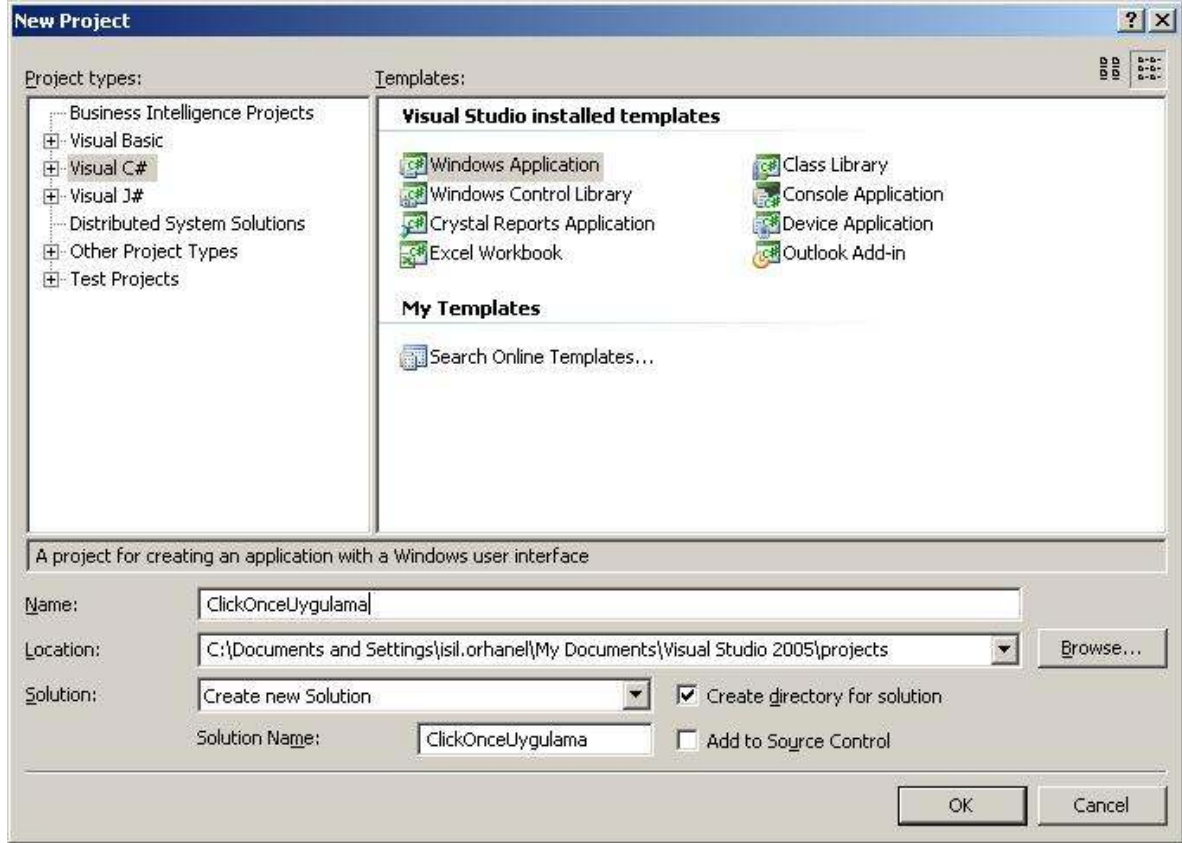


Geliştirdiğimiz Windows tabanlı uygulamaların dağıtımlarını ClickOnce teknolojisi kullanarak daha rahat ve sorunsuz bir şekilde gerçekleştirebiliriz. Bu makalemde ClickOnce teknolojisi hakkında bilgi vereceğim ve örnek bir ClickOnce uygulaması üzerinden ClickOnce teknolojisi ile yapılabilecekleri anlatacağım.

Windows uygulamalarında güncelleme yapılırken kullanıcı tüm uygulamayı kaldırıp yeniden kurmak durumunda kalır. Ayrıca eğer kullanıcı bilgisayarda admin değilse uygulama kurma haklarına sahip olmadığından uygulamayı kuramaz. Eğer uygulama birden fazla makineye kurulacaksa, exe dosyasının her makineye ayrı ayrı yüklenmesi gerekir. Kurulan veya kaldırılan bir uygulama çalışmakta olan diğer uygulamanın dll'lerini silerek veya değiştirerek uygulamanın çalışmamasına neden olabilir veya sonradan kurulan bir uygulama bizim geliştirdiğimiz uygulamaya biraz önce bahsettiğim şekilde bir etkide bulunabilir. Bunlar ve bunlar gibi karşılaşılması olası problemler yüzünden Windows uygulamaları yerine web uygulamaları geliştirmek gibi bir çözüme gidilebilir ancak ClickOnce daha pratik bir çözüm sağlamaktadır. ClickOnce Windows projelerinin esnekliğini web projelerinin kolay dağıtılabirliğiyle birleştiren bir teknolojidir. Çünkü biraz önce yazdığım Windows uygulaması geliştirmenin dezavantajlarını ClickOnce kullanarak uygulamanızı publish ettiğinizde yaşamazsınız. Geliştirilen uygulama ClickOnce ile bir web server'dan, ağ üzerinde paylaşılan bir yerden veya CD'den kullanıcılara dağıtılabir. Uygulama client'lara kurulabildiği gibi, makineye kurulum yapmaya gerek olmadan herhangi bir ağ yolundan da çalıştırılabilir. Böylece kullanıcının bilgisayarında cache'lenir ancak kurulumu yapılmaz. Uygulama güncellemelerinin olup olmadığı belirlenen zamanlarda veya uygulamanın her çalıştırıldığında kontrol edilebilir. Uygulamanın yüklü olması gereken en alt versiyon, yapılan güncelleştirmenin kullanıcı tarafından kurulmasının zorunlu olup olmadığı gibi seçenekler de ClickOnce teknolojisi sayesinde çok kolay bir şekilde ayarlanabilir.

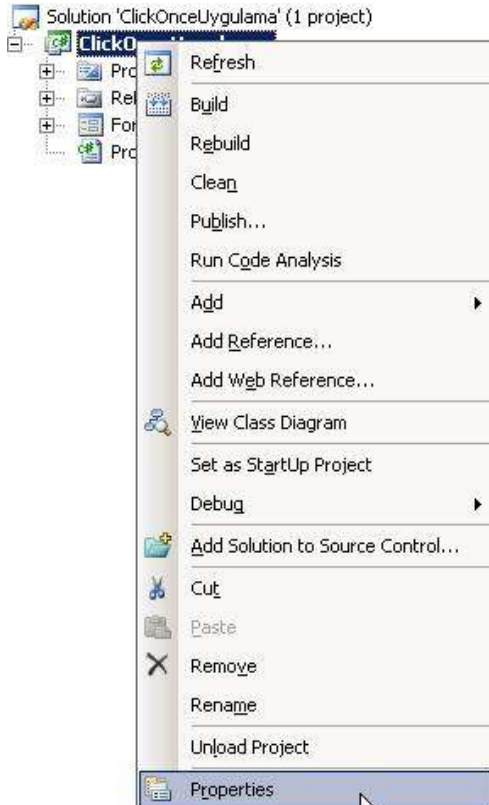
ClickOnce ile deploy edilen uygulamalarda iki adet XML dosyası bulunur. Bu dosyalar application manifest ve deployment manifest dosyalarıdır. Application manifest dosyası uygulamayı geliştiren kişi tarafından oluşturulur ve uygulama ile ilgili dosyaların, gerekli izinlerin, güncelleştirme ayarlarının tutulduğu dosyadır. Deployment manifest dosyası ise uygulamanın nasıl deploy edileceği bilgisinin, uygulamanın çalışacağı en düşük versiyon bilgisinin ve uygulamanın referans aldığı kütüphanelerin tanımlandığı dosyadır. Publish Wizard kullanılarak application ve deployment manifest dosyaları oluşturulabilir. Uygulama deploy edildikten sonra dosyalar belirlenmiş olan web server, CD veya paylaşımdaki bir yere kopyalanır. Application manifest ve uygulama dosyaları da deployment manifest dosyasının içinde belirtilen yerlere kopyalanırlar.

