

Windows Phone 7 Hileleri

İbrahim Ersoy

Expression
Blend

VISUAL
STUDIO 2010

Windows Phone
MARKETPLACE

ÖRNEK
UYGULAMA

Windows Phone 7 Hileleri

Ön Söz

Kısım-1: Windows Phone 7'de Uygulama Geliştirmeye Giriş

Bölüm-1: Microsoft'un Mobil Serüveni

Bölüm-2: WP7 ile neler yapabilirsiniz?

Bölüm-3: WP7 SDK ve Gerekli Araçlar

Bölüm-4: WP7 Aygıtını Etkinleştirme

Kısım-2: Windows Phone 7'de Uygulama Geliştirme

Bölüm-1: Silverlight ve Expression Blend

Bölüm-2: Kullanıcı Arabirimi Geliştirme

Bölüm-3: İmaj Tabanlı Yazı Efektleri

Bölüm-4: ScrollViewer Kullanımı

Bölüm-5: Farklı Panorama Efektleri

Bölüm-6: Kopyala-Yapıştır Özelliği

Bölüm-7: CapDetect Aracının Kullanımı

Bölüm-8: Sanal Klavye'yi Etkisizleştirme

Bölüm-9: Navigasyon

Bölüm-10: Görevler ve Seçiciler

Bölüm-11: Aygıt Bilgilerini Alma

Bölüm-12: Örnek Uygulama – Mesaj Servisi

Bölüm-13: Uygulamanızın Marketplace için Hazırlanması

İletişim

Kaynaklar

Önsöz

Bu kitabın hedef kitlesi, Mobil geliřtirmeye yeni bařlayan ancak Silverlight ile biraz uğrařmıř geliřtiricilerdir.

Bu mini e-kitabı yazmamdaki amaç bu konuda ücretsiz Türkçe kaynak sıkıntısının olmasından kaynaklanmaktadır. Öyle ki, kısa zaman önce bařladığım (1 Ağustos) bu kitaba paralel olarak olabildiğince çok ve yararlı, işin püf noktalarını, hilelerini içeren bilgileri yerleřtirmeyi hedefledim. Umarım başarılı olabilmifimdir bu konuda.

Umarım bu mini e-kitap sizi Windows Phone 7 uygulama geliřtirmeye teřvik edecektir. Neler yapabileceğinizi, hangi araçları kullanacağınızı, işin püf noktalarını ve uygulamanızı Marketplace e hazırlama gibi çok mühim konulara değindim.

Bu kitap ilerleyen dönemlerde güncellenecektir(*)

(*) Daha fazla yararlı içerik ekleyerek bir başucu kaynağı olması için gayret göstereceğim

Kısım-1: Windows Phone 7'de Uygulama Geliştirmeye Giriş

Bölüm-1:Microsoft'un Mobil Serüveni

Microsoft'un mobil cihazlar ile ilgili ilk macerası 1996 senesinde Windows CE 1.0 ı piyasaya sürmesiyle başladı.ARGE çalışmalarına 1990 senesinde başlanan Windows CE bundan 2 sene kadar sonra geliştirilmeye başlanmış ve 1996 senesinde nihayet piyasaya sürülmüştür.Tabi o dönem Microsoft Windows 95 i piyasaya süreceğinden Windows CE'de İşletim Sistemi olarak Windows 95 mimarisi kullanılırken Kullanıcı Arabirimi olarak WinPad adını verdikleri mimariyi kullandı.

Ardından 2000 yılında Microsoft, Pocket PC 2000 i çıkardı.Windows CE 3.0 tabanlı bu sistem Palm tarzı cihazlara yönelikti. Pocket PC 2000 Windows 98,Windows ME ve Windows 2000 İşletim Sistemlerindeki gibi bir görünümü vardı. Lakin Microsoft'un "Akıllı Telefon" diye tabir edilen bir ürünü henüz yoktu.

Ve 2001 senesine geldiğimizde Microsoft,Pocket PC 2002 i çıkarmış ve bu sistem ile beraber Microsoft un "Akıllı Telefonlara" uygulanabilen ilk modeli olma özelliğine sahip oldular.

2003 senesine geldiğimizde karşımıza artık Akıllı Telefonlara odaklanmış bir Microsoft çıkıverdi ki aynı sene Windows Mobile 2003 ürünlerini piyasaya sürdüler.Bu sistem Windows Mobile ailesinin ilk ferdi olup Windows CE 4.2 altyapısını kullanmaktadır.

Ve sene 2004 ü gösterdiğinde Microsoft,Windows Mobile 2003 SE'yi çıkarttı.Bu versiyon'da bir öncekine nazaran iyileştirmeler söz konusudur:Orientation dönüşümleri(Portrait-Landscape),yeni VGA boyutları ve WPA(Wi-Fi Protected Access) eklendi.

2005 senesine girdiğimizde Microsoft çitayı biraz daha yükselterek

Windows Mobile 5 I piyasaya sürdü.Windows Mobile 5 ile gelen yenilikler arasında .NET Mimarisi'nin bir parçası olan Compact Framework,Microsoft Office Mobile,DirectPush teknolojisi eklenmesi oldu.

2007 senesi Windows Mobile ailesinin 6.versiyonu olan Windows Mobile 6 telefonların piyasaya sürülmesi ile geçti.Windows Mobile 6 da dikkat çeken özellikler arasında; Windows CE 5.0 in kullanılması,Windows Live,Exchange 2007 ürünlerinin dahil edilmesi,.NET Compact Framework 2.0'ın bütünleşik olarak dahil edilerek yazılım geliştirilebilmesinin önünün açılması, SQL Server 2005 Compact Edition kullanarak Database uyumlu projelerin geliştirilmesini sağlamak ve VOIP desteği özelliklerin başını çekiyor.

Derken 2008'de Microsoft,Windows Mobile 6.1'i çıkartır.Lakin bu WM6 ya küçük güncellemeler içermektedir.Nedir bunlar;Microsoft OneNote 'un Mobil versiyonu ve "Domain Enroll" diye tabir edilen System Center Mobile Device Manager 2008 ürününün kullanılması sağlayarak Mobil cihazlarınızı yönetmenizi sağlayan sistem.

2009'a geldiğimizde Windows Mobile 6.5 ürünü piyasaya çıkar ama Microsoft un hiç bir zaman yol haritasında bulunmayan bir üründür.6.5 in çıkmasındaki amaç müşterileri "Windows Phone 7" diye tabir edilen yeni nesil Akıllı telefonları piyasaya çıkana kadar diğer Mobile İşletim Sistemi geliştiren firmalara kaptırmamak için ortaya çıkmış bir üründür.Ve bu ürünün tanıtımını da şu anda Microsoft CEO'su olan Steve Ballmer bizzat yapmıştır.

Windows Mobile 6.5 ile birlikte gelen yenilikler arasında:"Titanium" adını verdikleri Microsoft'un Zune ürününe benzer menu geçişleri,yepyeni bir GUI ile Hoşgeldin ekranı,"SkyBox" olarak lanse edilmiş şu anda kullandığımız SkyDrive benzeri bir Cloud sistemi,6.1 telefonların 6.5 e

yükseltme seçeneđi,Multitouch desteđi ve Office Mobile 2010 göze çarpan yenilikler arasındadır.

Ve 2010'da Microsoft Mobil İşletim Sisteminde gerek İşletim Sistemi seviyesinde gerekse de UI seviyesinde tasarım deđişikliğine giderek "Metro" adını verdiđi yeni bir UI kullanan "ZUNE" den sahip olduđu tecrübelerini de kullanarak Windows Phone 7 yi piyasaya sürdü.

Windows Phone 7 ile beraber Microsoft Mobil yazılım geliştirme stratejisini de deđiştirdi diyebiliriz.Uygulama Geliştiriciler için Silverlight kullanmaları istenirken,Oyun geliştirenler için XNA platformunu kullanmamız gerekmektedir.

Microsoft bir adım daha ileri giderek Windows Phone 7 için ilk update i olan Windows Phone 7.5 Mango yu çıkardı.

Mango kod adıyla tabir edilen Windows Phone 7.5 de öne çıkan güncellemeler arasında;Multitasking,Kullanıcı Arabirimi,Sosyal Ağ Entegrasyonu,Groups,Calendar,Camera ve daha adını sayamayacađız 500'ün üzerinde güncelleme bulunmaktadır.

Microsoft'un bize sunacađı en son mobil işletim sistemi olan Windows Phone 8 de geçtiğimiz günlerde tanıtıldı.Çok yeni bir teknoloji olduğundan ve daha SDK sı görücüye çıkmadığından ötürü kitabımızda Windows Phone 8 konusuna yer verilmemiştir.

Bölüm-2: WP7 ile neler yapabilirsiniz?

Windows Phone 7 SDK sını kullanarak uygulamalar "sandbox" olduğundan ötürü sistem tabanlı uygulamalar geliştiremezsiniz.

Silverlight kullanarak mobil uygulamalar geliştirebilirken,XNA kullanarak oyunlar geliştirebilirsiniz.

Önerim şudur ki 2012-2013 yılına damgasını vuracak Mobil Uygulamalar öngörüsü listeme bir göz atarak geliştirebileceğinizi düşündüğünüz bir fikri uygulayınız

En iyi 15 Mobil Uygulama Fikri:

- Mobil Bulut Servisleri
- Konum Tabanlı Servisler
- Mobil Arama
- Augmented Reality
- Mobil Sağlık Takibi
- Mobil Ödeme
- NFC Servisleri
- Mobil Reklam
- Mobil Anında Mesajlaşma
- Mobil Müzik-Video
- Sosyal Ağlar
- Mobil Alışveriş-Ticaret
- İçeriğe-Göre Servisler
- Nesne Tanıma
- Mobil E-Posta

Bölüm-3: WP7 SDK ve Gerekli Araçlar

WP7.1 SDK sını buradan indirerek projelerinizi geliştirebilirsiniz:

<http://www.microsoft.com/en-us/download/details.aspx?id=27570>

WP Icon Maker adlı uygulama size Marketplace için gerekli imaj dosyalarını oluşturmaktadır.

<http://wpiconmaker.codeplex.com/>

WP7 Emulator görünümünü beğenmediniz mi? Benim geliştirdiğim Desenleri kullanabilirsiniz:

<http://iersoy.com/post/2012/05/12/Windows-Phone-7-Emulator-Skins.aspx>

Bölüm-4: WP7 Aygıtını Etkinleştirme

Bu işlemi yapmak için Dev Center a girip ordan My Devices kısmında bulunan "Register a new Device" bölümünden aygıtınızı test amacıyla kullanmak üzere etkinleştirebilirsiniz.

Kısım-2: Windows Phone 7'de Uygulama Geliştirme

Bölüm-1: Silverlight ve Expression Blend

Bu bölümde Silverlight'ı baştan sona öğretmeye çalışmayacağım, çünkü zaten Windows Phone 'da bulunan Silverlight, Silverlight 4 ün bir benzeri olduğundan ötürü Silverlight 4 konusu ile ilgili kitapların, makalelerin okunması size Windows Phone 7 üzerinde uygulama geliştirirken kolaylık sağlayacaktır. Şu anda okuduğunuz kitabın olabildiğince az sayfada ve çok içerik-bilgi barındırmasına özen gösterdiğimden ötürü Silverlight konusunu dahil etmem e-kitabın sayfa adedini yükseltecektir. Bu yüzden Silverlight konusuna kitabımda değinmedim.

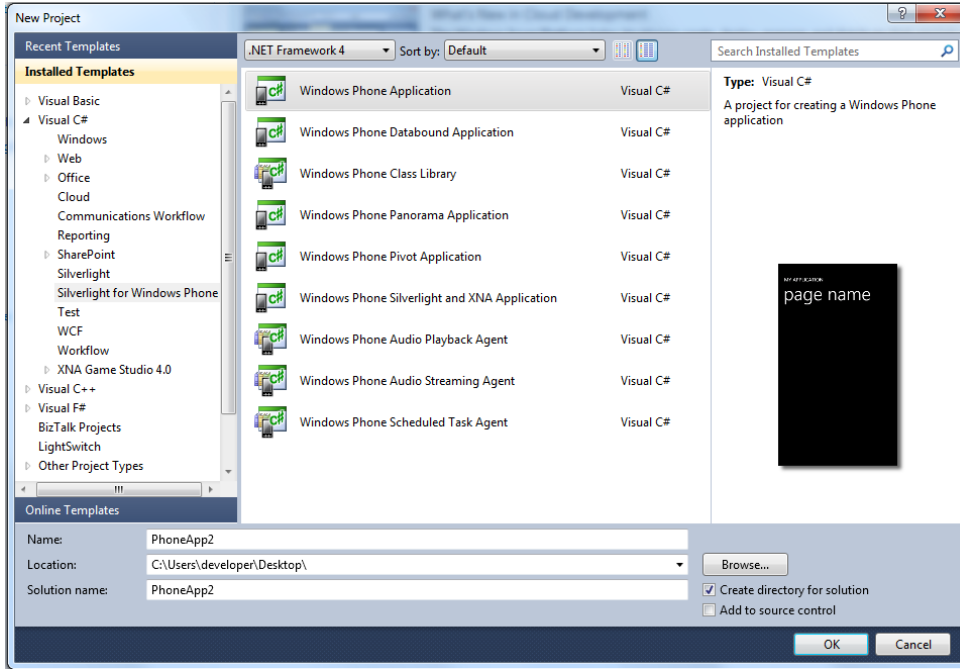
Silverlight, Microsoft'un zengin içerikli uygulamalar geliştirebilmemiz için biz yazılım geliştiricilere sunduğu bir teknolojidir.

Gerek Windows gerek Web gerekse de Mobil uygulamalar geliştirmemize olanak sağlar.

Expression Blend, Expression Studio ailesinin en kuvvetli bireylerinden biri olup XAML tabanlı animasyonlar, efektler oluşturmanıza yarayan bir üründür.

Bu bölümde sizlere Silverlight'ın ve Expression Blend ürünlerinin Mobil kısımda nasıl kullanılacağından bahsedeceğim.

Eğer ki en son WP7.1 SDK 'yı kurduysanız aşağıdaki gibi proje türlerini göreceksiniz:



Bu proje türlerini açıklayalım:

Windows Phone Application:

Standart bir Windows Phone Uygulaması geliştirebileceğimiz proje türü

Windows Phone Databound Application:

Liste ve Navigasyon fonksiyonlarını kullanan veri odaklı uygulamalar geliştirebileceğimiz proje türü

Windows Phone Class Library:

Farklı Windows Phone uygulamalarında da kullanabileceğiniz kod kütüphaneniz.

Windows Phone Panorama Application:

Panorama Uygulaması birkaç bölüme ayrılmış, 2 veya daha fazla sayfalardan oluşan, dokunarak geçiş sağlanan, Windows Phone Uygulaması diyince akla ilk gelen uygulama türüdür. Panorama Uygulamalarında sayfalar Panoramaltem olarak lanse edilir.

Windows Phone Pivot Application:

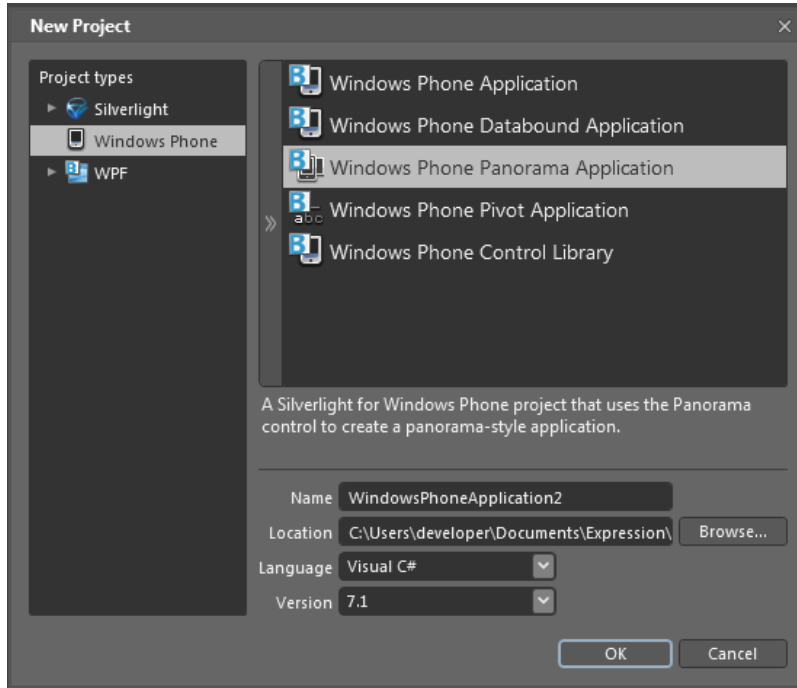
Pivot projeleri Panorama projelerine çok benzer.Tek farkla: Panorama Uygulaması ile birlikte aralarındaki fark Panorama proje tipinin tasarımsal yönü kuvvetli uygulamalarda kullanılması gerektiğidir.Nitekim "Hub" tarzı yapılar.Üstelik Panorama Background' una uygulanan efekt Pivot ta yoktur.

