1.1.2. Hide ().....	4
1.1.3. BringToFront ve SendToBack.....	5
1.2. “Project*Uygulamaadi Properties” Penceresi	7
1.3. SDI (Single Document Interface) ve MDI (Multiple Document Interface)	12
1.3.1. SDI Formlar	12
1.3.2. MDI Formlar.....	13
1.4. Windows Gezgini Tarzı Uygulamalar (Çok Panelli).....	18
UYGULAMA FAALİYETİ	20
ÖLÇME VE DEĞERLENDİRME	21
ÖĞRENME FAALİYETİ-2	22
2. MENÜ TASARIMI.....	22
2.1. “MainMenu” Nesnesi	22
2.2. “Insert New, Insert Separator ve Edit names” Menü Komutları	24
2.3. Menüye İmage Ekleme, Kısayol Atama ve Hazır Menü Kullanma	27
2.3.1. Menüye İmage Ekleme	27
2.3.2. Kısayol Atama	28
2.3.3. Hazır Menü Kullanma	29
2.4. Menü Komutları İçin Kod Yazma ve Menüler için Alt Menü Hazırlama	30
2.4.1. Menü Komutları için Kod Yazma	30
2.4.2. Menüler için Alt Menü Hazırlamak.....	31
2.5. Menü Elemanlarının Genel Özellikleri (text, name, checked, radio checked, shortcut, enabled, visible)	31
2.5.1. Komutları Seçili Duruma Getirmek ve Gruplamak	31
2.5.2. Komutları Gizlemek ve Kullanılmaz Hale Getirmek	33
UYGULAMA FAALİYETİ	34
ÖLÇME VE DEĞERLENDİRME	35
ÖĞRENME FAALİYETİ-3	36
3. YARDIMCI NESNELER	36
3.1. ToolStrip Nesnesi	36
3.1.1. Komutlar için Kod Yazma.....	40
3.2. StatusStrip Nesnesi	42
3.3. Context Menü Hazırlamak.....	46
3.4. ToolTip Nesnesi	48
3.5. Push Button, DropDown Menu Özelliği	52
3.5.1. PushButton (Basılır Düğme).....	52
3.5.2. DropDownMenu Özelliği	58
3.6. Diyalog Kutuları	59
UYGULAMA FAALİYETİ	63
ÖLÇME VE DEĞERLENDİRME	64

ÖĞRENME FAALİYETİ-4	65
4. YARDIM.....	65
4.1. Yardım Menüsü	65
4.2. “HTML Help Workshop” Programı	68
4.3. “HelpProvider” Nesnesi (F1 Tuşu Yardımı)	75
4.3.1. Nesnelerin “HelpString” Özellikleri.....	78
UYGULAMA FAALİYETİ	80
ÖLÇME VE DEĞERLENDİRME	81
MODÜL DEĞERLENDİRME	82
CEVAP ANAHTARLARI	84
ÖNERİLEN KAYNAKLAR.....	85
KAYNAKÇA	86

AÇIKLAMALAR

KOD	482BK0067
ALAN	Bilişim Teknolojileri
DAL/MESLEK	Veri Tabanı Programcılığı
MODÜLÜN ADI	Görsel Programlamada Ara Yüz
MODÜLÜN TANIMI	Görsel programlamada ara yüz hazırlama ile ilgili öğrenme materyalidir.
SÜRE	40/32
ÖN KOŞUL	Görsel Programlama Kurulumu modülünü bitirmiş olmak
YETERLİK	Görsel ara birim yapmak
MODÜLÜN AMACI	Genel Amaç Bu modül ile, gerekli ortam sağlandığında çok formlu uygulamalar, menü ve araç çubukları yapabileceksiniz. Amaçlar <ol style="list-style-type: none">1. Çok formlu uygulama geliştirebileceksiniz.2. Ana menü hazırlayabileceksiniz.3. Araç çubukları, durum çubuğu ve içerik menüsü hazırlayabileceksiniz.4. Kullanıcı ara yüzünü geliştirip yardım bilgileri ekleyebileceksiniz.
EĞİTİM ÖĞRETİM ORTAMLARI VE DONANIMLARI	Bilgisayar laboratuvarı ve bu ortamda bulunan görsel programlama için gerekli donanıma sahip bilgisayar, lisanslı işletim sistemi programı, kâğıt ve kalem hazır bulundurulmalıdır.
ÖLÇME VE DEĞERLENDİRME	Her faaliyet sonrasında o faaliyetle ilgili değerlendirme soruları ile kendi kendinizi değerlendireceksiniz. Modül içinde ve sonunda verilen öğretici sorularla edindiğiniz bilgileri pekiştirecek, uygulama örneklerini ve testleri gerekli süre içinde tamamlayarak etkili öğrenmeyi gerçekleştireceksiniz. Sırasıyla araştırma yaparak, grup çalışmalarına katılarak ve en son aşamada alan öğretmenlerine danışarak ölçme ve değerlendirme uygulamalarını gerçekleştiriniz.

GİRİŞ

Sevgili Öğrenci,

Programcılıkta karşımıza çıkacak sorunlar hiçbir zaman bizleri yıldırılmamalıdır. Önemli olan kişide öğrenme isteği olmasıdır ve öğrenen kişinin hemen vazgeçmemesidir. Yapılan işlerin hepsinde başarılı olunamayabilir. Ancak gayret, çalışma ve sabır sonrasında tüm sorunların üstesinden gelinebilir. Bu bazıları için kısa bazıları için ise uzun zaman alabilir. Hedefine ulaşmadan pes etmemen gerekir. Ne kadar çok uygulama, araştırma yapar iseniz kendinizi o kadar geliştirirsiniz ve ilerletirsiniz.

Bu modül ile çok formlu uygulamalar oluşturabilecek, Windows uygulamalarında sürekli olarak yararlandığınız menü ve araç çubuklarını kendiniz yapabileceksiniz. Modülü bitirdiğinizde anlamadığınız yerleri tekrar okuyup uygulayınız. Konuları tam olarak kavramadan diğer modüle geçmeyiniz.

Modülü başarı ile tamamlayacağınıza inanıyor ve modülün hayatınızda olumlu değişiklikler yapmasını arzu ediyorum, çalışmalarınızda başarılar diliyorum.

ÖĞRENME FAALİYETİ-1

AMAÇ

Bu modül ile gerekli ortam sağlandığında, çok formlu uygulama geliştirebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

- MDI ve SDI form ne demektir? Araştırınız.
- Ana ve çocuk formlar arasındaki farkları araştırınız.

Araştırma işlemleri için internet ortamı ve görsel programlama dilini anlatan kaynak kitaplardan faydalanınız.

1. MDI VE SDI UYGULAMALAR

1.1. “Show, Hide, BringToFront, SendToBack” Metotları

1.1.1. Show()

Visual Basic .NET’te uygulama çalıştığında seçilen formla uygulama başlar. Peki diğer formları uygulamaya nasıl getireceğiz? Bu sorunun cevabını öğrenmek için öncelikle projemize iki tane form ekleyelim. Form1 projemiz çalıştığında ekrana gelecek olan ilk formumuz olsun ve Form1 ekrana geldikten sonra Form2’yi görüntüleyelim. Form1 üzerine bir buton ekleyiniz ve butonun kod penceresine geçiniz. Kod penceresinde bir formu uygulamada göstermenin iki yolu vardır.

- **Form_ismi.Show ()**: Form bu şekilde uygulamada gösterildiğinde, form açıkken diğer formlara erişmek mümkündür. Formlar tıklanarak aktif hale getirilebilir.

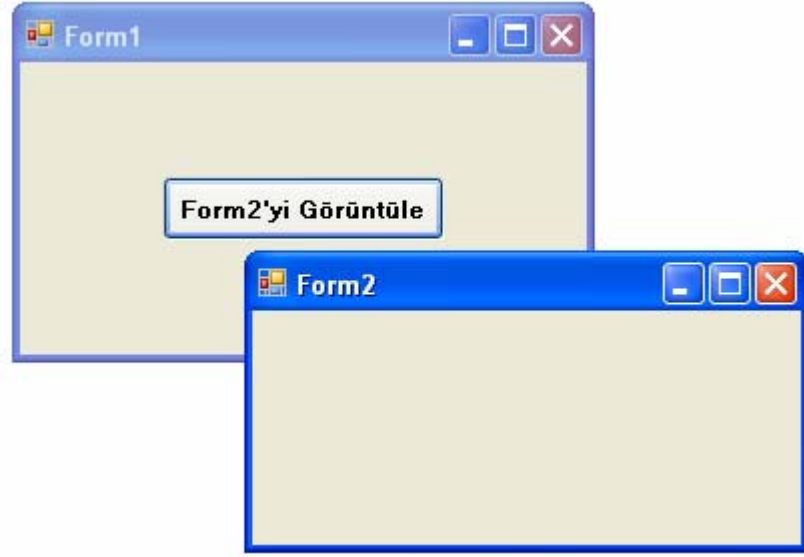
```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click,
```

```
Form2.Show() 'Form2'yi ekrana getirir
```

```
End Sub
```

Gerekli olan komut satırını yazdıktan sonra programınızı çalıştırınız.

Şimdi Form2'yi Görüntüle isimli butona tıklayınız. Form2'nin ekrana geleceğini göreceksiniz.



Resim 1.1: Show() komutunun kullanımı

- **Form_ismi.ShowDialog():** Form bu şekilde uygulamada gösterildiğin de, form açıkken diğer formlara erişmek mümkün değildir. Yani bu diyalog kutusuna benzer.

```
Private Sub Button1_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click,  
  
    Form2.ShowDialog() 'Form2'yi ekrana getirir ama diğer  
    formlara ulaşmaz  
  
End Sub
```

1.1.2. Hide ()

Uygulamalarınızda formları göstermeniz kadar gizlemenizde gerekecektir. Bir formu kaldırmak için de 3 yol vardır :

- **Form_ismi.Close ():** Açılan form kapatılır.

```
Private Sub Button2_Click(ByVal sender As System.Object,  
  
Form2.Close() 'Form2'yi kapatır  
  
End Sub
```

- **Me.Close()** : Bu formu hafızadan da siler.

```
Private Sub Button2_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click,  
    Me.Close() 'Formu kapatır.
```

```
End Sub
```

- **Form_ismi.Hide ()** : Form bu şekilde gizlenir. Ama hafızadan silinmez.

```
Private Sub Button2_Click(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles Button1.Click,ect,
```

```
    Form2.Hide() 'Form2'yi gizler.
```

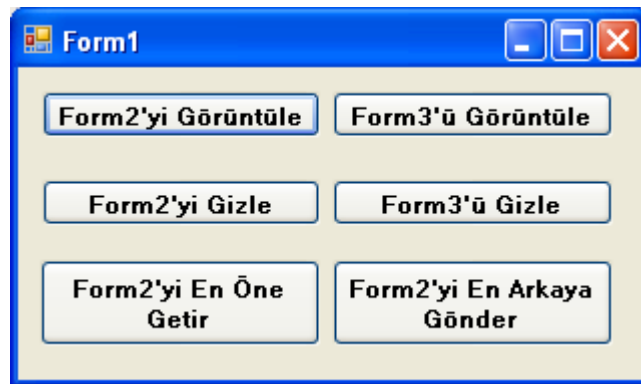
```
End Sub
```

1.1.3. BringToFront ve SendToBack

Uygulamalarımızda birden fazla form bulunuyorsa bu formları en öne getirmek ya da en arkaya gönderip birbiri arasında geçiş yapmanız gerekebilir.

- **Form_ismi.BringToFront ()**: Formu en öne getirir.
- **Form_ismi.SendToBack ()** : Formu en arkaya gönderir.

Şimdi projemize üç tane form ekleyiniz ve Form1'i Resim 1.2'de görüldüğü gibi düzenleyin.



Resim 1.2: BringToFront ve SendToBack özelliklerinin kullanımı

Daha sonra formda bulunan butonların kod kısımlarına aşağıdaki komut satırlarını yazınız.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click,
    Form2.Show() 'Form2'yi gösterir.
```

```
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click,
    Form3.Show() 'Form3'ü gösterir.
```

```
End Sub
```

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click,
    Form2.Hide() 'Form2'yi gizler.
```

```
End Sub
```

```
Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click,
    Form3.Hide() 'Form3'ü gizler.
```

```
End Sub
```

```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click,
    Form2.BringToFront() 'Form2'yi en öne getirir
```

```
End Sub
```


```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click,
    'BU BÖLÜMÜN KODLARINI SİZ YAZMAYA ÇALIŞINIZ.
```

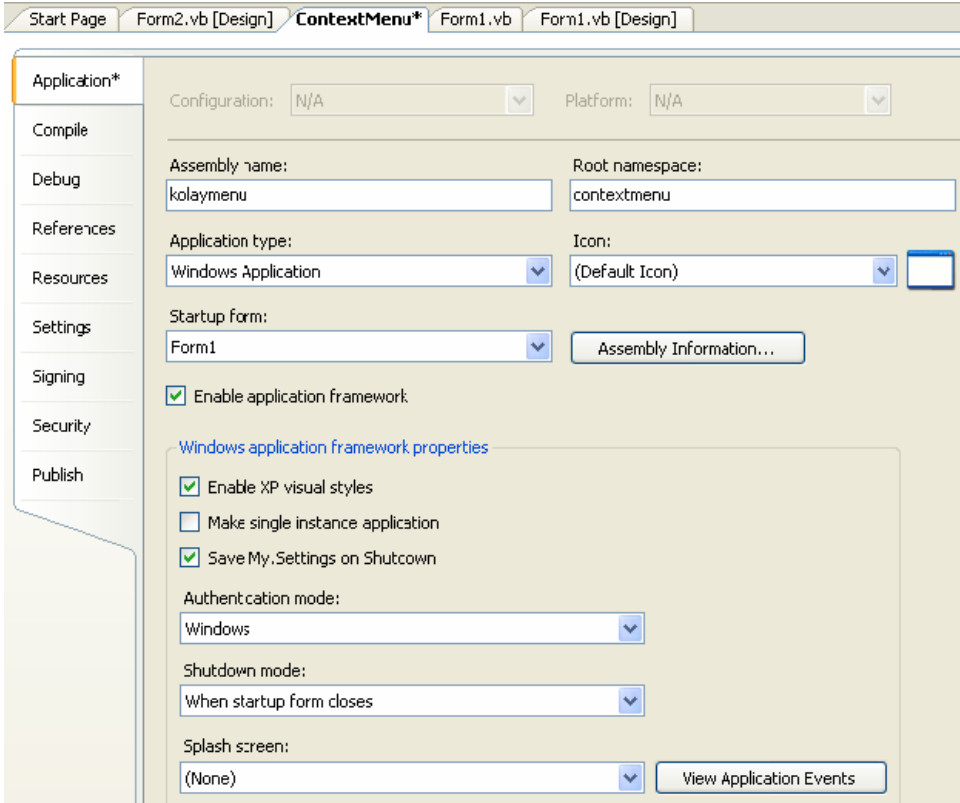
```
End Sub
```

```
End Class
```

Projenizi çalıştırıp sırasıyla butonları tıklayınız. Formlar arasında geçiş yapabildiğinizi göreceksiniz.

1.2. “Project*Uygulamaadi Properties” Penceresi

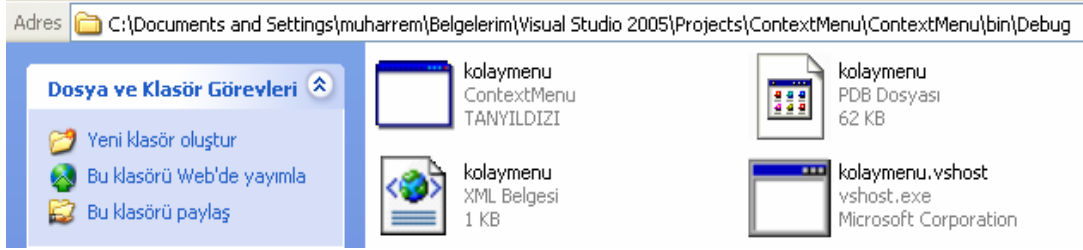
Bu pencerede proje ile ilgili özellikleri ayarlayabiliriz. Örneğin projede birden fazla form varsa çalışmaya hangi formdan başlanacağı, uygulamaya ekleyeceğimiz kaynakların neler olacağı, uygulama dosyalarının nerede yayınlayacağını vb. belirleyebiliriz. Project Properties penceresini açmak için Project menüsünden “Uygulamamızın adı Properties  ContextMenu Properties...” komutunu seçmeliyiz.



Resim 1.3: Project ... properties penceresinin görünümü

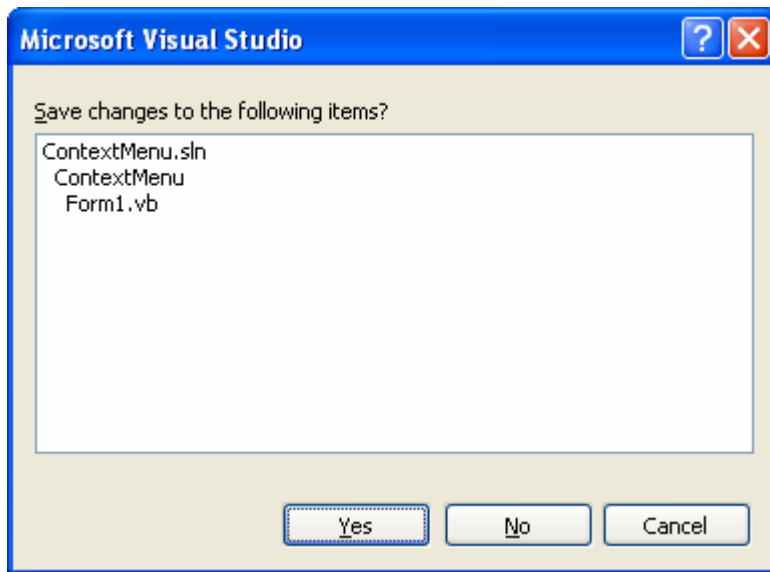
➤ **Application:** Project Properties penceresindeki Application sekmesinde aşağıdaki seçenekler bulunur.

- **Assembly Name:** Derleme uygulamamızda kullanacağımız ismi belirlememize yarar. Windows tabanlı bir uygulama için buraya istediğimiz herhangi bir ismi yazınca programımızın çalıştırılabilir bir uzantısı .exe olur (Kolaymenu yazınca program kolaymenu.exe’yi oluşturacaktır.)



Resim 1.4: Project ... properties ile oluşturulan .exe dosyaları

- **Root namespace:** Projemizdeki tüm dosyalar için taban ismi belirler.
- **Application Type:** Hazırlayacağımız uygulama tipini belirler.
- **Icon:** Derlediğimiz uygulama için Windows'ta gösterilecek icon resmini belirler.
- **Startup:** Projemizde birden fazla form varsa, programı çalıştırdığımızda hangi formun başlangıç formu olacağını belirlememizi sağlar.
- **Enable Visual Styles:** Windows XP stillerinin etkin olup olmayacağını belirler. Etkinse düğme şekilleri Windows XP gibi davranır.
- **Make Single Instance Application:** Bu seçenek aktif ise programımızın iki kez üst üste çalışmasını önler.
- **Save My Settings on Shutdown:** Aktif durumda ise programımızı kapatınca yapılan değişikliklerin kaydedilmesini sağlar.



Resim 1.5: Kaydetme Penceresinin Görünümü

- **Authentication Mode:** Aktif olan kullanıcı kimliğini tanımlar.
- **Shutdown Mode:** Projemizde birden fazla form varsa ve burada When Startup Form Closes seçildiğinde başlangıç formumuzu kapattığımızda programımız sonlanır. When Last Form Closes seçildiğinde ise en son formumuzu kapattığımızda programımız sonlanır.
- **Splash Screen:** Burada başlangıç formu haricinde diğer formlar bulunur. Bu formlardan birini seçip programımızı çalıştırdığımızda ekrana önce belirli bir süre seçilen form gelir. Daha sonra başlangıç formumuz görünür.

➤ **Compile:** Compile sekmesinde aşağıdaki seçenekler bulunur.

- **Build Output Path:** Programımızın .exe'sini oluşturmak istediğimizde dosyanın nereye oluşacağını gösteren kısımdır. İstersek değiştirebiliriz.
- **Option Explicit:** Projelerimizdeki değişkenlerin tanımlama yaparak mı yoksa yapmadan mı kullanabileceğimizi belirler. On durumunda ise tanımlama yaparız, Off durumunda ise değişkenleri tanımlama yapmadan kullanabiliriz.
- **Option Script:** On durumunda İnteger türü veriyi Long türü veriye dönüştürmeye izin verir.
- **Option Compare:** Eğer burada Binary seçilir ise sayıların birbiri ile karşılaştırılmasında şöyle bir sıralama olur.

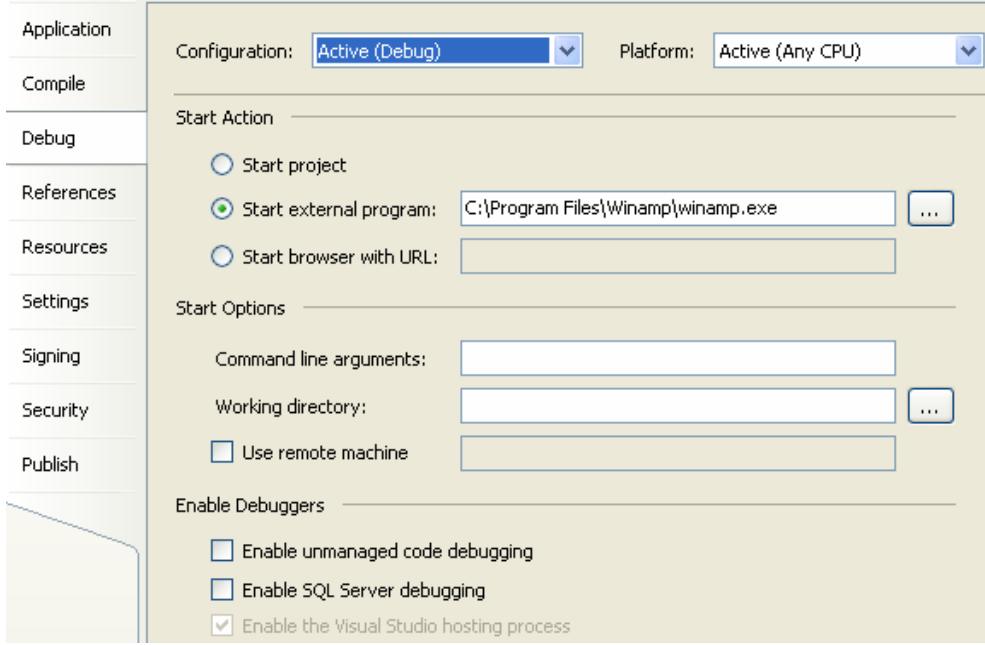
A<B<E<Z<a<b<c<e<z<Á<Ê<Ø<á<ê<ø

Text seçilir ise sıralama değişecektir.

(A=a)<(Á = á)<(B=b)<(E=e)<(Ê = ê)<(Z=z)<(Ø = ø)

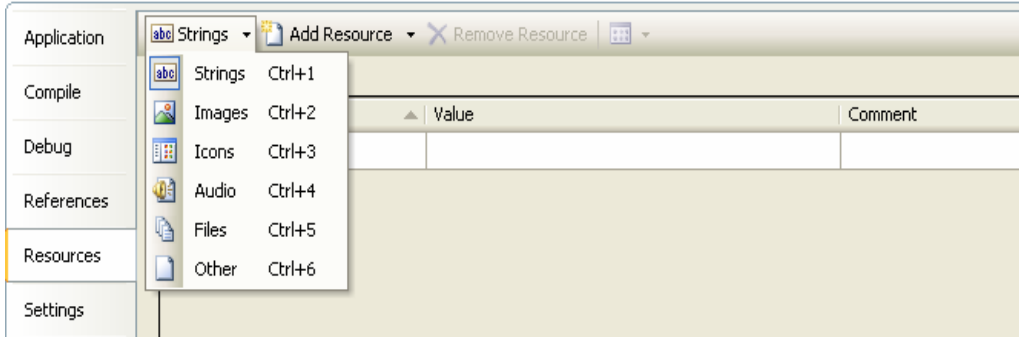
- **Condition ve Notification:** Condition bölümünde yazan isimlerin hata mı yoksa tehlike mi veya hiçbir şey sayılmayacağını belirlememizi sağlar.
- **Disable All Compiler Warnings:** Projemizdeki tüm uyarıları etkisiz hale getirir.
- **Treat All Compiler Warnings As Compilation Errors:** Projemizdeki tüm uyarıların hataymış gibi davranmasını sağlar.
- **Generate XML Documantion File:** XML doküman dosyası üretir.

- **Debug:** Debug sekmesi seçilince karşımıza aşağıdaki pencere ve seçenekler gelir.