Şimdi örnek bir uygulama oluşturalım, yeni versiyonlarını oluşturalım ve her farklı versiyonu ClickOnce ile publish edelim.



İlk versiyona hiçbir şey eklemeyen uygulamamızı deploy edelim.

ClickOnce ile uygulamamızı ilk kez deploy etmek için oluşturmuş olduğumuz projemize sağ tıklayarak Properties penceresini açıyoruz:



Açılan ekranda sol tarafta bulunan sekmelerden Publish 'e basarak projemizin ClickOnce ayarlarının yapılacağı ekranı açıyoruz:

The screenshot shows the Visual Studio Publish Wizard dialog box. On the left, there is a sidebar with tabs: Application, Build, Build Events, Debug, Resources, Settings, Reference Paths, Signing, Security, Publish, and Code Analysis. The main area is divided into several sections:

- Configuration:** N/A (dropdown)
- Platform:** N/A (dropdown)
- Publish Location:**
 - Publishing Location (web site, ftp server, or file path):** http://localhost/ClickOnceUygulama/ (text box)
 - Installation URL (if different than above):** (empty text box)
- Install Mode and Settings:**
 - The application is available online only
 - The application is available offline as well (launchable from Start menu)
 - Buttons: Application Files..., Prerequisites..., Updates..., Options...
- Publish Version:**
 - Major: 1, Minor: 0, Build: 0, Revision: 0 (spinners)
 - Automatically increment revision with each publish
 - Buttons: Publish Wizard..., Publish Now

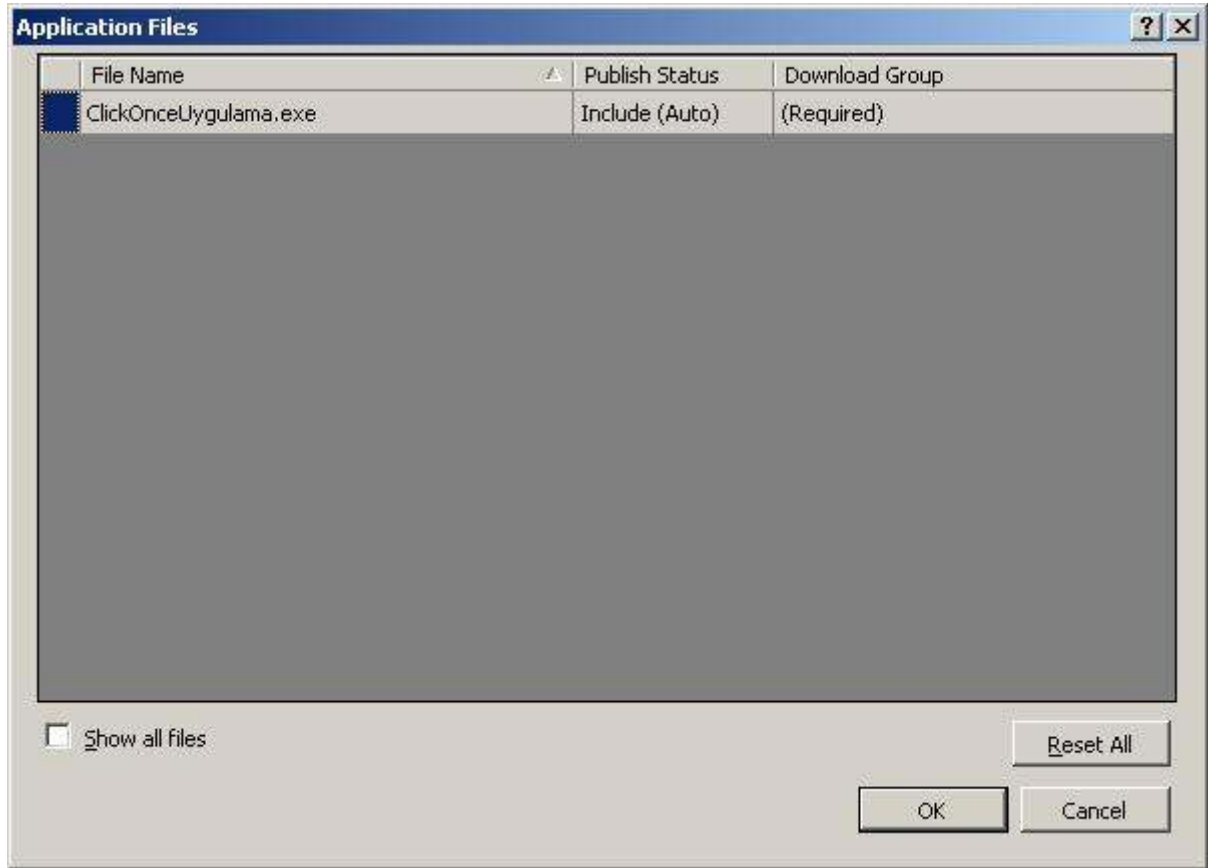
Bu ekrandaki Publish Location alanındaki Publishing Location uygulamanın son versiyonuna ait dosyaların ve uygulamanın son versiyonunun nereye kopyalanacağı bilgisinin, Installation URL ise uygulama kurulumunun online olarak indirilebileceği adres bilgisinin girildiği alanlardır.

Ekrandaki Install Mode and Settings alanında ise iki adet radiobutton bulunmaktadır. "The application is available online only" seçeneği seçilirse kullanıcı uygulamayı sadece yukarıda belirtilmiş olan Installation URL'e erişerek çalıştırabilir. Uygulama Start menüsüne ve Add/Remove Programs listesine eklenmez, sadece cache'lenerek çalıştırılır. Uygulama diğer bir seçenek olan "The application is available offline as well (launchable from Start menu)" seçilerek publish edilirse, kullanıcının uygulamayı çalıştırması için Installation URL'e erişmesine gerek kalmaz, uygulama Start menüsü altına ve Add / Remove Programs listesine eklenir ve Start menüsü üzerinden erişilerek çalıştırılabilir.

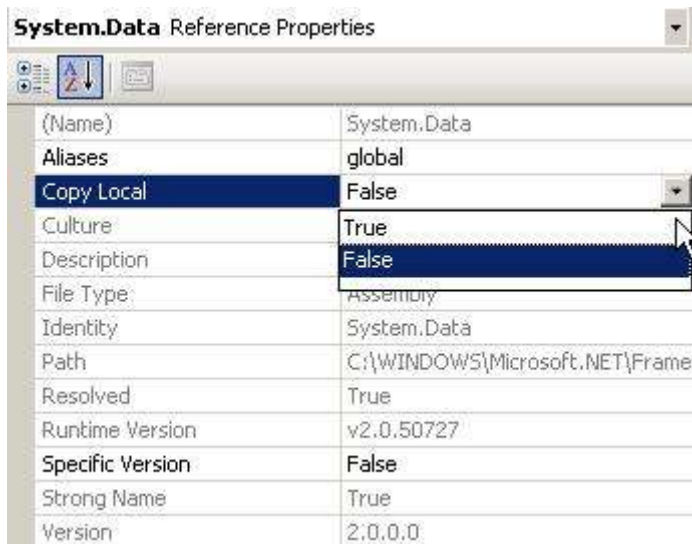
Bu alandaki dört adet buton ve ne işe yaradıkları aşağıda açıklanmıştır:

Application Files:

Application Files ekranında uygulamanın bileşenleri (dll, exe vs) listelenir ve kullanıcı bu bileşenlerin publish status ve download group ayarlarını yapar. Publish status, bileşenlerin uygulama publish edilirken uygulama ile beraber publish edileceğini (Include(Auto)) veya bu bileşenlerin uygulamanın çalışması için bir gereklilik olduğunu (Prerequisite) veya uygulama kurulumuna dahil edilmeyeceğini (Exclude) belirtir. Bu ekranda listelenen öğelerden uygulamanın .exe veya varsa .config dosyasının publish status'u ve download group'u değiştirilemez.