Windows Phone Silverlight and XNA Application:

Her zaman entegrasyonun olmasını istediğim proje tipi.Silverlight ve XNA oyunlarınızı tek bir yapıda birleştirebileceğiniz proje türüdür.Oyunlarınıza Silverlight ile Başlangıç Menüsü tasarımı yapabilirsiniz.En güzel özelliği de budur






















Diğer proje tipleri pek kullanılmadığından değinmeyeceğim.

Gelin şimdi de Blend 4 uygulamasında neler yapabiliyoruz bakalım:



Blend 4 ile XAML tabanlı uygulamalar geliştirilebileceğinden Silverlight proje tipleri görüntülenmektedir.Ve Visual Studio 2010 ile aynı tarz projeler oluşturabilir,tasarım işlemlerinizi bittikten sonra Visual Studio 2010 da bu projenizi açabilirsiniz.Tamamen bir entegrasyon söz konusu.Microsoft bu konuda çok başarılı bir geliştirme ortamı sunuyor biz yazılım geliştiricilere.

Gelin hep beraber ilk önce Windows Phone ortamında bulunan kontrolleri inceleyelim.

-  AdControl
-  Border
-  Button
-  Canvas
-  CheckBox
-  Ellipse
-  HyperlinkButton
-  Grid
-  Image
-  ListBox
-  Map
-  MediaElement
-  PasswordBox
-  RadioButton
-  Rectangle
-  ScrollView
-  Slider
-  StackPanel
-  TextBlock
-  TextBox
-  WebBrowser

AdControl:

Uygulamalarınızın içerisine reklam koymak isterseniz bu kontrolü kullanmanız gerekmektedir.Bu işlem için öncelikle bir "Ad id" almanız gerekir: <http://advertising.microsoft.com/mobile-apps>

Border:

Uygulamanızda Panel tarzı alanlar çizmenize olanak sağlar.

Button:

Klasik düğme kontrolü.Bas-İşlem Yap.

Canvas:

Panel tarzı çalışan,içerisinde bulunan öğelerin pozisyonlarını değiştirebileceğiniz lakin ekrana göre boyutlandırılabilir olmadığından ötürü tavsiye edilmeyen bir Panel kontrolüdür

CheckBox:

Çok seçenekli onay işlemleri için kullanılır.

Ellipse:

Yuvarlak,Elips şekiller çizmeye yarar.

HyperLinkButton:

Web sayfalarındaki linkler ile aynıdır.

Grid:

Sütun ve satır olarak bölünebilen,boyutlandırma sorunu yaşamayan en fazla kullanılan Panel kontrolüdür.

Image:

Bir resim dosyası eklemek isterseniz bu kontrolü kullanırsınız.

ListBox:

Birden fazla elemana sahip yapıların yani listelerin görüntülediği kontroldür.

Map:

Uygulamanıza BingMaps eklemeye yarar.

MediaElement:

Uygulamanızda medya dosyalarını çalıştırmanıza olanak sağlar.

PasswordBox:

Şifre girişi yapabileceğiniz özel bir veri giriş kontrolüdür.

RadioButton:

Birden fazla seçenek arasından sadece birini seçebileceğiniz bir kontroldür.

Rectangle:

Dikdörtgen ve Kare şekillerini çizebileceğiniz bir kontroldür.

ScrollView:

Belirli bir alan içerisinde bulunan ama görüntülenemeyen bilgilerin aşağı-yukarı hareket ettirerek görebilmenizi sağlayan bir kontroldür.

Slider:

Bu control belirli aralıklarda bulunan verileri seçmenize olanak sağlar. Bir örnek vermek gerekirse Windows da bulunan Ses uygulamasında sesin değerini belirlediğimiz o yapıya Slider denir.

StackPanel:

Panelimsi tarzda çalışan diđer bir control StackPanel'dir.Bu control içerisine dahil edilen nesnelere belirli istikamette alt alta veya yan yana dizebilir.

TextBlock:

Standart Label kontrolüdür.

TextBox:

Veri giriři yapabileceđimiz bir kontroldür

WebBrowser:

Uygulama içerisinden bir web sitesi görüntölemek istersek bunu kullanırız.

Bölüm-2: Kullanıcı Arabirimi Geliştirme

Kullanıcı Arabirimi Geliştirme konusu çok detaylı olduğundan ötürü elimden geldiğince basit anlatmaya çalışacağım.

Uygulamamızı etkileyici kılan etkenlerden birisi de tasarımıdır.Windows Phone üzerinde uygulama tasarımı yaparken iki methoddan faydalanırız.

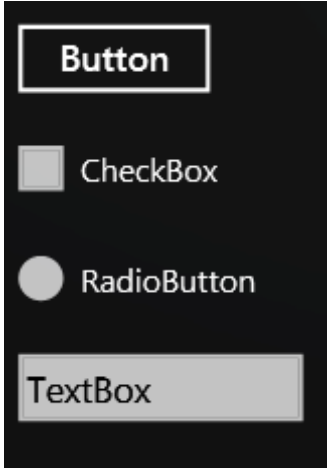
-**Klasik Yöntem**(İmaj Tabanlı)

-**XAML Yöntemi**(Renk Tabanlı)

Bu bölümde 4 adet kullanıcı arabirimi kontrolü tasarlayacağız:

- Button
- CheckBox
- RadioButton
- TextBox

Varsayılan tasarımlarına bakalım:



Bu Windows Phone 7 kontrollerinin varsayılan görüntüsüdür.Stiller ile oynayıp daha çekici bir tasarım yapmak isteyeceksinizdir.

Klasik Yöntem ile tasarımlara başlayalım:

Klasik Yöntem

Eğer Web Geliştiriciliği ile ilgilendiyseniz,CSS ile uygulamanızda kullanabileceğiniz stiller geliştirebildiğinizi bilirsiniz.Bildiğiniz üzere CSS kullanırken, kullanıcı arayüzüne kendi stilinizi giydirebilirsiniz.Bu stiller yine renk ve imaj tabanlı olabilir.

Web Geliştiricilerin sık kullandığı bu yöntemden yola çıkarak bu yönetime neden Klasik Yöntem dediğimi anlamışsınızdır.

Klasik Yöntem'de kontrolün cevap verdiği olaylara yönelik resim dosyalarına sahip olmamız gerekir. Örneğin bir Button için en azından 3 imaj dosyasına sahip olmanız gerekir:

- Button_Down (Button a basıldığında)
- Button_Over (Normal hali)
- Button_Disabled (Enabled özelliği "False" olarak ayarlandığında)

Button Stiline güzel bir örnek:

Button Down:



Button Over:

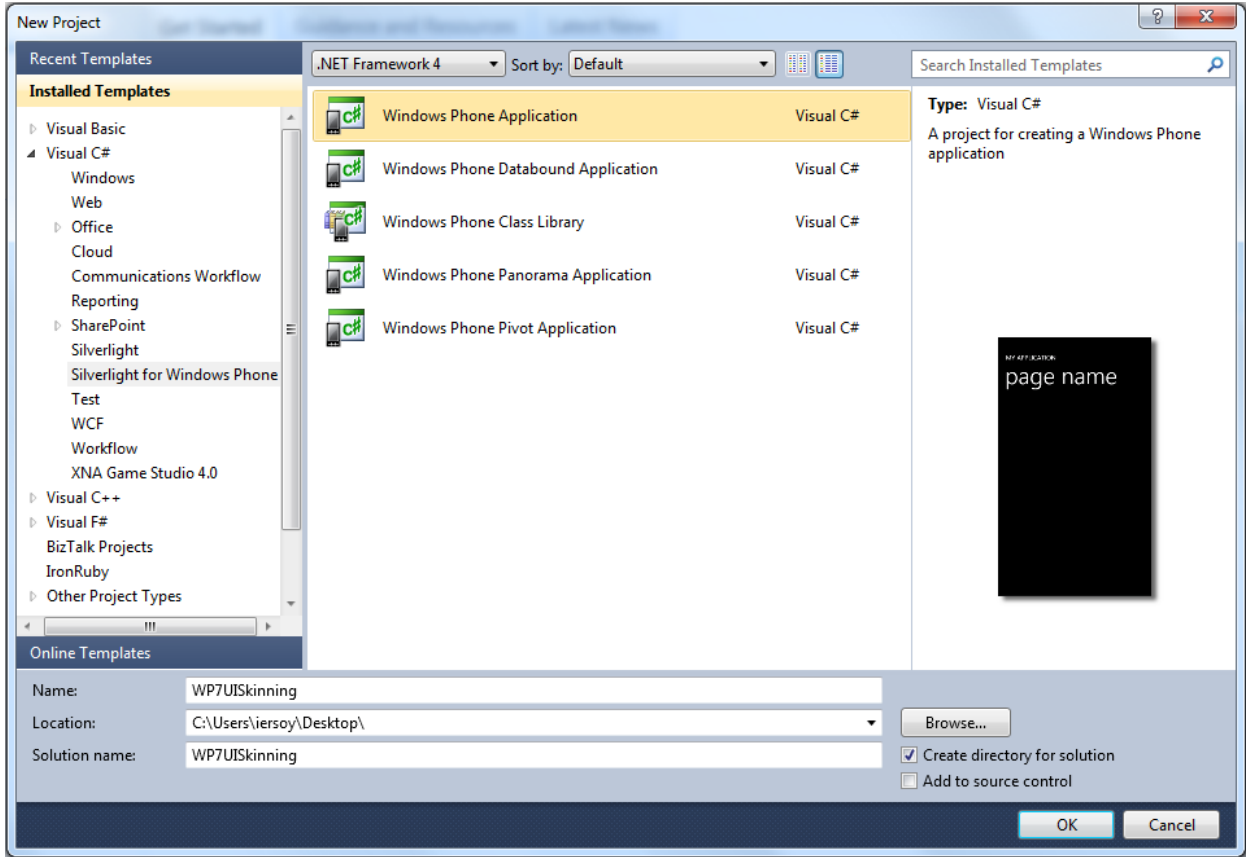


Button Disabled:



Ancak bir TextBox kontrolü için (eğer GotFocus-LostFocus olayları için de stil oluşturmayacaksanız) 1 adet imaj yeterli olacaktır.

Uygulamamıza stil giydirelim o halde:







Yeni bir Silverlight for Windows Phone 7 Uygulaması oluřturun

Projenize řu 4 kontrolü ekleyin: Button, CheckBox, RadioButton and TextBox

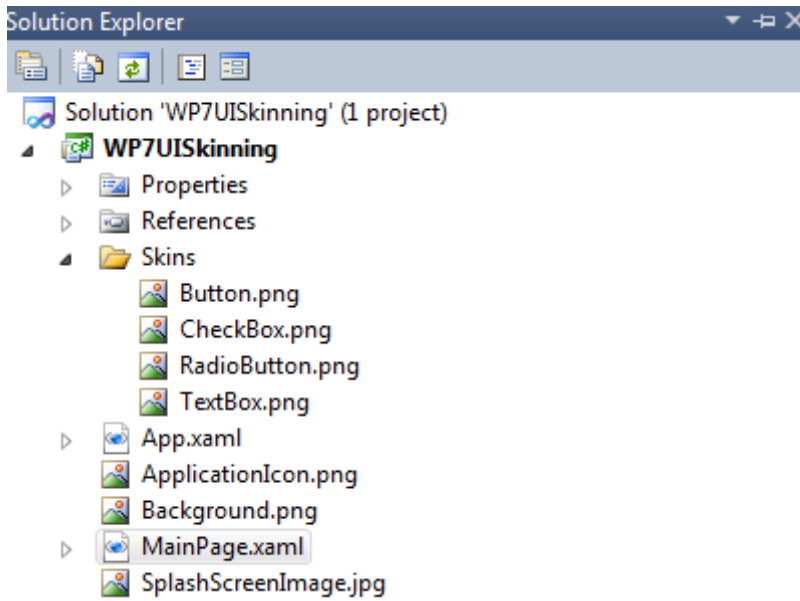


Evet aynen bu şekilde.

Şimdi de projenize aşağıdaki kontrollere ait imajları dahil edin:

-  (Button)
-  (CheckBox)
-  (RadioButton)
-  (TextBox)

Ben düzenli durmaları için "Skins" adlı yeni bir klasör oluşturdum:

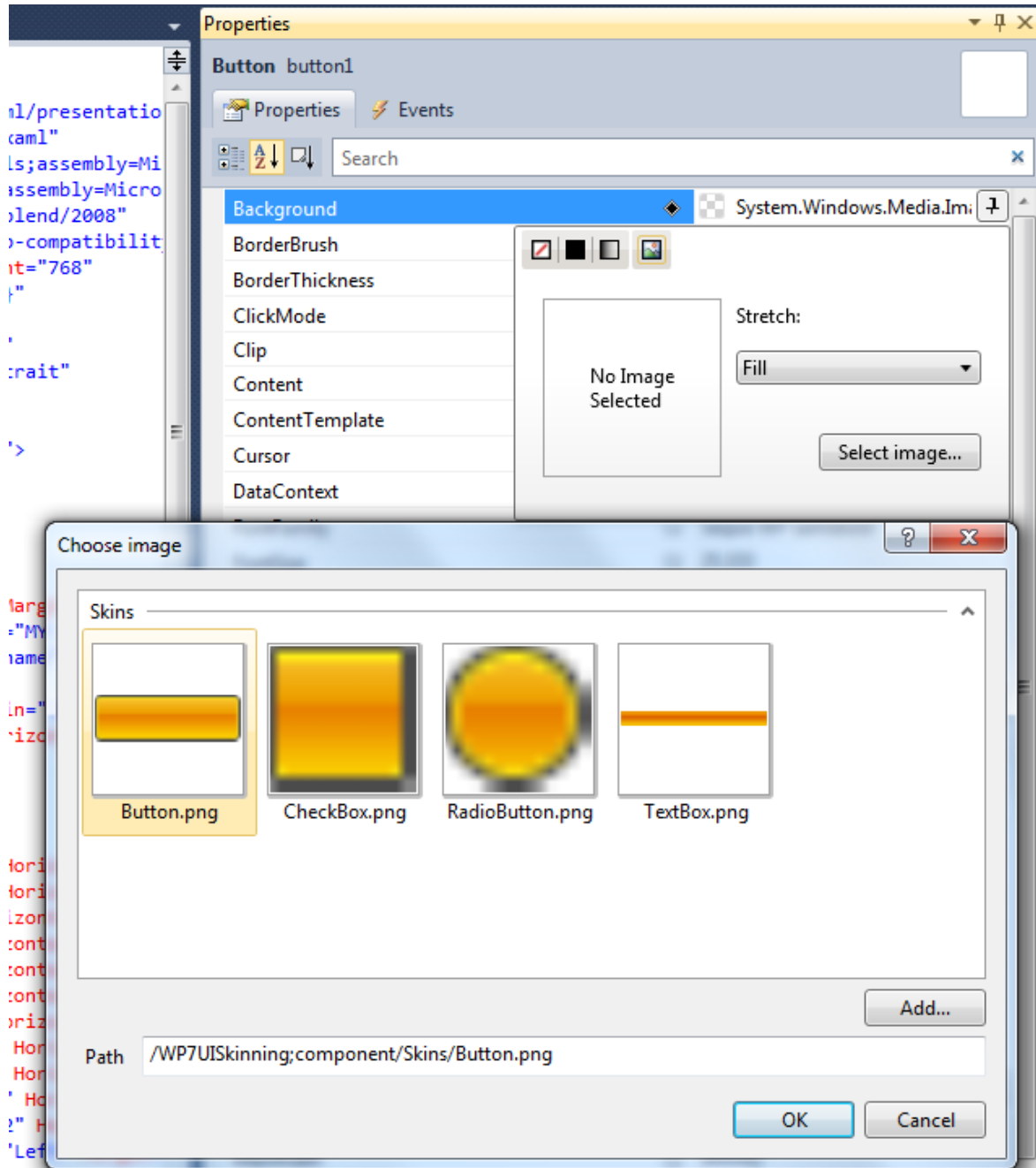


Şimdi önümüzde iki seçenek var:

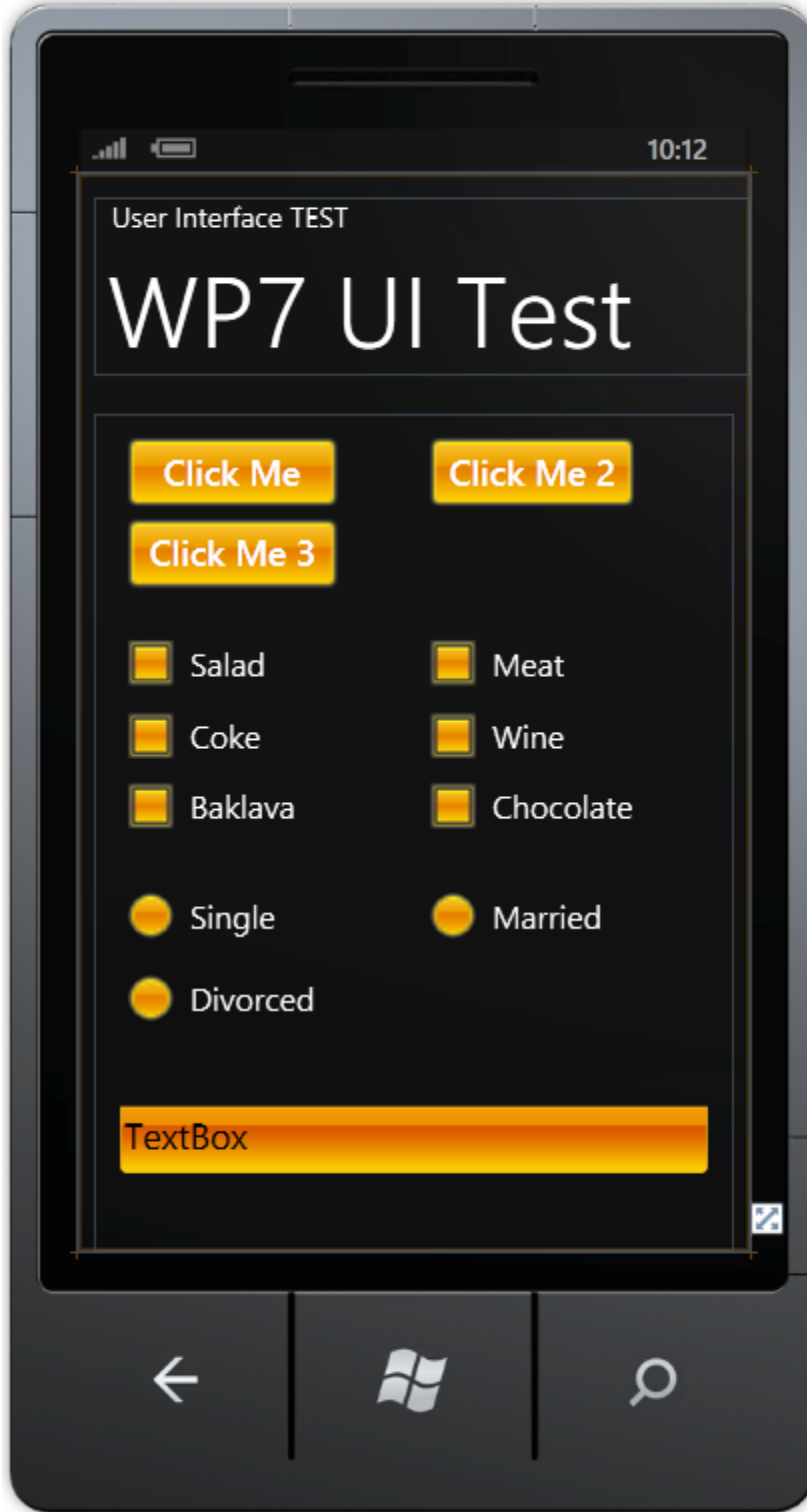
- 1-Manuel olarak stilleri giydirme
- 2-Kodlama ile stilleri giydirme

1) Eğer Manuel olarak bu işlemi yapmak isterseniz her bir kontrole ait Background özelliğini az önce projeye dahil ettiğimiz stil imajlarını ayarlayarak yapabilirsiniz. Bu arada dikkat edilmesi gereken diğer nokta; "BorderThickness" özelliğini "0" değerini vermeniz. Çünkü eğer tüm

kontrolün bizim stilimizde olmasını istiyorsak,XAML tabanlı köşe çizgilerini ortadan kaldırmamız gerekmektedir.Bu işlemi gerçekleştirmek için ilgili kontrole ait Background özelliğini açarak en sağda bulunan resim simgesine tıklayıp ilgili kontrole ait imajları seçmek olacaktır:



Aynı şekilde diğer controller üzerinde de denedikten sonra Tasarım Anında stilimizi görebiliriz.



Görsellik uygulama geliştirmede çok önemlidir ki bu sizin uygulamanızın indirilme-kullanma sayısını da etkileyen bir faktördür.

İlk halinden daha canlı görünüyor öyle değil mi?

2) Eğer kodlama ile yapmak isterseniz:

İlk önce her control için 4 adet Brush nesnesi oluşturun :

```
ImageBrush brush = new ImageBrush();  
ImageBrush brush_checkbox = new ImageBrush();  
ImageBrush brush_radiobutton = new ImageBrush();  
ImageBrush brush_textbox = new ImageBrush();
```

Daha sonra ilgili kod yerinde;

```
brush.ImageSource  
= new BitmapImage(new Uri(@"Skins/Button.png", UriKind.Relative));  
brush_checkbox.ImageSource  
= new BitmapImage(new Uri(@"Skins/CheckBox.png", UriKind.Relative));  
brush_radiobutton.ImageSource  
= new BitmapImage(new Uri(@"Skins/RadioButton.png", UriKind.Relative  
));  
brush_textbox.ImageSource  
= new BitmapImage(new Uri(@"Skins/TextBox.png", UriKind.Relative));
```

Şeklinde bir kodlama yaparsak tüm Brushlara imajları yerleştirmiş oluruz.

```
private void PhoneApplicationPage_Loaded(object sender, RoutedEventArgs e)  
{  
    foreach (UIElement uielement in ContentPanel.Children)  
    {  
        if (uielement.GetType().Name == "Button")  
        {
```

```
((Button)uielement).Background = brush;
((Button)uielement).BorderThickness = new Thickness(0);
}
else if (uielement.GetType().Name == "CheckBox")
{
    ((CheckBox)uielement).Background = brush_checkbox;
    ((CheckBox)uielement).BorderThickness = new Thickness(0);
}
else if (uielement.GetType().Name == "RadioButton")
{
    ((RadioButton)uielement).Background = brush_radiobutton;
    ((RadioButton)uielement).BorderThickness = new Thickness(0);
}
else if (uielement.GetType().Name == "TextBox")
{
    ((TextBox)uielement).Background = brush_textbox;
    ((TextBox)uielement).BorderThickness = new Thickness(0);
}
}
}
```

Kodu ile de tüm kontrollere stil giydirmiş oluruz.

XAML Yöntemi

Bu bölümde size kontrollere XAML ile renk tabanlı stilleri nasıl oluşturacağınızı göstereceğim.

Uygulamanızı oluşturduğunuzu varsayarak;

İlgili kontrollere ait Background özelliğine girerseniz aşağıdaki gibi simgeler göreceksiniz;



Ne amaçla kullanıldıklarını açıklayalım:



Bunu seçtiğinizde kontrole hiçbir renk,efekt eklemeyecektir(Transparan olacaktır). ContentPanel in rengi ne olursa bu kontrolün de rengi o olur.



Bunu seçtiğiniz tek bir renk efekti verirsiniz

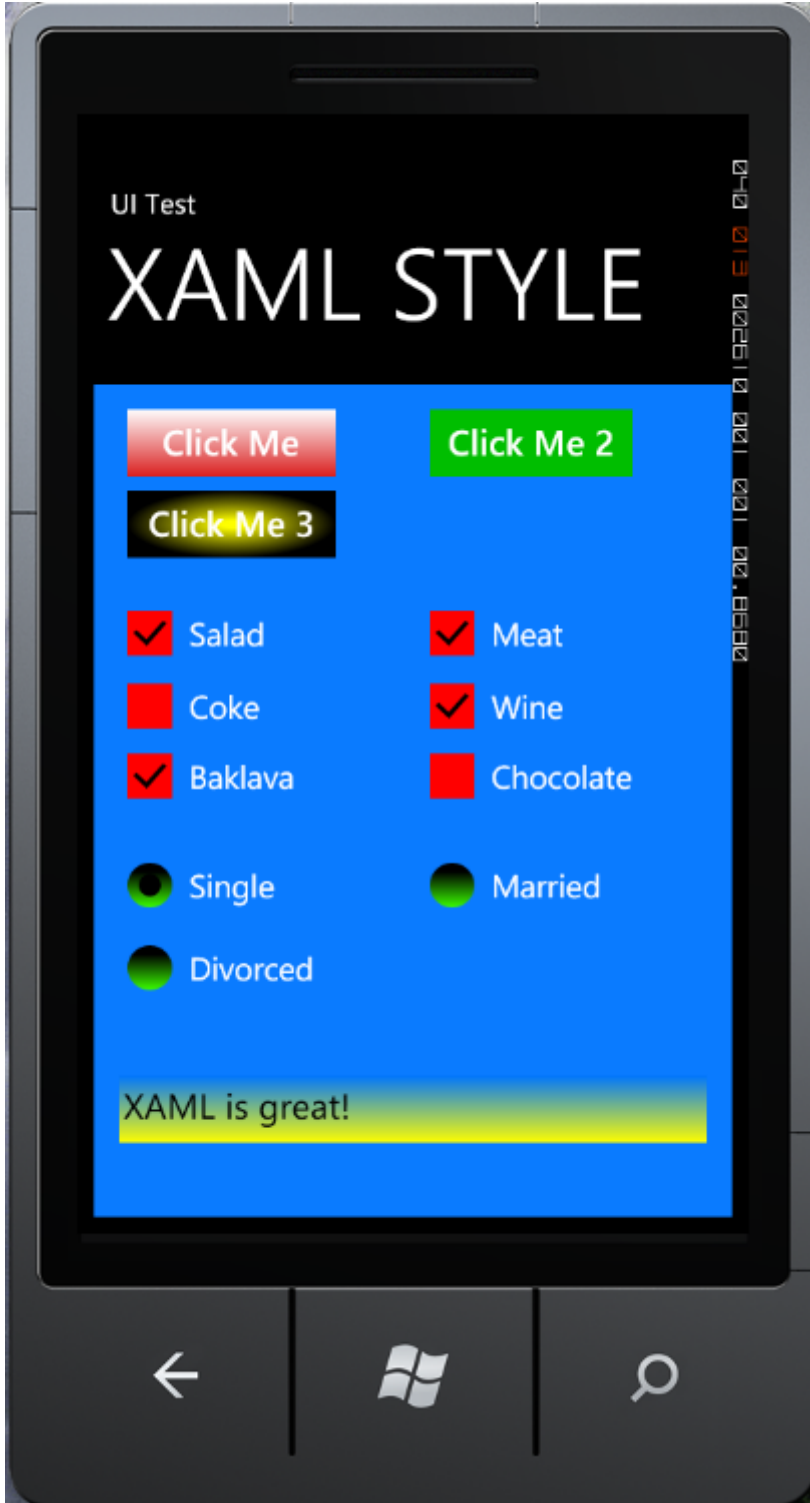


Bunu seçtiğinizde ise farklı renkler arası geçiş işlemi(Gradient efekti) gerçekleştirmiş olursunuz.



Bunu seçerseniz imaj eklersiniz.Lakin bu 1.yöntemdeydi.

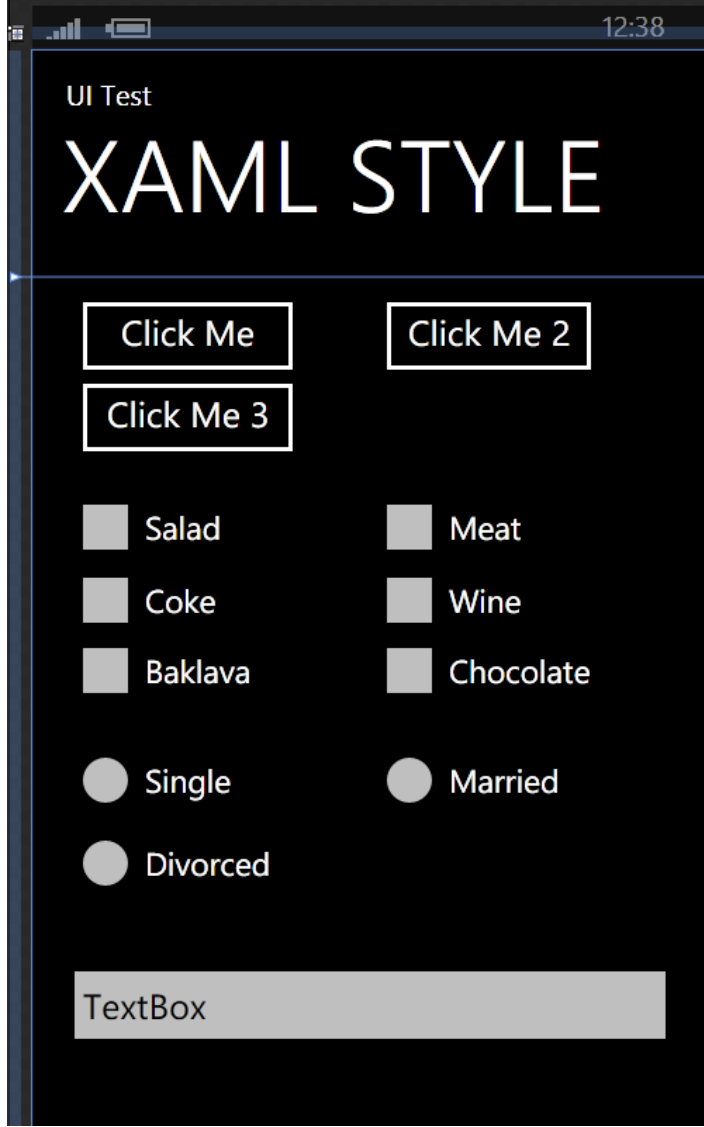
Biraz renkler ile oynadıktan sonra şöyle bir görüntü oluştu:



Gördüğünüz üzere bu efektleri oluşturmak 5 dakikamızı bile almadı. Ne kadar kolay olduğunu gördünüz.

Aynı şekilde Blend 4 aracını kullanarak da extradan efektler eklemeniz, şekillerin görüntüsünü değiştirmeniz, büyütmeniz, döndürmeniz mümkündür.

Projenizi Blend de açmak için projenize sağ tıklayıp "Open in Expression Blend" demeniz yeterli olacaktır.



Bildiğiniz üzere projemizin ilk stili bu şekildeydi.Hadi gelin biraz değişikliğe gidelim.

Aşağıda Döndürme,Küçültme gibi efektleri ekledik.





XAML tabanlı stil oluşturmak isterseniz gerek Visual Studio 2010 gerekse de Blend 4 bunu sağladığından ötürü istediğiniz stili verebilirsiniz.

Bölüm-3: İmaj Tabanlı Yazı Efektleri

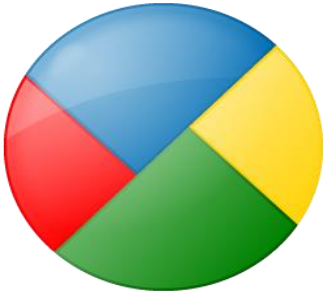
Giriş

Windows Phone 7'de yazılar üzerinde imaj tabanlı gradient efektler oluşturabileceğinizi biliyor muydunuz?

Gelin bunu nasıl yapabiliyoruz hep beraber görelim.