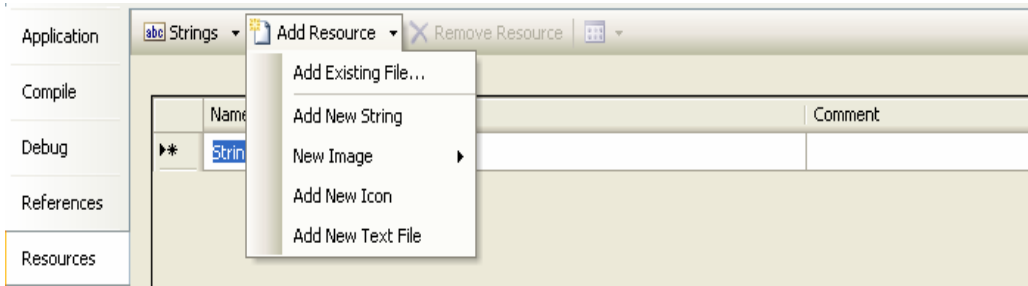
Resim 1.6: Debug penceresinin görünümü

- **Start Project:** Programımızı çalıştırdığımızda sadece projenin çalışmasını sağlar.
 - **Start External Program:** Projemizle beraber çalışmasını istediğimiz bir programın seçilmesini sağlar. Örneğin Winamp seçilince hem projemiz hem de Winamp çalışacaktır.
 - **Start Browser In URL:** Programımız çalıştığında kendi belirleyeceğimiz bir WEB adresinin de açılmasını sağlar.
- **Resources:** Resources sekmesinde aşağıdaki seçenekler bulunur.
- **Strings:** Uygulamamıza ekleyeceğimiz kaynakları belirleyebiliriz. Bunlar string, image, ses dosyası ve icon olabilir.



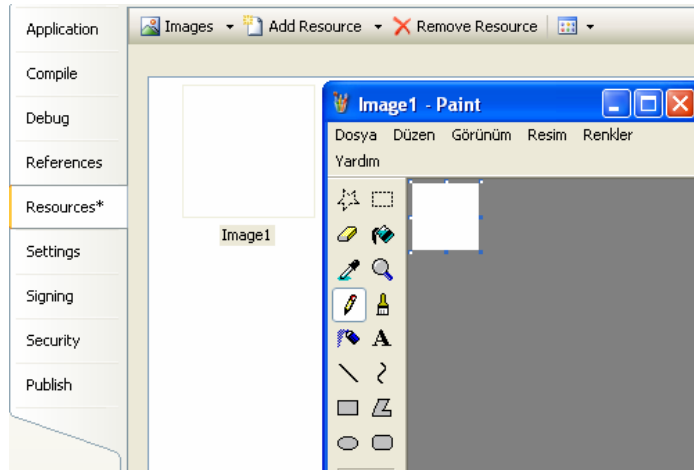
Resim 1.7: Resources/Strings penceresinin görünümü

Bunu yapmak için, önce kaynak izlemesinden nasıl bir kaynak ekleyeceğimizi belirleriz. Bu kısımda kategorilere ayrılmış kaynakları da görebiliriz. Örneğin, bir image dosyası eklediğimizi düşünelim. İlk önce kaynak seçiciden image seçelim.



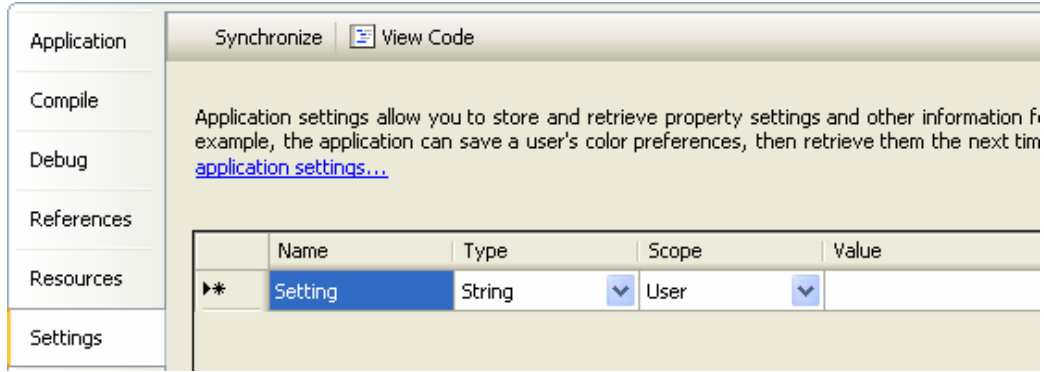
Resim 1.8: Resources/Add Resource penceresinin görünümü

Sonra Add Source (Kaynak Ekle)'den Add Existing File (Var Olan Dosya Ekle) seçeneğini seçtiğimizde dosyanın uzantısına göre VB. NET bu dosyayı uygun kategorilere atayacaktır veya yeni bir image seçtiğimizde bu image oluşturulur ve istersek bu image'nin üzerine gelip Edit yaparak Paint programında düzenleyebiliriz.



Resim 1.9: Resources ve Paint penceresinin görünümü

- **Settings:** Settings sekmesi, uygulama ayarlarımızın daha dinamik olması için özelliklerin ve diğer bilgilerin saklanması ve istenildiğinde tekrar kullanılabilmemize müsaade eder. Örneğin, bir uygulamamızdaki renk tercihlerini kaydedebilir ve bir sonraki uygulamamızda tekrar erişip kullanabiliriz.



Resim 1.10: Settings penceresinin görünümü

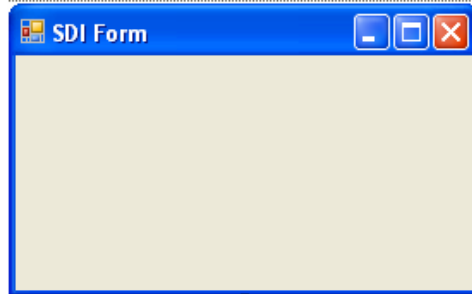
- **Publish:** Bu sekme, uygulama dosyalarımızın nerede yayınlanacağını belirlememizi sağlar. Bu bir internet adresi, FTP adresi veya dosya yolu olabilir.

1.3. SDI (Single Document Interface) ve MDI (Multiple Document Interface)

İki tür esas arayüz vardır. SDI (single document interface) tek formlu arayüzler ve MDI (multiple document interface) çok formlu arayüzler olarak adlandırılırlar

1.3.1. SDI Formlar

Tek başına çalışan formlara SDI form denir. Normalde yeni bir projeye başlarken projede bulunan form ve daha sonra eklenen formlar SDI formlardır. Bu formlar herhangi bir forma bağlı olmaksızın kendi başlarına çalışır. SDI arayüze örnek olarak WordPad programını verebiliriz. WordPad'de bir belge açabilmek için açık olan belgeyi kapatmanız gerekir. Yani tek formlu bir arayüz vardır.

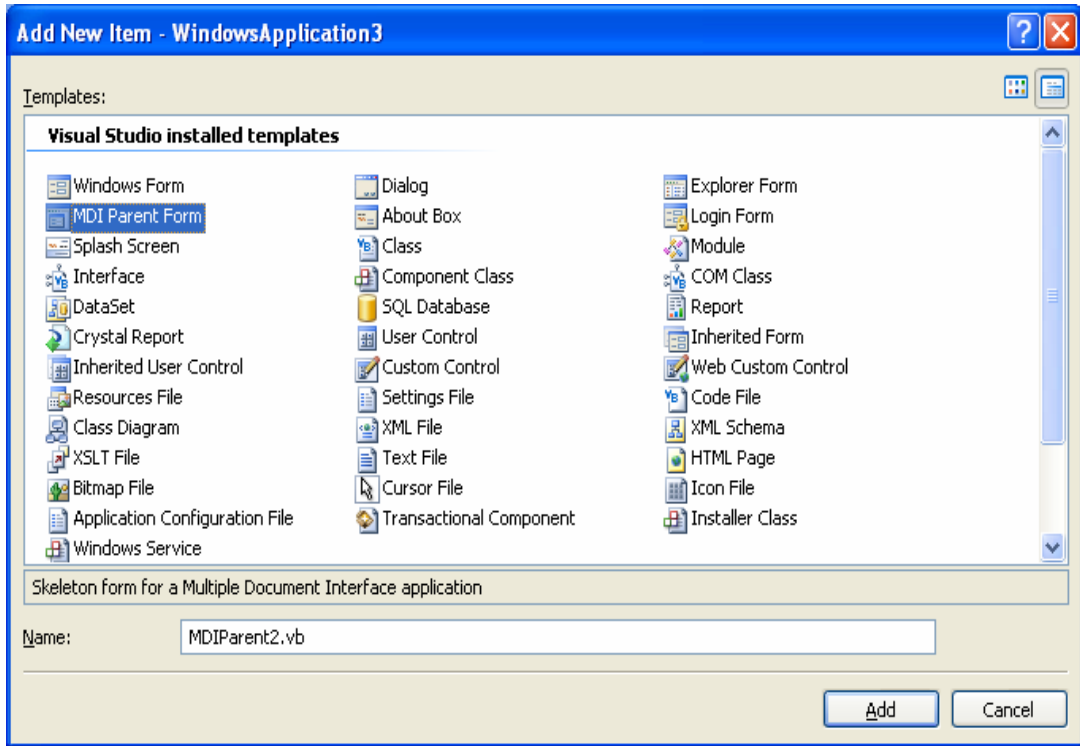


Resim 1.11: Settings penceresinin görünümü

1.3.2. MDI Formlar

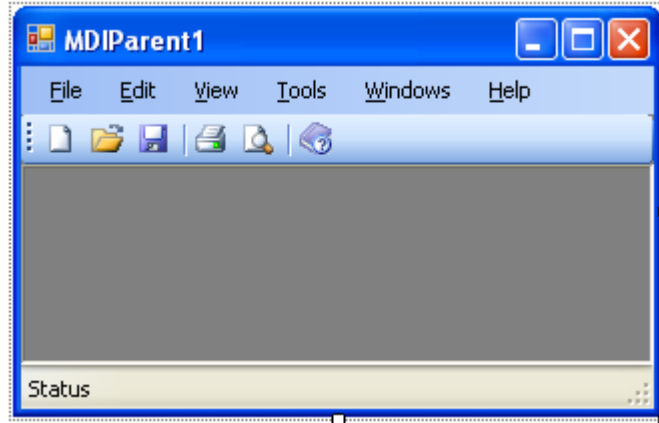
Bir MDI uygulaması, kullanıcıya aynı anda birçok formu gösterme imkânı vermektedir. MDI uygulaması bir MDI formu ve bunun içinde çok sayıda alt formdan oluşmaktadır. MDI formuna Parent (ebeveyn) form, içindeki formlara ise Child (çocuk) form denir. Excel ya da Word gibi Ofis programları ise MDI arayüze sahiptir ve aynı anda birden fazla dökümanın görünmesine izin verir. MDI arayüz programlarını bir ana pencere içinde birden fazla çocuk (child) pencere barındırabilen programlar olarak kabul etmek gerekir.

Genellikle WEB uygulamalarında veya Windows ortamındaki bazı paket programlarda sürekli olarak bir butona ya da bir düğmeye tıkladığımızda yeni formlar ekrana gelmektedir. Bu formlar child form adını alır.



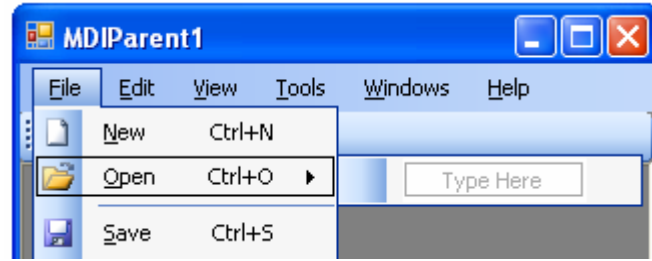
Resim 1.12: Add News Item penceresinin görünümü

Bu işlemleri yaptıktan sonra ekrana MDIParent1 adında bir ana form gelecektir (Resim 1.15).



Resim 1.13: MDI Parent form görünümü

Gördüğünüz gibi Visual Basic .NET'in oluşturduğu MDI ana formları üzerinde araç çubukları ve menüler bulunmaktadır. Çünkü buton gibi kontroller ana formlarda yer alamaz. Bu formlar genellikle menüler aracılığıyla yönetilir. Oluşturulan bu ana forma eklenen çocuk formları menü ve araç çubukları aracılığıyla çağrılır.



Resim 1.14: MDI Parent forma menü ekleme

Şimdi ana form üzerine bir form ekleyelim. Daha sonra Resim 1.16'daki File menüsü altında bulunan Open komutunun kod penceresine geçip aşağıdaki komut satırlarını yazalım.

```
Private Sub OpenFile(ByVal sender As Object, ByVal e As EventArgs)

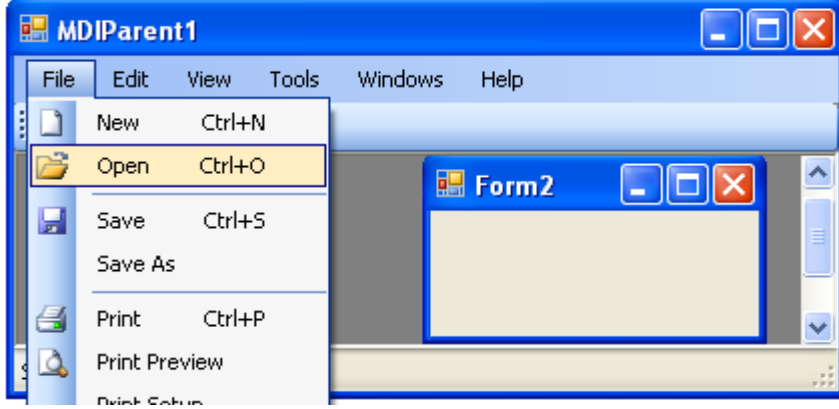
    Dim Form2 As New Form2() ' Yeni formumuzu tanımlıyoruz.

    Form2.MdiParent = Me ' Form2'yi MDIParent ile çocuk formu yapıyoruz.

    Form2.Show() 'Form2 yi görüntülüyoruz.

End Sub
```

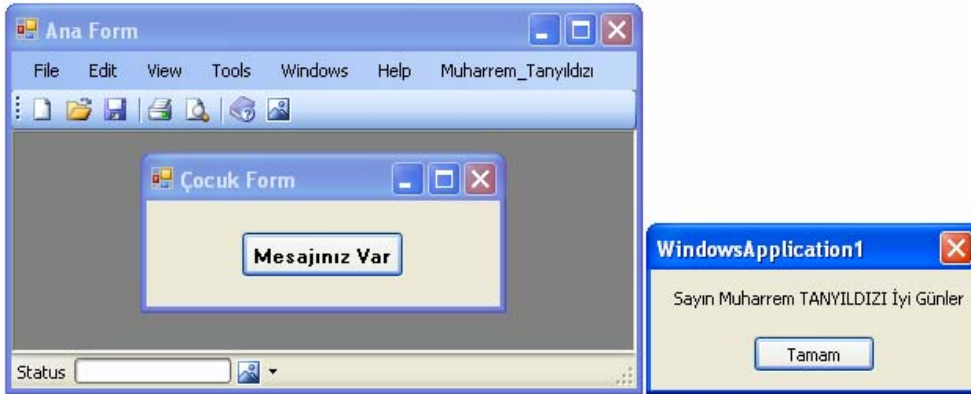
Programınızı çalıştırdıktan sonra Open komutunu tıkladınca Form2 ana form üzerinde açılacaktır.



Resim 1.15: Open komutunun kullanımı

Çocuk formlar üzerinde de istediğiniz menü ve nesnelere kullanabilirsiniz. Şimdi Form2 üzerine bir buton ekleyelim ve çalışma anında bu düğmenin üzerine tıklama yapıldığı zaman yapılacak işlemleri belirlemek için aşağıdaki satırları yazalım.

```
Private Sub Button1_Click(ByVal sender As System.Object, EventArgs)
Handles Button1.Click, MsgBox("Sayın Muharrem TANYILDIZI İyi
Günler")
End Sub
```

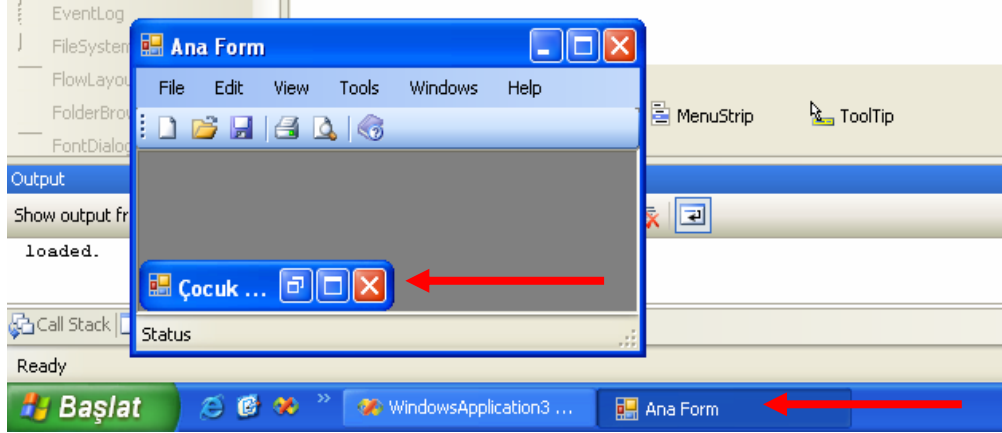


Resim 1.16: Çocuk form ile MessageBox kullanımı

Program çalıştırıldığında önce File / Open ile Form2 açılacak daha sonra Form2 deki butona tıkladınca buradaki işlem gerçekleşecektir.

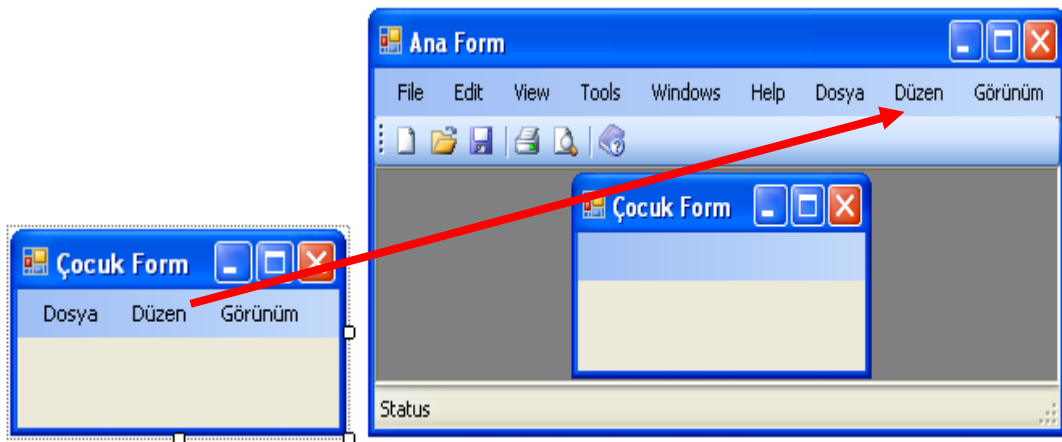
MDI ana ve çocuk formların özellikleri şöyle sıralanabilir:

- Programın çalışması sırasında çocuk formlar, ana formunun çalışma alanını kullanır.
- Ana form simge durumuna getirildiğinde görev çubuğunda görülür. Çocuk formlar simge durumuna getirildiğinde ise ana form içinde görüntülenir.



Resim 1.17. Ana ve çocuk formların görünümü

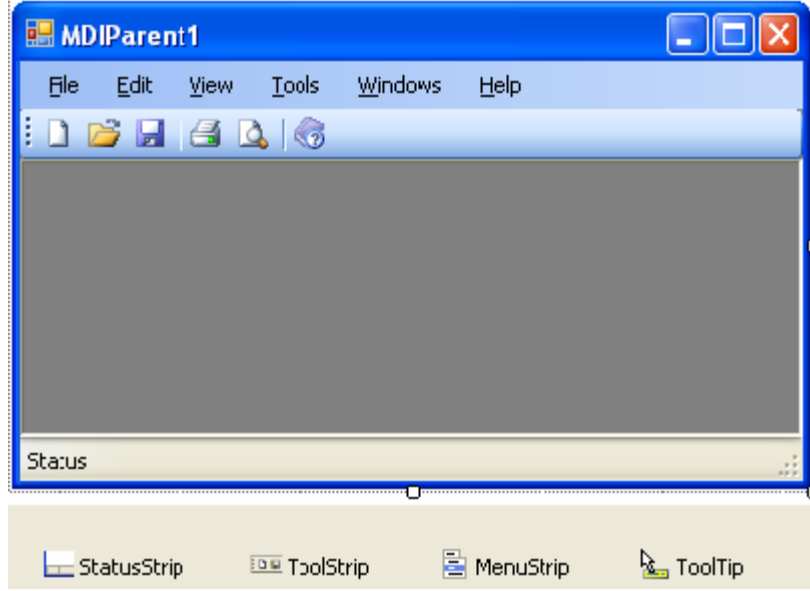
- Bir çocuk form ekranı kapladığında, başlığı ana formunun başlığı olur.
- Aktif çocuk formunun menüleri, ana formunun menü çubuğunda görüntülenir.



Resim 1.18: Çocuk Forma ait menünün ana forma yerleştirilmesi

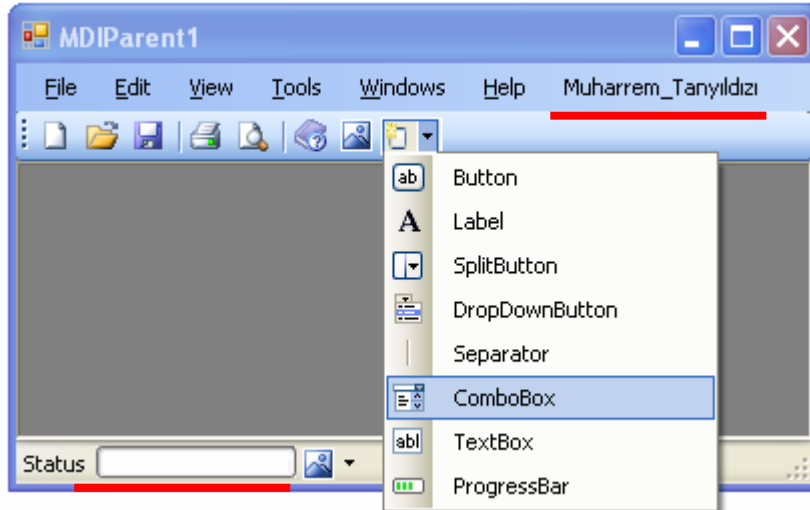
- Çalışma anında yeni çocuk formlar oluşturulabilir.

- Visual Basic .NET tarafından oluşturulan ana form üzerindeki menü ve araç çubukları üzerinde isterseniz değişiklik yapabilirsiniz.



Resim 1.19: MDI Parent form ve kontrollerin görünümü

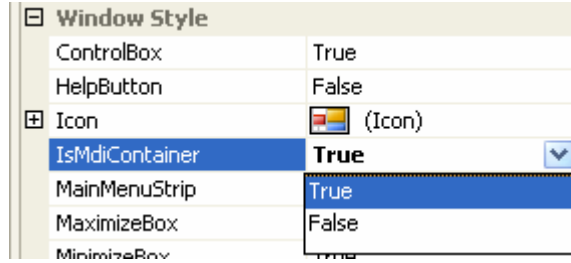
Programınızın alt tarafında StatusStrip, ToolStrip, MenüStrip ve ToolTip seçenekleri ile ihtiyaç duyduğunuz nesnelere ekleyip çıkartabilirsiniz. Bu seçeneklerden hangisini seçerseniz form üzerinde ona ait olan işlemler aktif olacaktır. Resim 1.20'de değişiklik yapılan form görünmektedir.



Resim 1.20: MDI Parent Forma yeni menü ve durum çubuğu ekleme

1.4. Windows Gezgin Tarzı Uygulamalar (Çok Panelli)

Bir uygulama içinde çok panelli proje tasarımı ve kullanımını inceleyelim. Yeni bir proje açıp Form1 ve Form2 adında iki adet form oluşturalım. Form1 MDI Parent (ebevenyn) formu, Form2 ise MDI Child (çocuk) formu olsun. Form1'i parent yani ana form yapmak için Properties penceresinde IsMdiContainer özelliğini true vermemiz gerekir (Resim 1.12).



Resim 1.21: IsMdiContainer özelliği

Artık Form1 Parent (ana) form oldu. Daha sonra ana formunuzdan Child (çocuk) formu açmak için ana formun üzerine bir buton yerleştirebiliriz.

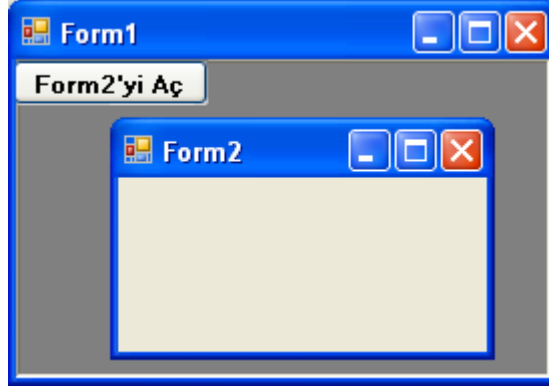


Resim 1.22: Form görünümü

Bu butonun OnClick olayına
Dim Form2 As New Form2()
ve Form2'nin Parent özelliğine Form1'i verelim.
Form2.MdiParent = Me
ve **Show()** metoduyla ikinci formumuzu çağırıyoruz.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal...  
  
    Dim Form2 As New Form2() ' Yeni formumuzu tanımlıyoruz  
  
    Form2.MdiParent = Me ' Form2'yi MDIParent ile çocuk formu yapıyoruz.  
  