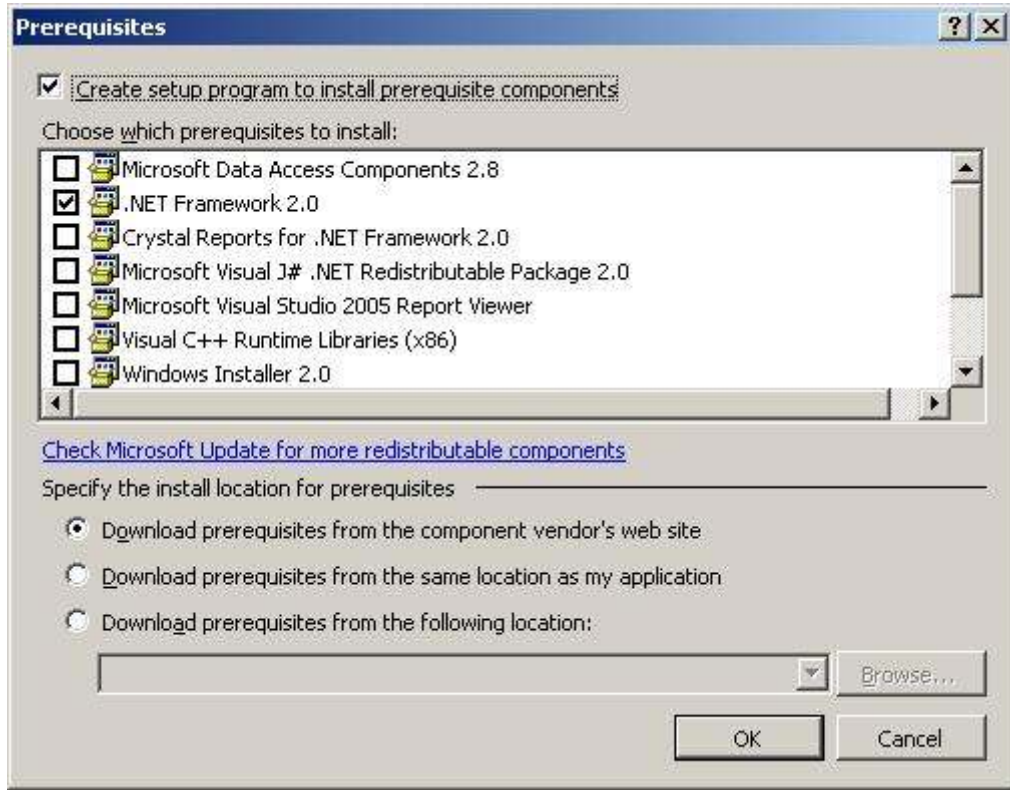


Ayrıca projemize eklemiş olduğumuz dll dosyalarının refereranslarının CopyLocal özellikleri True olarak ayarlanmışsa, dll (Include(Auto)) olarak işaretlenir ve kurulumda uygulamanın klasörüne kopyalanır. CopyLocal özelliği False ise dll'in kurulumdan önce GAC'e eklenmiş olması veya developer tarafından el ile Include durumuna getirilmiş olması gerekir:

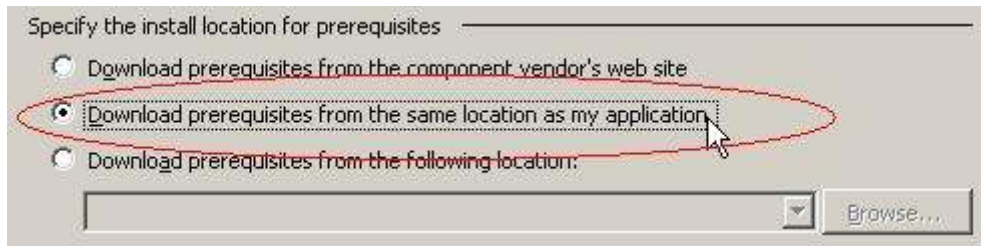


Prerequisites:

Uygulamamızın çalışması için gerekli olan araçları bu ekrandan işaretlediğimizde, uygulamamız kurulmadan önce seçili olan gerekli dosyalar makinada kurulu değilse öncelikle onların kurulumu tamamlanır ve sonrasında uygulama kurulumu başlar.

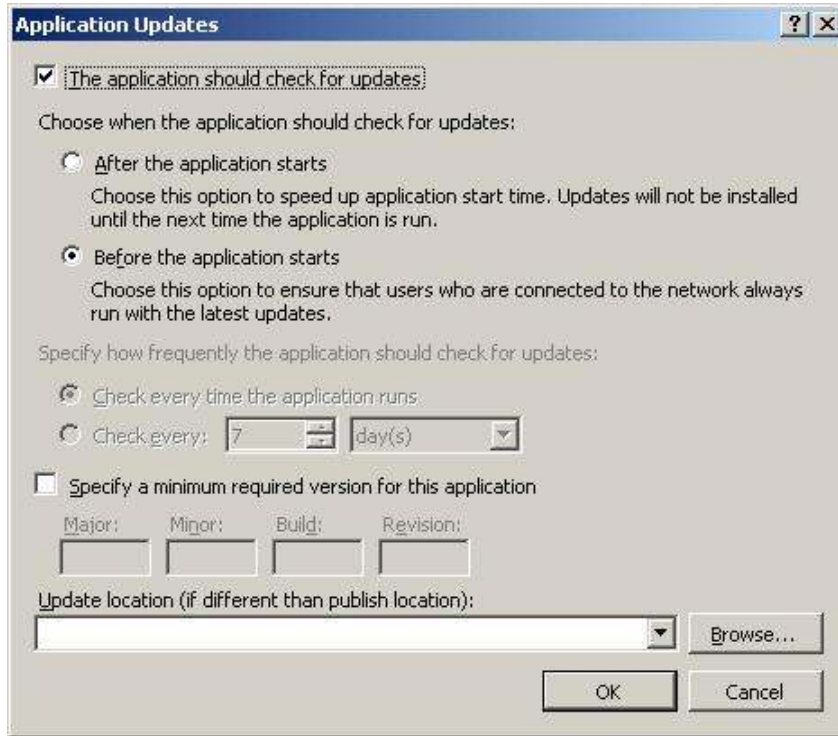


Ekrandaki listenin altında yer alan 3 radiobutton ise bu araçların nereden yükleneceğini seçtiğimiz alandır. Bu radiobutton'lardan bize uyanı seçerek uygulama için gerekli bu araçların, dosyaları yayınlayan firmanın web sitesinden, uygulamamızın publish edildiği lokasyondan veya bizim belirlediğimiz bir lokasyondan indirilmesini sağlayabiliriz. Ben örnek projemizde bu araçların uygulamamızın publish edildiği lokasyondan indirilmesi seçeneğini işaretledim:



Updates:

Updates ekranında uygulamamızın güncelleştirmelerinin olup olmadığının kontrol edilmesi veya edilmemesi, eğer ediliyorsa uygulama çalıştırıldıktan sonra mı önce mi kontrol edileceği, hangi sıklıkta güncelleştirmelerin kontrol edileceği, uygulamanın çalışacağı minimum versiyon ayarının yapılması ve publish edilen lokasyondan farklı olarak bir de update lokasyonu kaydedilmesi bu ekrandan yapılır.