Adım 1 : İlk önce yeni bir Windows Phone 7 uygulaması oluşturun ve aşağıdaki resim gibi çok renkli imaj dosyasını projenize dahil edin

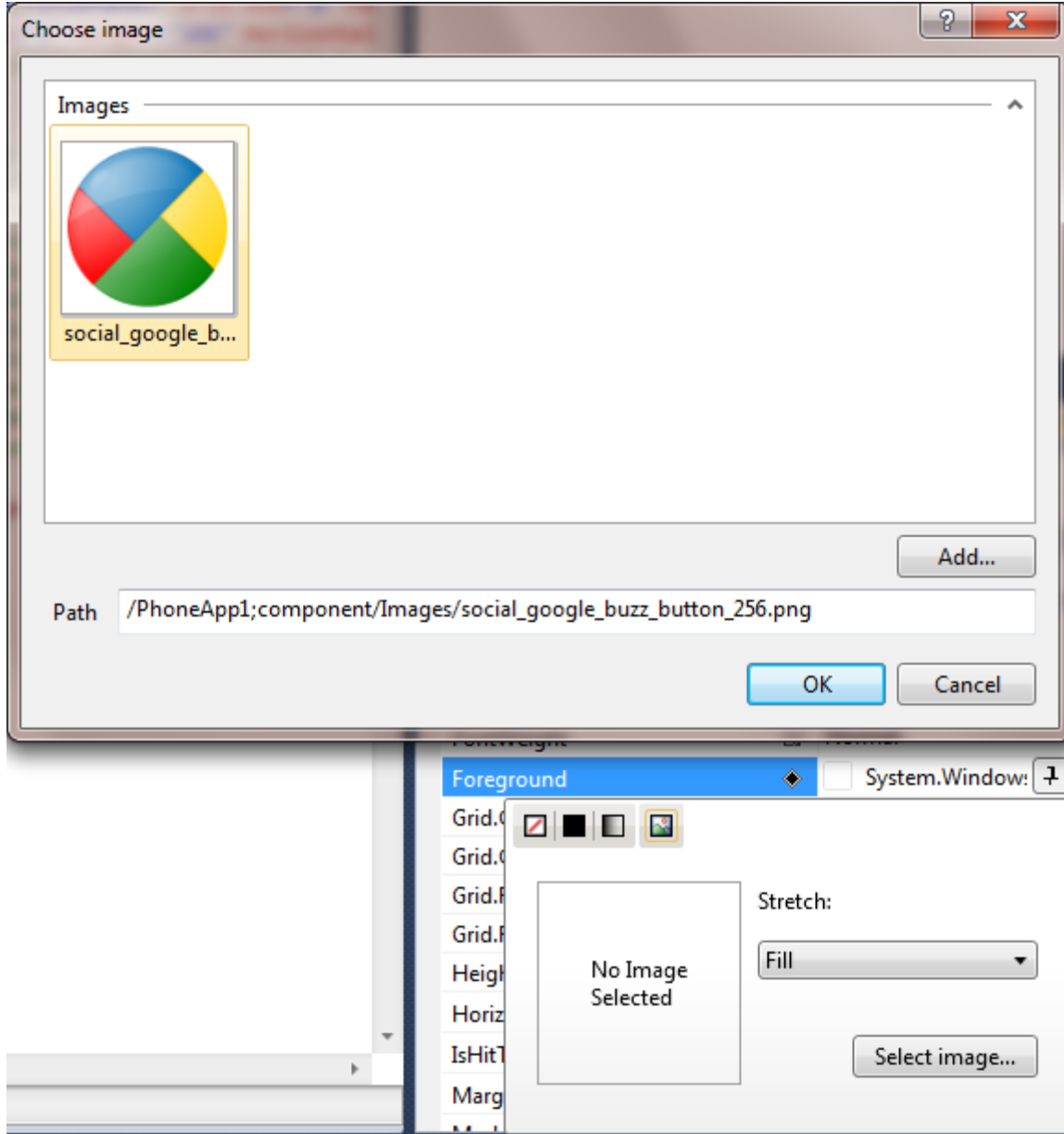
Google Buzz simgesi



Adım 2 : Projenize bir adet yazı ekleyin.



Adım 3 : Şimdi Foreground değerini imaj olacak şekilde değiştirin.



Adım 4 : Stretch değerini UniformToFill 'a çevirmeniz daha gerçekçi bir gradient efekti yakalamanızı sağlayacaktır.Gelin ne yaptık bir göz atalım:



Umarım hořunuza gitmiřtir ve renk tabanlı efektler vermek yerine bu yöntemi uygularsınız.

Bölüm-4: ScrollView Kullanımı

Projenizde ScrollView kullanmak için içi nesnelere ile doldurulmuş bir Grid'e ihtiyaç duyarsınız. ScrollView kullanımındaki hile; Grid'in boyutunu ScrollView'den büyük ayarlamaktan geçer ki aşağı-yukarı işlemleri gerçekleştirilebilsin.

Hemen bir örnek ile konuyu netliğe kavuşturalım:

Yeni bir Windows Phone 7 projesi oluşturduğunuz diyelim.

Projenize ilk önce bir ScrollView ekleyiniz ve boyutunu Height "450" ve Width "450" olacak şekilde oluşturalım:

```
<ScrollView Height="446" HorizontalAlignment="Left"
Margin="12,12,0,0" Name="scrollView1" VerticalAlignment="Top"
Width="456">
```

```
</ScrollView>
```

Daha sonra bu ScrollView içerisine bir adet Grid ekleyelim ve Height değerini 700 yapalım:

```
<Grid Height="700">
```

```
</Grid>
```

Şimdi de bu Grid'in içerisine birkaç nesne ekleyelim:

```
<ScrollView Height="450" HorizontalAlignment="Left"
```

```
Margin="12,12,0,0" Name="scrollView1" VerticalAlignment="Top"
Width="450">
<Grid Height="700">
<Button Width="200" Height="100" Content="Ornek"
Margin="6,552,274,48" />
<Button Content="Ornek" Height="100" Margin="274,552,6,48"
Width="200" />
<CheckBox Content="CheckBox" Height="72"
HorizontalAlignment="Left" Margin="24,37,0,0"
Name="checkBox1" VerticalAlignment="Top" />
<CheckBox Content="CheckBox" Height="72"
HorizontalAlignment="Left" Margin="24,115,0,0"
Name="checkBox2" VerticalAlignment="Top" />
<PasswordBox Height="72" HorizontalAlignment="Left"
Margin="220,37,0,0" Name="passwordBox1"
VerticalAlignment="Top" Width="175" />
<TextBlock Height="30" HorizontalAlignment="Left"
Margin="105,281,0,0" Name="textBlock1" Text="Örnek"
VerticalAlignment="Top" />
</Grid>

</ScrollView>
```

Uygulamayı çalıştıralım ve aşağı kaydığımızda 2 adet button göreceksinizdir.

Bölüm-5: Farklı Panorama Efektleri

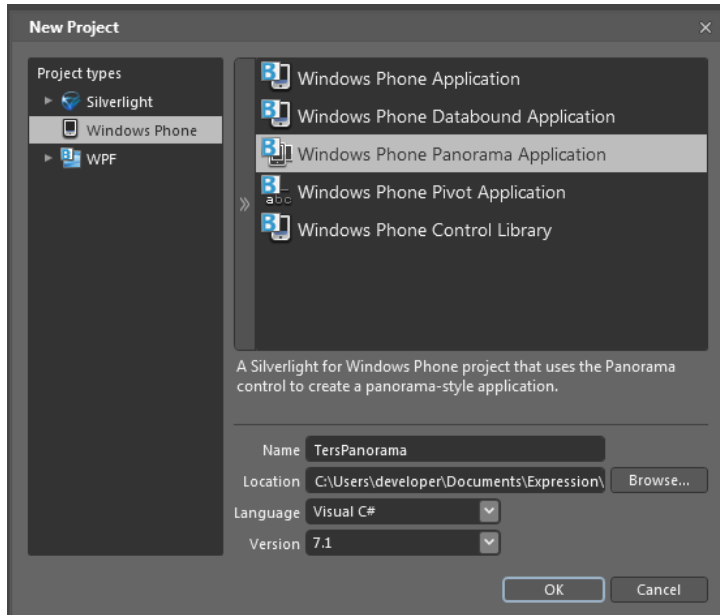
Bu bölümde sizlere Windows Phone 7 alışagelmış olarak kullanılan Panorama projesine alternatif olarak farklı bir yöntem sunacağım.Eğer alternative yöntemler ile Panorama tarzı bir yapı isterseniz,birazdan bahsedeceğim yöntemi uygulayabilirsiniz.Ters Panorama adını verdiğim yöntemden bahsedeyim biraz

Ters Panorama

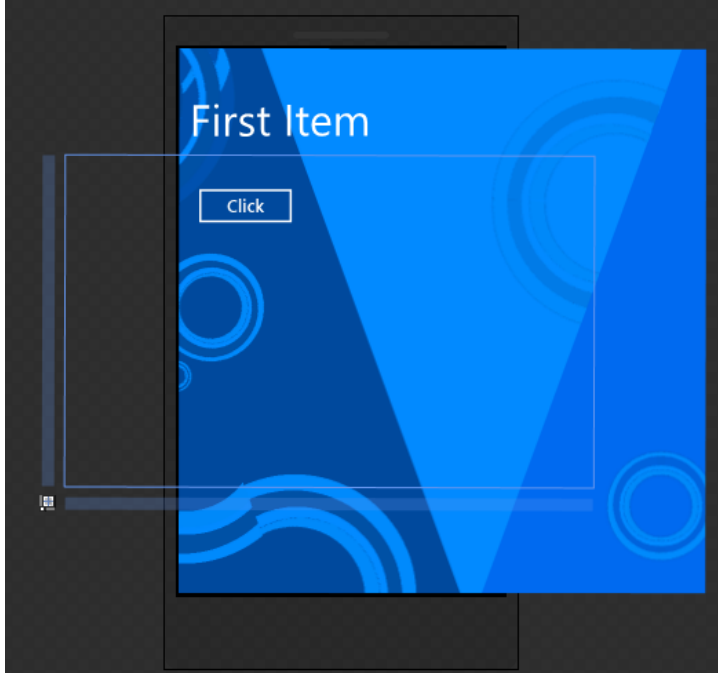
Ters Panorama'nın çalışma mantığında yanlara doğru sürükleme işlemi değil de yukarıdan aşağıya doğru sürükleme işlemi yapılarak gerçekleştirilen bir Panorama Efektidir.Bu tarz bir projeyi vektörel bir işlem olduğundan ötürü Expression Blend ile yapmak gerekir.

O halde yapımına başlayalım.

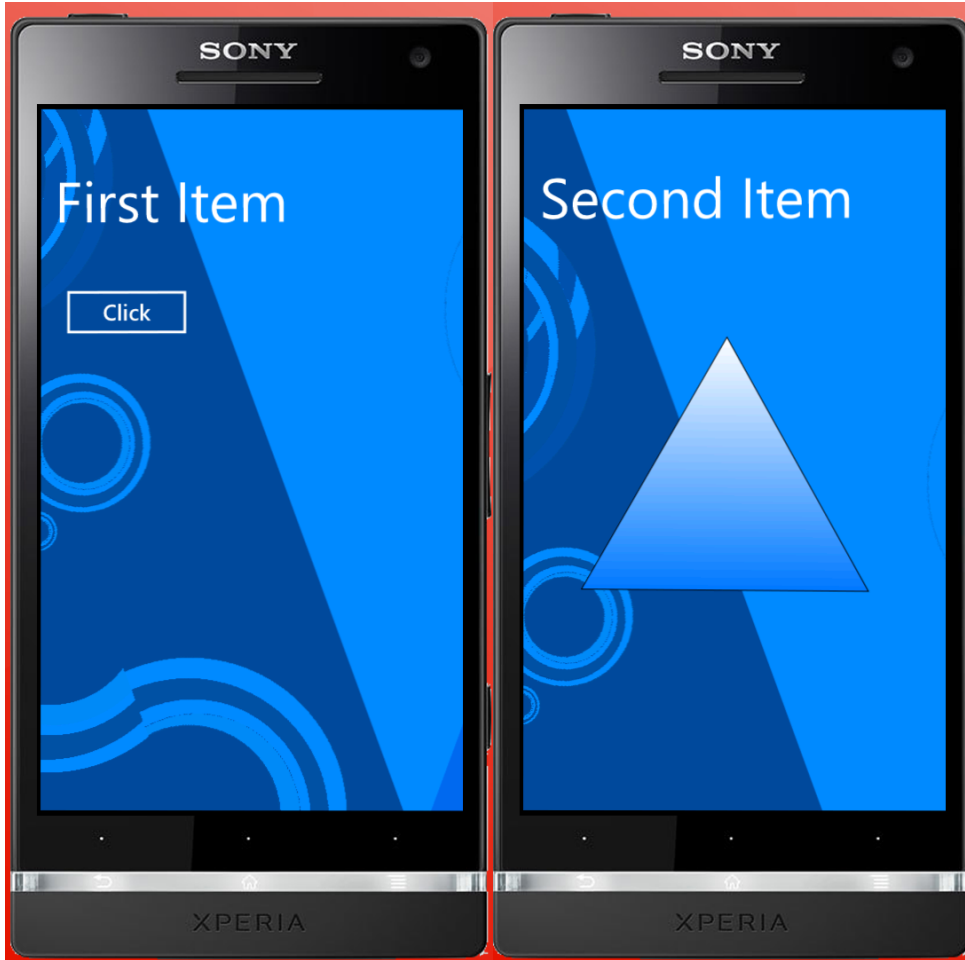
Yeni bir Panorama Uygulaması oluşturun:



Daha sonra LayoutRoot u seçerek 90 derece saga ya da sola döndürün:



Uygulamanızın üzerinde de birkaç değişiklik yaparak; Panorama Headerlar kaldırılması onun yerine TextBlock nesnelere ile hangi Panorama sayfasında olduğunun belirtilmesi ve ilgili Panoramaltem lara Grid eklenerek kontrollerin eklenmesi gibi güncellemeler yaparsak istediğimiz yapıya kavuşacağız.



Varsayılan Panorama yine bildiğimiz şekliyle uygulanır ancak bu sefer dikey olarak.

Yardımcı olması sebebiyle uygulamaya ait kodları paylaşıyorum:

```
<phone:PhoneApplicationPage
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:shell="clr-namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
    xmlns:phone="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
    xmlns:controls="clr-namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone.Controls"
    s
```

```
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
xmlns:es="clr-
namespace:Microsoft.Expression.Shapes;assembly=Microsoft.Expression.
Drawing"
mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="800"
x:Class="TersPanorama.MainPage"
d:DataContext="{d:DesignData
SampleData/MainViewModelSampleData.xaml}"
FontFamily="{StaticResource PhoneFontFamilyNormal}"
FontSize="{StaticResource PhoneFontSizeNormal}"
Foreground="{StaticResource PhoneForegroundBrush}"
SupportedOrientations="Portrait" Orientation="Portrait"
shell:SystemTray.IsVisible="False">
```

```
<!--LayoutRoot is the root grid where all page content is placed-->
<Grid x:Name="LayoutRoot" Background="Transparent"
RenderTransformOrigin="0.5,0.5" Margin="-17.009,17.021,17.009,16.997"
UseLayoutRounding="False" d:LayoutRounding="Auto">
    <Grid.RenderTransform>
        <CompositeTransform Rotation="-89.841"/>
    </Grid.RenderTransform>

    <!--Panorama control-->
    <controls:Panorama Title="" Margin="-154.007,163.986,-
156.015,-163.986" RenderTransformOrigin="0.5,0.5">
        <controls:Panorama.RenderTransform>
            <CompositeTransform Rotation="-0.096"/>
        </controls:Panorama.RenderTransform>
        <controls:Panorama.Background>
            <ImageBrush
ImageSource="PanoramaBackground.png"/>
        </controls:Panorama.Background>

    <!--Panorama item one-->
```

```

        <controls:Panoramaltem Header=""
RenderTransformOrigin="0.5,0.5">
            <Grid Margin="0,-114,0,0">
                <Button Content="Click"
HorizontalAlignment="Right" Margin="0,150.654,101.323,0"
VerticalAlignment="Top" RenderTransformOrigin="0.5,0.5"
Width="158.179">
                    <Button.RenderTransform>
                        <CompositeTransform
Rotation="89.891"/>
                    </Button.RenderTransform>
                </Button>
                <TextBlock HorizontalAlignment="Right"
Margin="0,193.876,-78.937,0" TextWrapping="Wrap" Text="First Item"
VerticalAlignment="Top" FontSize="64"
RenderTransformOrigin="0.5,0.5">
                    <TextBlock.RenderTransform>
                        <CompositeTransform
Rotation="89.767"/>
                    </TextBlock.RenderTransform>
                </TextBlock>
            </Grid>
        </controls:Panoramaltem>

```

```

        <controls:Panoramaltem Header=""
RenderTransformOrigin="0.5,0.5">
            <Grid Margin="0,-113.993,0,0">
                <TextBlock HorizontalAlignment="Right"
Margin="0,241.821,-128.49,0" TextWrapping="Wrap" Text="Second Item"
VerticalAlignment="Top" FontSize="64"
RenderTransformOrigin="0.5,0.5">
                    <TextBlock.RenderTransform>
                        <CompositeTransform
Rotation="89.767"/>
                    </TextBlock.RenderTransform>
            </Grid>
        </controls:Panoramaltem>

```

```
</TextBlock>

    <es:RegularPolygon InnerRadius="1"
Margin="216.752,174.93,188.933,393.919" PointCount="3" Stretch="Fill"
Stroke="Black" RenderTransformOrigin="0.5,0.5">
        <es:RegularPolygon.Fill>
            <LinearGradientBrush
EndPoint="0.5,1" StartPoint="0.5,0">
                <GradientStop
Color="White" Offset="0"/>
                <GradientStop
Color="#FF0079FF" Offset="1"/>
            </LinearGradientBrush>
        </es:RegularPolygon.Fill>

        <es:RegularPolygon.RenderTransform>
            <CompositeTransform
Rotation="90.385"/>
        </es:RegularPolygon.RenderTransform>
    </es:RegularPolygon>

</Grid>
</controls:Panoramaltem>
</controls:Panorama>
</Grid>
</phone:PhoneApplicationPage>
```

Bölüm-6: Kopyala-Yapıştır Özelliđi

Bildiđiniz üzere Windows Phone 7 çıkmadan önce "WishList" adını verdikleri Türkçe olarak "Dilek Listesi" diye tabir edilen bir konu açılmıştı MSDN Forumunda.