    Form2.Show() 'Form2 yi görüntülüyoruz  
  
End Sub
```

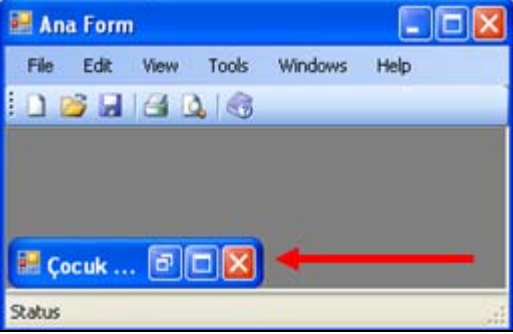
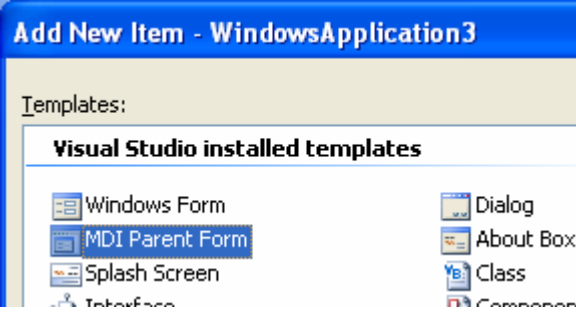

Kodu çalıştırıp Form2'yi Aç diye adlandırılan butona tıkladığımızda aynı pencere içinde Form2 açılacaktır.



Resim 1.23: Form2'nin görünümü

Projemize MDI Parent (Ana form) ekleyebilmemiz için ikinci bir yolda, Project / Add Windows Form komutunu tıkladıktan sonra Add New Item diyalog kutusundan MDI Parent form seçeneğini işaretleyip Add tuşunu tıklamamızdır (Resim 1.14).

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. Bir formu Properties penceresinden ana form haline getiriniz. Ana formun içerisinde çocuk form açılmasını sağlayınız.</p> 	<p>Ana form ile çocuk formu arasındaki farklılıklar öğrenildikten sonra uygulamaya geçilmelidir.</p>
<p>2. Ana form içerisinde iki tane çocuk form oluşturabilmek için gerekli olan kodları defterinize yazınız. Aşağıdaki koddan faydalanınız.</p> <pre>Dim Form2 As New Form2() Form2.MdiParent = Me</pre>	<p>Ana form ile çocuk formu arasındaki farklılıklar öğrenildikten sonra uygulamaya geçilmelidir.</p>
<p>3. Add Project komutu ile yeni bir Ana form oluşturunuz.</p> 	<p>Ana form ile çocuk formu arasındaki farklılıklar öğrenildikten sonra uygulamaya geçilmelidir.</p>

ÖLÇME VE DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki cümlelerde verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. Kod penceresinde yazılan “Form2.ShowDialog()” kodu ile 'Form2 ekrana gelir ama diğer formlara ulaşmaz. ()
2. SDI formlar (single document interface) tek formlu arayüzler olarak adlandırılır. ()
3. Ana form ve çocuk formlar simge durumuna getirildiğinde görev çubuğunda görüntülenir. ()
4. Project / Add Windows Form komutunu tıkladıktan sonra Add New Item diyalog kutusundan MDI Parent Form seçeneğini işaretleyip Add tuşunu tıklayınca ana form oluşturabiliriz . ()
5. Projemizde birden fazla form varsa hangi formun açılış formu olacağını uygulamamızın adı Properties penceresindeki Start Project ile belirleyebiliriz . ()

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız.

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

ÖĞRENME FAALİYETİ-2

AMAÇ

Bu modül ile gerekli ortam sağlandığında, Ana Menü hazırlayabileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

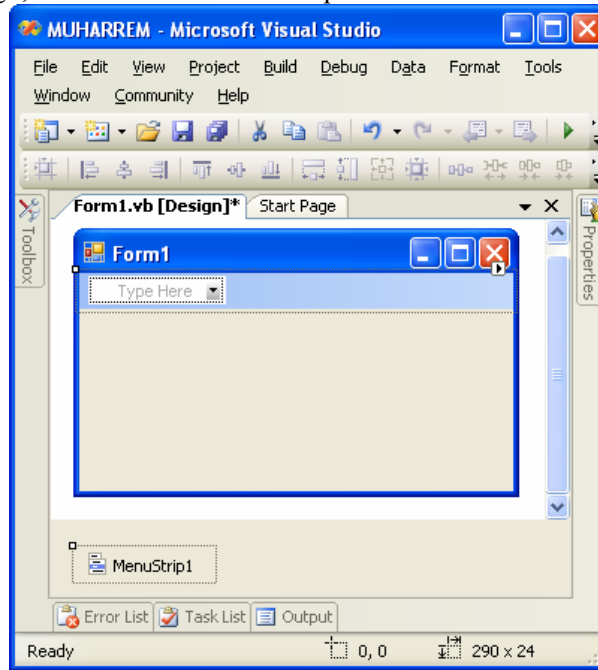


- Daha önce kullandığınız Windows uygulamalarının hepsinde kullanılan menü ve araç kutularında bulunan komutların görevlerini araştırınız.
- Menülerde bulunan tüm komutlar niçin araç kutularında bulunmaz? Araştırınız.
- Ana menü ve kısayol menüleri arasındaki farkı araştırınız.

2. MENÜ TASARIMI

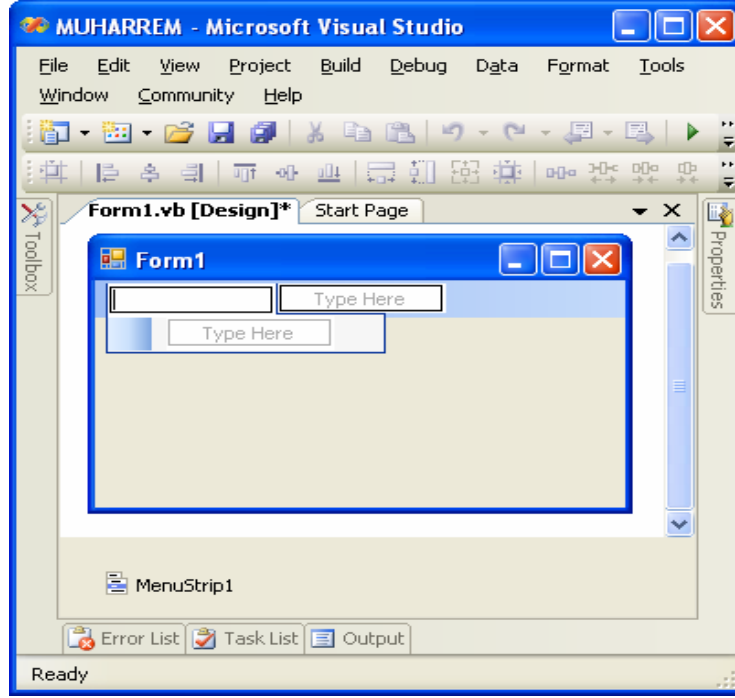
2.1. “MainMenu” Nesnesi

Visual Basic .NET projelerinde menü hazırlamak için **MenuStrip** kontrolünden yararlanırsınız. MenuStrip kontrolünü seçip formun üzerine tıkladığımızda üst kısımda menü çubuğu, alt kısımda ise MenuStrip nesnesini temsil eden bir düğme görürüz.



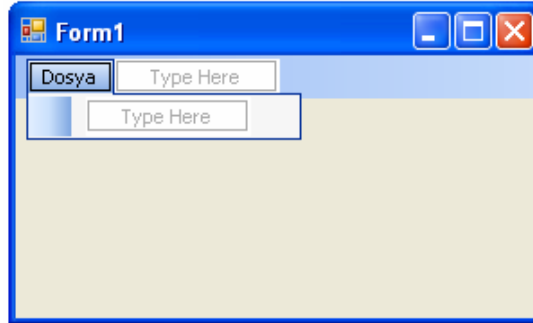
Resim 2.1: MenuStrip kontrolünü forma ekleme

Menüde yer almasını istediğimiz ilk seçeneği menü çubuğunda **Type Here** ile işaret edilen yere yazmalıyız. Type Here seçeneğine tıkladığımızda menü çubuğu aşağıdaki hali alır.



Resim 2.2: MenuStrip kontrolü

Yukarıdaki işlemi yaptıktan sonra hem birinci menü için hem de ilk menüdeki ilk komut için yer ayrılmış olur. Aşağıdaki ekran görüntüsü ilk menünün metni girildikten sonra alınmıştır.



Resim 2.3: MenuStrip kontrolüne menü eklenmesi

2.2. “Insert New, Insert Separator ve Edit names” Menü Komutları

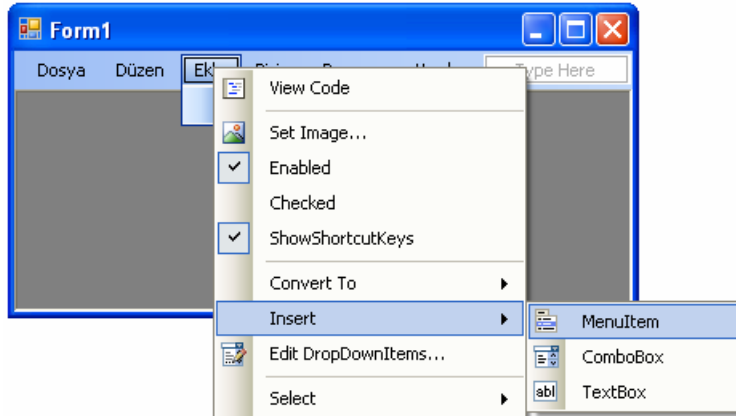
Eklenen menülere yeni menüler eklenebilir. Menü sayısı arttığında menüleri gruplamak için ara çizgiler de verilebilir. Daha önce eklenmiş bir menü adı düzeltilebilir. Şimdi bu işlemlerin nasıl yapıldığını inceleyelim.

İlk menünün metni girildikten sonra artık bu menüde bulunmasını istediğimiz komutları ve ikinci menünün metnini girebiliriz. Aşağıda verilen ekran görüntüsü ikinci menü girildikten sonra alınmıştır.

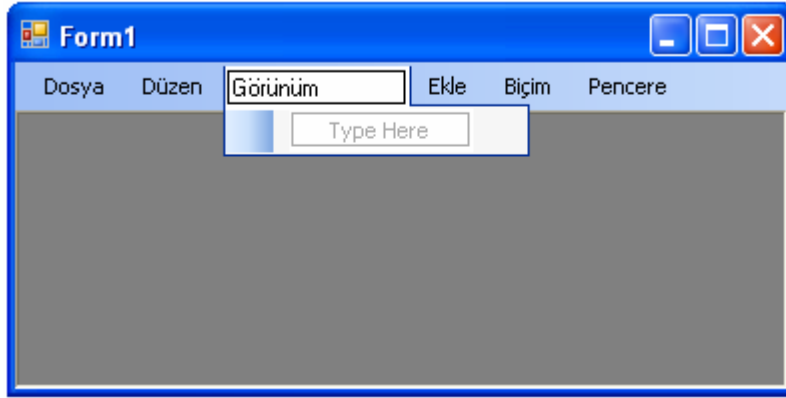


Resim 2.4: MenuStrip kontrolüne menü eklenmesi

Bu şekilde istediğimiz sayıda açılan menüyü menü çubuğuna dahil edebiliriz. Menüleri oluştururken unuttuğumuz ve daha sonra araya eklemek istediğimiz bir menü elemanı olursa bunun için öncelikle hangi menü elemanından önce gelmesini istiyorsak onun üzerine sağ tıklayıp Insert/MenuItem komutunu kullanmalıyız. Daha sonra istenilen menü elemanı araya eklenebilir.

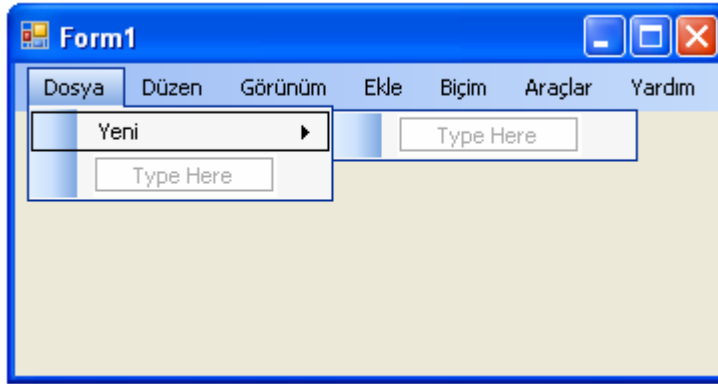


Resim 2.5: MenuStrip kontrolünde araya menü eklenmesi



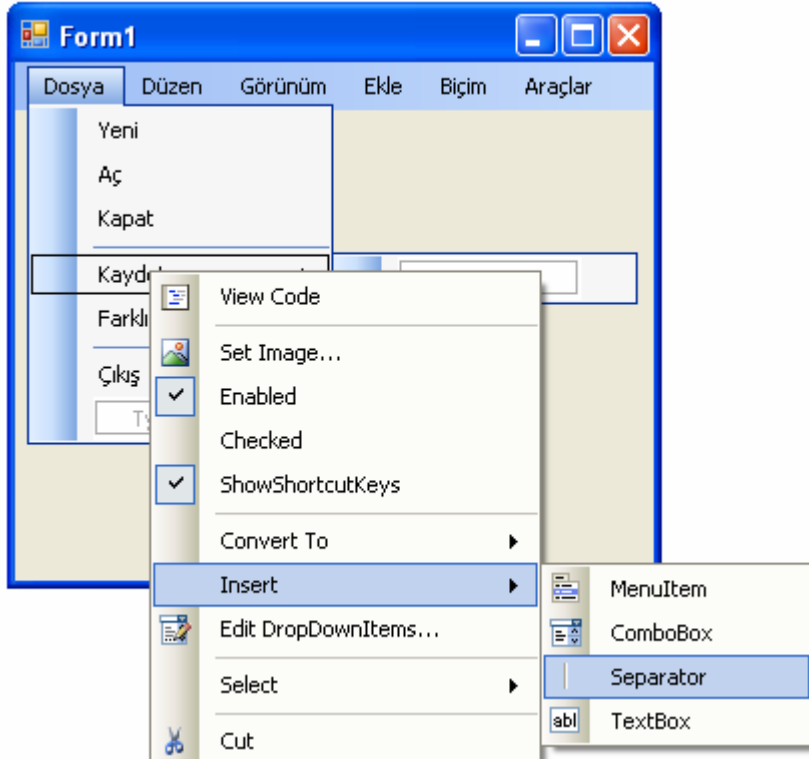
Resim 2.6: MenuStrip kontrolünde araya eklenen menü elemanın görünümü

Açılan menüler hazırlandıktan sonra menülerde bulunacak komutları hazırlayalım. Hangi menü için komut hazırlamak istiyorsak o menünün üstüne tıklama yapmamız gerekir. Dosya menüsünde ilk yer alacak komutun başlığını girmek için “Type Here” yazan yere tıklayıp komut metnini girebiliriz. Komut başlığı olarak girilen metinler komut nesnesinin (MenuItem) **Text** özelliğine yazılır.

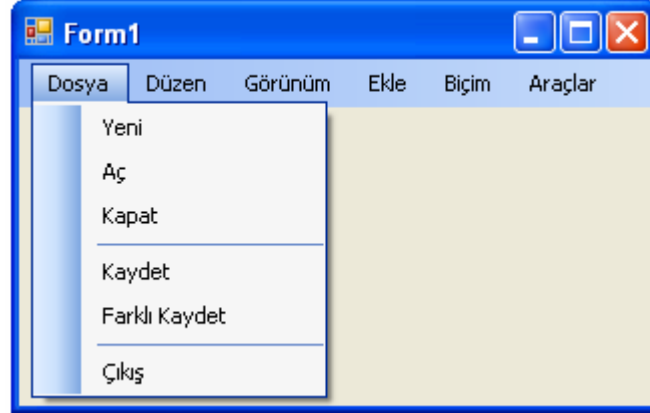


Resim 2.7: MenuStrip kontrolünde komut eklenmesi

Benzer şekilde menüde bulunmasını istediğimiz komutları tanımlayabiliriz. Komutların arasına çizgi koyup gruplamak istersek ekleme noktası komut metni yazılan yerde iken kısa yol menüsünü açıp Insert Separator komutunu vermeliyiz. Aşağıdaki verilen ekran görüntüsü Dosya menüsünde bulunmasını istediğimiz komutları yazdıktan sonra alınmıştır.



Resim 2.8: MenuStrip kontrolünde Seperator eklenmesi

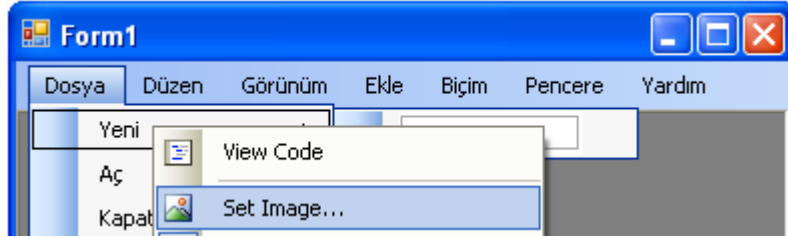


Resim 2.9: Seperator eklendikten sonraki menü görünümü

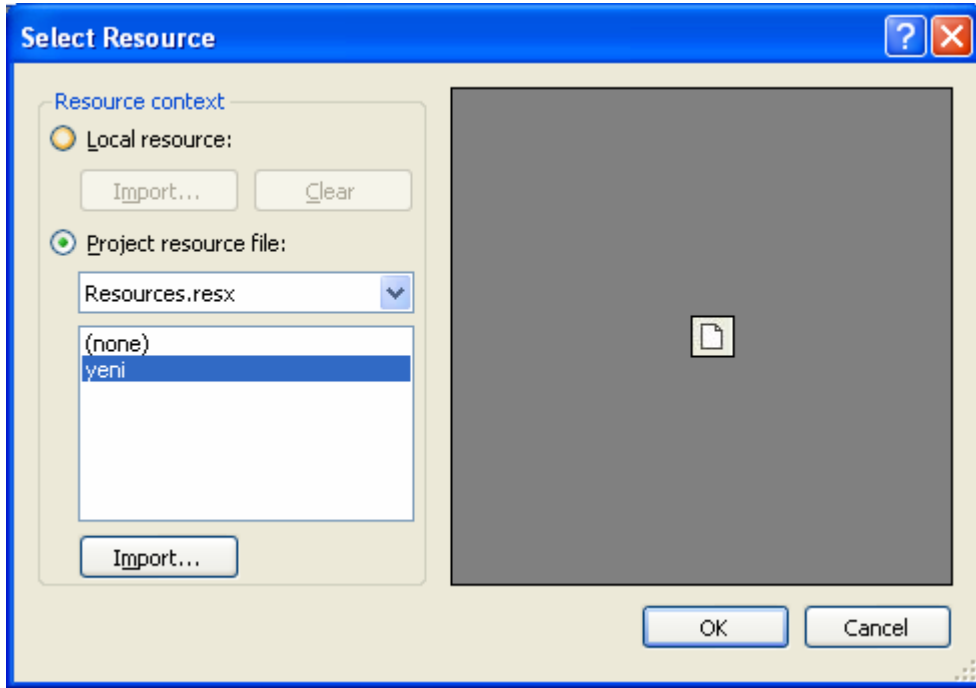
2.3. Menüye İmage Ekleme, Kısayol Atama ve Hazır Menü Kullanma

2.3.1. Menüye İmage Ekleme

Menü hazırlarken Set Image komutundan yararlanabiliriz. Böylece komutların yanına onunla ilişkilendirilmiş resimleri yerleştirebiliriz.



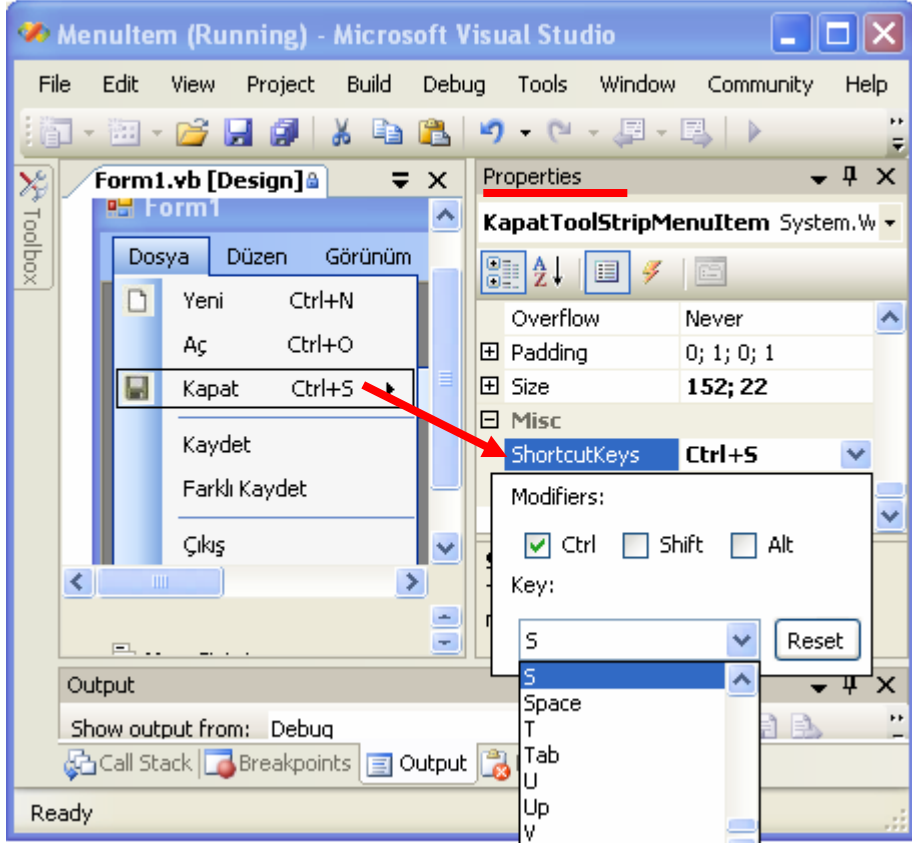
Resim 2.10: Set Image ile komutun yanına resim eklenmesi



Resim 2.11: Set Image ile komutun yanına resim eklenmesi

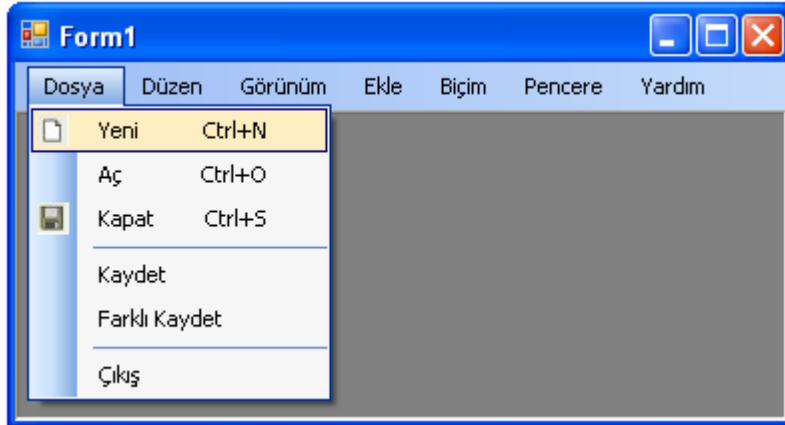
Komutlara kısayol tuşu tanımlamak için MenuItem nesnesinin **Shortcut** özelliğinden yararlanır. Dosya menüsünün "Kaydet" komutuna kısayol tuşu eklemek istersek Kaydet komutunun propertiesine aşağıdaki görülen şekilde işlem yapmalıyız.

2.3.2. Kısayol Atama



Resim 2.12: Kısayol eklenmesi

Örneğin burada Ctrl' yi işaretledikten sonra Key kısmına "S" harfini yazarsanız Resim 2.13'deki görüntü oluşacaktır. Burada bulunan Reset düğmesi ise atamış olduğunuz kısayolu iptal etmeye yarar.



Resim 2.13: Hazırlanan menünün görünümü

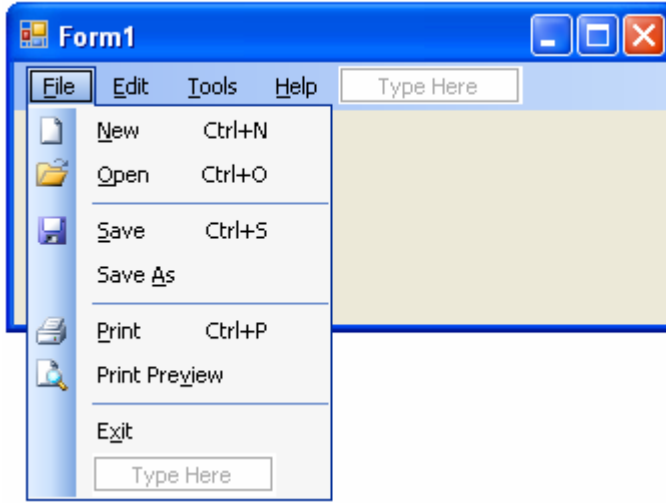
2.3.3. Hazır Menü Kullanma

Menüleri kendimiz oluşturmak yerine Visual Basic .NET tarafından bize sunulan hazır menü ve alt seçeneklerini kullanabiliriz. Bunu yapmak için forma öncelikle bir MenuStrip eklememiz gerekir. Daha sonra MenuStrip1 seçilir ve menünün sağ üst köşesinde bulunan küçük ok işareti tıklanır. Böylece MenuStripTasks diyalog kutusu açılır.



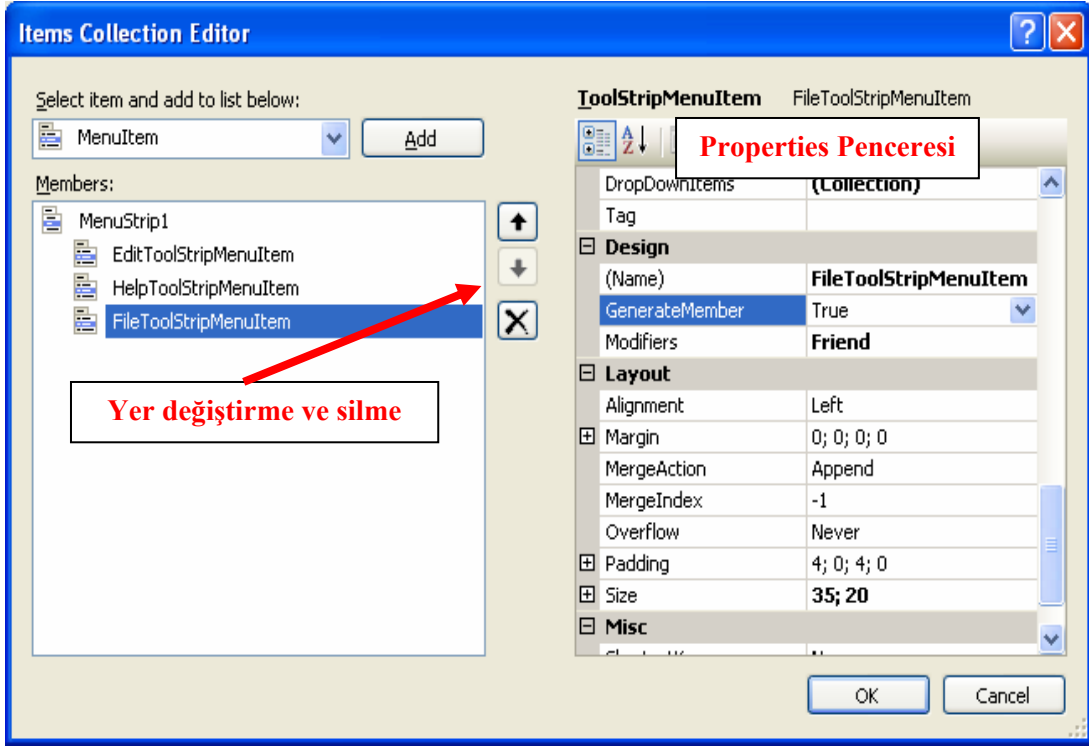
Resim 2.14: MenuStripTasks diyalog kutusu

Eğer Insert Standart Items seçeneğine tıklarsak üzerinde istediğimiz değişikliği yapabileceğimiz menümüz hazır olarak karşımıza gelir.



Resim 2.15: MenuStripTasks ile hazırlanan menü

MenuStrip Tasks diyalog penceresindeki Edit Items seçeneğini tıklarsak bu sefer karşımıza Items Collection Editor penceresi gelecektir. Bu pencerede MenuStrip1'in ve menü elemanlarının özellikleri listelenmektedir. Properties penceresindeki istenilen değişiklikleri buradan da yapılabiliriz. Ayrıca menü elemanlarının yerlerini değiştirip istediğimiz elemanı çıkartabiliriz.



Resim 2.16: Items Collection Editor Diyalog Kutusu

2.4. Menü Komutları İçin Kod Yazma ve Menüler için Alt Menü Hazırlama

2.4.1. Menü Komutları için Kod Yazma

Menüdeki komutların işlevsel olabilmesi için kod yazılması gerekir. Komutun üzerine çift tıklama yapıldığında hem ilgili komut için bir yordam hazırlanır hem de kod sayfası ekrana getirilir. Çıkış komutu çift tıkladığında aşağıdaki yordam hazırlanır.