Application Updates [?] [X]

The application should check for updates

Choose when the application should check for updates:

After the application starts
Choose this option to speed up application start time. Updates will not be installed until the next time the application is run.

Before the application starts
Choose this option to ensure that users who are connected to the network always run with the latest updates.

Specify how frequently the application should check for updates:

Check every time the application runs.

Check every:

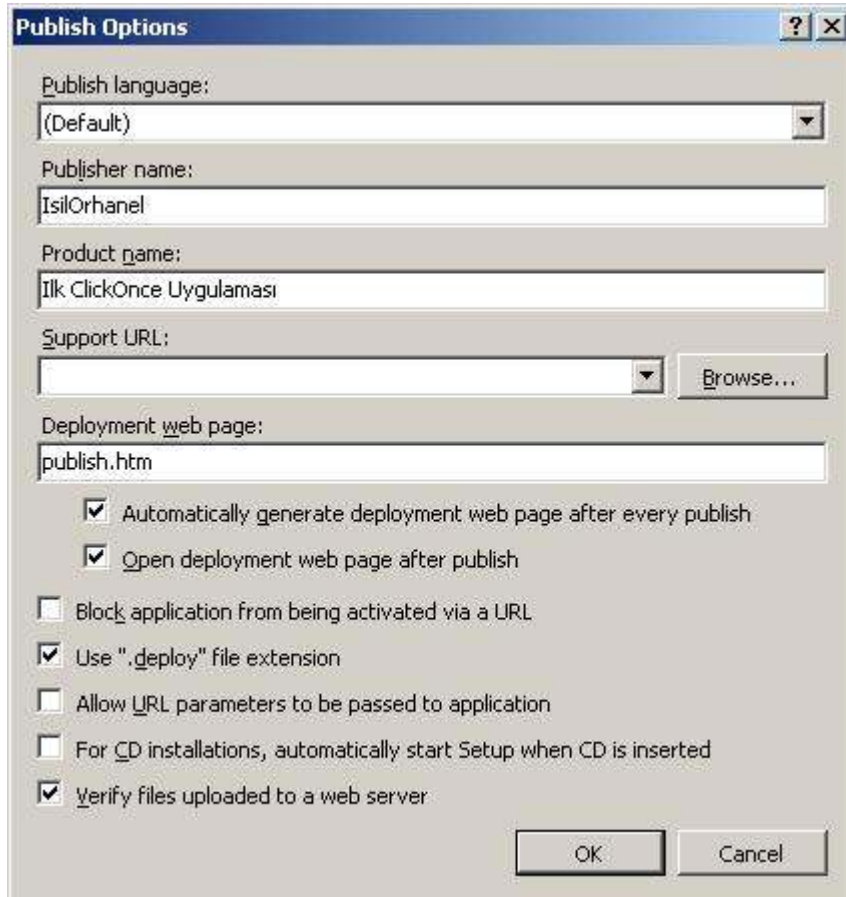
Specify a minimum required version for this application

Major: Minor: Build: Revision:

Update location (if different than publish location):

Options:

Bu ekranda çeşitli publish seçeneklerini ayarlayabiliriz. Ekrandaki Publisher name alanı program kurulduktan sonra Start - Programs altındaki menüde görülecek isimdir. Product name ise uygulamanın menüsünün altında görülecek olan uygulamanın kendisinin adıdır.



Publish Options [?] [X]

Publish language:

Publisher name:

Product name:

Support URL:

Deployment web page:

Automatically generate deployment web page after every publish

Open deployment web page after publish

Block application from being activated via a URL

Use ".deploy" file extension

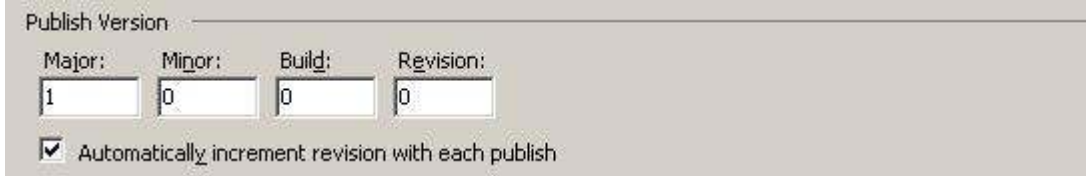
Allow URL parameters to be passed to application

For CD installations, automatically start Setup when CD is inserted

Verify files uploaded to a web server

Options ekranında ayarlanabilecek diğer önemli bir özellik de uygulamamızın client'lar tarafından kurulurken kurulacak olan dizindeki hangi sayfadan bu kurulumla ulaşılabileceğinin ayarlandığı yer olan Deployment web page özelliğidir. Deployment web page varsayılan olarak publish.htm olarak adlandırılmıştır, fakat bu ekrandan varolan başka bir sayfa ile değiştirilebilir.

Publish ekranındaki başka bir alan da Publish Version alanıdır. Burada yer alan checkbox'ı işaretlediğimizde her deployment'ta uygulamamızın versiyonu otomatik olarak 1 artar. İstersek textbox'lara kendimiz de versiyon numarasını yazarak deployment'tan önce versiyonu belirleyebiliriz.