Listenin en başında da bu özellik vardı.Hatırlayanlar bilir.

Bu özellik İşletim Sisteminin bir parçası olarak çalıştığından ötürü olacak ki kodlama yaparak bu özelliđin değerlerine erişemiyoruz.

Ancak ben yine de bu özelliđin nasıl kullanıldığını göstermek isterim.



Gördüğünüz üzere uygulamamız içerisinde bir adet TextBox kullanıyoruz.Gelin bu kontrolümüzden bir harf seçerek kopyala-yapıştır nasıl yapılır görelim.İlk önce bir kelime seçiyoruz:



Görmüş olduğunuz düğme kelimeleri kopyalamaya yarar. Sağa-sola kaydırarak kopyalanacak kelimeyi uzatabilirsiniz.



Düğmeye bastıktan sonra kelimeniz kopyalanmış oldu.



Şimdi Sanal Klavyenin en üst sol köşesinde bulunan düğmeye bakın.Bu düğme de Yapıştırma düğmesidir.Bastığınız zaman kelimeyi ilgili yer olarak nereyi seçtiyseniz oraya yapıştırır.



Şekilde görüldüğü gibi!

Bölüm-7: CAPDETECT Aracının Kullanımı

CapDetect Aracı "WManifest.xml" adlı dosyayı uygulamanın ihtiyaç duyduğu niteliklere göre tekrar düzenlemenize olanak sağlar.

CapDetect Aracı parameter olarak xap dosyasının bulunduğu yolu alır.
Kullanımı:

CapabilityDetection <rules path> <file path>
şeklindedir.

Sisteminizde bu aracı bulabileceğiniz yolu verelim hemen:

32 bit işletim sisteminde -> %ProgramFiles%\Microsoft SDKs\Windows Phone\v7.0\Tools\CapDetect

64 bit işletim sisteminde -> %ProgramFiles (x86)%\Microsoft SDKs\Windows Phone\v7.0\Tools\CapDetect

Şimdi bu aracı kullanalım:

Komut isteminde CapDetect aracının bulunduğu yere gidin ve aşağıdakine benzer bir komut giriniz:

```
CapabilityDetection Rules.xml  
"C:\Users\iersoy\Desktop\WindowsPhoneGame2\WindowsPhoneGame2  
\WindowsPhoneGame2\bin\WindowsPhone\Debug"
```

Yolu kendi uygulamanıza uyarlamamız gerekecektir.

Yalnız bir şeye dikkatinizi çekerim!

(") işaretini yolu belirlerken kullanmak zorundasınız aksi takdirde hata alacaksınız.

Örnek olsun diye uygulamamın WManifest.xml dosyasından 4 adet niteliği sildim:

```
<Capability Name="ID_CAP_MICROPHONE" />
<Capability Name="ID_CAP_MEDIALIB" />
<Capability Name="ID_CAP_GAMERSERVICES" />
<Capability Name="ID_CAP_PHONEDIALER" />
```

Lakin aracı kullandıktan sonra yeniden eklendiler.

```
<?xml version="1.0" encoding="utf-8"?>
<Deployment
xmlns="http://schemas.microsoft.com/windowsphone/2009/deployment
"
AppPlatformVersion="7.0">
  <App xmlns="" ProductID="{bfd638f3-4ca0-4043-9871-
d34bc559eeb6}"
Title="WindowsPhoneGame2"
RuntimeType="XNA" Version="1.0.0.0" Genre="Apps.Normal"
Author="" Description="" Publisher="">
  <IconPath IsRelative="true" IsResource="false">
GameThumbnail.png</IconPath>
  <Capabilities>
    <Capability Name="ID_CAP_NETWORKING" />
    <Capability Name="ID_CAP_LOCATION" />
    <Capability Name="ID_CAP_SENSORS" />
    <Capability Name="ID_CAP_MICROPHONE" />
    <Capability Name="ID_CAP_MEDIALIB" />
    <Capability Name="ID_CAP_GAMERSERVICES" />
    <Capability Name="ID_CAP_PHONEDIALER" />
    <Capability Name="ID_CAP_PUSH_NOTIFICATION" />
    <Capability Name="ID_CAP_WEBBROWSERCOMPONENT" />
    <Capability Name="ID_CAP_IDENTITY_USER" />
    <Capability Name="ID_CAP_IDENTITY_DEVICE" />
  </Capabilities>
  <Tasks>
    <DefaultTask Name="_default" />
```

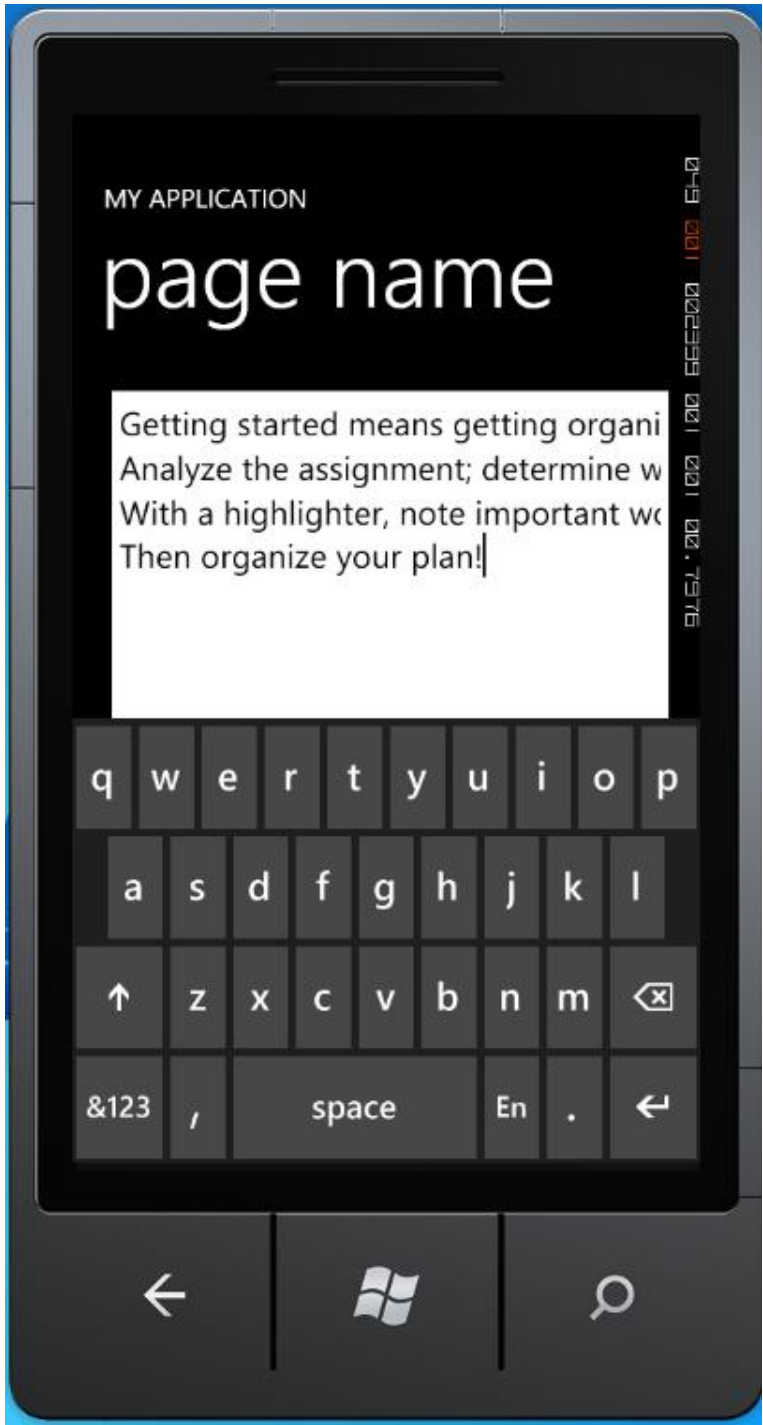
```
</Tasks>
<Tokens>
  <PrimaryToken TokenID="WindowsPhoneGame2Token"
TaskName="_default">
  <TemplateType5>
    <BackgroundImageURI IsRelative="true" IsResource="false">
Background.png</BackgroundImageURI>
    <Count>0</Count>
    <Title>WindowsPhoneGame2</Title>
  </TemplateType5>
</PrimaryToken>
</Tokens>
</App>
</Deployment>
```

Bu araç oldukça yararlıdır.Kullanmanızı tavsiye ederim.

Bölüm-8: Sanal Klavye'yi Etkisizleştirme

Sanal Klavyeyi etkisizleştirmek için birkaç yöntem bulunmaktadır(InputScope gibi)

Lakin ben farklı bir yöntem önereceğim.



MY APPLICATION

page name

Getting started means getting organi
Analyze the assignment; determine w
With a highlighter, note important wc
Then organize your plan!

q w e r t y u i o p
a s d f g h j k l
↑ z x c v b n m ↵
&123 , space En . ↵



İlgili TextBox nesnenize ait değeri:

`IsHitTestVisible="False"` (in XAML)

ya da

`textBox1.IsHitTestVisible = false;` (dynamically)

şeklinde değiştirdiğiniz vakit Sanal Klavye bir daha görüntülenemeyecektir.

Bölüm-9: Navigasyon

Windows Phone da Navigasyon dediğimiz sayfalar arası geçişler için NavigationService adlı sınıftan faydalanırız.

Projenizde iki adet Portrait ya da Landscape sayfalar olduğunu varsayarsak;

1.sayfadan 2.sayfaya gitmek için aşağıdaki gibi bir kodlama kullanmamız gerekecektir.

```
NavigationService.Navigate(new Uri("/Page2.xaml", UriKind.Relative));
```

Böylelikle uygulama çalıştığında ikinci sayfaya geçiş yapmış olacağız.

Bölüm-10: Görevler ve Seçiciler

Bu bölümde sizlere Task&Choosers diye tabir edilen Görevler ve Seçicileri açıklayacağım.

Windows Phone 7 de extra bazı işleri gerçekleştirmek için küçük programcıklar mevcuttur.Bu programcıklar vasıtasıyla çok kolay bir şekilde kullanıcı ile etkileşime girebilirsiniz.Bu küçük programcıklara biz "Tasks&Choosers" deriz.

XNA ya da Silverlight farketmeksizin bu işlemleri gerçekleştirebilirsiniz.Bu özellikleri kullanabilmek için öncelikle Microsoft.Phone u reference olarak eklemeli ardından projeye Microsoft.Phone.Tasks ı dahil etmeniz gerekir.

Bu bölümde bahsedeceğimiz Görevler ve Seçiciler:

- Email Address Chooser Task
- MarketPlace Search Task
- MarketPlace Hub Task
- MediaPlayer Launcher
- Phone Call Task
- Phone Number Chooser Task
- Photo Chooser Task
- Search Task
- SMS Compose Task
- WebBrowser Task

Email Address Chooser Task:

Bu seçici size iletişim listenizden birini seçmenizi sağlar.

Yeni bir değişken tanımlayın:

```
EmailAddressChooserTask ect = new EmailAddressChooserTask();
```

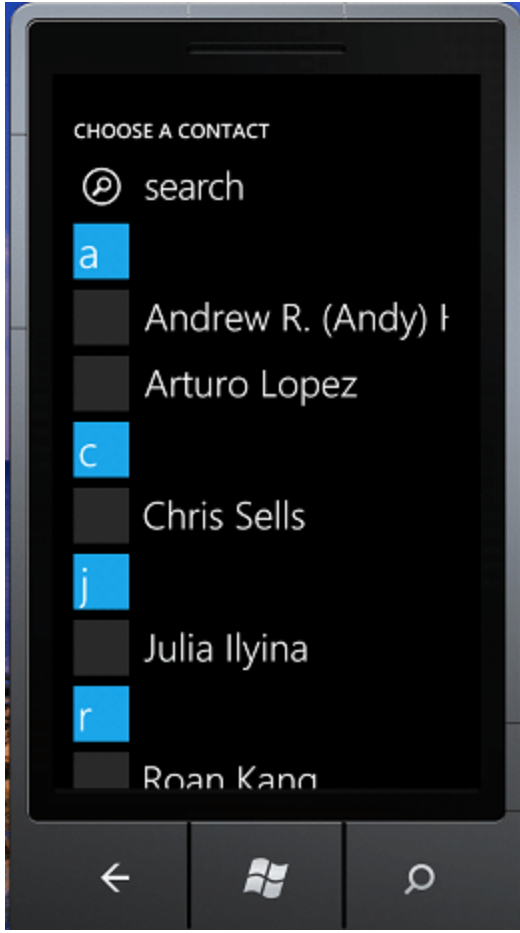
Ve çağırın:

```
ect.Completed += new EventHandler<EmailResult>(ect_Completed);  
ect.Show();
```

Şu Eventhandler l da eklemeyi unutmayınız:

```
public void ect_Completed(object sender, EmailResult e)  
{  
    if (e.TaskResult == TaskResult.OK)  
    {  
        Debug.WriteLine(e.Email);  
    }  
    else  
    {  
    }  
}
```

Şimdi gelin sonucu görelim



Burası İletişim Listeniz.Buradan birini seçelim örneğin Julia Ilyina

Sonuç:

Julia@example.com

MarketPlace SearchTask:

MarketPlace Search Task görevi size Marketplace den Müzik veya Uygulama aramanıza olanak sağlar.

Ancak öncelikle Aranacak kelimeleri ve tipini belirtmeniz gerekiyor

Öncelikle MarketplaceSearchTask görevi oluşturalım:

```
MarketplaceSearchTask mst = new MarketplaceSearchTask();
```

Sonra da bir konuyla ilgili müzik arayalım:

```
mst.ContentType = MarketplaceContentType.Music;  
mst.SearchTerms = "usain";  
mst.Show();
```

Bu Usain Bolt ile ilgili Müzik albümlerini karşımıza getirecektir.

MARKETPLACE SEARCH

usain



Usain Bolt Tribute
album



Usain Bolt Tribute
song



Usain Bolt Nuh Ling
album



Usain Bolt Nuh Ling
song



Usain Bolt
song



MarketPlace Hub Task:

Diğer bir MarketPlace Görevi ise MarketPlaceHubTask 'dir.

Bu size Zune MarketPlace i getirecektir. Bir Panorama Uygulaması şeklinde çalışır

Bu işlemi gerçekleştirmek için

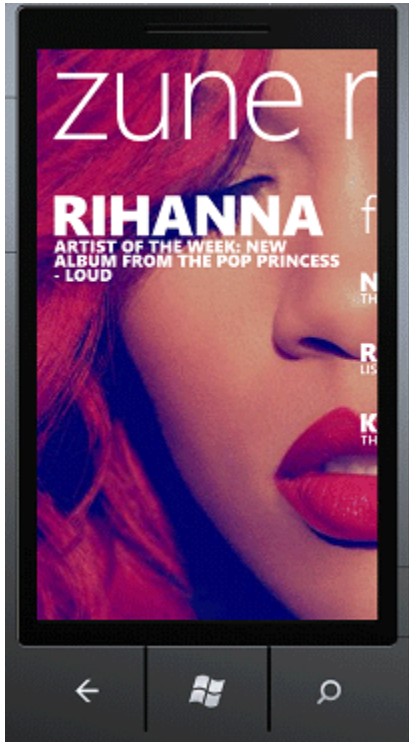
İlk önce MarketplaceHubTask nesnemizi oluştururuz:

```
MarketplaceHubTask mht = new MarketplaceHubTask();
```

Daha sonra da bunu görüntüleriz.