```
Private Sub ÇıkışToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ÇıkışToolStripMenuItem.Click
```

```
End Sub
```

Menüdeki komutlara ÇıkışToolStripMenuItem, YeniToolStripMenuItem gibi isimleri Basic otomatik olarak verir. Bu adları Properties penceresindeki **Name** özelliğinden değiştirebiliriz. Çalışma anında Dosya menüsündeki Çıkış komutu verildiğinde Çıkış komutuna ait yordamı aşağıdaki gibi düzenleyelim.

```
Private Sub ÇıkışToolStripMenuItem_Click(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
ÇıkışToolStripMenuItem.Click
```

```
End Sub
```

```
MessageBox.Show("Çıkış Komutunu Verdiniz")
```

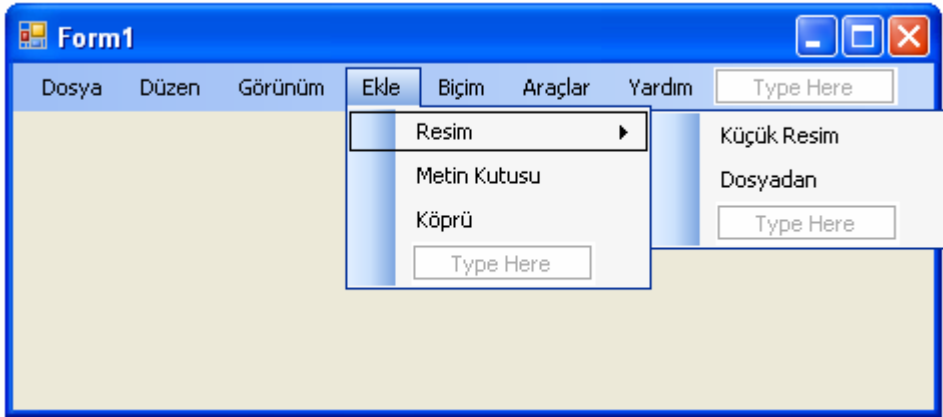
```
End Sub
```



Resim 2.17: Mesaj kutusu

2.4.2. Menüler için Alt Menü Hazırlamak

Alt menü hazırlamak için daha önce hazırlamış olduğumuz projedeki ekle menüsünün resim komutunu seçelim. Resim komutuna tıkladığımızda sağ tarafında komut adı yazılacak Type Here kutusu açılır. Buraya gerekli alt mönüleri yazdıktan sonra ekran görüntüsü aşağıdaki gibi olur.

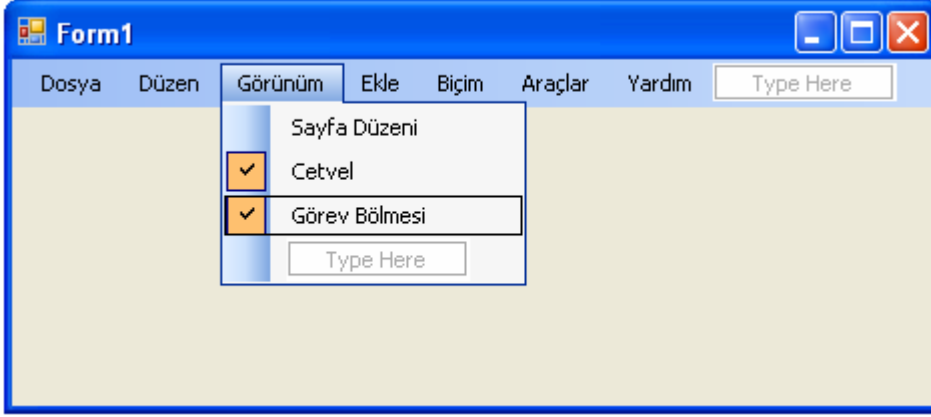


Resim 2.18: Alt menü hazırlama

2.5. Menü Elemanlarının Genel Özellikleri (text, name, checked, radio checked, shortcut, enabled, visible)

2.5.1. Komutları Seçili Duruma Getirmek ve Gruplamak

Bazen açılan menüdeki komutlardan bazılarını devrede olduğunu işaret etmek için komut metninin önüne onay işareti konulmaktadır. Buna örnek göstermek için Görünüm menüsüne aşağıdaki komutları dahil edelim.



Resim 2.19: Seçili duruma getirilen komutların görünümü

Yazacağımız kod ile çalışma anında bu komutlardan hangisi seçilirse, o komuta onay işareti konulacak eğer onay işareti varsa kaldırılacaktır. Bu işlem ToolStripMenuItems nesnesinin veya komutun **Checked** özelliği ile yapılmalıdır. Komut metninin önüne onay işareti koymak için yapılması gereken ToolStripMenuItems nesnesine ait Checked ve CheckOnClick özelliklerine True değeri aktarmaktadır. Eğer Çetvel komutuna bu işlemi uygulamak istersek aşağıdaki kod satırlarını Çetvel' in Click yordamına eklememiz gerekir. Aynı işlemleri istediğimiz kadar komuta uygulayabiliriz.

```
Private Sub ÇetvelToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs)
    If (Çetvel.Checked <> True) Then

        Çetvel.Checked = False
    Else
        Çetvel.Checked = True
    End If
EndSub
```



Resim 2.20: Seçili duruma getirilen komutların görünümü

Bazen de bir menüdeki sadece bir komutu aktif yapmak isteyebiliriz. Bu durumda gruplanacak komutların **RadioCheck** true değerlerini aktarmalıyız. RadioCheck özelliği ancak komutun Checked özelliği true olduğunda işlevsel olmaktadır.

```
Private Sub YeniToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As System.EventArgs)
    YeniToolStripMenuItem.Checked = True
    AçToolStripMenuItem.Checked = False

End Sub

Private Sub AçToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As System.EventArgs)
    AçToolStripMenuItem.Checked = True

    YeniToolStripMenuItem.Checked = False
End Sub
```

2.5.2. Komutları Gizlemek ve Kullanılmaz Hale Getirmek

Çalışma anında bazen bazı komutların menüde görünmesine gerek yoktur. Bu komutu gizlemek için komuta ait **Visible** false yapılır. Örneğin düzen menüsündeki “Geri Al” komutunu listeden kaldırmak için Click yordamına aşağıdaki komutu satırlarını eklemeliyiz.

```
Private Sub GeriAlToolStripMenuItem_Click(ByVal sender As

    GeriAlToolStripMenuItem.Visible = False

End Sub
```

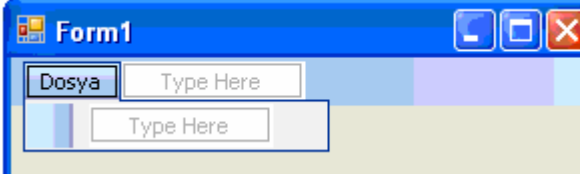

Bazen de kullanılmayan komutu menüde soluk renkte göstermek isteyebiliriz. Bunun için **Enabled** özelliğinden yararlanmalıyız. Örneğin düzen menüsündeki “Geri Al” komutunu listede soluk renkte göstermek için Click yordamına aşağıdaki komut satırlarını eklemeliyiz.

```
Private Sub GeriAlToolStripMenuItem_Click(ByVal sender As
    GeriAlToolStripMenuItem.Enabled = False
End Sub
```



Resim 2.21: Gizli duruma getirilen komutların görünümü

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. Bir form açarak üzerine menüleri ve komutları yerleştiriniz.</p> 	<p>Windows uygulamalarında sık kullanılan menüleri, komutları ve kısayolları öğrendikten sonra oluşturunuz.</p>
<p>2. Eklediğiniz komutların kodlarını yazınız. Örnek bir alt program alta verilmiştir.</p> <pre>Private Sub ÇıkışToolStripMenuItem_Click(ByVal sender As Object, ByVal e As System.EventArgs) End Sub</pre>	<p>Bunu yapmak için komutların görevleri iyi bilinmelidir.</p>
<p>3. Eklediğiniz komutları gruplayınız.</p> 	<p>Seperatoru kullanmalısınız.</p>

ÖLÇME VE DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki cümlelerde verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. Kısayol menülere ContextMenuStrip adı verilmektedir. ()
2. Çalışma anında bazen bazı komutların menüde görünmesine gerek yoktur. Bu komutu gizlemek için komuta ait Visible True yapılır. ()
3. RadioCheck özelliği ancak komutun Checked özelliği False olduğunda işlevsel olmaktadır. ()
4. Menü elemanlarını oluştururken yerlerini değiştirip istediğimiz elemanı çıkartabiliriz. ()
5. MenuStripTasks diyalog kutusu ile hazır menü oluşturabiliriz. ()

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız.

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

ÖĞRENME FAALİYETİ-3

AMAÇ

Bu modül ile gerekli ortam sağlandığında, araç çubukları, durum çubuğu ve içerik menüsü hazırlayabileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

- Araç çubuklarının görevlerini araştırınız.
- Durum çubuğunun görevlerini araştırınız.



3. YARDIMCI NESNELER

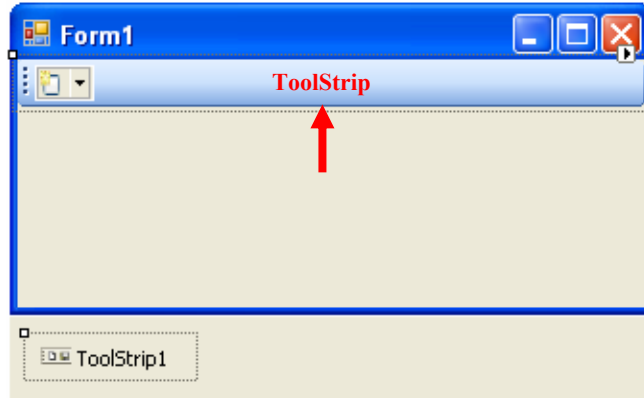
3.1. ToolStrip Nesnesi

ToolStrip kontrolü ile uygulamamıza araç çubuğu ekleyebiliriz. Windows uygulamalarında sürekli olarak karşımıza çıkan resimli butonlar olan ToolStrip'ler genellikle uygulamalarda çok kullanılan menü seçeneklerine kolay erişimi sağlar.



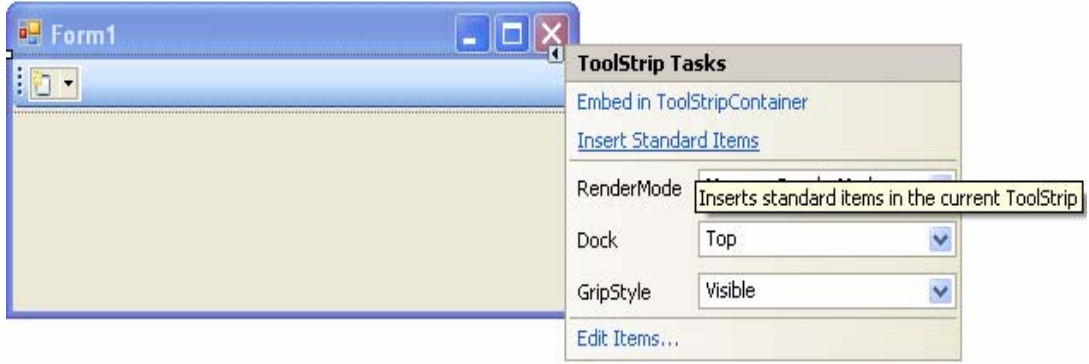
Resim 3.1: Standart araç çubuğu

Projelerimize ToolStrip eklemek için Toolbox penceresinden ToolStrip kontrolünü seçeriz. ToolStrip kontrolünü seçip formun üzerine tıkladığımızda üst kısımda araç çubuğu, alt kısımda ise ToolStrip nesnesini temsil eden bir düğme görürüz.



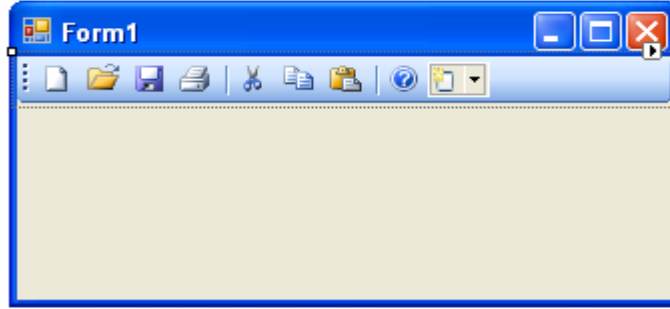
Resim 3.2: Forma ToolStrip eklenmesi

Araç çubuğunu kendimiz oluşturmak yerine Visual Basic .NET tarafından bize sunulan hazır araç çubuğu seçeneklerini kullanabiliriz. Bunu yapmak için tıpkı menü oluştururken yaptığımız gibi ToolStrip1 seçip ve sağ üst köşesinde bulunan küçük ok işaretine tıklarız. Böylece ToolStripTasks diyalog kutusu açılır.



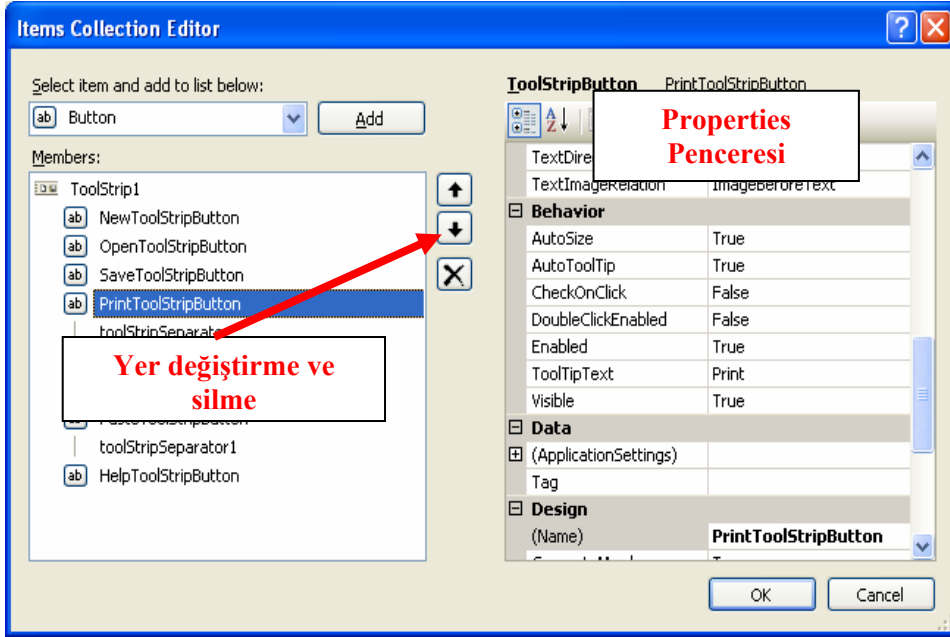
Resim 3.3: ToolStrip Tasks diyalog kutusu

Eğer Insert Standart Items seçeneğine tıklarsak üzerinde istediğimiz değişikliği yapabileceğimiz araç çubuğumuz hazır olarak karşımıza gelecektir.



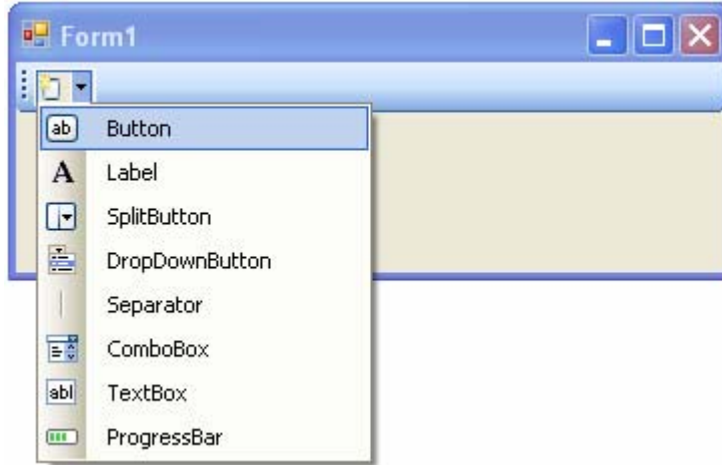
Resim 3.4: Araç Çubuğunun Forma Yerleştirilmesi

ToolStrip Tasks diyalog penceresindeki Edit Items seçeneğini tıklarsak bu sefer karşımıza Items Collection Editor penceresi gelecektir. Bu pencerede ToolStrip1'in ve araç çubuğu elemanlarının özellikleri listelenmektedir. Properties penceresindeki istenilen değişiklikleri buradan da yapılabiliriz. Ayrıca araç çubuğu elemanlarının yerlerini değiştirip istediğimiz elemanı çıkartabiliriz.



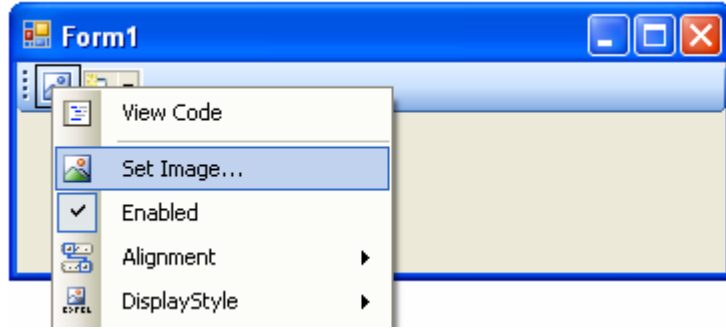
Resim 3.5: Items Collection Editor penceresi

Eđer araç çubuęunu kendimiz oluřturmak istiyorsak ToolStrip üzerinde bulunan seeneklerden istedięimizi ekleyebiliriz.



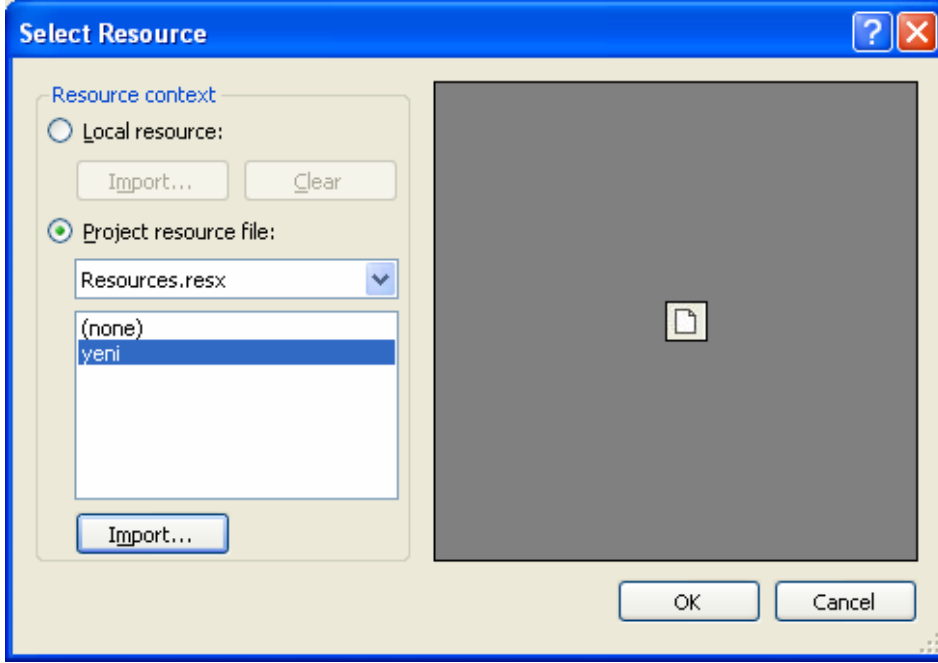
Resim 3.6: ToolStrip seenekleri

Bu seeneklerden Button' u ekleyelim. Menülerdeki komutları temsil eden butonlar genellikle resim řeklinde dir. Bunun iin bir butonu ekledikten sonra üzerine saę tıklayarak Set Image komutunu iřletiyoruz.



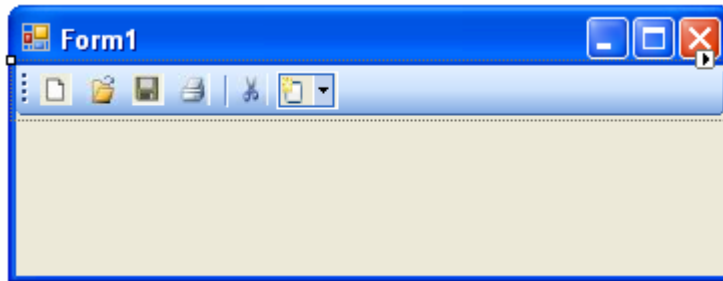
Resim 3.7: Set Image komutu

Set Image komutunu tıkklayınca karşımıza buton için seçeceğimiz resmi belirleyeceğimiz Select Resource penceresi gelecektir.




Resim 3.8: Select Resource penceresi

Select Resource penceresinde istediğimiz resmi Import (Ekle) yaptıktan sonra OK tuşuna tıkklayınca oluşturduğumuz araç çubuğu aşağıdaki görüntüyü alacaktır.

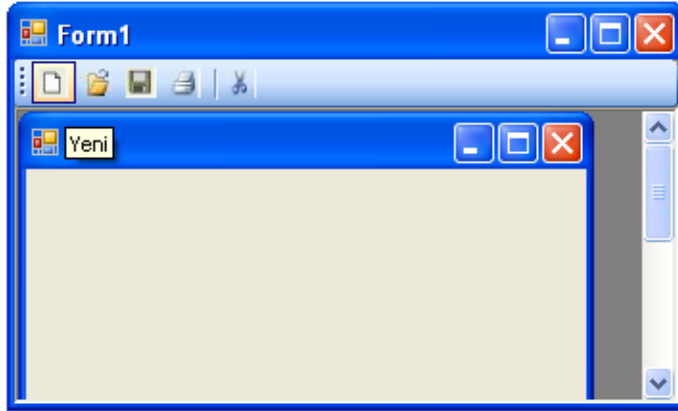


Resim 3.9: Yeni Araç çubuğumuz

3.1.1. Komutlar için Kod Yazma

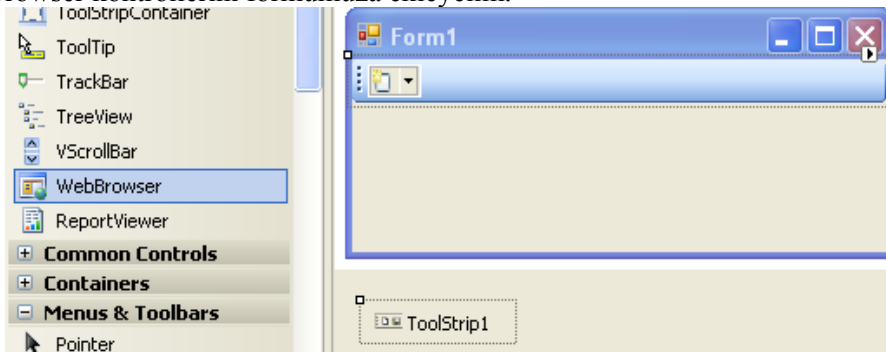
Araç çubuğundaki komutların işlevsel olabilmesi için kod yazılması gerekir. Örneğin Yeni komutunu temsil eden  düğmenin görevi, hangi programda çalışıyorsak onunla ilgili yeni bir sayfa açmaktır. Bunun için komut düğmesi ile ilgili gerekli komut satırlarını yazmadan önce formumuzu ana form yapmalıyız. Daha sonra gerekli olan kodları yazabiliriz.