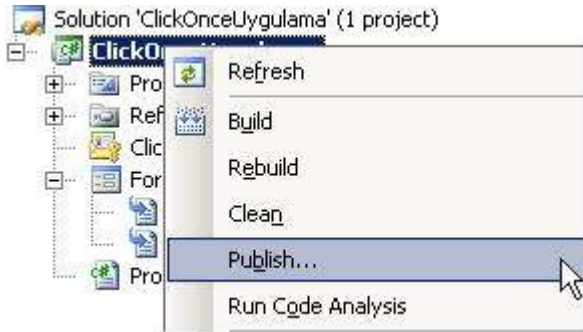
Publish Version: _____

Major:	Minor:	Build:	Revision:
1	0	0	0

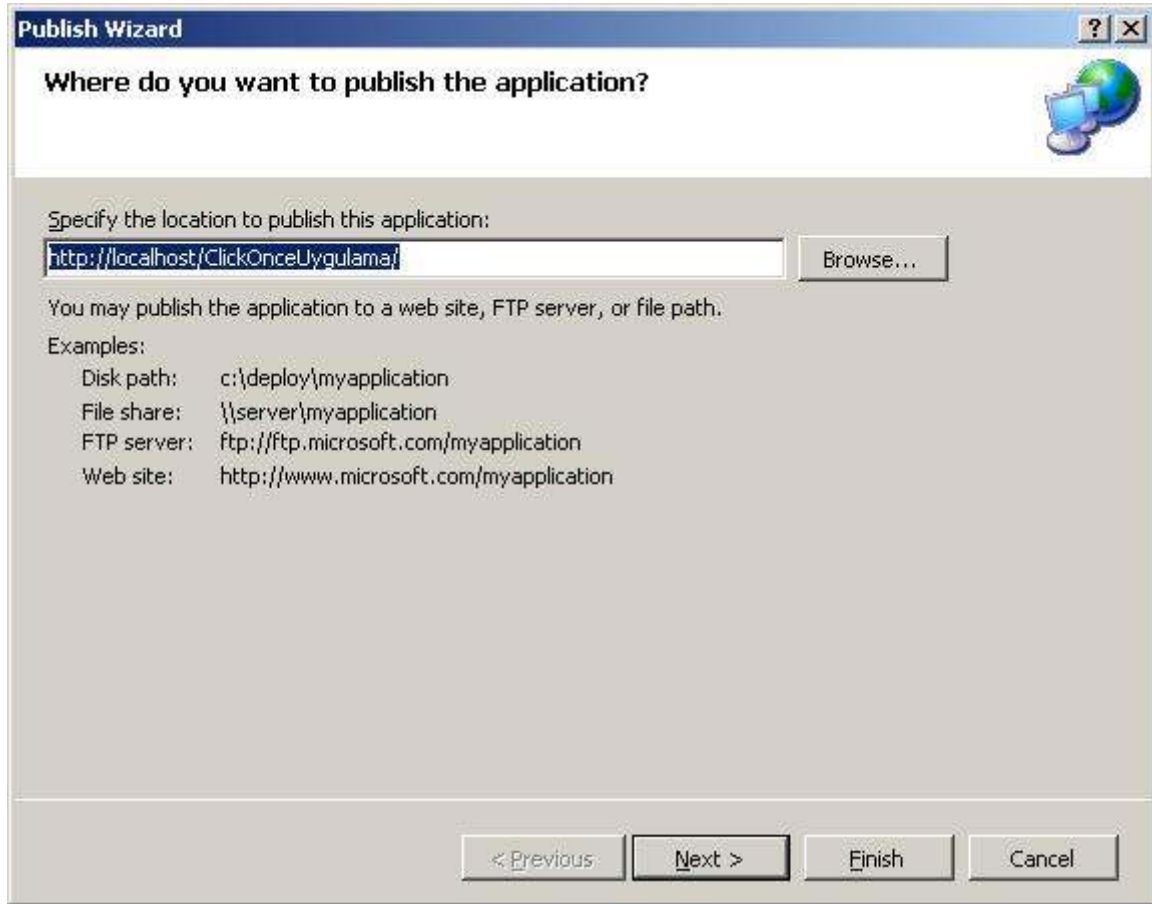
Automatically increment revision with each publish

Bütün bu ayarlardan sonra artık uygulamamız publish edilmeye hazır hale gelir ve ekrandaki Publish Now butonuna basarak uygulamamızı publish edebiliriz.

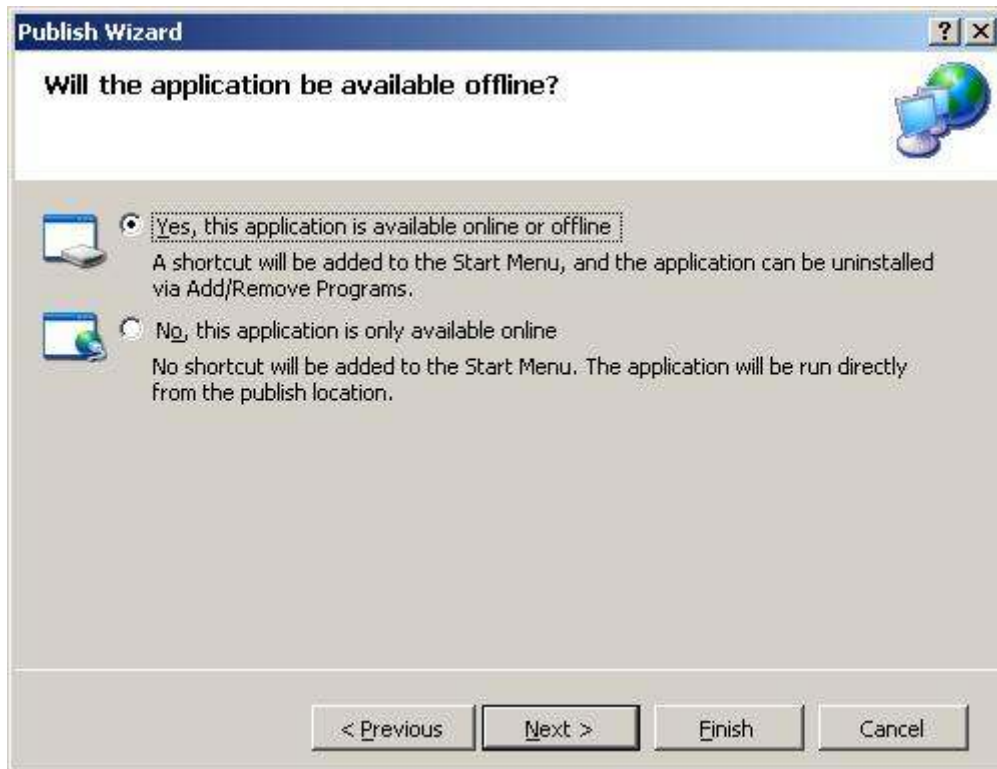
Şimdi size uygulamalarınızı publish edebileceğiniz alternatif bir yöntemden bahsedeceğim. Uygulamamızı Publish ekranı yerine projemize sağ tıkladığımızda açılan Publish... menü öğesi ile veya yine Publish ekranında yer alan Publish Wizard... butonunu kullanarak da publish edebiliriz:



Publish'e tıkladığımızda ilk olarak bize uygulamamızın dağıtımının yapılacağı web adresi, ftp adresi veya dosya yolunun girileceği ekran çıkar:



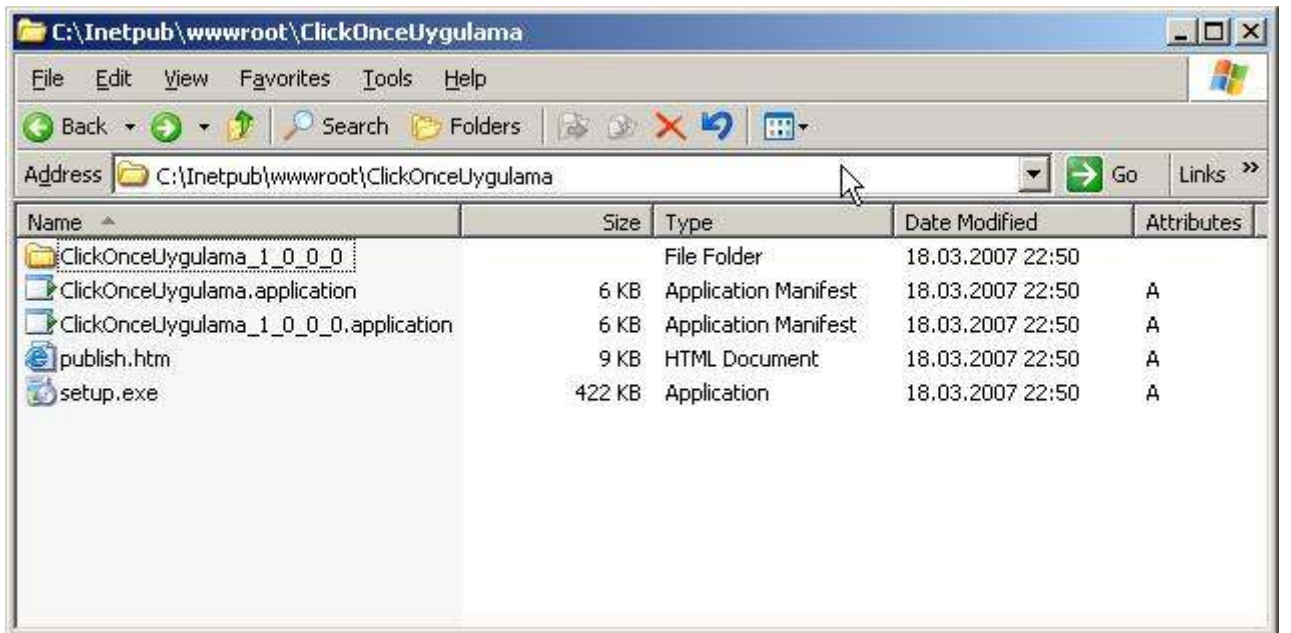
Next'e basıp ilerlediğimizde ise uygulamanın biraz önce de belirttiğim gibi belirli bir lokasyona erişilerek mi (online) veya client bilgisayarlara kurularak mı (offline) çalışacağı seçilir:



Son ekrana geldiğimizde ise artık uygulamamız publish edilmeye hazır hale gelir ve Finish butonuna tıkladığımızda publish edilir:



Publish işlemi başladığında ClickOnce Belirttiğimiz Publish Location altına uygulamanın adını taşıyan yeni bir dizin oluşturur ve gerekli dosyaları bu konuma upload eder. Oluşturulan dosyalar aşağıda görüldüğü gibidir:



.application uzantılı XML dosyalar uygulamanın ilgili versiyonunun indirileceği adresin bilgilerini tutar:

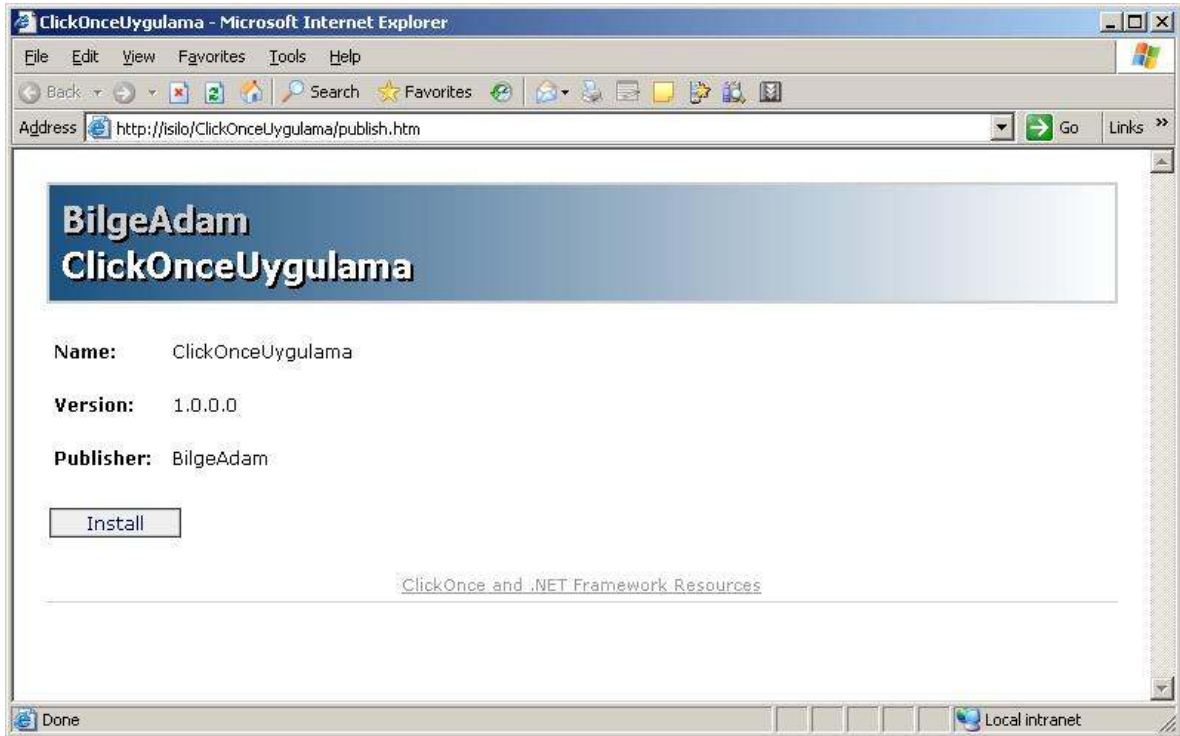
```
<description asmv2:publisher="BilgeAdam" asmv2:product="ClickOnceUygulama"
xmlns="urn:schemas-microsoft-com:asm.v1" />
<deployment install="true" mapFileExtensions="true">
  <subscription>
    <update>
      <beforeApplicationStartup />
    </update>
  </subscription>
  <deploymentProvider
codebase="http://isilo/ClickOnceUygulama/ClickOnceUygulama.application" />
</deployment>
```

publish.htm dosyası uygulamanın ve uygulamayı kurmadan önce sistemde kurulu olması gereken bileşenlerin kurulumlarına ait linklerin olduğu sayfadır.

setup.exe dosyası ise uygulamanın kurulum dosyasıdır.

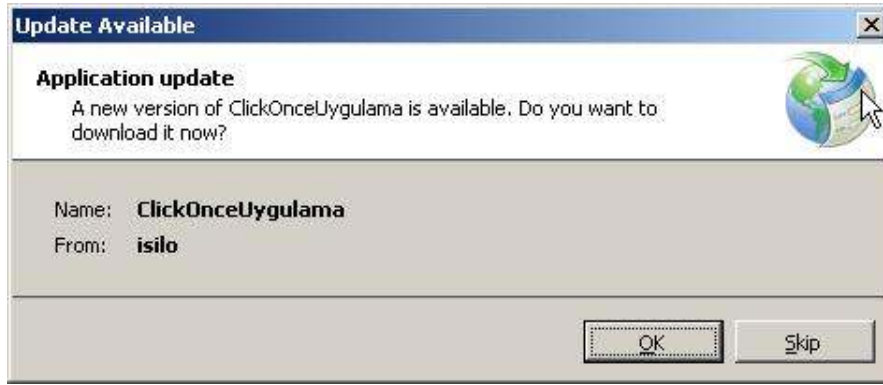
ClickOnceUygulama_1_0_0_0 klasöründe ve her versiyon artırımında versiyon numarasına göre isimlendirilip yeniden oluşacak olan bu klasörde ise uygulamanın ilgili versiyonunun dosyası ve application manifest dosyası bulunur.

Uygulama upload edildikten sonra oluşturulan publish.htm sayfası otomatik olarak açılır:

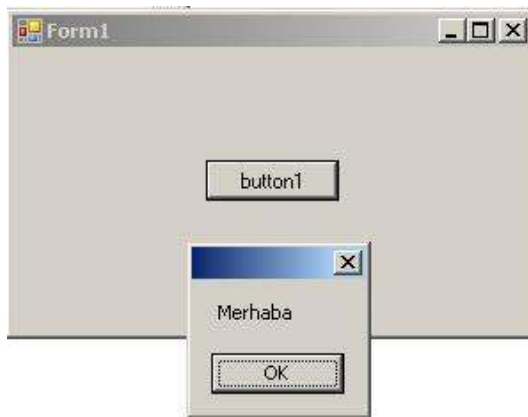


Sayfadaki Install butonuna basıldığında güvenlik uyarısı gelebilir. Install butonuna basıp devam ettiğimizde uygulama bilgisayarımıza kurulmuş olacaktır ve Start menüsü altından çalıştırılabilir.