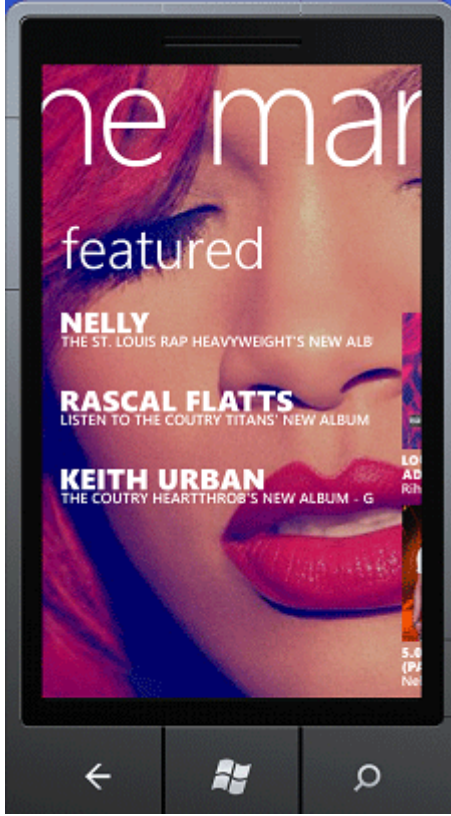
```
mht.ContentType = MarketplaceContentType.Music;  
mht.Show();
```

Çalıştırdığımızda şöyle bir ekran gelecektir:



1.sayfa her zaman bize Haftanın Sanatçısını getirir.

Diğer sayfalarda ise benzer albümler yer alır.



MediaPlayer Launcher:

MediaPlayerLauncher adı üstünde MediaPlayer ı çalıştırmanıza olanak sağlar.

İlk önce MediaPlayerLaucher nesnesini oluşturuyoruz:

```
MediaPlayerLauncher mpl = new MediaPlayerLauncher();
```

Daha sonra da herhangi bir video dosyasını oynatmasını söylüyoruz:

```
mpl.Controls = MediaPlayerPlaybackControls.All;  
mpl.Location = MediaPlayerLocationType.Install;  
mpl.Media  
= new Uri("http://www.samplewebsitehere.com/sample.wmv"); //This is a  
sample.Please change it as your needs.  
mpl.Show();
```

Durdurma,İleri-Geri Sarma,Duraklatma gibi fonksiyonları da bulunmaktadır

Phone Call Task:

Herhangi bir numarayı aramanızı sağlar.

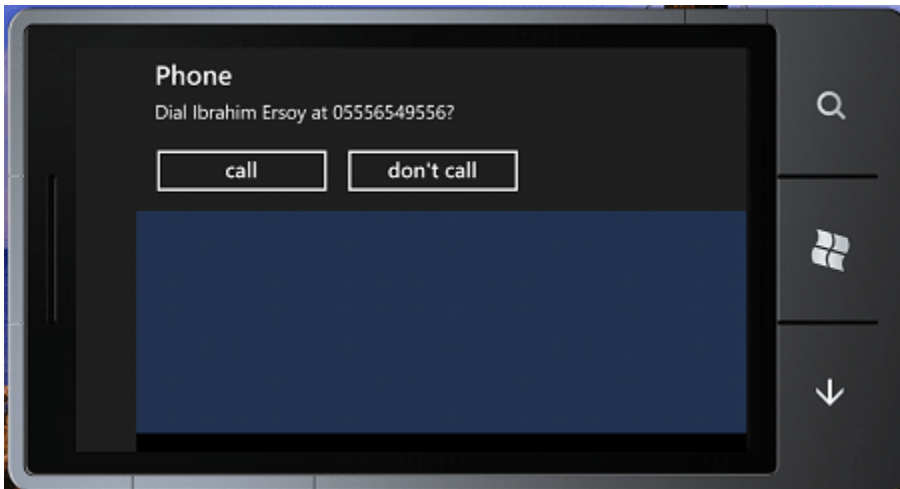
İlk önce PhoneCallTask nesnesini oluştururuz:

```
PhoneCallTask pct = new PhoneCallTask();
```

Daha sonra da aranacak numarayı gireriz:

```
pct.DisplayName = "İbrahim Ersoy";  
pct.PhoneNumber = "05556549556"; //Sahte numara  
pct.Show();
```

Uygulamayı çalıştırdığımızda verilen Numara ve ismi arayacaktır:



Phone Number Chooser:

İletişimlerimizden telefon numarası seçmemize yarar.

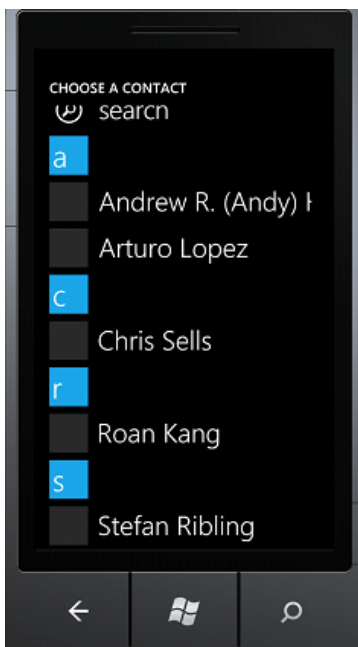
İlk önce PhoneNumberChooserTask nesnesini oluştururuz:

```
PhoneNumberChooserTask pnct = new PhoneNumberChooserTask();
```

Sonuçları görmek için bir olay ekler ve görüntüleriz:

```
pnct.Completed  
+= new EventHandler<PhoneNumberResult>(pnct_Completed);  
pnct.Show();
```

```
public void pnct_Completed(object sender, PhoneNumberResult e)  
{  
    Debug.WriteLine(e.PhoneNumber);  
}
```



Chris Sells i seçtiğimde bana : (206)555-0003 değeri döndürecektir.

Photo Chooser:

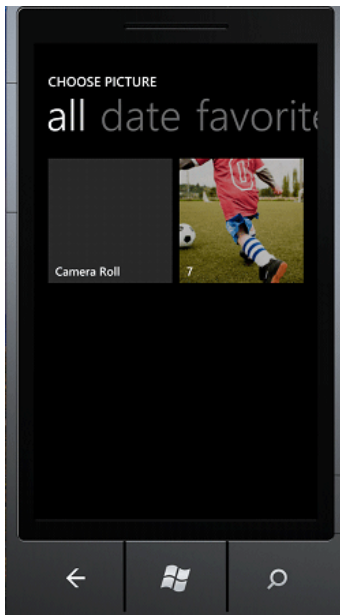
Bu seçici, aygıtımızdan bir fotoğraf seçmemize olanak sağlar. İlk önce PhotoChooserTask seçicisini oluştururuz:

```
PhotoChooserTask pct = new PhotoChooserTask();
```

Daha sonra bir olay ekler ve sonucu yazdırırız:

```
pct.Completed  
+= new EventHandler<PhotoResult>(photochooser_Completed);  
pct.Show();  
public void photochooser_Completed(object sender, PhotoResult e)  
{  
    Debug.WriteLine("Size: " + e.ChosenPhoto.Length + " bytes");  
    Debug.WriteLine("File Name: " + e.OriginalFileName);  
}
```

Sonuç:



Size: 85524 bytes

File Name: \Applications\Data\314F9DB7-9953-48CE-847B-A4D5FE25B058\Data\PlatformData\PhotoChooser-168c35b3-a7e3-4355-9124-ca3d25b93481.jpg

Search Task:

Bu görev Bing Arama Motorunu kullanarak herhangi bir şey aramanıza olanak sağlar.

İlk önce SearchTask nesnesi oluştururuz:

```
SearchTask st = new SearchTask();
```

Daha sonra da aranacak kelimeleri gireriz:

```
st.SearchQuery = "iersoy xna";  
st.Show();
```

ve sonuç:



SMS Compose Task:

Bu görev SMS göndermenize yarar.

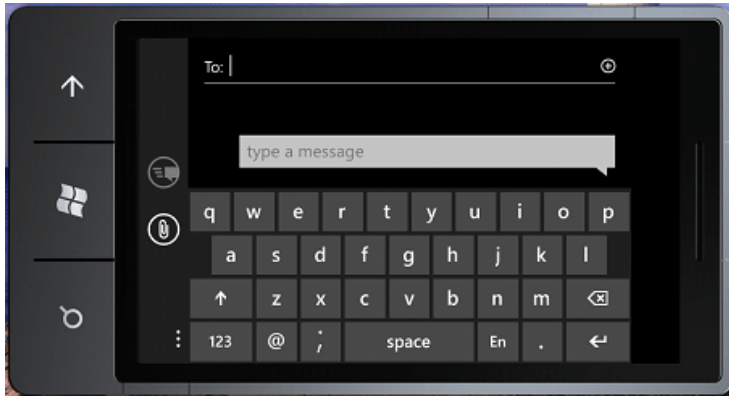
İlk önce SmsComposeTask görevi oluştururuz:

```
SmsComposeTask smsct = new SmsComposeTask();
```

Daha sonra da SMS yazma ekranını gösteririz:

```
smsct.Show();
```

Sonuç:



Web Browser Task:

WebBrowserTask görevi kısaca websitesi görüntülemeye yarar.

İlk önce bir WebBrowserTask nesnesi oluştururuz:

```
WebBrowserTask wbt = new WebBrowserTask();
```

Daha sonra açılacak websitesini girer ve görüntüleriz:

```
wbt.URL = "http://www.csharpcorner.com";
```

```
wbt.Show();
```

Sonuç:



Bölüm-11: Aygıt Bilgileri

Windows Phone 7 aygıtınıza ait bilgilere ulaşmak mı istiyorsunuz? Size bunu nasıl yapacağınızı birazdan anlatacağım.

Emulator kullandığımdan ötürü olacak ki gerçekçi veriler alınamayacaktır bu sebepten ötürüdür ki gerçek bir aygıt üzerinde test etmeniz yararınıza olacaktır

O halde başlayalım.

Örneğimizde aşağıda gördüğünüz bilgilere erişmeye çalışacağız:

- Manufacturer
- Device Name
- Device ID
- Firmware Version
- Hardware Version
- Total Memory
- Application Current Memory Usage
- Application Peak Memory Usage

Bu örneğimizde DispatcherTimer vasıtasıyla her 3 saniyede bir bilgileri güncelleyen 8 adet Textblock nesnesine sahip bir uygulama geliştireceğiz.

Uygulamamızın görünüşü aşağıdaki gibi olacaktır:



- Manufacturer = **textBlock2**
- Device Name = **textBlock4**
- Device ID = **textBlock6**
- Firmware Version = **textBlock8**
- Hardware Version = **textBlock10**
- Total Memory = **textBlock12**
- Application Current Memory Usage = **textBlock14**
- Application Peak Memory Usage = **textBlock16**

Yukarıda ilgili textblock nesnelere çekeceğimiz bilgileri eşleşmiş olarak görmekteyiz.

İlk öncelikle yapacağımız işlem bir DispatcherTimer nesnesi oluşturmak olacaktır.

`DispatcherTimer` timer;

Tabi bunu dahil etmek için System.Windows.Threading namespace'ini projenize eklemeniz gerekmektedir.

Device ID bilgisini string formatında çekmek için bir değişken tanımladık:

```
public static string val;
```

Device ID'yi byte tipinde döndüren ve bunu "val" adlı string değişkene aktaran bir fonksiyon geliştirdik.

```
public static byte[] GetDeviceUniqueId()
{
    byte[] result = null;
    object uniqueId;
    if (DeviceExtendedProperties.TryGetValue("DeviceUniqueId", out uniqueId))
    {
        result = (byte[])uniqueId;
    }
    val=Convert.ToBase64String(result);
    return result;
}
```

Yukarıdaki kodun kaynağı :<http://www.nickharris.net/2010/09/windows-phone-7-how-to-find-the-device-unique-id-windows-live-anonymous-id-and-manufacturer/>

Device ID yi direct çekseydik bize byte tipinde veriler gönderecekti.Dolayısıyla burada yaptığımız aslında Unique ID yi bulmak oldu.

Şimdi yapacağımız işlemde ise DispatcherTimer nesnesi çalıştırmak ve DeviceUniqueId yi bulmaktır.

```
timer = new DispatcherTimer();
timer.Interval = new TimeSpan(0, 0, 3);
timer.Tick += new EventHandler(timer_Tick);
timer.Start();
GetDeviceUniqueId();
```

Tabi timer_Tick fonksiyonumuzu da unutmayalım:

```
void timer_Tick(object sender, EventArgs e)
{
    try
    {
        textBlock2.Text
        = DeviceExtendedProperties.GetValue("DeviceManufacturer").ToString();
        textBlock4.Text
        = DeviceExtendedProperties.GetValue("DeviceName").ToString();
        textBlock6.Text = val;
        textBlock8.Text
        = DeviceExtendedProperties.GetValue("DeviceFirmwareVersion").ToString();
        textBlock10.Text
        = DeviceExtendedProperties.GetValue("DeviceHardwareVersion").ToString();
        textBlock12.Text
        = DeviceExtendedProperties.GetValue("DeviceTotalMemory").ToString();
        textBlock14.Text
        = DeviceExtendedProperties.GetValue("ApplicationCurrentMemoryUsage").ToString();
    }
}
```

```
textBlock16.Text
=DeviceExtendedProperties.GetValue("ApplicationPeakMemoryUsage").T
oString();
}
catch (Exception ex)
{
}
}
}
```

DeviceExtendedProperties kullanarak WP7 Aygıtına ait bilgilere erişebiliyoruz.Parametreleri hakkında daha fazla bilgi isterseniz aşağıdaki linke bir gözatmanızda fayda var:

[http://msdn.microsoft.com/en-us/library/ff941122\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff941122(v=VS.92).aspx)

Uygulamamızı çalıştırdıktan sonra bilgilerimize ulaşılmış olarak görürüz:



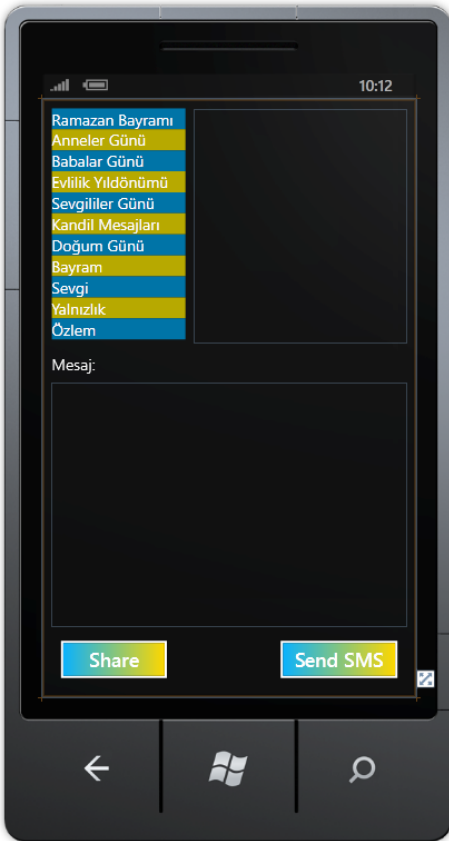
Daha önceden de dediğim gibi Emulator üzerinde test edildiğinden ötürü Device ID anlamsız bir biçimde karşımıza çıkmaktadır.Ama eminim ki eğer bir Windows Phone 7 aygıtında denerseniz istediğiniz sonucu alacaksınızdır.

Bölüm-12: Örnek Uygulama – Mesaj Servisi

Bu bölümde sizlerle “Mesaj Servisi” adlı örnek bir uygulama geliştireceğiz. Bu uygulama esasında arkadaşlarınıza, sevdiklerinize gerek SMS olarak gerekse de Sosyal Medya hesaplarınızı kullanarak güzel mesajlar paylaşmanıza olanak sağlar.

Uygulamayı geliştirmeye başlayalım o halde. Windows Phone 7’de yeni bir proje oluşturun.