```
Private Sub Yeni_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Yeni.Click  
    ' Yeni bir çocuk form oluşturalım.  
    Dim ChildForm As New System.Windows.Forms.Form  
  
    ' Formu Ana Form İçinde oluşturmaya yarar.  
    ChildForm.MdiParent = Me  
  
    'Formu göstermeye yarar.  
    ChildForm.Show()  
  
End Sub
```



Resim 3.10: Araç Çubuğundaki Komutun Çalıştırılması

İkinci bir örnek olarak ToolStrip bileşenlerini kullanarak kendi WEB tarayıcımızı oluşturalım. Kullanıcı ara yüzünü oluşturmak için önce ToolStrip, ve son olarak da WebBrowser kontrollerini formumuza ekleyelim.



Resim 3.11: Form Üzerine ToolStrip ve web browser eklenmesi

ToolStrip'e tıklayarak açılan kontrol listesinden önce Label, sonrasında TextBox ve Button seçelim. Böylece ToolStrip üzerine üç kontrol eklemiş oluruz. ToolStrip'e eklediğimiz Label'a tıklayalım ve Properties penceresinden Text özelliğine "Adres :"' yazalım. Ardından Button kontrolüne tıklayalım ve yine Properties penceresinden Text özelliğine 'Git' yazalım. Ardından Button kontrolüne tıklayalım ve yine Properties penceresinden Text özelliğine 'Git' yazalım. Sonra yine Button için 'DisplayStyle' özeliğini 'Text' olarak deęiřtirelim. Sonra da formun başlık çubuęuna tıklayalım ve Form1'in Text özelliğine 'Web Gezgini' yazalım. Bu kadar kolay bir řekilde ara yüzümüzü tamamlamış olduk.



Resim 3.12: Form arayüzünün oluşturulması

Son olarak da işlevi yerine getirecek olan kodu yazalım.

```
Private Sub ToolStripButton1_Click(ByVal sender As Object, ByVal e As System.EventArgs) Handles ÇıkışToolStripMenuItem.Click
```

```
WebBrowser1.Navigate(ToolStripTextBox1.Text) ' web sayfasına bağlanmayı sağlıyor
```

```
End Sub
```

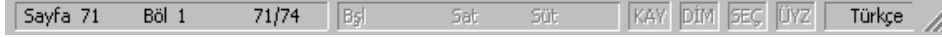


Resim 3.13: WEB sayfasına bağlanma

3.2. StatusStrip Nesnesi

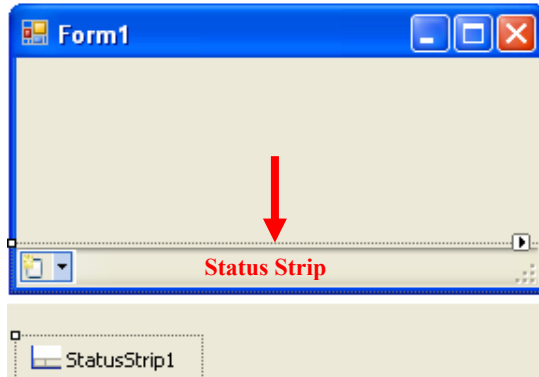
Genellikle formun alt tarafında yer alan ve o andaki işlemle ilgili çeşitli durum bilgilerini görüntülemek için kullanılır.

Birçok programda pencerelerin en altında o anki durumu gösteren bazı bilgiler bulunur. Örneğin Word programının en alt satırında bazı bilgiler veren bir durum çubuğu bulunur.



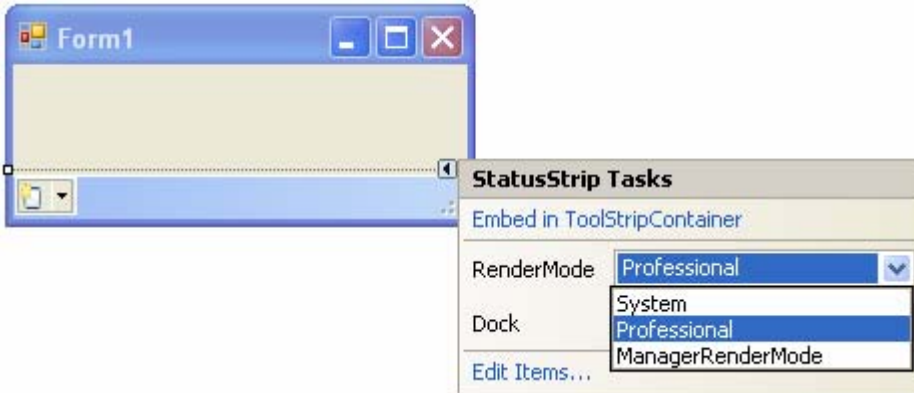
Resim 3.14: Durum çubuğu

Bu tip durum çubuklarını Toolbox penceresindeki StatusStrip kontrolünü kullanarak kolayca oluşturabiliriz. StatusStrip kontrolünün forma yerleştirildiğindeki görüntüsü resim 3.15'teki gibidir.



Resim 3.15: Form üzerine StatusStrip (Durum Çubuğu) eklenmesi

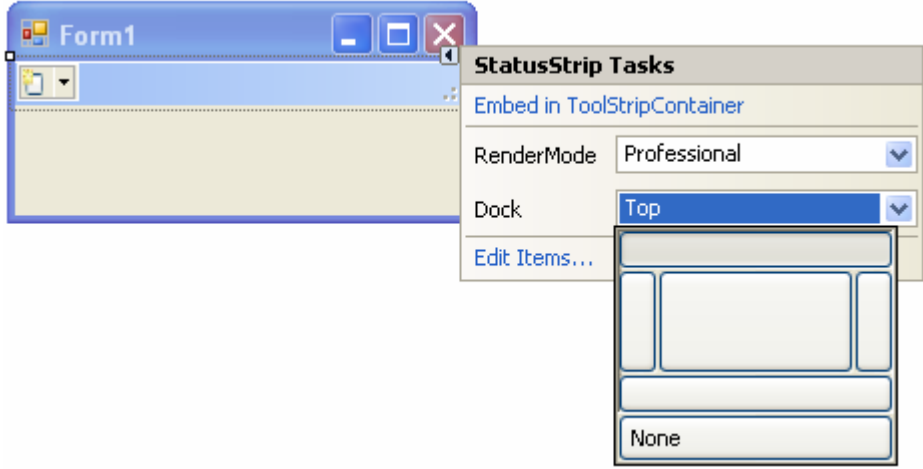
Durum çubuklarında menü ve araç çubuklarını projemize eklerken yararlandığımız standart bir seçenek yoktur. StatusStrip'i seçip ve sağ üst köşesinde bulunan küçük ok işaretine tıklayınca karşımıza gelen StatusStripTasks diyalog kutusunda Insert Standarts Items seçeneği bulunmamaktadır.



Resim 3.16: StatusStrip Tasks diyalog kutusu

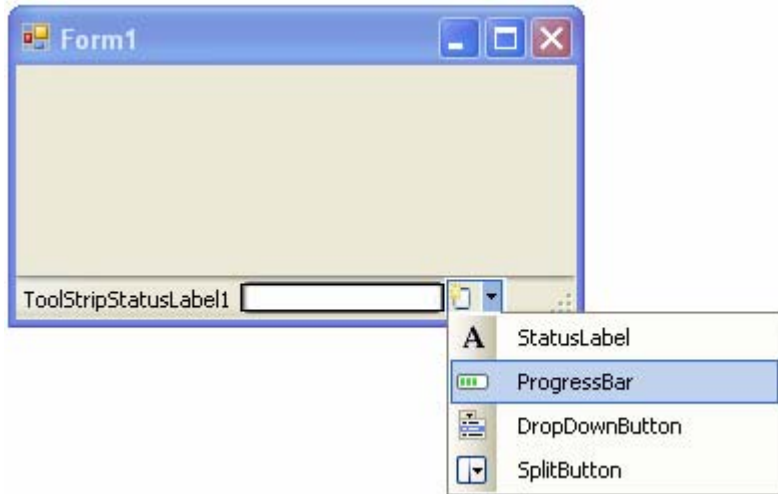
StatusStripTasks diyalog kutusunda bulunan RenderMode seçeneği ile StatusStrip'in görünümünü değiştirebiliriz.

Dock seçeneği ile ise StatusStrip'in formun neresinde konumlanacağını belirleyebiliriz.



Resim 3.17: StatusStrip Tasks diyalog kutusu

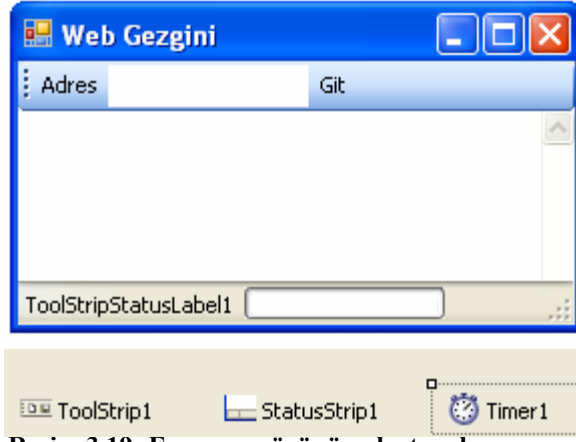
Durum çubuğuna etiket, ilerleme çubuğu, aşağı açılan düğme vb. ekleyebiliriz.



Resim 3.18: StatusStrip seçenekleri

Örnek olarak ToolStrip bileşenlerini kullanarak oluşturduğumuz WEB tarayıcımıza bir durum çubuğu ekleyelim.

ToolStrip'e tıklayarak açılan kontrol listesinden önce Label, sonrasında TextBox ve Button seçelim. Böylece ToolStrip üzerine üç kontrol eklemiş oluruz. ToolStrip'e eklediğimiz Label'a tıklayalım ve Properties penceresinden Text özelliğine "Adres :." yazalım. Ardından Button kontrolüne tıklayalım ve yine Properties penceresinden Text özelliğine 'Git' yazalım. Ardından Button kontrolüne tıklayalım ve yine Properties penceresinden Text özelliğine 'Git' yazalım. Sonra yine Button için 'DisplayStyle' özeliğini 'Text' olarak değiştirelim. Daha sonra StatusStrip'e tıklayarak bir Label bir tanede Progress Bar ekleyelim. Ayrıca formun üzerine bir de Timer kontrolü ekleyelim. Sonra da formun başlık çubuğuna tıklayalım ve Form1'in Text özelliğine 'Web Gezini' yazalım.



Resim 3.19: Form arayüzünün oluşturulması

Formumuzun ara yüzünü bu şekilde oluşturduktan sonra kod penceresine geçerek gerekli olan komut satırlarını yazalım.

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles Me.Load
    ToolStripStatusLabel1.Text = "Form Yüklendi" ' Label1 formun
    yüklendiği anda uyarı yazısı yazılıyor

    Timer1.Stop() ' Timer (zaman) işeyişini furduruyoer

End Sub

Private Sub ToolStripButton1_Click(ByVal sender As System.Object,

    ToolStripProgressBar1.Value = 0 ' Her yeni web sayfasına
    bağlandığında progres bas değerini sıfırlıyor

    Timer1.Start() ' zamanı başlat

    WebBrowser1.Navigate(ToolStripTextBox1.Text) ' web sayfasına
    bağlanmayı sağlıyor

End Sub
```

```
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As As System.EventArgs) Handles Timer1_Tick
```

```
    ToolStripProgressBar1.Value = ToolStripProgressBar1.Value + 10  
    '' progress barın her saniyede 10 birim artmasını sağlıyor
```

```
    ToolStripStatusLabel1.Text = ToolStripTextBox1.Text + " Sayfa  
Yükleniyor"
```

```
    If ToolStripProgressBar1.Value = 100 Then  
        ToolStripProgressBar1.Value = 0  
    End If  
End Sub
```

```
Private Sub WebBrowser1_DocumentCompleted(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles WebBrowser1_DocumentCompleted
```

```
    '' bu kısım web sayfası yüklendiği anda zamanı durdurmak için kullanıldı
```

```
    Timer1.Stop()  
    ToolStripStatusLabel1.Text = ToolStripTextBox1.Text + " Sayfa  
Yüklendi"  
End Sub
```

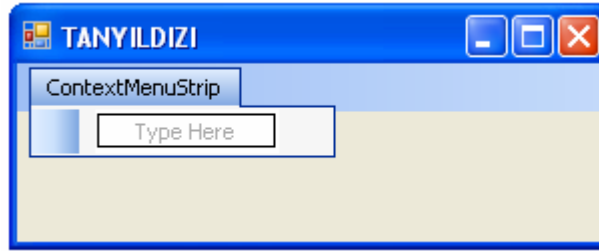
Kod penceresi de hazırlandıktan sonra formumuzu çalıştırıp Adres satırına açılmasını istediğimiz WEB sayfasının adresini yazıp Git düğmesine basınca sayfa açılacak ve durum çubuğunda sayfanın durumu ile ilgili bilgiler gelecektir.



Resim 3.20: WEB sayfası

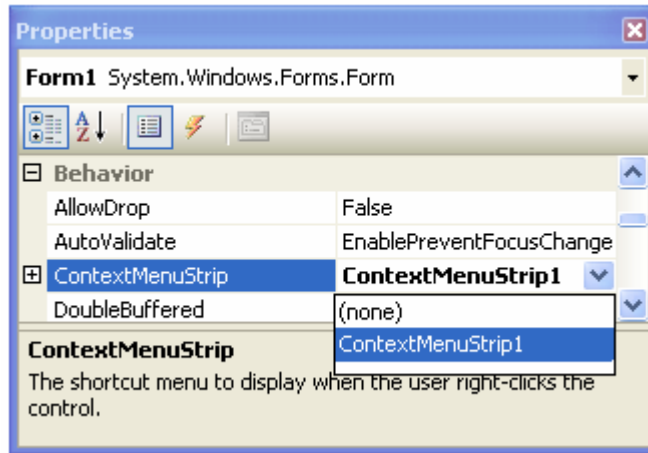
3.3. Context Menü Hazırlamak

Açılan menüleri alternatif olarak hazırlanan kısa yol menüleri çok kullanışlıdır. Genelde komutun hangi menüde olduğunu bilmediğimiz durumda işimize yaramaktadır. Visual Basic projelerindeki kısayol menüleri **ContextMenuStrip** adı verilmektedir. Kısa yol menüsünü projeye eklemek için Toolbox’ ta ContextMenuStrip seçilir ve forma eklenir. Eklemeyi yaptıktan sonra Text Here yazan yere komutları ekleyebiliriz.



Resim 2.21: ContextMenüStrip’in görünümü

Kısa yol menüler dahil edildiği form aktif edildiği zaman görünmez. Çünkü çalışma anında ContextMenuStrip’ ler yapılan tanımlamaya uygun olarak bir düğme veya metin kutusunun üzerinde farenin sağ tuşu ile tıklama yapıldığında ekrana gelmektedir. Hangi nesnenin üzerinde farenin sağ tuşu ile tıklama yapıldığında ekrana ContextMenuStrip’ in açılmasını istiyorsak o nesnenin **ContextMenuStrip** özelliğine hazırladığımız kısa yol menüsünün ismini aktarmalıyız. Formun üzerinde farenin sağ tuşu ile tıklama yapıldığı zaman açılmasını istediğimiz ContextMenuStrip ontextMenu nesnesinin adını formun ContextMenuStrip özelliğine aktaralım.



Resim 2.22: ContextMenüStrip özellikler penceresi

Bu işlemden sonra form çalıştırılıp ve üzerinde sağ tuş tıklanırsa aşağıdaki ekran görüntüsü elde edilir.



Resim 2.23: Hazırlanan ContextMenüStrip'in görünümü

ContextMenülerdeki komutların üzerine çift tıklanarak o yordam hazırlanır ve içine yazılan komutlar çalışma zamanında üzerine tıklandığında çalıştırılır. Bunu bir örnekle açıklayalım.

Formumuza bir label ve ContextMenu ekleyelim. Daha sonra ContextMenu'ye Arial ve Times New Roman adında iki tane komut ekleyelim.



Resim 2.24: Hazırlanan formun görünümü

Daha sonra projenin kod kısmına aşağıdaki komut satırlarını yazıp programı çalıştıralım

```
Private Sub ArialToolStripMenuItem_Click(ByVal sender As Object,  
ByVal e As System.EventArgs) Handles ArialToolStripMenuItem_Click
```

```
Me.Label1.Font = New Font("Arial", Label1.Font.Size)
```

```
End Sub
```

```

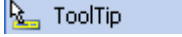
Private Sub TimesToolStripMenuItem_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles TimesToolStripMenuItem_Click
    Me.Label1.Font = New Font("Times New Roman", Label1.Font.Size)
End Sub

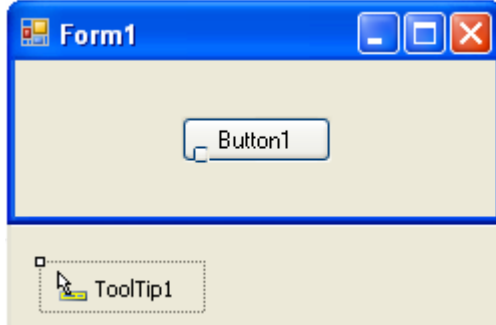
```



Resim 2.25: Contextmenüstrip çalışma anı

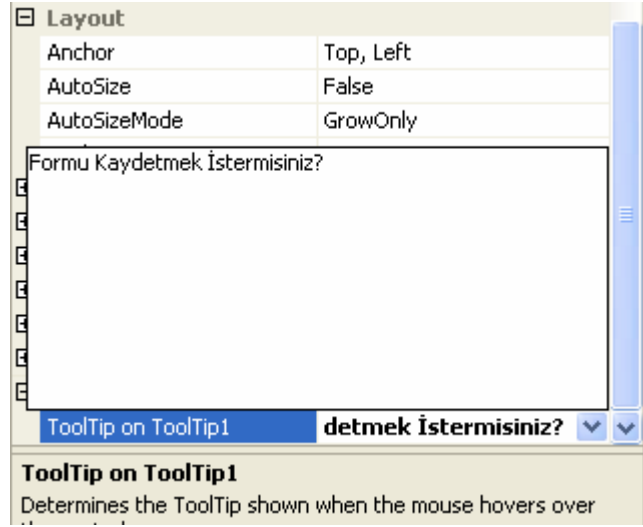
3.4. ToolTip Nesnesi

ToolTip kontrolü ile uygulamadaki herhangi bir kontrole (Tool Tip)  ipucu ekleyebiliriz. Bu ipucu metni, imleç bir kontrolün üzerinde bekletildiğinde kontrolün yanında veya altında görüntülenir ve genellikle ilgili kontrolün amacının ya da işlevinin anlaşılmasına yardımcı olması için kullanılır. Forma eklendiğimizde formun üzerine değil, alt kısmına yerleşecektir.



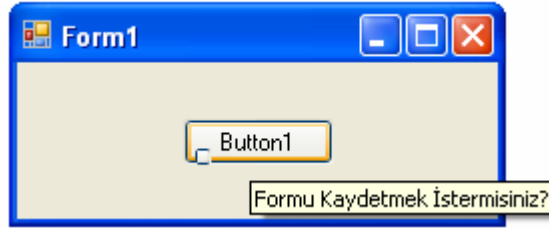
Resim 3.21: ToolTip nesnesi

Formumuza bir ToolTip ekledikten sonra hangi kontrol için ipucu metni görüntüleyeceksek onu seçmemiz gerekir. Daha sonra Properties penceresinde ToolTip on ToolTip1 etiketli bir özellik göreceğiz. Bu özellik forma eklediğimiz ToolTip kontrolünün ismidir.



Resim 3.22: ToolTip nesnesi özellikleri

Button1 kontrolü için istediğiniz ToolTip metnini yazınız. Uygulamanızı çalıştırıp imleci kontrolün üzerinde beklettiğinizde yazdığınız metin kontrolün yanında görüntülenecektir.

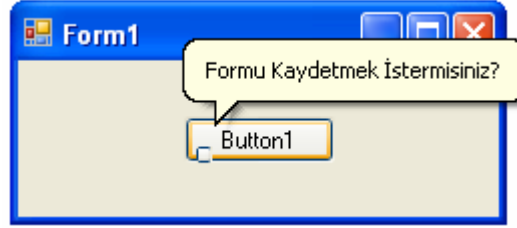


Resim 3.23: ToolTip nesnesinin uygulanması

ToolTip nesnesinin Properties penceresindeki özellikleri ve değerleri şunlardır:

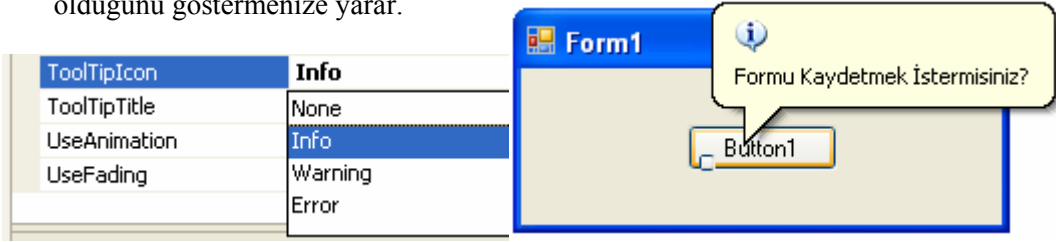
- **Active:** False değerinde ipucu metni görüntülenmez.
- **Automatic Delay:** Metnin tekrarlanma sayısını etkiler.
- **BackColor:** Zemin rengini değiştirir.
- **ForeColor:** Yazı rengini değiştirir.
- **InitialDelay:** Mouse kontrolün üzerine geldiğinde metnin ne kadar sürede görüneceğini belirlemenize yarar.

- **IsBalloon:** Bu özelliği True yaparsanız ipucu metni balon içerisinde görüntülenir.



Resim 3.24: ToolTip nesnesinin kullanımı

- **ReShowDelay:** Mouse'yi kontrolün üzerinden çektikten sonra metnin ne kadar daha görüneceğini belirlemenize yarar.
- **ToolTipIcon:** İpucu metninin bilgi amaçlı mı, uyarı mı yoksa hata bildirimini mi olduğunu göstermenize yarar.



Resim 3.25: ToolTip nesnesinin kullanımı

ToolTip kontrolünü kod penceresi yardımıyla da ekleyebiliriz. Formun kod penceresine gelerek gerekli olan kodları yazalım.

```
Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load

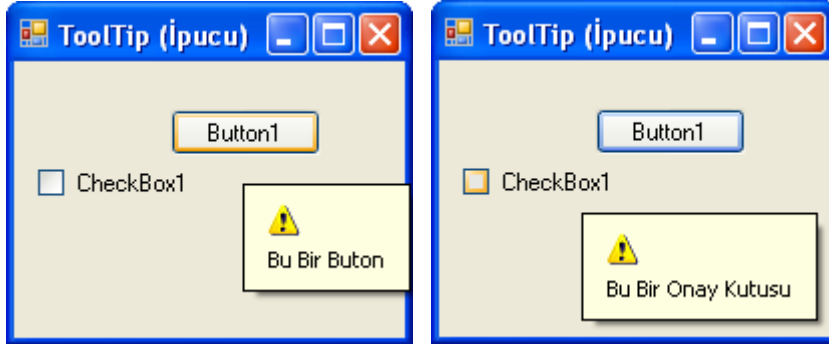
    'ToolTip nesnesinin eklenmesi
    Dim toolTip1 As New ToolTip()

    'ToolTip nesnesinin özellik ayarları
    toolTip1.AutoPopDelay = 5000
    toolTip1.InitialDelay = 1000
    toolTip1.ReshowDelay = 500
    toolTip1.ToolTipIcon = ToolTipIcon.Warning

    ' ToolTip'in sürekli olarak aktif olmasını sağlar
    toolTip1.ShowAlways = True

    ' Button ve Checkbox nesnelerinde görünecek ipuçları
    toolTip1.SetToolTip(Me.Button1, "Bu Bir Buton")
    toolTip1.SetToolTip(Me.CheckBox1, "Bu Bir Onay Kutusu")

End Sub
```

Resim 3.26: ToolTip nesnesinin kullanımı

İkinci bir örnek olarak bir kontrole birden fazla ipucu metni ekleyelim. Hazırlamış olduğumuz form üzerindeki CheckBox'ın kod penceresine aşağıdaki kodları yazalım.

```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged

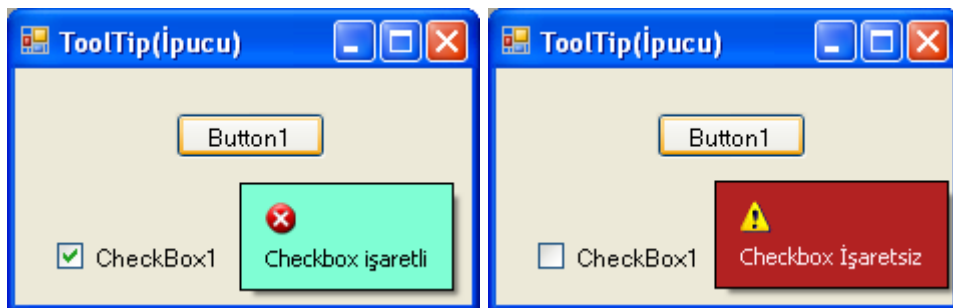
    If CheckBox1.Checked = True Then 'CheckBox nesnesi işaretli

        ToolTip1.Active = True'ToolTip1 aktif
        ToolTip1.BackColor = Color.Aquamarine
        ToolTip1.ForeColor = Color.Black
        ToolTip1.SetToolTip(Me.Button1, "Checkbox işaretli")
        ToolTip2.Active = False
    Else

        ToolTip1.Active = False
        ToolTip2.Active = True
        ToolTip2.BackColor = Color.Firebrick
        ToolTip2.ForeColor = Color.LavenderBlush
        ToolTip2.SetToolTip(Me.Button1, "Checkbox İşaretsiz")

    End If

End Sub
```

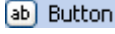


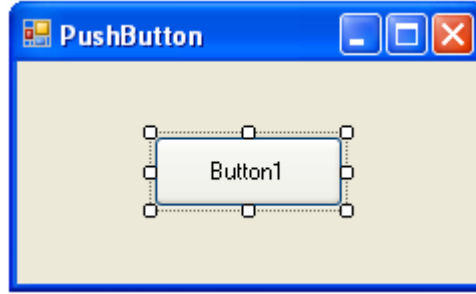
Resim 3.27: ToolTip nesnesinin kullanımı

3.5. Push Button, DropDown Menu Özelliği

3.5.1. PushButton (Basılır Düğme)

Hemen hemen bütün iletişim kutuları tıklanıldığı zaman bir olayın gerçekleştiği bir basılır düğme (pushbutton) içerir. Bazen bir işleme onay vermek, bir arama işlemini başlatmak veya yardım istemek için kullanılır.