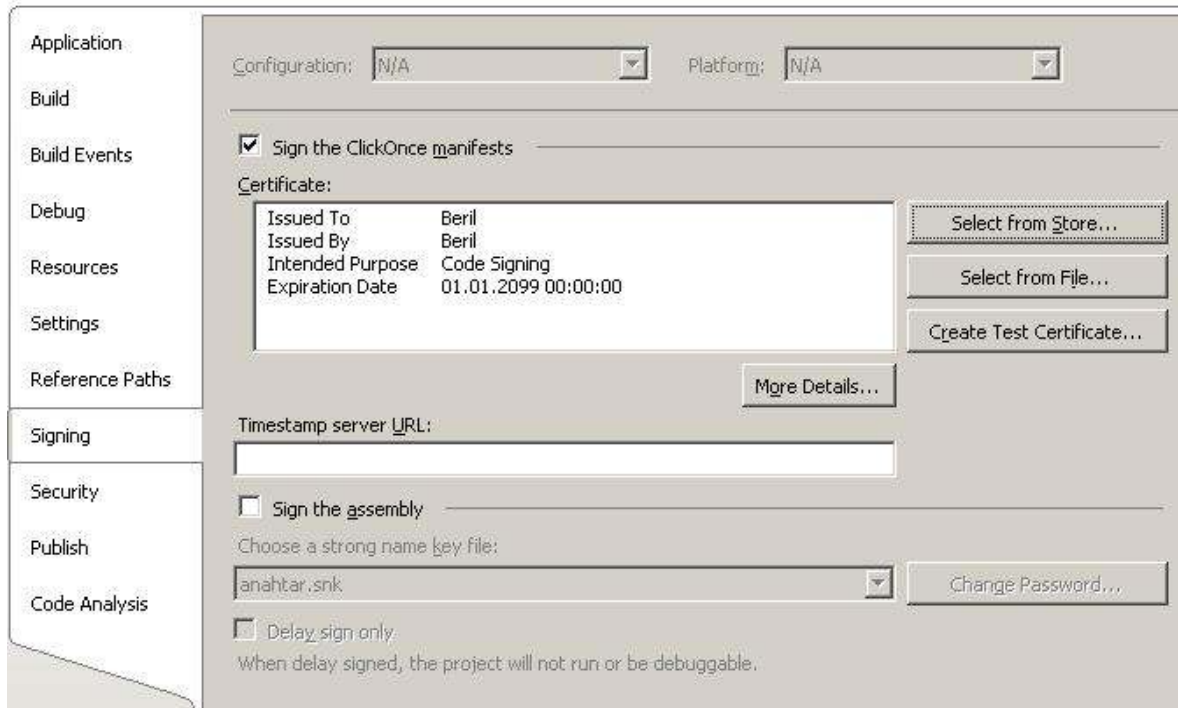
İkinci versiyonda ise uygulamamıza tıkladığında "Merhaba" şeklinde messagebox gösteren bir buton ekleyelim ve yeniden publish edelim. Uygulamamızı publish ettikten sonra ilk çalıştırmak istediğimizde aşağıdaki uyarı mesajı görüntülenir:



Bu ekran uygulamamızın yeni versiyonunun yüklenmesi için onay verilmesi gereken ekrandır. OK butonuna basıldığında uygulamanın güncelleştirilmesi yapılır ve en son versiyon client'a kurulur:

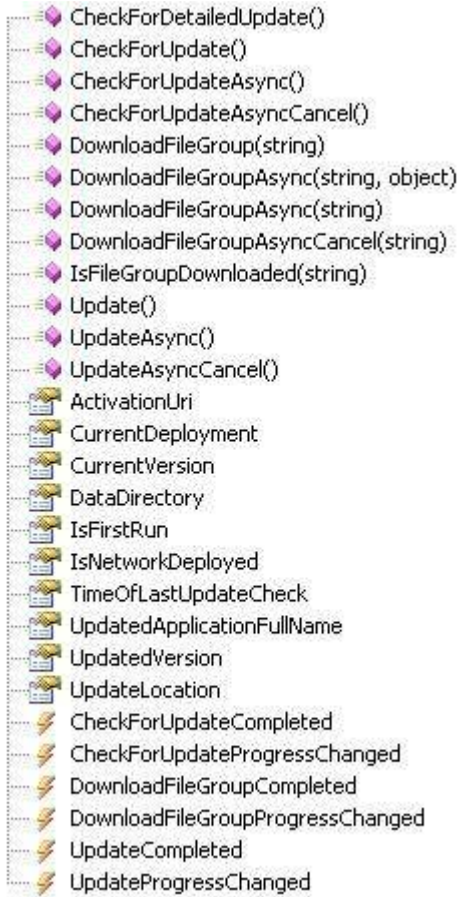


Proje özelliklerine tıkladığımızda açılan ekrandaki Signing sekmesi ile uygulamamızı imzalamamız gerekir. Bu ekranda uygulamanın manifest dosyalarını imzalamak zorunludur, eğer istenirse assembly de imzalanabilir:



Ekranın ilk kısmında yer alan Sign the ClickOnce manifests alanındaki üç buton yardımıyla uygulamamızı çeşitli şekillerde imzalayabiliriz. Select From Store butonu kullanıcının kayıtlı olan mevcut sertifikaları arasından bir tanesinin kullanılarak belirlenmiş bir süreye kadar (Expiration Date) uygulamanın imzalanmasını sağlar. Select From File mevcut bir .pfx dosyasını kullanarak uygulamanın imzalanmasını sağlar. Create Test Certificate butonu ise uygulamaya özel test amaçlı sertifika oluşturulmasını sağlar.

System.Deployment.Application namespace'indeki en önemli class ApplicationDeployment class'ıdır. Bu class yardımıyla uygulamanın update'lerinin kontrolü, uygulamanın kurulu versiyon bilgileri, uygulama ile indirilecek olan dosyaların kontrolü gibi işlemler yapılabilir. Aşağıda bu namespace'e ait önemli property'ler, metodlar ve görevleri listelenmiştir:



Property'ler	
ActivationUri :	Uygulamanın manifest dosyasının çağrıldığı URL'i verir.
CurrentDeployment:	Uygulamanın mevcut ApplicationDeployment özelliklerinin okunabileceği property'dir.
CurrentVersion:	Uygulamanın çalışan mevcut deployment versiyon bilgisinin alınabileceği property'dir.
DataDirectory :	ClickOnce uygulamasının data klasörünün yolunu verir.
IsFirstRun:	Uygulamanın yüklü versiyonunun bilgisayarda ilk defa çalıştırılıp çalıştırılmadığı bilgisinin okunabileceği property'dir.

IsNetworkDeployed:	Uygulamanın bir ClickOnce uygulaması olup olmadığı bilgisinin okunabileceği property'dir.
TimeOfLastUpdateCheck:	Uygulamanın en son ne zaman güncelleme kontrolünün yapıldığı bilgisinin okunabileceği property'dir.
UpdatedApplicationFullName	Güncelleme sonrası uygulamanın isminin okunabileceği property'dir.
UpdatedVersion:	Yüklü olan uygulama versiyon bilgisinin okunabileceği property'dir.
UpdateLocation:	Uygulamanın update edildiği lokasyon bilgisinin okunabileceği property'dir.
<u>Metod'lar</u>	
CheckForDetailedUpdate	CheckForUpdate ile aynı işlemi yapar ancak daha detaylı bilgi döndürür.
CheckForUpdate:	Yeni bir güncelleştirme olup olmadığını kontrol eder.
CheckForUpdateAsync	Yeni bir güncelleştirme olup olmadığını asenkron olarak kontrol eder.
CheckForUpdateAsyncCancel	Asenkron yapılan güncelleştirme kontrolünü iptal eder.
DownloadFileGroup	Belirli bir gruba ait dosyaları indiren metoddur.
DownloadFileGroupAsync	Belirli bir gruba ait dosyaları asenkron olarak indiren metoddur.
DownloadFileGroupAsyncCancel	Belirli bir gruba ait dosyaları asenkron indirirken iptal edilmesi için kullanılan metoddur.
IsFileGroupDownloaded	Belirtilen isimdeki FileGroup'un istemciye indirilip indirilmediğinin kontrol edilebileceği metoddur.
Update	Uygulamanın en son versiyonunun indirilmesi ve kurulması işlemi başlatır.
UpdateAsync	Uygulamanın en son versiyonunun indirilmesi ve kurulması işlemi asenkron olarak başlatır.
UpdateAsyncCancel	Asenkron olarak başlatılmış uygulamanın en son versiyonunun indirilmesi ve kurulması işlemi iptal eder.
<u>Event'lar</u>	
CheckForUpdateCompleted	CheckForUpdateAsync metodu tamamlandığında tetiklenir.
CheckForUpdateProgressChanged	CheckForUpdateAsync metodunda bir güncelleme bulunduğu tetiklenir.
DownloadFileGroupCompleted	FileGroup download'ı bittiğinde tetiklenir.
DownloadFileGroupProgressChanged	DownloadFileGroupAsync metodunda bir filegroup download işlemi bulunduğu tetiklenir.
UpdateCompleted	Uygulamanın güncelleştirmesi bittiğinde kullanılacak event.
UpdateprogressChanged	UpdateAsync metodunda bir güncelleme işlemi başladığında tetiklenir.