Uygulamamızın bu şekilde bir tasarıma sahip olduğunu varsayalım:



Uygulama genel itibariyle en üst sağ köşedeki Listbox1 den kategori seçmemizi sağlayacak ardından bu kategorilere ait veriler hemen yanında bulunan Listbox2 de görüntülenecek. Buna ilaveten Listbox2 de seçilen mesajlar Mesaj adlı bölümde geniş bir şekilde gösterilecektir. En son olarak da ilgili mesajı sevdiklerinize göndermenize ve paylaşmanıza olanak sağlayacaktır. Bu uygulamanın çalışma mantığı bu şekildedir.

XAML hali:

```
<phone:PhoneApplicationPage
  x:Class="MesajServisi.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:phone="clr-
namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone"
  xmlns:shell="clr-
namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
  mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768"
  FontFamily="{StaticResource PhoneFontFamilyNormal}"
  FontSize="{StaticResource PhoneFontSizeNormal}"
  Foreground="{StaticResource PhoneForegroundBrush}"
  SupportedOrientations="Portrait" Orientation="Portrait"
  shell:SystemTray.IsVisible="True">

  <Grid x:Name="LayoutRoot" Background="Transparent">
    <ListBox Height="301" HorizontalAlignment="Left"
Margin="12,12,0,0" Name="listBox1" VerticalAlignment="Top"
Width="172">
      <ListBoxItem Content="Ramazan Bayramı"
Background="#FF0074A7"></ListBoxItem>
      <ListBoxItem Content="Anneler Günü"
Background="#FFB7A900"></ListBoxItem>
```

```
<ListBoxItem Content="Babalar Günü" Background="#FF0074A7"
/>
<ListBoxItem Content="Evlilik Yıldönümü"
Background="#FFB7A900" />
<ListBoxItem Content="Sevgililer Günü"
Background="#FF0074A7" />
<ListBoxItem Content="Kandil Mesajları"
Background="#FFB7A900" />
<ListBoxItem Content="Doğum Günü"
Background="#FF0074A7" />
<ListBoxItem Content="Bayram" Background="#FFB7A900" />
<ListBoxItem Content="Sevgi" Background="#FF0074A7" />
<ListBoxItem Content="Yalnızlık" Background="#FFB7A900" />
<ListBoxItem Content="Özlem" Background="#FF0074A7" />
</ListBox>
<Button Content="Send SMS" Height="72"
HorizontalAlignment="Left" Margin="294,684,0,0" Name="button1"
VerticalAlignment="Top" Width="174">
<Button.Background>
<LinearGradientBrush EndPoint="1,0.5" StartPoint="0,0.5">
<GradientStop Color="#FF0AB3FF" Offset="0" />
<GradientStop Color="#FFFFD800" Offset="1" />
</LinearGradientBrush>
</Button.Background>
</Button>
<Button Content="Share" Height="72" HorizontalAlignment="Left"
Margin="12,684,0,0" Name="button2" VerticalAlignment="Top"
Width="160">
<Button.Background>
<LinearGradientBrush EndPoint="1,0.5" StartPoint="0,0.5">
<GradientStop Color="#FF0AB3FF" Offset="0" />
<GradientStop Color="#FFFFD800" Offset="1" />
</LinearGradientBrush>
</Button.Background>
</Button>
```

```
<ListBox Height="301" HorizontalAlignment="Left"
Margin="194,12,0,0" Name="listBox2" VerticalAlignment="Top"
Width="274" />
  <TextBlock Height="315" HorizontalAlignment="Left"
Margin="12,363,0,0" Name="textBlock1" Text=""
VerticalAlignment="Top" Width="456" TextWrapping="Wrap" />
  <TextBlock Height="30" HorizontalAlignment="Left"
Margin="12,327,0,0" Name="textBlock2" Text="Mesaj:"
VerticalAlignment="Top" />
  </Grid>

</phone:PhoneApplicationPage>
```

Her bir kategori için 5'er tane mesaj eklememiz gerekiyor.Dolayısıyla bu işlem için static class kullanmayı - kolayıma gittiğinden :-) - uygun gördüm:

```
using System;  
using System.Net;  
using System.Windows;  
using System.Windows.Controls;  
using System.Windows.Documents;  
using System.Windows.Ink;  
using System.Windows.Input;  
using System.Windows.Media;  
using System.Windows.Media.Animation;  
using System.Windows.Shapes;  
using System.Collections.Generic;
```

```
namespace MesajServisi
```

```
{
```

```
class Mesajlar
```

```
{
```

```
public static List<string> ramazan = new List<string>
```

```
{
```

```
    "Benim ömrümde ırmaklar vardır sularında hayallerimi  
yüzdürdüğüm, benim ömrümde sevdiklerim vardır bayramlar ayrı  
geçince üzüldüğüm. Bayramınız mübarek olsun!",
```

```
    "Kainatın yaratıcısı ve alemlerin Rabbi yüce Allah'a sonsuz  
şükürler olsun! Ramazan Bayramı bereketiyle, bolluğuyla gelsin, tüm  
insanlık için hayırlara vesile olsun",
```

```
    "Heyecan ve özlemle beklenen kutsal Ramazan Bayramı geldi,  
hoşgeldin. Mübarek bayram ulusumuza sağlık, huzur, mutluluk, bolluk ve  
bereket getirsin. Hayırlı bayramlar dileğiyle.",
```

```
    "Sema kapılarının açık olduğu bugünde heybenizde tohum  
tohum dua menekşeleri saçmanız temennisiyle hayırlı bayramlar.",
```

"Bugün Bayram! Mübarek Ramazan Bayramı. Tüm inananlar birbirlerine daha çok yakınlaşsın, dargınlıklar ortadan kalksın, kardeşlik ve dostluk duyguları daha da kuvvetlensin. Tüm insanlar neşe ve mutluluk denizinde yüzsün. Bugün sevinç günü, kederleri bir yana bırakıp mutlu olalım. Ramazan Bayramı.nı doya doya yaşayalım. Hayırlı bayramlar! Her şeye kadir olan Yüce Allah, bizleri, doğru yoldan ve sevdiklerimizden ayırmasın! Hayırlı ve bereketli Ramazan Bayramları dileğiyle."

};

```
public static List<string> anneler = new List<string>  
{
```

"Her zaman senin karşında masum ve sevgine muhtaç bir çocuk ruhuyla dururum. Çünkü sen benim annemsin. Beni benden çok tanıyorsun, bilensin. Bana sarıldığın zaman tüm dertlerimi yok edensin. Anneler günün kutlu olsun ey aziz kadın, annem!",

"Sen evimizin kraliçesi, başımızın tacısın.. en aziz varlığımız. Anneler günün kutlu olsun annem.",

"Her şeye değer senin sonsuz sevgin.. Annem annem.. Seni çok arıyorum.. Çok özledim. Anneler günün kutlu olsun biricik meleğim.",

"Annem senin sevgin dünyamı ısıtan tek güneştir. Hiç ışığın eksilmeyecek biliyorum. Varlığınla mutluyum. Anneler günün kutlu olsun sevgili annem.",

"Sınırsız bir sevgi, anlatılmaz bir sevgiyle beni seven annem, sana layık olmak için yaşıyorum. Anneler günün kutlu olsun."

};

```
public static List<string> babalar = new List<string>  
{
```

"Çok sevdiğim canım babacığma kucak dolusu sevgilerimi sunar, Babalar gününü kutlarım.",

"Sen güllere özenme güller sana özensin. Üzme tatlı canını sen güllerden güzelsin. Özgürlük kadar özel güller kadar güzelsin. BENİM GÜZEL BABACIĞIM",

"Üzerine sabır tohumu ekip sevgiyle suladığın gülünün bilmeni istediği bir şey var. SENİ ÇOK SEVİYORUM CANIM BABACIĞIM.",

"ATESİN YAKTIGINDAN, GUNESİN HAREKET ETTIGINDEN, GERCEĞİN BİR YALAN OLDUGUNDAN KUSKULAN FAKAT BENİM SENİ

SEVDIGIMDEN ASLA KUSKULANMA BABACIGIM, BABALAR GUNUN KUTLU OLSUN.",

"Babacığım seni çok seviyorum. İyi ki varsın, iyi ki yanımdasın, iyi ki babamsın. Babalar gunun kutlu olsun..."

};

```
public static List<string> evlilik = new List<string>
```

```
{
```

"öyle bir anda girdinki hayatıma tam her şey karardı derken ışığım oldun, öyle çok sevdimki seni tam hayatın uçurumuna geldim derken tek dayanağım sen oldun sevgilim seni çok seviyorum seni herşeyden ve herkesten çok seviyorum ve artık hayatımın geri kalanını senile geçirmek istiyorum şimdi sana soruyorum gökyüzündeki ebedi yıldızım BENİMLE EVLENİR MİSİN?",

"Sarı giyer güneş olursun, Mavi giyer deniz olursun, Siyah giyer matem olursun, Belki birgün beyaz giyer benim olursun",

"benim biricik sevgilim hayatımın anlamı evimin kadını kalbimin yarısı olmaya ve benimle evlenmeye evet der misin",

"Geceme güneş olsan, Kalbimi görmüş olsan, Unuturdum cihanı, Bana bir gün eş olsan.",

"Bir Kızın En Güzel Verebileceği Cegiz Öpülmemis Dudaklarıdır"

```
};
```

```
}
```

```
}
```

Her bir kategoriye string argüman alan Liste olarak oluşturduk ve mesajları bu listelere ekledik.

Şimdi gelelim kontrollere ait olaylara.

Listbox1 yani Kategoriler değiştiğinde bu listelere ait verilerin hemen Listbox2 ye eklenmesini istiyoruz. Dolayısıyla bu durumda şöyle bir kod yazmamız makul olacaktır:

```
private void listBox1_SelectionChanged(object sender,
SelectionChangedEventArgs e)
```

```
{
    listBox2.Items.Clear();
    if (listBox1.SelectedIndex == 0)
    {
        foreach (string c in Mesajlar.ramazan)
        {
            listBox2.Items.Add(c);
        }
    }

    else if (listBox1.SelectedIndex == 1)
    {
        foreach (string c in Mesajlar.anneler)
        {
            listBox2.Items.Add(c);
        }
    }

    else if (listBox1.SelectedIndex == 2)
    {
        foreach (string c in Mesajlar.babalar)
        {
            listBox2.Items.Add(c);
        }
    }
}
```

```
else if (listBox1.SelectedIndex == 3)
{
    foreach (string c in Mesajlar.evlilik)
    {
        listBox2.Items.Add(c);
    }
}
}
```


Şimdi ise Listbox2 yani mesajlardan birisi seçildiğinde mesaj alanine yazdıracak kodu ekleyiyoruz:

```
private void listBox2_SelectionChanged(object sender,
SelectionChangedEventArgs e)
{
    try
    {
        textBlock1.Text = listBox2.SelectedItem.ToString();
    }

    catch (Exception ex)
    {

    }

}
```

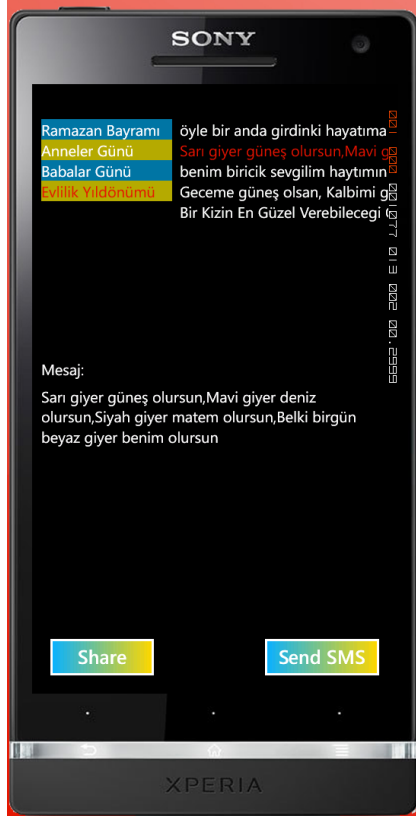
Son olarak da SMS olarak gönderme ve Paylaşma kodlarını ekleyelim:

```
private void button2_Click(object sender, RoutedEventArgs e)
{
    //Paylaşma
    try
    {
        ShareStatusTask st = new ShareStatusTask();
        st.Status = textBlock1.Text;
        st.Show();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata! Kategori ve Mesajı seçmelisiniz");
    }
}

private void button1_Click(object sender, RoutedEventArgs e)
{
    //SMS Gönderme
    try
    {
        SmsComposeTask sct = new SmsComposeTask();
        sct.Body = textBlock1.Text;
        sct.Show();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata! Kategori ve Mesajı seçmelisiniz");
    }
}
```

Buraya kadar uygulamamız tamamlandı.

Nasıl olmuş bir bakalım:



Uygulamamız tüm işlevselliği ile karşınızda.

Şimdi bu uygulamamızı Marketplace için hazırlayalım.

Bölüm-13: Uygulamanızın Marketplace için Hazırlanması

Bir önceki bölümde hep beraber basit bir WP7 uygulaması geliştirdik.

Bu bölümde sizlere bu uygulamayı Marketplace e nasıl hazırlayacaksınız onu göstereceğim.

İhtiyacımız olanlar:

- 1-Uygulama Iconları**
- 2-Uygulamanın Debug Mode'dan çıkarılması**
- 3-Uygulamanın yerleştirme değerinin güncellenmesi**
- 4-Light-Dark Tema Uyumluluğu**
- 5-Uygulamanın Local Olarak Test Edilmesi**
- 6-Uygulamanın Device üzerinde Deploy edilerek test edilmesi**

1-Uygulama Iconları:

Uygulamanızı geliřtirdiniz ve Marketplace e gndereceksiniz.Lakin ilk once sahip olmanız gereken uygulamaya ait iconlardır.

Bu iřlemi kolayca yerine getirmek iin yardımcı bir ara mevcuttur:

WP Icon Maker <http://wpiconmaker.codeplex.com/>

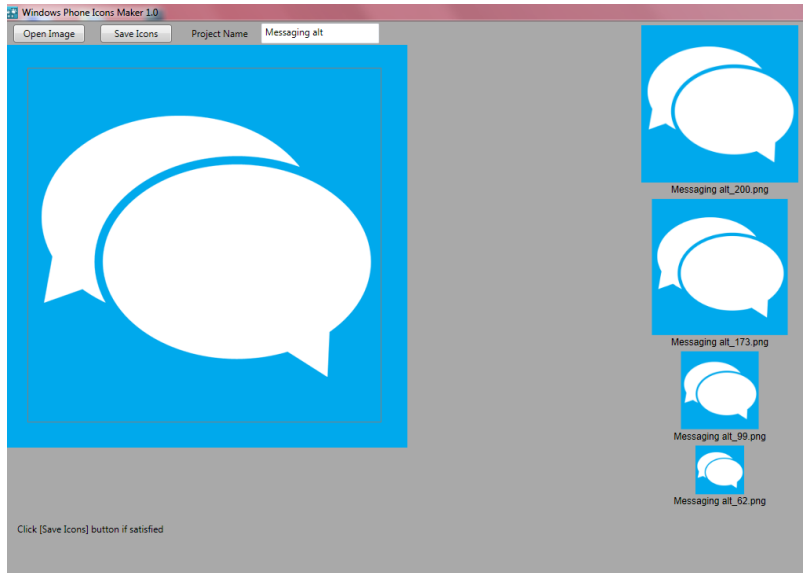
Bu uygulama vasıtasıyla aynı grntye sahip iconlar oluřturabilirsiniz ki bu testi gemek iin gereklidir – tm uygulama iconları aynı grntye sahip olmalıdır –

Kullanımı ok basittir:

1-Uygulamayı aın

2-Bir resim dosyasını ieri taşıyın ya da uygulama ierisinden aın.(farketmez)

3-Daha sonra resimde bir alan sein

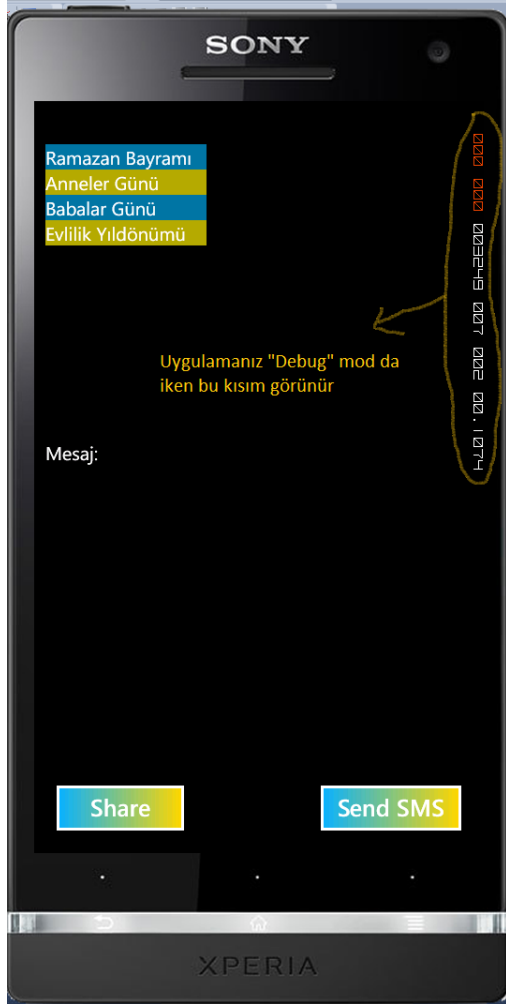


“Save Icons” a basınca Uygulama iconlarınız oluřturulacaktır.