Butonu da diğer kontroller gibi ToolBox'tan  seçerek kullanabiliriz. Bir butonun forma eklenmiş hali aşağıdaki gibidir.



Resim 3.28: Push Buton

Aşağıdaki ilk şekilde düğmenin adını Çıkış olarak değiştirdik. Şekilde görüldüğü gibi Çıkış sözcüğünün başındaki "&"sembolü kendinden sonra gelen harfin altını çizerek kısayol tuşu oluşturur. Diğer şekilde ise düğmenin proje içindeki değişken adını Çıkış yapalım.

Text	&Çıkış
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True
UseVisualStyleBackColor	True
UseWaitCursor	False
Behavior	
Data	
Design	
(Name)	Çıkış

Resim 3.29: Push Button özellikleri

Bu işlemlerden sonra Çıkış adındaki düğmenin üzerine çift tıklayarak kod penceresini açalım ve buradaki yordama aşağıda görülen kodu ekleyelim.

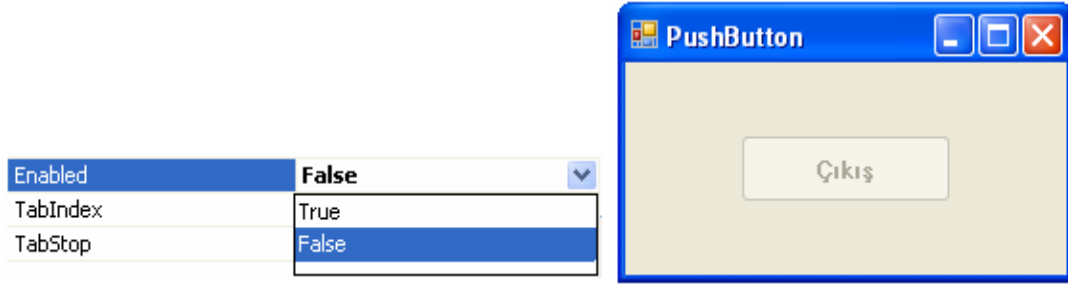
```
Private Sub Çıkış_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Çıkış.Click
```

```
Global.System.Windows.Forms.Application.Exit()
```

```
End Sub
```

Bu kodu yazıp Çıkış düğmesine basdıktan sonra program sona erecektir.

Formumuzdaki düğmeyi pasif hale getirmek için Enabled özelliğine False değerini atamalıyız. Bu işlemten sonra form görünümü resim 3.30 gibi olacaktır.



Resim 3.30: Push Button özellikleri

Enabled özelliğini Properties ten değiştirmek kullanışlı değildir. Örneğin veri tabanı ile ilgili bir proje yaptığımızı düşünürsek, veri tabanında kayıt yoksa güncelleme butonunun aktif olmasına gerek yoktur. Bu durumda Enabled özelliğini kod yazarak değiştirmeliyiz.

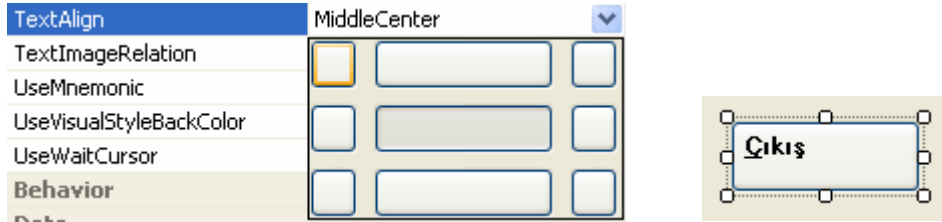
```
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Me.Load
```

```
    Çıkış.Enabled = False
```

```
End Sub
```

Programa bu kodu eklersek form yüklenirken Çıkış düğmesi pasif olur.

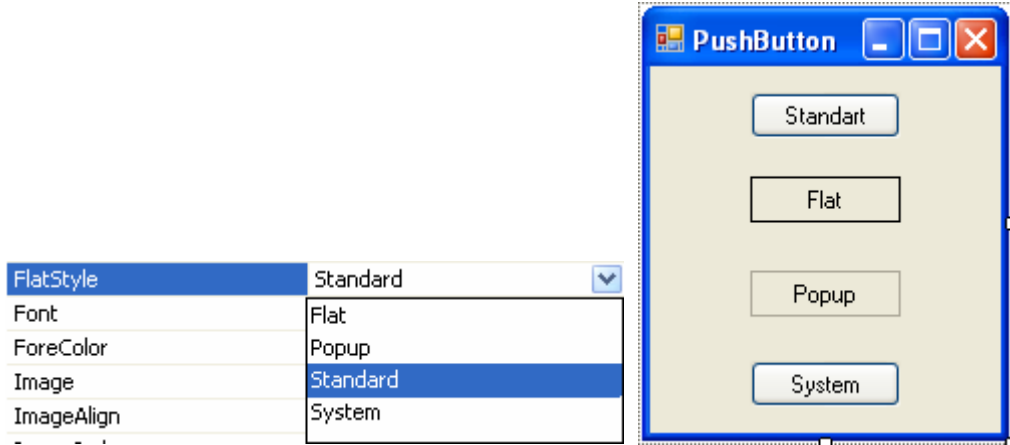
Şimdi de **TextAlign** özelliği ile butonun üzerine yazılan yazının yerinin belirlenmesi özelliğini görelim.



Resim 3.31: Push Button özellikleri

Görüldüğü gibi mouse ile seçtiğimiz kenardan yazı yazılmaya başlayacaktır ve o kenara formatlanacaktır.

Butonların bir başka özelliği ise **FlatStyle** özelliğidir. Dört farklı değer alabilir. Bunlar Standart, flat, popup ve sistemdir. Görünüm şekilleri aşağıdaki gibidir.



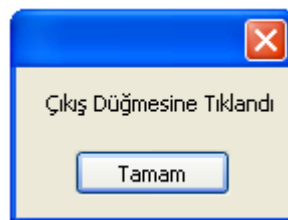
Resim 3.32: Push Button çeşitleri

3.5.1.1. PerformClick() Metodu

Bildiğiniz üzere butonların çalışması için üzerine tıklamak gerekiyordu. Ancak PerformClick() metodu ile düğme tıklanmış gibi bir etki yapabiliriz. Bunun için bir form oluşturup ve bir düğme ekleyelim. Butonun Click yordamına MessageBox sınıfı ile “buton tıklandı” yazalım. Formun Load olayına aşağıdaki kodu yazalım.

```
Private Sub Çıkış_Click(ByVal sender As System.Object ByVal e As System.EventArgs) Handles Çıkış_Click
    MessageBox.Show("Çıkış Düğmesine Tıklandı")
End Sub
Private Sub Form1_Load(ByVal sender As System.Object,
    Çıkış.PerformClick()
End Sub
```

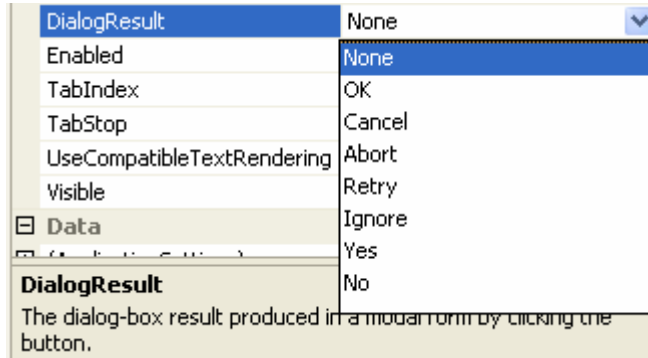
Bu şekilde proje çalıştırıldığında button1 form yüklenirken işletilmiş olacak ve ekrana “Çıkış Düğmesine Tıklandı” yazacaktır.



Resim 3.33: Push Button

3.5.1.2. DialogResult Özelliđi

Düğmeleri kullanma amacımızı belirlerken DialogResult özelliđinden yararlanabiliriz. DialogResult özelliđi ShowDialog metodu ile ekrana getirilen formlardaki düğmelerde işlevsel olabilmektedir. Bunu bir örnek ile anlatacađız. Önce projemize iki tane form ve Form2'ye de iki tane buton ekleyip DialogResult özelliklerini deđiştirelim. Buton1'in DialogResult özelliđini "Yes", Buton2'nin özelliđini de "No" yapalım. Daha sonra Form1'in FormClosing yordamını seçip kod penceresine gerekli kodlarımızı yazalım.



Resim 3.34: Push Button Dialog Result özellikleri

```
Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As System.
```

```
Form2.ShowDialog()
```

```
DialogResult = (Form2.DialogResult)
```

```
If (Form2.DialogResult = Windows.Forms.DialogResult.Yes) Then  
    MessageBox.Show("Evet Düğmesine Tıklandı")
```

```
End If
```

```
If (Form2.DialogResult = Windows.Forms.DialogResult.No) Then  
    MessageBox.Show("Hayır Düğmesine Tıklandı")
```

```
End If
```

```
End Sub
```

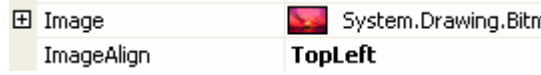
Form1' in Closing yordamına yazılan kodlar Form1 kapatılırken işleneceđi için Form1 kapatılırken Form2'nin butonlarından gelen DialogResult deđişkenini beklemek zorundadır. Aksi takdirde kapatılmaz.

3.5.1.3. TabIndex ve TabStop Özellikleri

Formlarımızın üzerinde birçok nesne bulunabilir. Programımızı kullanan kullanıcı Tab tuşu ile bu nesnelerin arasında dolaşmak isteyecektir. Buna göre nesnelerin sıra numarası TabIndex özelliğinde bulunur. Visual Basic nesneleri eklerken otomatik olarak sıra verir. Ama biz bu sırayı TabIndex özelliği ile değiştirebiliriz. Eğer bu sırada atlanmasını istediğimiz bir nesne olursa bu nesnenin TabStop özelliğini False yapmalıyız.

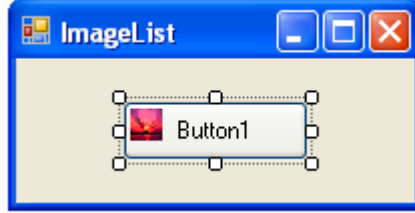
3.5.1.4. Düğmelerin Üzerine Resim Yerleştirmek

Düğmelerin üzerine resim yerleştirmek için **Image** özelliğinden yararlanılır.




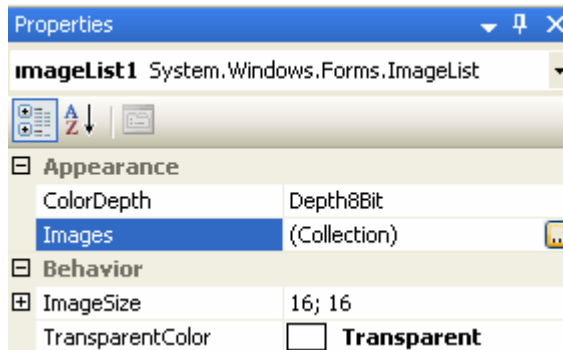
Resim 3.35: Push Button Image özelliği

Buradaki üç nokta (...) ikonuna tıklandığı zaman resim dosyası eklemek için ekrana diyalog kutusu gelir. Buradan resim dosyası seçilir. Hemen altındaki **ImageAlign** özelliği ile resmin butondaki yeri belirlenir.



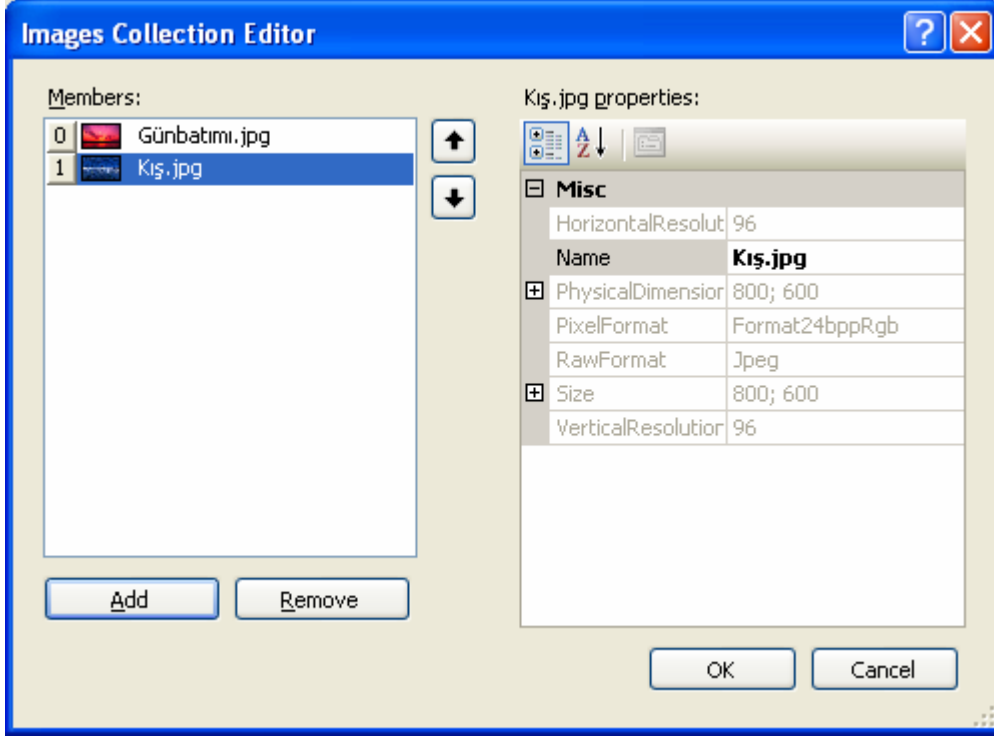
Resim 3.36: Push Button'a resim yerleştirme

Butonlara resim eklerken Toolbox'taki ImageList  ImageList kontrolünden de yararlanabiliriz. ImageList kontrolü formun üzerinde görünmeyip alt tarafa yerleşir. ImageList kontrolünün özellikler penceresindeki Images değerinin koleksiyonuna girerek projelerimizde kullanmak istediğimiz imajeleri ekleyebiliriz.



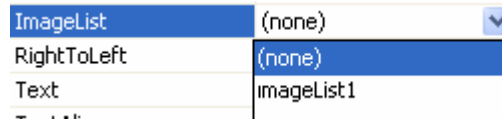
Resim 3.37: ImageList özellikleri

ImageList deęerinin yanındaki üç noktaya “...” tıklayınca karşımıza ekleme çıkarma yapabileceğimiz Image Collection Editor penceresi gelecektir.



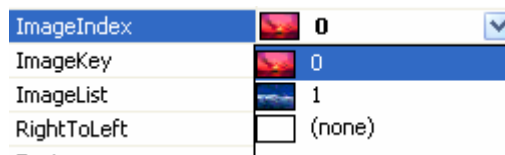
Resim 3.38: Images Collection Editor

Bu pencerede istediğimiz resimleri ekledikten sonra Button'nun özellikler penceresine geçelim. Daha önce sadece “None” deęeri bulunan ImageList özellięi artık ImageList1 deęerine de sahip.



Resim 3.39: ImageList

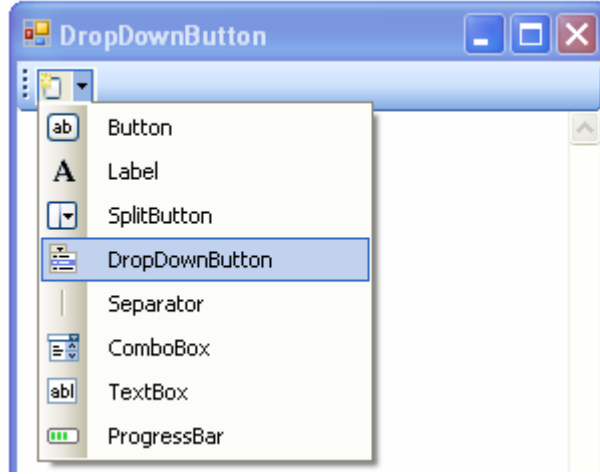
ImageList1 seçilince Button'nun ImageIndex deęerinde daha önce ImageList1'e eklediğimiz resimler numarası ile görünecektir. İstenilen resim seçilince butonun üzerine eklenecektir.



Resim 3.40: ImageList

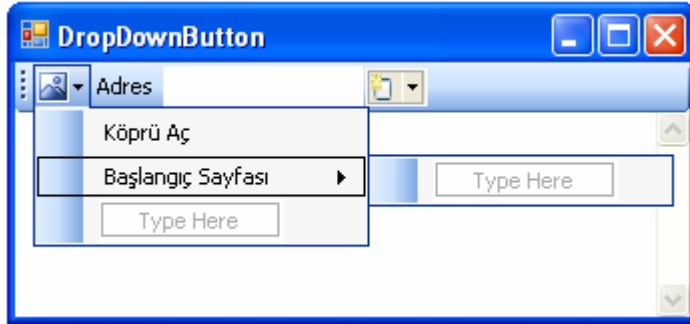
3.5.2. DropDownMenu Özelliği

DropDownButton, bize kayan bir menü özelliği sunmaktadır. Formumuzda DropDownButton kullanabilmek için öncelikle bir ToolStrip veya StatusStrip eklememiz gerekmektedir.



Resim 3.41: DropDownButton Ekleme

DropDownButton tıpkı bir menü gibidir. Daha önce MenuStrip de yaptığımız gibi komutlar ekleyebiliriz. Bunu bir örnekle anlatalım. Formumuza önce ToolStrip, WebBrowser ekleyelim. Daha sonra ToolStrip'e bir DropDownButton, Label ve TextBox ekleyelim. DropDownButton'a "Köprü Aç" ve Başlangıç Sayfası" adında iki komut yazdıktan sonra, Label'in adını Adres olarak değiştirelim ve kod penceresine gerekli olan kodlarımızı yazalım.



Resim 3.42: DropDownButton ile menü oluşturma

```
Private Sub KöprüAçToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As Sys
```

```
WebBrowser1.Navigate(ToolStripTextBox1.Text) ' web sayfasına  
bağlanmayı sağlıyor
```

```
End Sub
```



```
Private Sub BaşlangıçSayfasıToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BaşlangıçSayfasıToolStripMenuItem_Click
```

```
    ToolStripTextBox1.Text = "http://www.google.com.tr"
```


```
    WebBrowser1.Navigate(ToolStripTextBox1.Text) ' web sayfasına bağlanmayı sağlıyor
```

```
End Sub
```

Gerekli olan kodlarımızı yazıp programımızı çalıştırdığımızda, Köprü Aç denilince TextBox'a yazılan adres açılır ya da kod penceresine yazılan başlangıç sayfasına göre diğer komutu işletebiliriz.

3.6. Diyalog Kutuları

➤ Klasöre Gözet Diyalog Kutusu (FolderBrowserDialog)

FolderBrowserDialog kontrolü  FolderBrowserDialog, hazırlayacağımız projelerde tıpkı Windows Gezgininde olduğu gibi klasörleri taramamızı sağlar. Form üzerine buton ve FolderBrowserDialog kontrollerini ekledikten sonra kod penceresine aşağıdaki satırı yazalım.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
    FolderBrowserDialog1.ShowDialog()
```

```
End Sub
```

Daha sonra programımızı çalıştırıp butona tıklayınca karşımıza Klasöre Gözet diyalog kutusu gelecektir.



Resim 3.43: Klasöre Gözet diyalog kutusu

➤ Yazı Tipi Diyalog Kutusu (FontDialog)

FontDialog kontrolü `FontDialog`, geliştireceğimiz uygulamalarda yazı tipini belirlememizi sağlayan bir kontroldür. Toolbox üzerinden seçtiğimiz FontDialog kontrolünü, form üstüne bıraksak bile, formun altında görüntülenecektir. Form üzerine buton, textbox ve fontdialog kutusu ekledikten sonra kod penceresini aşağıdaki gibi düzenleyelim.



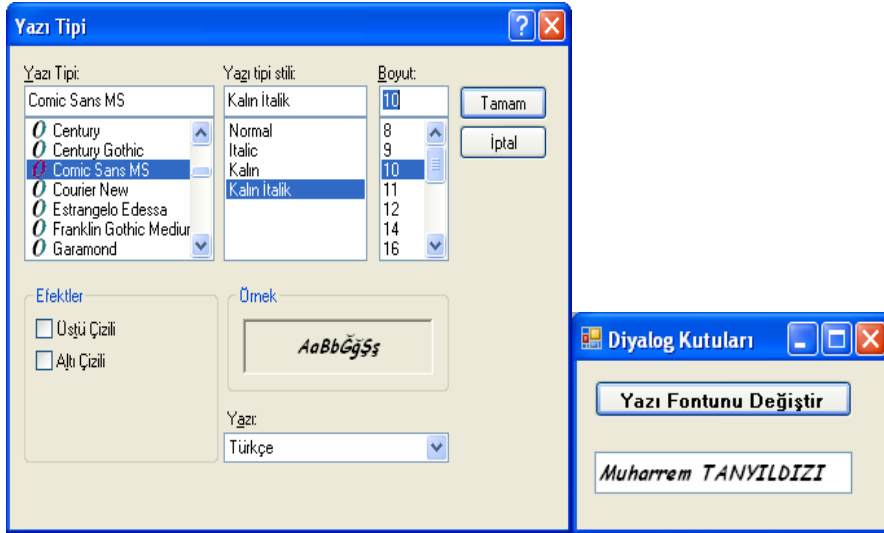
Resim 3.44: Font Dialog

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Me.Load
    TextBox1.Text = "Muharrem TANYILDIZI"
End Sub
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    FontDialog1.ShowDialog()
    TextBox1.Font = FontDialog1.Font
End Sub
```


Gerekli olan kodları yazıp programımızı çalıştırınca ekrana gelen formda “Yazı Fontunu Değiştir” butonuna tıklayalım.

Şimdi karşımıza Yazı Tipi diyalog kutusu gelecektir. İstedığımız yazı tipini seçip Tamam tuşuna tıklarsak TextBox’ımızdaki yazı fontu değişecektir.