Şimdi örnek bir uygulamada System.Deployment.Application namespace'i ile yapılabileceklere göz atalım. Yeni bir Windows uygulaması oluşturup, formu aşağıdaki gibi

dizayn edelim:



Görüldüğü üzere ilk buton ile uygulamamızın publish edilmiş yeni bir versiyonunun olup olmadığını kontrol edeceğiz ve yeni versiyon varsa ClickOnce ile kurulumunu yapacağız. İkinci buton ile ise uygulamamızın yüklü versiyonu, güncelleme lokasyonu gibi bilgilerini label'lara yazdıracağız. Formumuza System.Deployment.Application namespace'ini import ettikten sonra öncelikle güncelleme olup olmadığı kontrolünün koduna bakalım:

```
private void btnGuncellemeVarMi_Click(object sender, EventArgs e)
{
    ApplicationDeployment ad;
    try
    {
        ad = ApplicationDeployment.CurrentDeployment;
        if (ad.CheckForUpdate()) //uygulamamızın güncellemesinin olup olmadığının kontrolünü yapıyoruz.
        {
            if (MessageBox.Show("Uygulamanın yeni versiyonu bulunmakta. Kurmak ister misiniz?", "Güncelleme Bulundu", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            {
                ad.Update(); //güncelleştirme indiriliyor ve kuruluyor.
                MessageBox.Show("Uygulama güncellemesi tamamlandı. Uygulama yeniden başlatılacak.");
                Application.Restart();
            }
        }
        else
        {
            MessageBox.Show("Güncelleştirme bulunamadı.");
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show("Uygulama güncelleştirirken hata oluştu." + ex.Message);
    }
}
```

Bu noktada karşılaştığım bir problemden bahsetmekte fayda var. Uygulamamın bu

buton ile update kontrolünü yapması ve kendisinin update için check etmemesi için publish ekranındaki Updates butonuna tıkladığımda açılan formda "The application should check for updates" checkbox'ındaki check'i kaldırdım. Sonrasında uygulamamı publish ettim ve web sayfasından yeni versiyonu yükledim. Çalıştırdığımda ise aşağıdaki hatayı aldım:

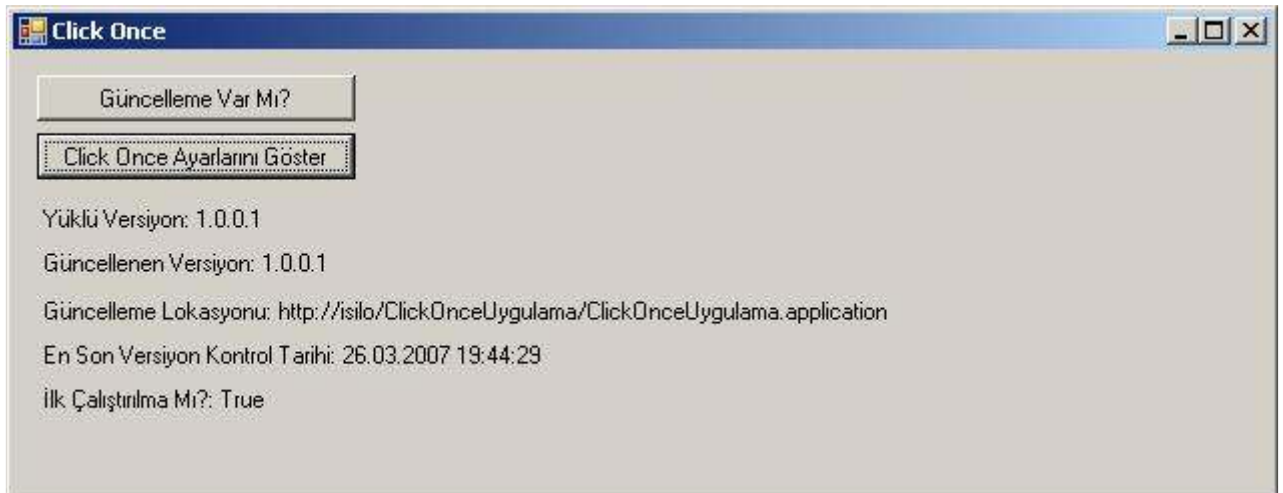
Application cannot be updated programmatically unless the deployment manifest includes the <deploymentProvider> element.

Bu hatanın oluşmasını engellemek için ise yapmanız gereken yine Updates ekranındaki Update Location alanına gidip bir lokasyon yazmanızdır. Bir önceki makalemde bu alana publish lokasyonundan farklı bir lokasyonda uygulamamızı update etmek istersek giriş yapabileceğimizi yazmıştım fakat bu sorunla karşılaşırsanız çözümü bu alana ister aynı publish lokasyon adresini ister başka bir adresi yazmak, uygulamanızı yeniden publish etmek ve web sayfasından kurmak olacaktır.

Şimdi sırada formdaki diğer buton yardımıyla uygulamamızın ClickOnce özelliklerini almaya geldi:

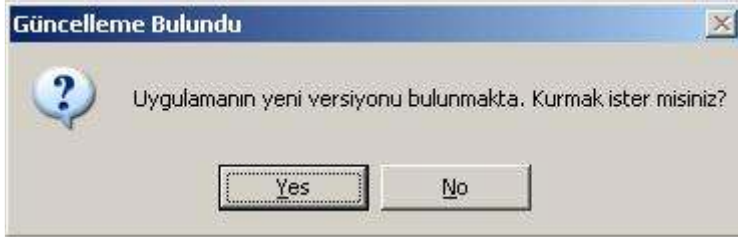
```
private void btnClickOnceAyarlari_Click(object sender, EventArgs e)
{
    ApplicationDeployment ad;
    ad = ApplicationDeployment.CurrentDeployment;
    lblYukluVersiyon.Text = "Yüklü Versiyon: " + ad.CurrentVersion.ToString();
    lblGuncellenenVersiyon.Text = "Güncellenen Versiyon: " +
ad.UpdatedVersion.ToString();
    lblUpdateLocation.Text = "Güncelleme Lokasyonu: " +
ad.UpdateLocation.ToString();
    lblEnSonVersiyonCheckTarihi.Text = "En Son Versiyon Kontrol Tarihi: " +
ad.TimeOfLastUpdateCheck.ToString();
    lblFirstRun.Text = "İlk Çalıştırılma Mı?: " + ad.IsFirstRun.ToString();
}
```

Artık uygulamamızı publish edip çalıştırabiliriz. "Güncelleme Var Mi?" butonuna bastığımızda yeni bir güncelleme olmadığına dair mesaj çıkacaktır. "ClickOnce Ayarlarını Göster" butonuna bastığımızda ise formdaki labellarda aşağıdaki gibi bilgiler görülecektir.



Uygulamamızı bir daha publish edip çalıştırdıktan sonra "Güncelleme Var Mı?" butonuna tıklarsak yeni bir versiyonunun mevcut olduğuna ve kurmak isteyip

istemediđimize dair bir messagebox ıkacaktır.



Onayladıktan sonra uygulamanın son hali indirilir ve kurulur.