řimdi bu iconları projenize dahil edin ve daha nceden varolan iconlar ile deęiřtirilmesini saęlayın.

2-Uygulamanın Debug Mod'dan çıkarılması

Uygulamanızı test ederken kaç FPS kullandığını görebilirsiniz.



Debug Mod'dan çıkmak için App.xaml.cs dosyasından aşağıdaki kodu kaldırınız:

```
Application.Current.Host.Settings.EnableFrameRateCounter = true;
```

Bir dahaki Build ettiğinizde uygulamanızda FPS görüntülenmeyecektir.

3-Uygulamanın yerelleştirme deęerinin güncellenmesi

Bu işlemleri gerçekleştirmek için AssemblyInfo.cs dosyasını açarak;

```
[assembly: NeutralResourcesLanguageAttribute("en-US")]
```

deęerini

```
[assembly: NeutralResourcesLanguageAttribute("en")]
```

olarak deęiştiriniz.

4-Light-Dark Tema Uyumluluđu

Bu ciddi bir sorundur.Ve uygulamanız eđer bu uyumluluđu sađlamıyorsa testi geemez

Normalde uygulamamızın Light-Dark tema ile sorunu yoktur.Ancak,eđer uygulamanızda Menü işlemleri için ApplicationIcon kullandıysanız,

Arka planı Transparan ve icon rengi beyaz olan iconlar kullanmanız size Light-Dark tema uyumluluđu sađlayacaktır.

Örnek Uyumlu iconlar için daha önceden hazırladıđım 150 adet icon a göz atabilirsiniz:

<http://iersoy.com/post/2012/06/27/150-Theme-Aware-Icons-WP7.aspx>

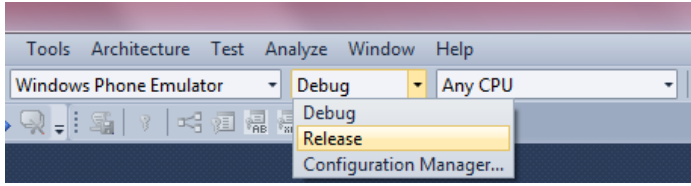
Eđer ki uygulamanızdaki kontrollerde Transparan efekti kullanıldıysa bazı kontroller Light temaya geiş yaptıđında görünmeyebilir.Bu durumda kontrollere ya arkaplan rengi ya da imaj giydirmek mantıklı bir çözüml olacaktır.

5-Uygulamanın Local Olarak Test Edilmesi

Uygulamanızı yerel bilgisayarınızda test etmek isterseniz,Marketplace Test Kit bu konuda size yardımcı olabilir.

Not: Marketplace Test Kit sadece Windows Phone 7.1 projesi ile çalışır.

Marketplace Test Kit i açmak için öncelikle "Release" moda getirin:



Ardından projenize "Clean" komutunu vererek oluşturulan xap dosyalarını kaldırmasını söyleyin.

Sonra Release mod da iken projenizi Build edin.Bu işlemde hemen sonra projenize sağ tıklayıp "Open Marketplace Test Kit" deyin.

Burada sizden birkaç dosyayı eklemeniz istenecek:

- 173X173 ebatında icon
- 99X99 ebatında icon
- 200X200 ebatında icon
- Maksimum 8 adet Ekran görüntüsü

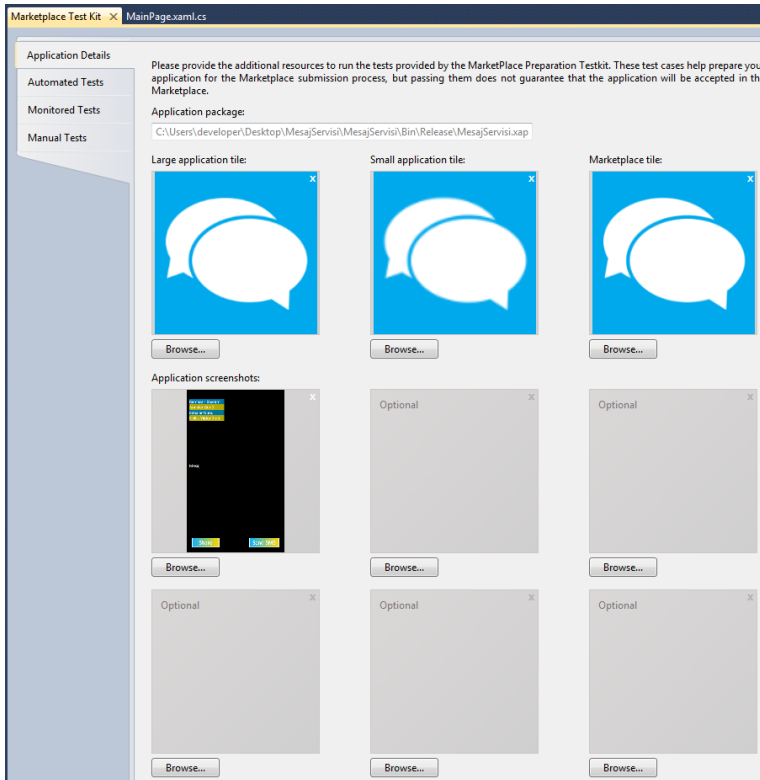
icon dosyaları zaten daha önceden WP Icon Maker ile oluşturulan dosyalar idi.

Ekran görüntüsü almak için uygulamayı çalıştırınız.Ardından;

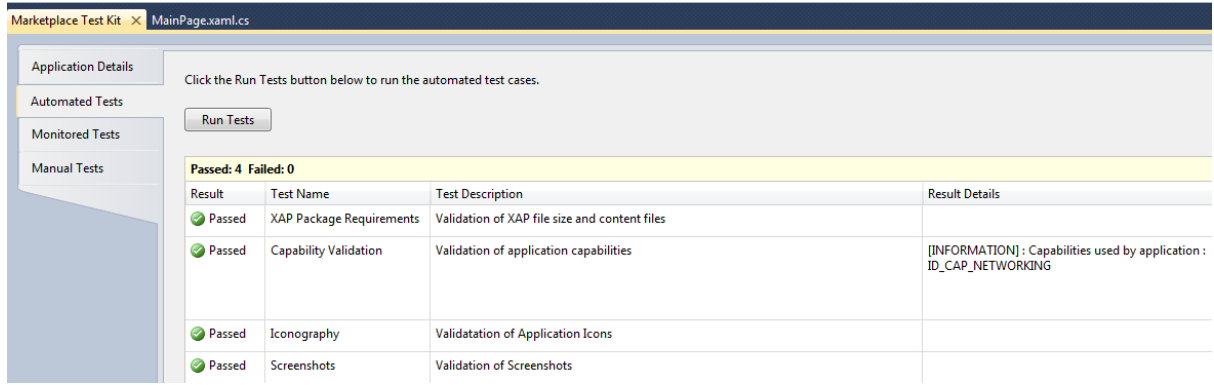


Emulator ile birlikte çalışan araç setinde bulunan ">>" şekilli simgeye tıklayın.Buradan Additional Tools penceresi çıkacaktır.Screenshot sekmesine giriş yaptıktan sonra öncelikli olarak "Capture" sonra "Save" düğmelerine tıklayarak ekran görüntüsünü alın.

Daha sonra Marketplace Test Kit e geri dönüp istenilen verileri girin:



En son olarak da Testi çalıştırmak için;



Marketplace Test Kit x MainPage.xaml.cs

Application Details

Automated Tests

Monitored Tests

Manual Tests

Click the Run Tests button below to run the automated test cases.

Run Tests

Passed: 4 Failed: 0

Result	Test Name	Test Description	Result Details
Passed	XAP Package Requirements	Validation of XAP file size and content files	
Passed	Capability Validation	Validation of application capabilities	[INFORMATION] : Capabilities used by application : ID_CAP_NETWORKING
Passed	Iconography	Validation of Application Icons	
Passed	Screenshots	Validation of Screenshots	

Automated Tests e girip "Run Tests" komutunu verin.Eğer testi geçtiyseniz,Marketplace de Test edildiğinde projenizin "Testing Failed" durumuna düşme olasılığını oldukça düşürmüş olursunuz.

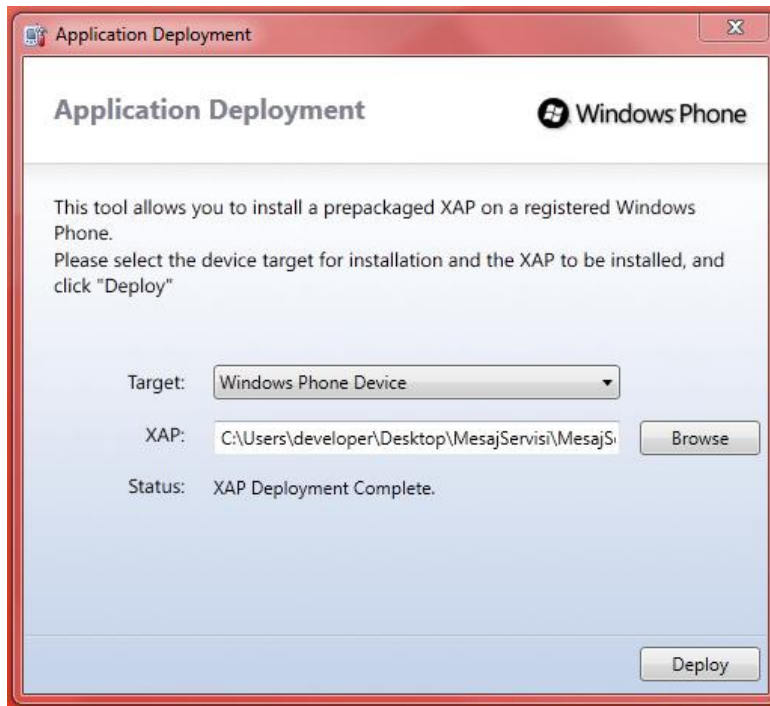
6-Uygulamanın Device üzerinde Deploy edilerek test edilmesi

Bu işlem aslında WP7 Unlocked aygıtı olanlar içindir ve opsiyoneldir.Lakin Marketplace de yayınlanmasını bekleyemeyecek durumda iseniz veya sevdiklerinize,arkadaşlarınıza veya patronunuza yazdığınız uygulamayı göstermek isterseniz,bu yöntemi uygulamaktan kaçınmayın derim şahsen.

Aygıtınızı bilgisayara bağlayın,eğer ilk defa bağlayacaksanız en son ZUNE uygulamasının yüklendiğinden emin olun.

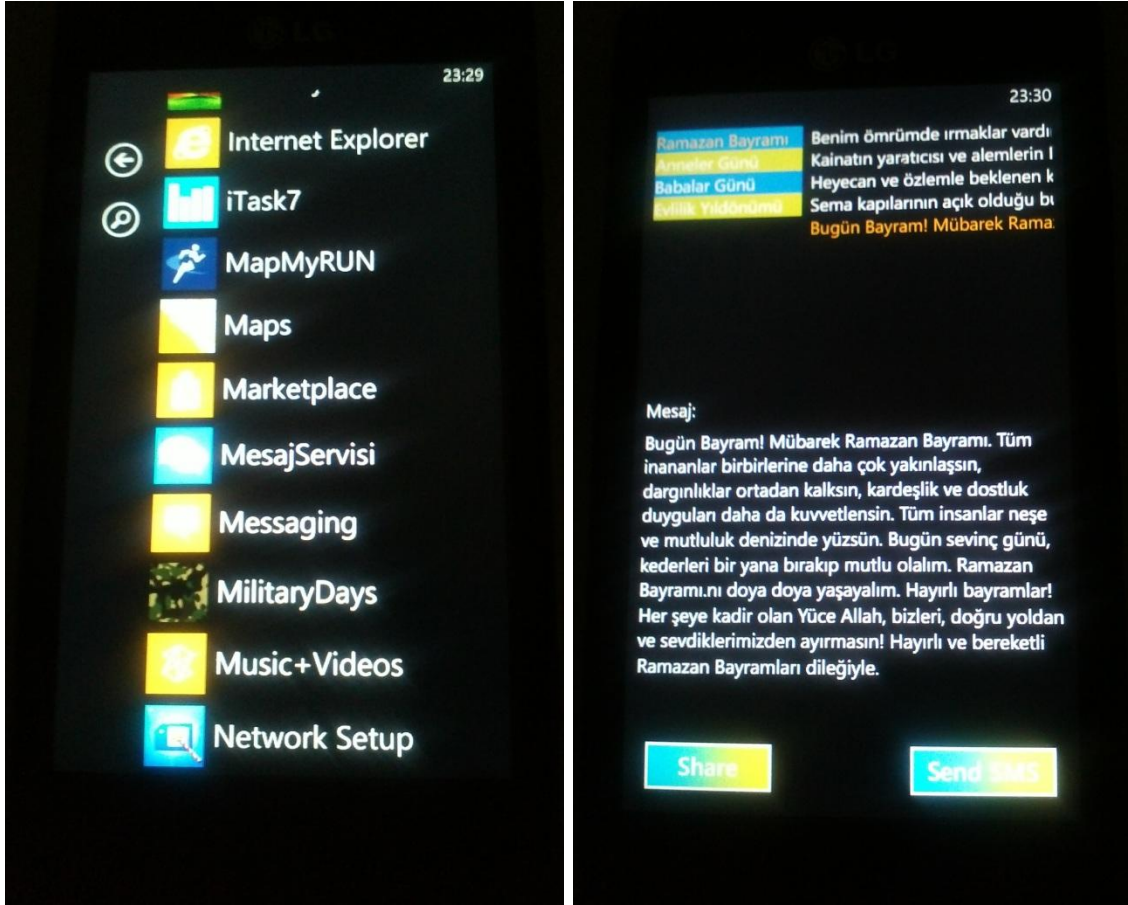
Yüklemediyseniz yükleyin: <http://www.zune.net/en-US/products/software/download/default.htm>

Windows Phone 7 SDK ile birlikte dağıtılan Application Deployment uygulamasını çalıştırın:



Target kısmında "Windows Phone Device" ı seçin, XAP kısmına ise "Release" klasöründeki xap dosyanızı ekleyin.

Ardından uygulamanızı Deploy edin.Başarılı bir şekilde WP7 aygıtınıza yüklenecektir:



Uygulamaları gerek Deploy ederken gerekse de Marketplace e gönderirken daima "bin->Release" klasörü altında bulunan xap dosyası seçilir.Bu konuya dikkat edilmelidir.Aksi takdirde testi geçemez.

"error: 0x89740006 " diye bir hata alırsanız; uygulamanızı Ekran kilitli iken Deploy ettiğinizden bu hatayı alırsınız.Ekran kilidini açarak tekrar Deploy ettiğinizde uygulamanızı artık telefonunuzda görebilirsiniz.

"error: 0x89731812" diye bir hata alırsanız şayet uygulamanız ya ZUNE üzerinden senkronize olmamıştır ya da halen daha kilitli ekrandadır.

İletişim

iersoy.com adlı kişisel blogumda paylaştığım üzere Facebook, Twitter veya Google+ üzerinden bana ulaşabilirsiniz.

Kaynaklar

Windows Mobile Tarihçesi için:

http://en.wikipedia.org/wiki/Windows_Mobile

Device ID Öğrenme: <http://www.nickharris.net/2010/09/windows-phone-7-how-to-find-the-device-unique-id-windows-live-anonymous-id-and-manufacturer/>

WP Icon Maker <http://wpiconmaker.codeplex.com/>

DeviceExtendedProperties [http://msdn.microsoft.com/en-us/library/ff941122\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/ff941122(v=VS.92).aspx)

Son Söz

Okuduđunuz için çok teŖekkürler!