Resim 3.45: Font Dialog

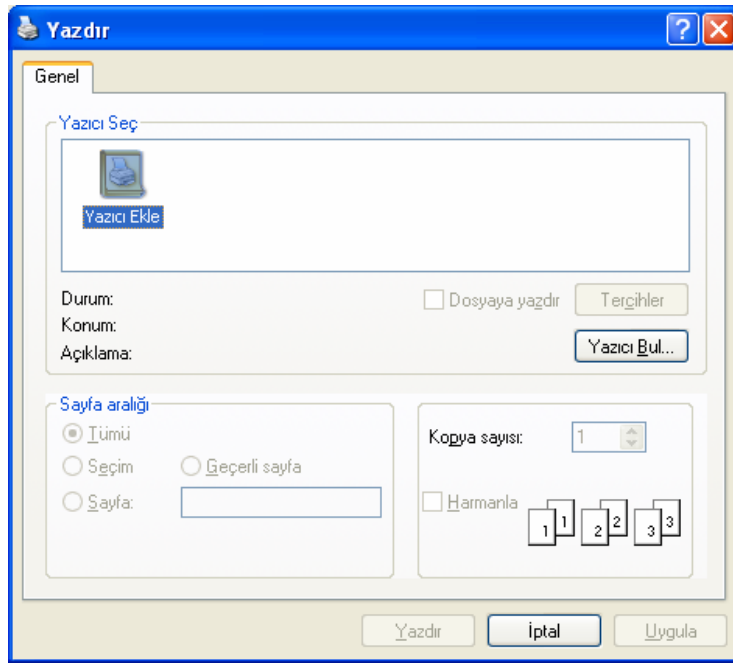
➤ Dosya Açma Diyalog Kutusu (OpenFileDialog)

OpenFileDialog kontrolü  `OpenFileDialog`, herhangi bir Windows uygulamasında Dosya/Aç komutunun kullanılması ile karşımıza çıkan Dosya Aç diyalog kutusunun projelerimizde kullanmamızı sağlayan bir kontroldür. Bu diyalog kontrolü de diğer kontroller gibi form altında görüntülenir. Form üzerine buton ve OpenFileDialog kontrolü ekledikten sonra kod penceresini aşağıdaki gibi düzenleyebiliriz.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    OpenFileDialog1.ShowDialog()
End Sub
```


➤ Yazdırma Diyalog Kutusu (PrintDialog)

PrintDialog kontrolü  `PrintDialog`, herhangi bir Windows uygulamasında Dosya/Yazdır komutunun kullanılması ile karşımıza çıkan Yazdır diyalog kutusunun projelerimizde kullanmamızı sağlayan bir kontroldür. Diğer diyalog kutularının gösterilmesi yöntemiyle ekrana getirebiliriz.

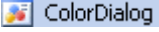


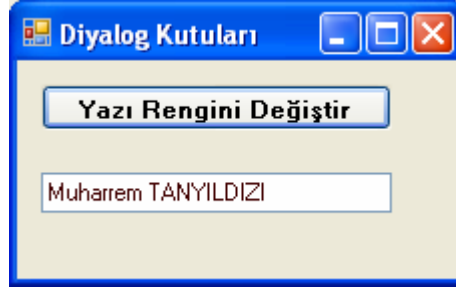
Resim 3.46: Print Dialog

➤ Kaydetme Diyalog Kutusu (SaveFileDialog)

SaveFileDialog kontrolü  `SaveFileDialog`, herhangi bir Windows uygulamasında Dosya/Kaydet komutlarında ekrana gelen Dosya Kaydetme arayüzünü uygulamalarda kullanmamızı sağlayan bir kontroldür.

➤ Renk Diyalog Kutusu (ColorDialog)

ColorDialog kontrolü , geliştirdiğimiz uygulamalarda kullanabileceğimiz yazı rengini belirlememizi sağlayan bir kontroldür. Belirlediğimiz rengi kontrol özelliklerinde (buton rengi, formun zemin rengi, vb. gibi) kullanabileceğimiz gibi, grafik uygulamalarında da kullanabiliriz. Bu diyalog kontrolü de diğer kontroller gibi form altında görüntülenir.

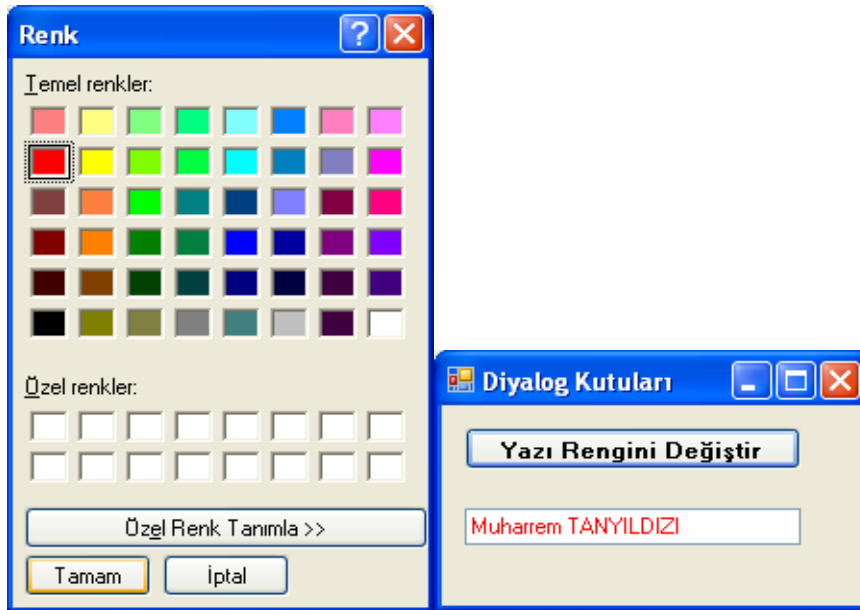


Resim 3.47: Color Dialog

Formu tasarladıktan sonra kod penceresine aşağıdaki satırları yazalım.


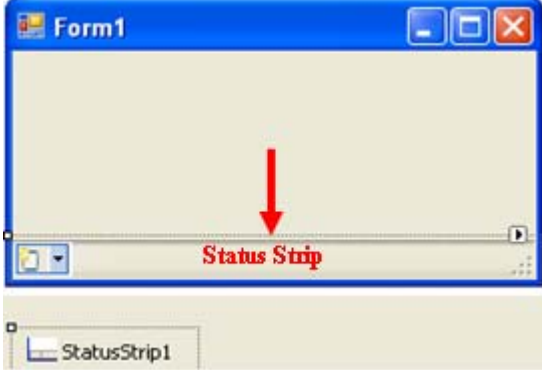

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    ColorDialog1.ShowDialog()
    TextBox1.ForeColor = ColorDialog1.Color
End Sub
```

Şimdi programımızı çalıştırabiliriz.



Resim 3.48: Color Dialog

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. Form üzerine araç çubukları ekleyiniz.</p> 	<p>ToolStrip kontrolünü kullanınız.</p>
<p>1. Formunuza durum çubuğu ekleyiniz.</p> 	<p>Status Strip kontrolünü kullanınız.</p>
<p>3. Forma Context menü ekleyiniz.</p> 	<p>Sağ buton kullanarak form üzerinde erişim sağlayabilirsiniz.</p>

ÖLÇME VE DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki cümlelerde verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. ToolStrip kontrolü ile uygulamamıza araç çubuğu ekleyebiliriz. ()
2. Dock seçeneği ile StatusStrip’in görünümünü belirleyebiliriz. ()
3. StatusStripTasks diyalog kutusunda bulunan Insert Standarts Items seçeneği ile standart bir durum çubuğu oluşturabiliriz.()
4. Items Collection Editor penceresi ile araç çubuğu elemanlarının yerlerini değiştirip istediğimiz elemanı çıkartabiliriz. ()
5. ToolTip özelliklerinden olan **InitialDelay**, Mouse kontrolün üzerine geldiğinde metnin ne kadar sürede görüneceğini belirlemenize yarar.()
6. PerformClick() metodu ile düğme tıklanmış gibi bir etki yapabiliriz ()
7. DropDownButton, bize kayan bir menü özelliği sunmaktadır ()
8. ToolTip kontrolü ile uygulamadaki herhangi bir kontrole başlık ekleyebiliriz ().
9. Komutların arasına çizgi koyup gruplamak istersek ekleme noktası komut metni yazılan yerde iken kısa yol menüsünü açıp Insert Seperator komutunu vermeliyiz.()

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız.

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

ÖĞRENME FAALİYETİ-4

AMAÇ

Bu modül ile gerekli ortam sağlandığında, kullanıcı ara yüzünü geliştirip yardım bilgileri ekleyebileceksiniz.

ARAŞTIRMA

Bu faaliyet öncesinde hazırlık amaçlı aşağıda belirtilen araştırma faaliyetlerini yapmalısınız.

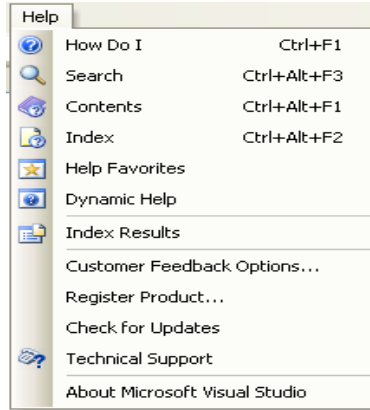


- Yardım menüleri neden kullanılır. Araştırınız.
- Kullandığınız programlarda yardım alabilmek için kullanabileceğiniz komutları ve kısayol tuş kombinasyonlarını araştırınız.
- *.chm dosya uzantıları hangi tür dosyaları içerir, araştırınız.

4. YARDIM

4.1. Yardım Menüsü

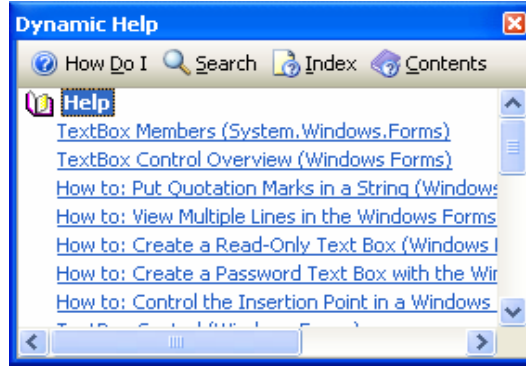
Help (yardım) menüsü her düzeydeki programcının ihtiyaç duyacağı bir olanaktır. Help menüsünün alt menüleri aşağıdaki gibidir.



Resim 4.1: Help menüsü

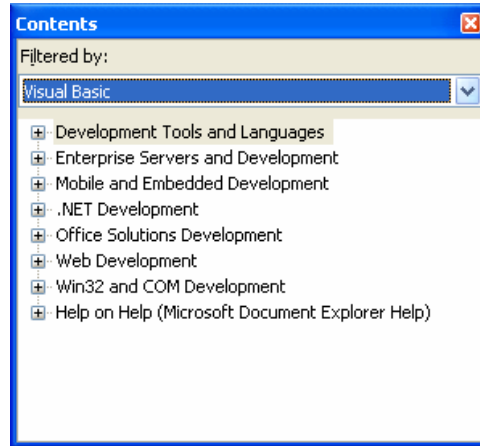
- **How Do I:** İnternet bağlantımız olduğu zaman bize online yardım imkanı sunar. Ctrl+F1 kısayol kombinasyonu ile bu komuta ulaşabiliriz.
- **Search :** MSDN Online, Codezone Community, Questions, Local Help olmak üzere dört farklı şekilde bütün Help belgelerinde arama yapmamızı sağlar. Kısayol tuş kombinasyonu Ctrl+Alt+F3 şeklindedir.

- **Dynamic Help:** Visual Basic .NET ortamındaki herhangi bir kontrol için içerik hassasiyetli yardım (context sensitive) biçimidir. Araç kutusundan herhangi bir kontrolü seçtikten sonra Help menüsü Dynamic Help komutu seçilince Dynamic Help penceresi ekrana gelecektir. Dynamic Help penceresi Visual Basic .NET ortamındaki aktif olan nesneye yönelik bilgileri bize sunar. Pencerede açılan seçeneklerden birini seçince program bizi yardım penceresine götürecektir.



Resim 4.2: Dynamic Help

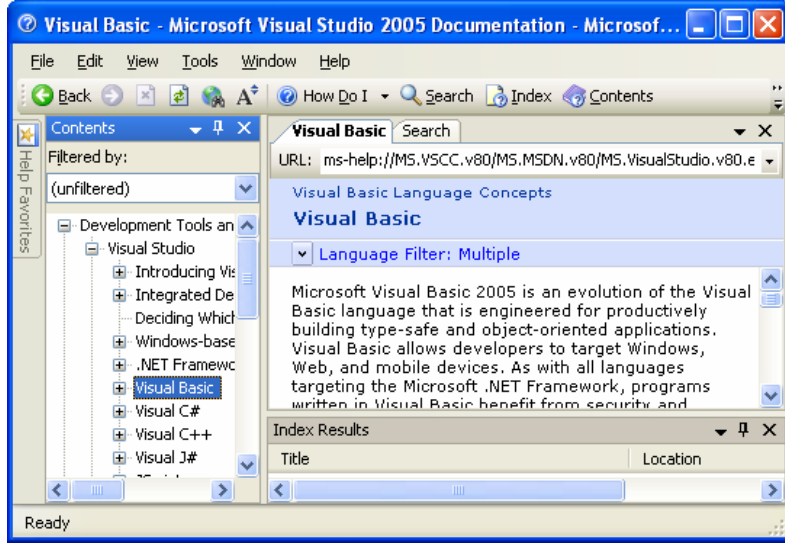
- **Contents:** Bu pencere çeşitli bağlantıları kullanarak bilgilere erişmemizi sağlar. Contents penceresini Help menüsü Contents komutunu kullanarak açabileceğimiz gibi Ctrl+Alt+F1 kısayol tuş kombinasyonu ile de açabiliriz. Pencere açılınca karşımıza birbiri ile bağlantılı konuların bulunduğu bir görüntü çıkar.



Resim 4.3: Contents Ana Pencere

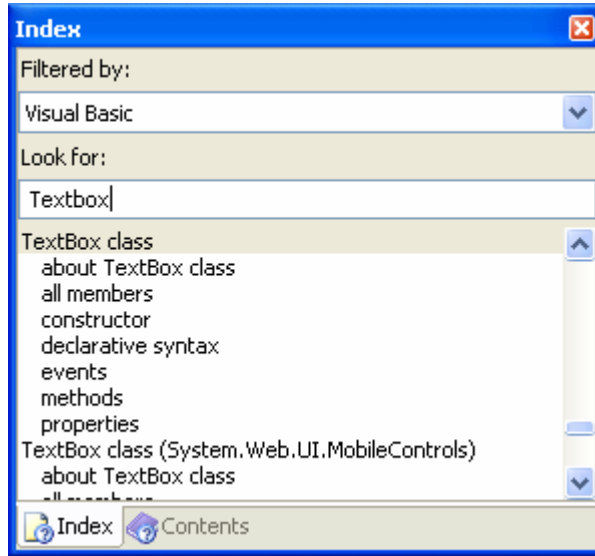
Pencerenin üst kısmında bulunan Filtered By seçeneği ile konuları filtreleyerek sadece istediğimiz konularda yardım alabiliriz.

Konuların yanında bulunan artı (+) işaretlerine tıklayarak genişletebiliriz. Aradığımız konuyu tıklayınca konu ile ilgili Help sayfasına gidebiliriz.



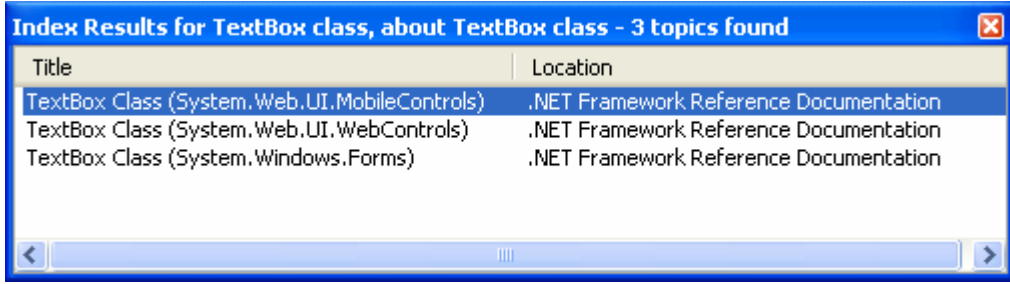
Resim 4.4: Contents Alt seçenekler

- **Index:** Bu pencere çeşitli bağlantıları kullanarak belirli bilgileri aramamızı sağlar. Index penceresini Help menüsü Index komutunu kullanarak açabiliriz. Ayrıca Ctrl+Alt+F2 kısayol tuş kombinasyonu ile de açabiliriz.



Resim 4.5: Index

Tıpkı Contents penceresinde olduğu gibi Index penceresi de filtreleme özelliğine sahiptir. Look for kutusuna aradığımız metni girince indeksli bir liste karşımıza çıkar. Bu listeden aradığımız konuyu tıklayınca Index Results penceresi açılacaktır.



Title	Location
TextBox Class (System.Web.UI.MobileControls)	.NET Framework Reference Documentation
TextBox Class (System.Web.UI.WebControls)	.NET Framework Reference Documentation
TextBox Class (System.Windows.Forms)	.NET Framework Reference Documentation

Resim 4.6: Index Results

Bu pencerede aradığımız konu ile ilgili listelerden birini tıklayınca görüntülemek istediğimiz konu açılır.

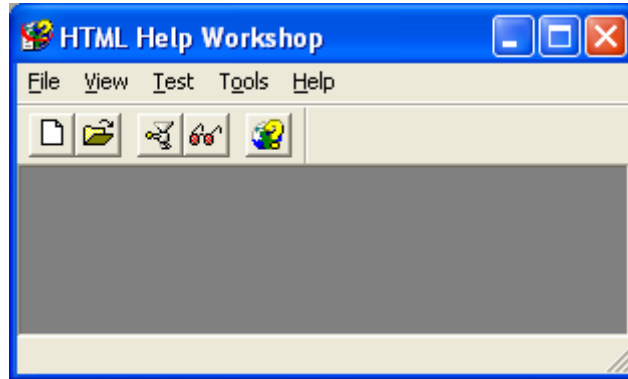
4.2. “HTML Help Workshop” Programı

Visual Basic .NET ile yapacağımız programları sadece biz kullanmayacağımız için diğer kullanıcılara kolaylık sunması açısından yardım menüsü de oluşturmamız gerekmektedir.

Yardım için gerekli olan her sayfayı herhangi bir HTML editöründe ayrı bir HTML dosyası olarak hazırladıktan sonra HTML Help Workshop programı ile HTML dosyalarını birleştirerek bir HTML Help oluşturabiliriz.


HTML Help yardım dosyamızı hazırlamak için öncelikle *.htm , *.html formatında içeriğimizi hazırlamamız gerekmektedir.

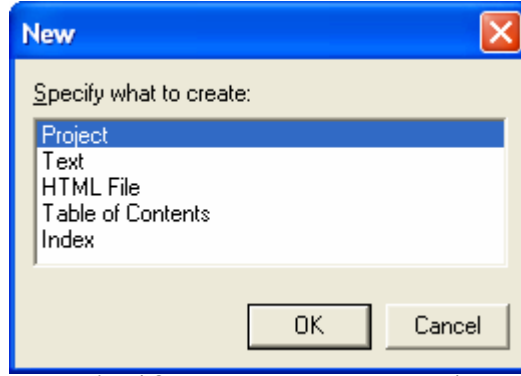
Daha sonra HTML Help Workshop programını çalıştırıp hazırladığımız dokümanları birleştirmeye başlayalım.



Resim 4.7: HTML Help Workshop

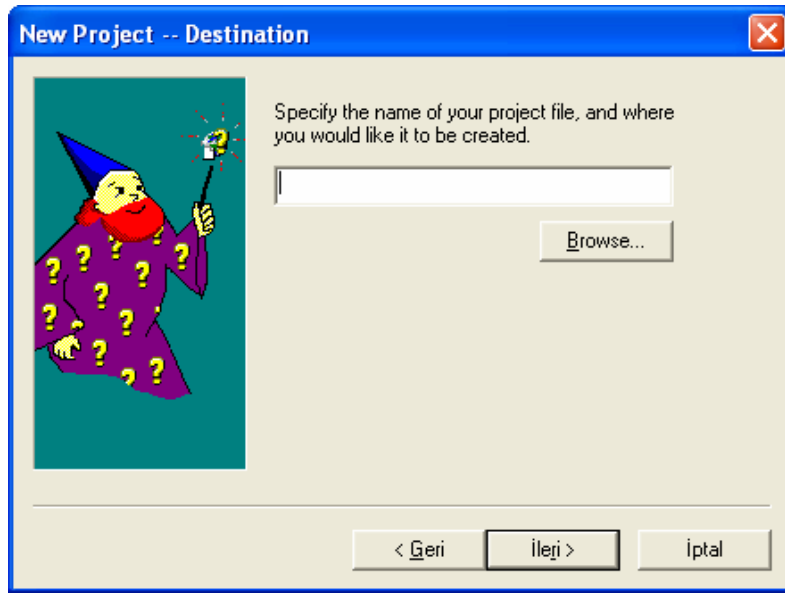
➤ **Yardımlı Dosyasının Hazırlanması**

File menüsünden New diyerek ya da araç çubuğundan  Yeni simgesini tıklayalım.



Resim 4.8: HTML Help Workshop/New

Sonrada Project'i seçerek OK butonuna basarak New Project sihirbazını ekrana getirelim.



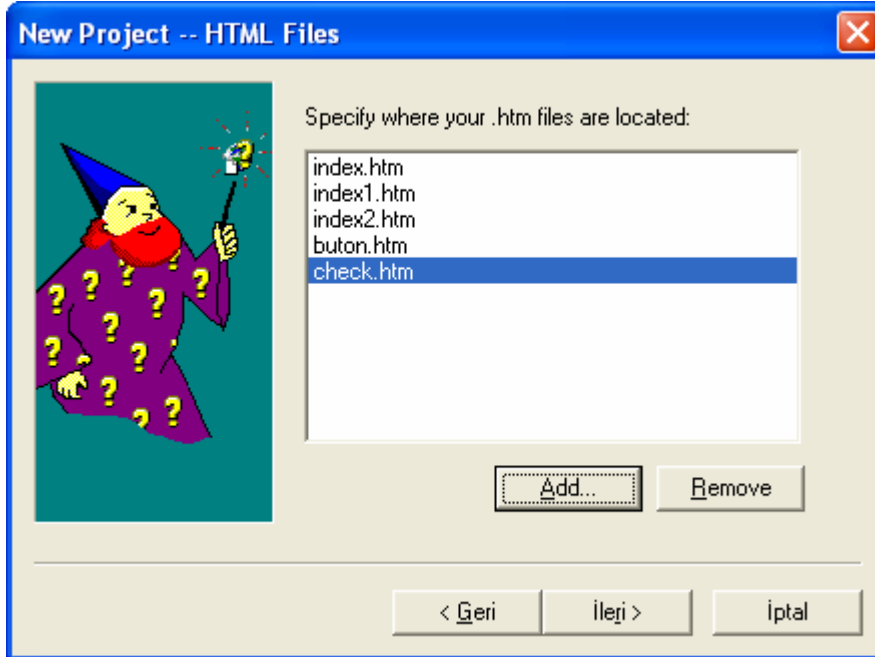
Resim 4.9: HTML Help Workshop/New Project

Browse butonuna basarak oluşturacağımız *.cfm dosyamızın oluşturulacağı yeri ve o dosyaya vereceğimiz adı belirtelim. Daha sonra ise bize hangi dosyalarımızın hazır olduğunun sorulduğu ekran gelecektir. Bizim şu anda sadece HTML dosyalarımız hazır, onun için HTML files (.htm) seçeneğini seçip Next butonuna basıyoruz.



Resim 4.10: HTML Help Workshop/New Project

Bir sonraki ekranda Add... butonuna basarak Help dosyası için hazırladığımız HTML formatındaki dosyalarımızı ekliyoruz. Daha sonra Next butonuna basarak bir sonraki sayfaya geçiyoruz.

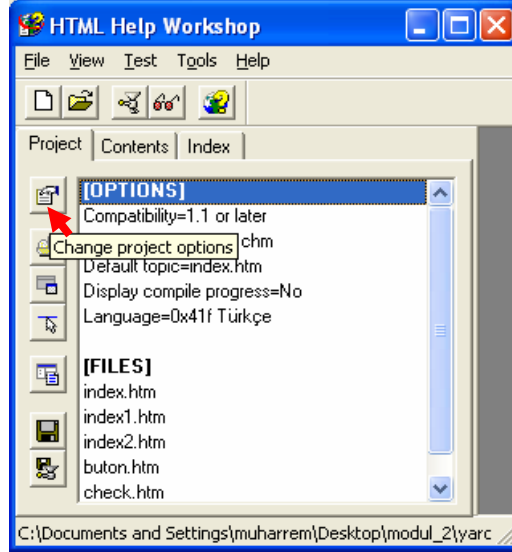


Resim 4.11: HTML Help Workshop/New Project

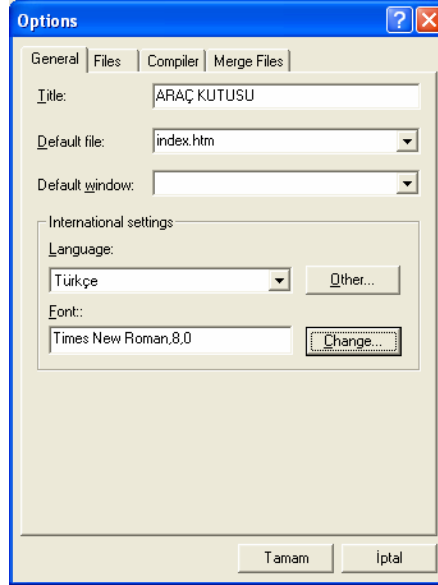
Dosyalarımızı ekledikten sonra Finish butonuna basarak işlemimizin bu kısmını bitirebiliriz.

➤ Options / Seçenekler

Şimdi yardım dosyamız için gerekli options ayarlarımızı tamamlayacağız. Bunun için sol tarafta bulunan **Change Project Options** butonuna tıklayalım ve Options ekranını açıp ayarlarımızı yapalım.




Resim 4.12: HTML Help Workshop/Options

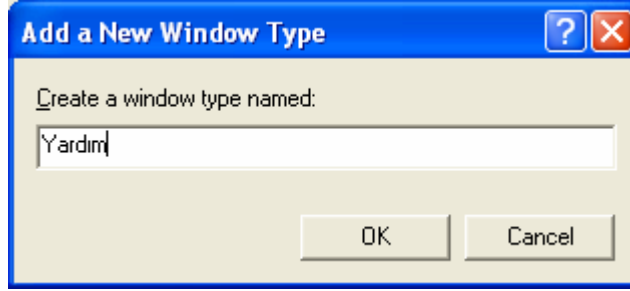


Resim 4.13: HTML Help Workshop/Options

- Title :** Yardım dosyamızın başlığı
- Default File :** Yardım dosyamız ilk açıldığında görüntülenmesini istediğimiz HTML dosyası
- Language :** Dil seçeneği
- Font :** Tercih edilen font

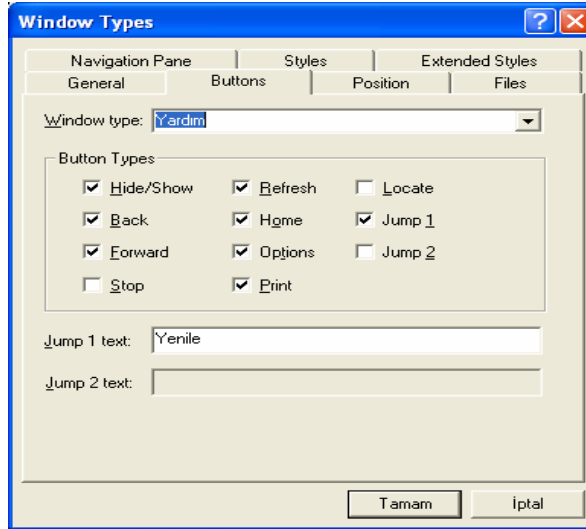
➤ Windows Ayarları

Yardım dosyamızın görüntüleneceği pencerenin ayarlarını yapmak için **Add/modify windows definitions**  butonuna tıklayalım. Karşımıza gelen ekrandan pencerenin ismini girelim.



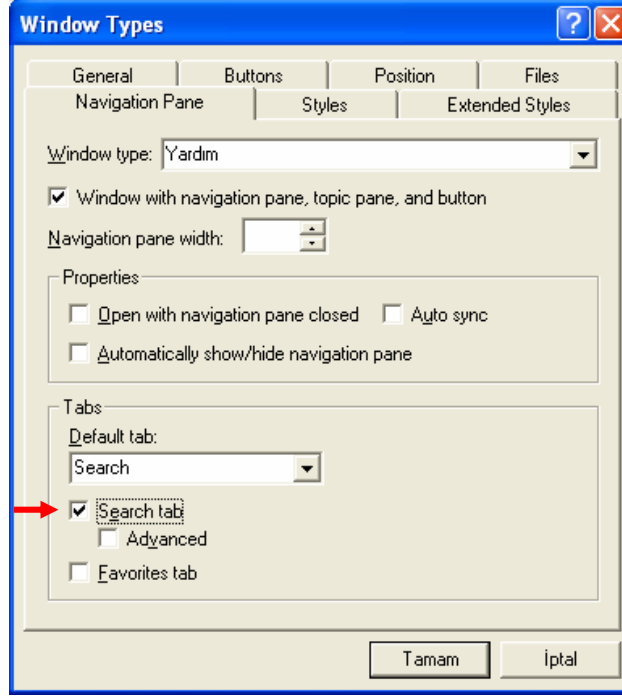
Resim 4.14: HTML Help Workshop/Windows

Title Bar Text kısmına pencerede görünecek başlığı girdikten sonra Buttons sekmesindeki Buttons Type kısmından pencerede görünmesini istediğimiz butonları ayarlayabiliriz. Standart butonların yanında Jump1 ve Jump2 adında 2 tane buton vardır. Bunlar işaretledikten sonra istediğimiz gibi ayarlayabiliriz.



Resim 4.15: HTML Help Workshop/Windows

Yardım dosyamızda arama seçeneği olması için Navigation Pane sekmesinde önce “Windows with navigation...” kısmını sonra da sayfanın altındaki **Search Tab** kısmını işaretleyelim.



Resim 4.16: HTML Help Workshop/Windows

Tamam butonuna basarak pencere ayarlarını da bitirelim.


➤ **Contents (içerik) Dosyasının Oluşturulması**

Şimdi Contents sekmesine geçelim. İçerik dosyamızı belirtmediğimiz için bize yeni bir içerik dosyası oluşturmayı veya mevcut bir içerik dosyasını kullanmayı önerecektir. Yeni bir içerik dosyası oluşturmak Create a new contents file seçeneğini, mevcut içerik dosyasını kullanmak için ise Open an existing contents file seçeneğini işaretleyip OK butonuna basalım.



Resim 4.17: HTML Help Workshop/Windows

Eğer yeni bir içerik oluşturmak istersek bu Contents dosyasını proje dosyasıyla aynı klasöre kaydetmemiz gerekir. Bundan sonra içerik dosyamıza konu başlığı veya sayfa ekleyebiliriz.

Öncelikle Insert a page  butonuna basıp bir sayfa ekleyelim. Entry Title kısmına ekranda gözükecek ismi girdikten sonra, Add butonuna basarak projeye eklediğimiz ilgili HTML dosyasını seçelim.

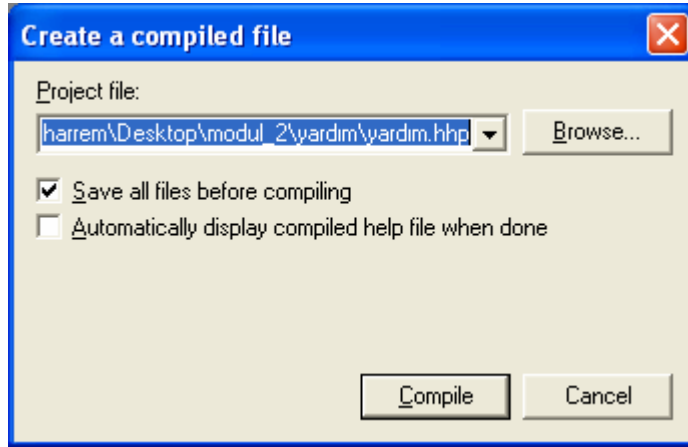
Bir konu başlığı eklemek için ise Insert a Heading butonuna bastıktan sonra Entry title kısmına ekranda gözükecek ismi yazalım.

➤ **Index (İndeks) Dosyasının Oluşturulması**


Index sekmesine geçip tıpkı Contents'te olduğu gibi, bize yeni bir index dosyası oluşturmayı veya mevcut bir index dosyasını kullanmayı önerir. Yeni bir index dosyası oluştururken seçeneğini (Create a new index file) seçip OK butonuna basalım.

➤ ***.chm Dosyasının Oluşturulması**

Project kısmına geçip, Change Project Options butonuna basınız. Default Window kısmında ayarlarımızı yaptığımız pencereyi seçelim.

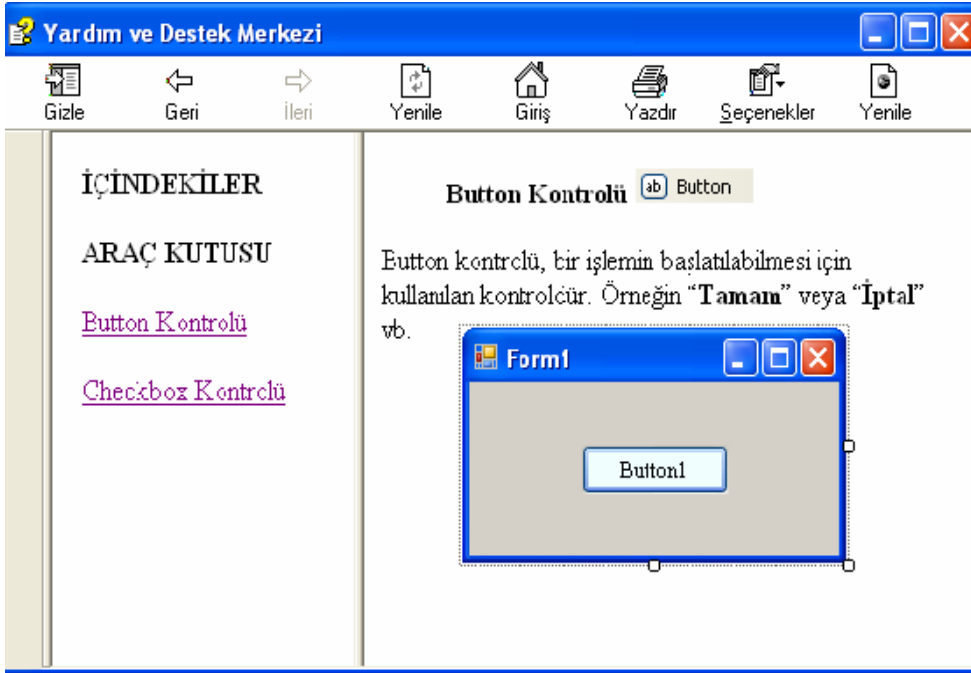


Resim 4.18: HTML Help Workshop

Son işlem olarak araç çubuğundan Compiles a Help Project File  butonuna basıp HTML yardım (.chm) dosyasını oluşturalım.



Yardım dosyamız oluşturduğunda klasördeki görüntüsü Resim 4.19 gibi olacaktır. Bu dosyayı açınca karşımıza hazırlamış olduğumuz yardım belgesi gelecektir.

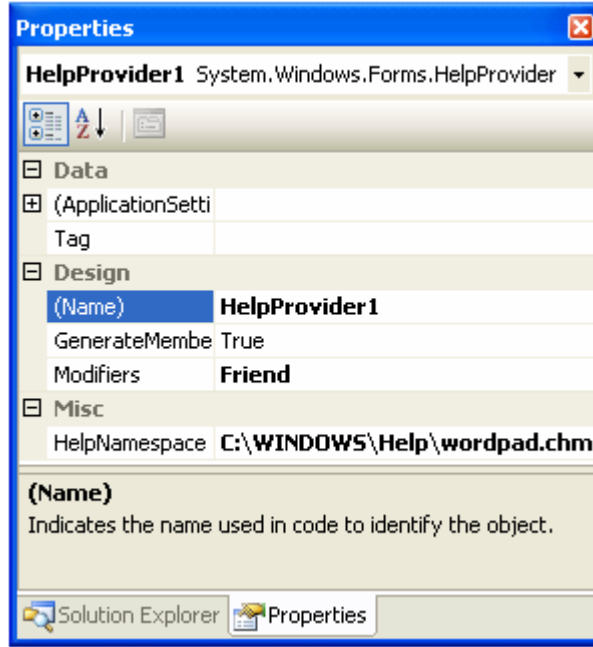


Resim 4.19: HTML Help Workshop

4.3. "HelpProvider" Nesnesi (F1 Tuşu Yardımı)

Bir bilgisayar kullanıcısı olarak F1 yardım tuşunun önemini inkâr edemeyiz. Hangi programı kullanıyorsak kullanalım bilmediğimiz merak ettiğimiz bir noktada yardım ve destek merkezlerine ulaşmamızı sağlar. Bizler de hazırlayacağımız projelerimizde diğer kullanıcılara yardımı götürmeliyiz.

Help Provider nesnesi F1 tuşunun kullanılmasına olanak tanır. Araç kutusundan Help Provider **F1 HelpProvider** nesnesine tıklayınca sayfanın alt tarafına yerleşir. Help Provider nesnesinin Properties penceresinde F1 tuşuna basınca açılmasını istediğimiz *.chm uzantılı yardım dosyasını HelpNamespace özelliğine ekleyelim.



Resim 4.20: Help Provider Properties

Daha sonra formun Properties penceresine giderek Help Provider nesnesini ekledikten sonra gelen özellikleri inceleyelim ve gerekli değişiklikleri yapalım.

- **ShowHelp on Help Provider:** Değerinin True olması gerekir. Çünkü False değerini alırsa F1 tuşu çalışmayacaktır.
- **HelpNavigator on HelpProvider:** F1 tuşuna basınca açılacak olan yardım dosyasının içindekiler, dizin, arama sayfası seçeneklerinden hangisi ile açılacağını belirlememize yarar.
- **Associate Index:** Yardım penceresinin Dizin sekmesini açar. Eğer “HelpKeyword on HelpProvider” özelliğine bir kelime yazılmışsa, yazılan bu kelimenin ilk harfine göre dizini düzenler.



Resim 4.21: Help Navigator/Associate Index

- **Topic:** Yardım penceresinin Dizin sekmesini açar. Konuları alfabetik sıraya göre en baştan gösterir.
- **Table Of Contents:** Yardım penceresinin İçindekiler sekmesini açar.
- **Index:** Yardım penceresinin Dizin sekmesini açar. “HelpKeyword on HelpProvider” özelliğine bir kelime yazılmışsa, yazılan bu kelimenin tamamına göre dizini düzenler.
- **Find:** Yardım penceresinin Ara sekmesini açar.



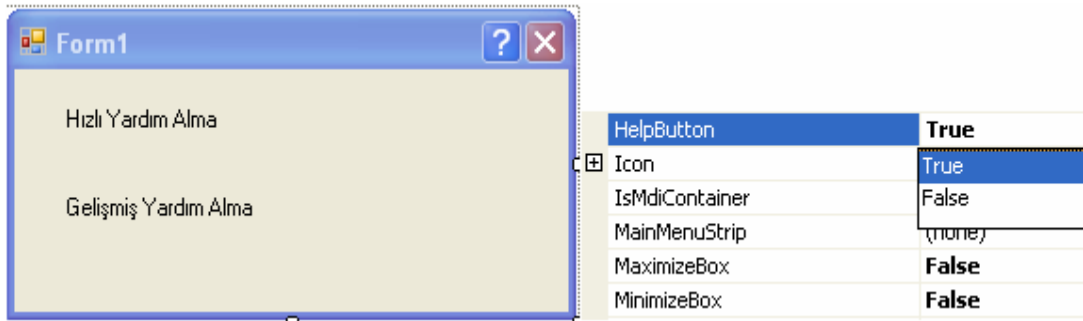
Resim 4.22: Help Navigator/Find

- **Keyword Index:** Tıpkı Index gibi Yardım penceresinin Dizin sekmesini açar. “HelpKeyword on HelpProvider” özelliğine bir kelime yazılmışsa, yazılan bu kelimenin tamamına göre dizini düzenler.
- **TopicId:** Yardım penceresinin Dizin sekmesini açar. Eğer “HelpKeyword on HelpProvider” özelliğine yazılan kelimenin tamamına göre dizini düzenler.

4.3.1. Nesnelerin “HelpString” Özellikleri

Help Provider nesnesini kullanarak yardım almanın tek yolu F1 tuşu değildir. Şimdi yeni bir form oluşturalım. Formumuza iki tane Label ve Help Provider nesnesini ekleyelim. Label1’in Text’ini “Hızlı Yardım Alma” ve Label2’nin Text değerini de “Gelişmiş Yardım Alma” olarak değiştirelim.

Formumuzun Properties penceresindeki Windows Style altında bulunan MaximizeBox ve MinimizeBox değerlerini False, Help Button değerini ise True yapalım.



Resim 4.23: Örnek Formda Help Provider

Bu işlemlerden sonra formun görüntüsü Resim 4.23 gibi olacaktır. Daha sonra Label1’in HelpString on HelpProvider değerine istediğimiz bir yardım metnini yazalım.


HelpKeyword on HelpProvider1	
HelpNavigator on HelpProvider1	AssociateIndex
HelpString on HelpProvider1	WordPad belgesi oluşturmak, açmak veya
ShowHelp on HelpProvider1	True

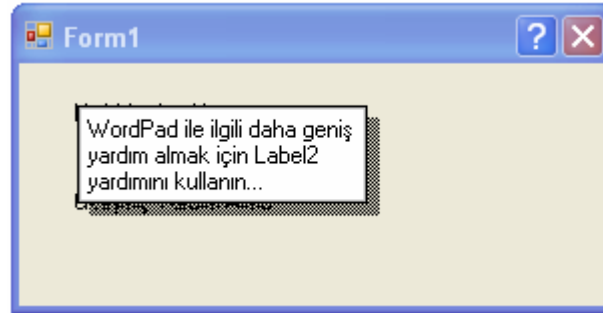
Resim 4.24: Help String

Label2 için ise, Help Provider nesnesine bir önceki örnekte olduğu gibi HelpNamespace özelliğine bir *.chm dosyası ekleyelim ve açılacak yardım dosyasının ilk görüntüsüne içindekiler sayfasını ayarlayalım.

HelpKeyword on HelpProvider1	
HelpNavigator on HelpProvider1	TableOfContents
HelpString on HelpProvider1	
ShowHelp on HelpProvider1	True

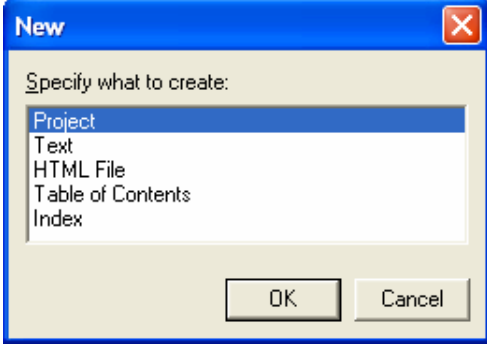
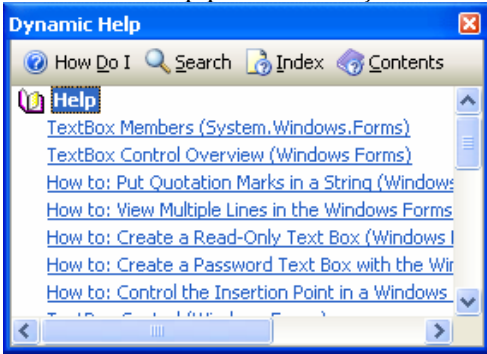
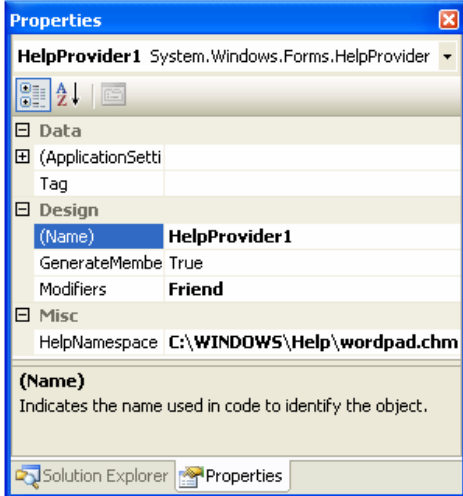
Resim 4.25: Help Keyword

Artık gerekli olan ayarlamalarımızı yaptık. Formumuzu çalıştıralım ve formun sağ üst köşesinde bulunan  simgesine tıklayıp Label1'in üstüne götürelim.



Resim 4.26 Help Provider'in son hali

UYGULAMA FAALİYETİ

İşlem Basamakları	Öneriler
<p>1. HTML Help Workshop programı ile bir chm dosyaları oluşturunuz.</p> 	<p>Öncelikle Html sayfası hazırlamayı öğrenmelisiniz. Bunun için FrontPage kullanılabilir.</p>
<p>2. Dinamik Help penceresini açınız.</p> 	
<p>2. HelpProvider nesnesinin özelliklerini değiştiriniz.</p> 	

ÖLÇME VE DEĞERLENDİRME

ÖLÇME SORULARI

Aşağıdaki cümlelerde verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. **Search** komutunun Kısayol tuş kombinasyonu Ctrl+Alt+F1 şeklindedir.()
2. **How Do I** komutuna ALT+F2 kısayol kombinasyonu ile bu komuta ulaşabiliriz. ()
3. **Statik Help** Visual Basic .NET ortamındaki herhangi bir kontrol için içerik hassasiyetli yardım (context sensitive) biçimidir.()
4. **HelpNavigator on HelpProvider:** F1 tuşuna basınca açılacak olan yardım dosyasının içindekiler, dizin, arama sayfası seçeneklerinden hangisi ile açılacağını belirlememize yarar.()
5. HTML Help Workshop / Options altında bulunan Default File seçeneği ile Yardım dosyamızın başlığını ayarlayabiliriz.. ()
6. **Topic:** Yardım penceresinin İçindekiler sekmesini açar. Konuları alfabetik sıraya göre en baştan gösterir..()
7. **Add/modify windows definitions** ile Yardım dosyamızın görüntüleneceği pencerenin ayarlarını yapabiliriz.()

DEĞERLENDİRME

Cevaplarınızı cevap anahtarı ile karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt yaşadığınız sorularla ilgili konuları faaliyete geri dönerek tekrarlayınız.

Tüm sorulara doğru cevap verdiyseniz diğer faaliyete geçiniz.

MODÜL DEĞERLENDİRME

A. PERFORMANS TESTİ (YETERLİK ÖLÇME)

Modül ile kazandığımız yeterliği aşağıdaki kriterlere göre değerlendiriniz.

DEĞERLENDİRME KRİTERLERİ	Evet	Hayır
Add Windows form ile SDI ve MDI formlar ekleyebildiniz mi?		
Bir formu IsMDIContainer özelliği ile ana form yapabildiniz mi?		
Çok formlu projelerde açılış formunu değiştirebildiniz mi?		
MenuStrip ile Menü oluşturabildiniz mi?		
ToolStrip ile Araç çubuğu oluşturabildiniz mi?		
StatusStrip ile Durum çubuğu oluşturabildiniz mi?		
ToolTip ile nesnelere ipucu metni ekleyebildiniz mi?		
Sağ tuş kısayol menüleri oluşturabildiniz mi?		
Menü elemanlarına kısayol oluşturabildiniz mi?		
Seperator ile komutları gruplayabildiniz mi?		
Set Image ile menü komutlarına resim ekleyebildiniz mi?		
ImageList kontrolünü kullanabildiniz mi?		
Flat Style ile yeni butonlar oluşturabildiniz mi?		
Help String özelliğini kullanabildiniz mi?		
“?” yardımını kullanabilme		
Help Navigator seçeneklerini kullanabilme		
HTML Help WorkShop Programını kullanabilme		
Help Provider nesnesini kullanarak F1 yardımı sağlama		

B. ÖLÇME SORULARI

Aşağıdaki cümlelerde verilen ifadeye göre parantez içine doğru ise “D”, yanlış ise “Y” yazınız.

1. Set Image ile komutların yanına onunla ilişkilendirilmiş resimleri yerleştirebiliriz. ()
2. Formumuzu parent yani ana form yapmak için Properties penceresinde IsMDIContainer özelliğini true vermemiz gerekir. ()
3. Klasöre Gözet Diyalog Kutusu “OpenFileDialog” olarak Toolbox’ta bulunur. ()
4. Butonların **FlatStyle** özelliği ile butona resim ekleyebiliriz. ()
5. Genellikle formun alt tarafında yer alan ve o andaki işlemle ilgili çeşitli durum bilgilerini görüntülemek ToolTip kontrolü kullanılır. ()
6. Help Provider nesnesi F2 tuşunun kullanılmasına olanak tanır. ()
7. Yardım dosyamızda arama seçeneği olması için Navigation Pane sekmesinde önce “Windows with navigation...” kısmını sonrada sayfanın altındaki **Search Tab** kısmını da işaretlenmelidir. ()
8. **ShowHelp on Help Provider** değeri False ise F1 tuşu kullanılabilir.()

DEĞERLENDİRME

Yaptığınız değerlendirme sonucunda eksikleriniz varsa öğrenme faaliyetlerini tekrarlayınız.

Modülü tamamladınız, tebrik ederiz. Programlamada önemli bir konu olduğundan belli zamanlarda bu modülü tekrar gözden geçiriniz.

Öğretmeniniz size çeşitli ölçme araçları uygulayacaktır. Öğretmeninizle iletişime geçiniz.

CEVAP ANAHTARLARI

ÖĞRENME FAALİYETİ-1 CEVAP ANAHTARI

1	D
2	D
3	Y
4	D
5	Y

ÖĞRENME FAALİYETİ-2 CEVAP ANAHTARI

1	D
2	Y
3	Y
4	D
5	D

ÖĞRENME FAALİYETİ-3 CEVAP ANAHTARI

1	D
2	D
3	D
4	D
5	D
6	D
7	D
8	Y
9	D

ÖĞRENME FAALİYETİ-4 CEVAP ANAHTARI

1	Y
2	Y
3	Y
4	D
5	Y
6	Y
7	D

MODÜL DEĞERLENDİRME CEVAP ANAHTARI

1	D
2	D
3	Y
4	Y
5	Y
6	Y
7	D
8	Y

Cevaplarınızı cevap anahtarları ile karşılaştırarak kendinizi değerlendiriniz.

ÖNERİLEN KAYNAKLAR

- www.yazgelistir.com
- www.vbturk.net
- www.programlama.com
- www.yazilimgrubu.com
- www.findikkurdu.com
- www.yazilimuzmani.com
- www.godoro.com
- www.ceturk.com
- www.dotnetturk.com
- www.startvbdotnet.com
- www.vbasicmaster.com
- <http://bilisim-kulubu.com>
- <http://msdn.microsoft.com/netframework/>
- <http://www.blogcu.com/suattuncer/VB-NET>
- www.wikipedia.org
- www.freevbcode.com



KAYNAKÇA

- HAYDAR Yalçın, **Adım Adım Microsoft Visual Basic .NET**, Arkadaş Yayınları, Ankara, 2002.
- HALVORSON Michael, **Microsoft Visual Basic .NET Step By Step**, Microsoft Pres, A Divicion of Microsoft Corporation One Microsoft Way Redmond, 2002.
- <http://www.ceturk.com/default.asp>
- <http://www.msakademik.net/makaleler.aspx?grup=VBN>
- <http://www.mutasyon.net/dersoku.asp?id=86>
- <http://www.bmssoftware.net/programlama/vbnet/vbnet02.aspx>
- Mastering, Visual Basic .NET
Evongelos Petroustos, SYBEX, Inc., Alameda, CA, 2002