

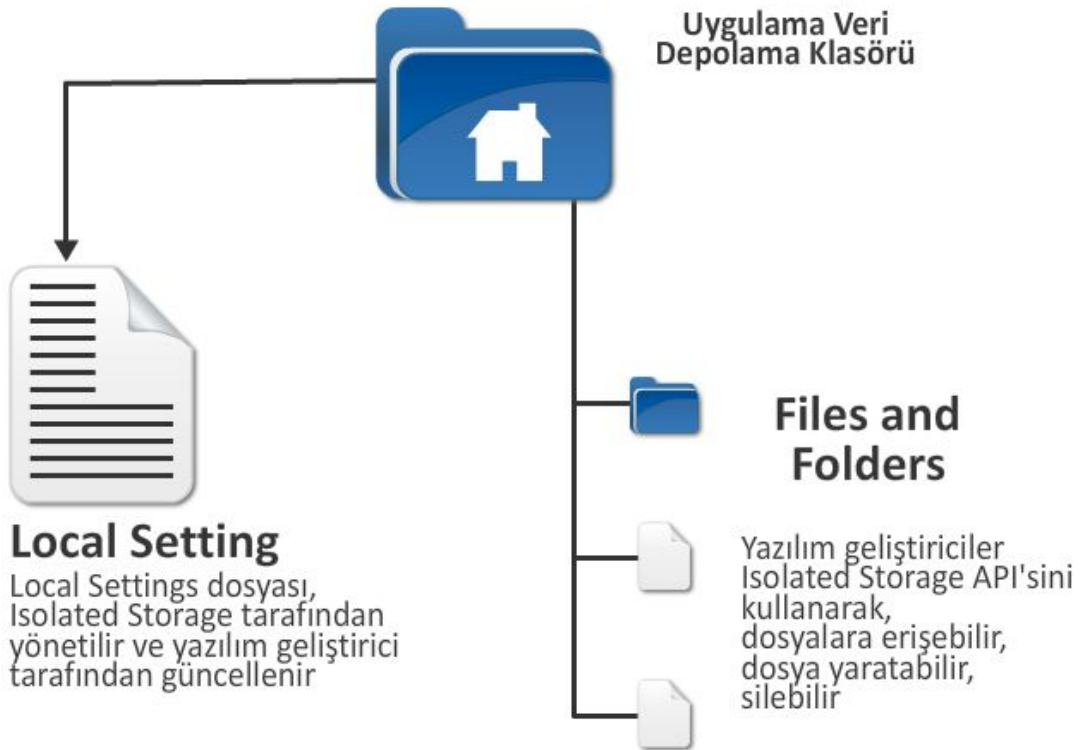
Isolated Storage, izole edilmiş depolama anlamına gelir ve Windows Phone 8 uygulamasının kalıcı ya da geçici olarak saklaması istenen verileri için kullanılabilecek alanları sağlar.

Isolated Storage'ın en sık kullanımı Ölüm (Tombstoning) modu durumunda, uygulamadaki verilerin kaydedilmesidir. Ölüm moduna giren bir uygulamaya ait durum (State) bilgileri tutulmayacağından, bunları yerel bir konumda kaydedip (IsolatedStorage gibi) uygulama tekrar açıldığında yüklemek sağlıklı bir çözüm olacaktır. Ölüm (Tombstoning) Modu, "Windows Phone 8 Uygulama Yaşam Döngüsü" bölümünde ayrıntılı olarak incelenmiştir.

Isolated Storage, Windows Phone 8 ile ortaya çıkmış bir kavram değildir. Silverlight 2'den beri kullanılmaktadır ve en temel anlamda veri ya da dosyaların yerel dosya sisteminde saklanması bir yoldur. İsminde Isolated (izole edilmiş) geçer, çünkü içindeki veriye yalnızca ilgili uygulama erişebilir. Bu yüzden, eğer iki uygulama arasında veri paylaşmak isteniyorsa, Isolated Storage kullanılamayacaktır. Bu gibi istekler için farklı veri kaynakları (Windows Azure, MS SQL Server vb.) kullanmak gerekir. Aynı cihazdaki farklı uygulamalar birbirleriyle etkileşime geçemez ve birbirleri ile paylaşımda bulunamaz.

Veriyi yerel olarak saklamanın 2 yöntemi vardır. İlk yöntem **IsolatedStorageSettings**, ikinci yöntem **IsolatedStorageFile** olarak adlandırılır.

Mantıksal Klasör Yapısı



İlk olarak **IsolatedStorageSettings**'i inceleyelim. Birçok durumda, ihtiyacı karşılayacak olan yöntem bu olacaktır. Bu, bir **Dictionary** (sözlük) içinde **Key-Value** (anahtar-değer) ikilisi şeklinde veri saklama yöntemidir. Uygulama başlatılıp durdurulsa da, cihaz kapatılıp açılrsa da burada saklanan veri –silininceye ya da uygulama cihazdan kaldırılınca kadar– kalıcı olacaktır. Aşağıda basit bir örnek verilmiştir:

```
using System;

using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Shapes;
using Microsoft.Phone.Controls;
using System.IO.IsolatedStorage;

namespace Windows8IsolatedStorage
{
    public partial class MainPage : PhoneApplicationPage
    {
        IsolatedStorageSettings ayarlar = IsolatedStorageSettings.ApplicationSettings;

        public MainPage()
        {
            InitializeComponent();
            AyarlariCek ();
        }

        private void AyarlariCek()
        {
            //Koleksiyon içerisinde ilgili Key'in olup olmadığı kontrol ediliyor
            if (ayarlar.Contains("kayitliAyar"))
```

```
{
    string veri = ayarlar["kayitliAyar"].ToString();
}
else
{
    ayarlar.Add("kayitliAyar", "Merhaba");
}
}
```

Yukarıdaki örnekte, **IsolatedStorageSettings.ApplicationSettings** ile uygulamanın ayarları **ayarlar** ismindeki nesneye eşitlenir.

ApplicationSettings içine **Key-Value** ikilisi eklemek için **Add** metodu kullanılır. **Add** metodu ilk parametre olarak **Key**, ikinci parametre olarak **Value** değerini alır. Tahmin edileceği gibi, koleksiyon içerisine aynı **Key** yalnızca bir defa eklenebilir. Aynı **Value** bir kereden fazla eklenebilir. Eklenmiş olan bir Key-Value ikilisinin **Value** değerini çekmek için, koleksiyon içinden ilgili **Key** ismi verilir.

Ayarlar ismindeki nesneden **indexer** yardımı ile **key** ismi verilerek ilgili **key**'e ait değer, **veri** isimli string değişkene eşitlenir. Bu, Key vererek Value çekme işlemidir.

IsolatedStorageSettings kullanımı kolay bir yöntemdir. Key-Value ikilisi olarak veri az kod ile saklanabilir ve çekilebilir.

Şimdi de diğer yöntem olan **IsolatedStorageFile**'i inceleyelim. IsolatedStorageFile, IsolatedStorageSettings'in aksine, yerel bir dosya üzerinde verileri saklamanıza olanak sağlayan bir yapıdır. IsolatedStorageSettings kullandığımızda uygulamaya ait ayarlarda yerel değerler saklayabilirsiniz. IsolatedStorageFile'da istediğiniz türde dosya oluşturup içeriğine hızlıca erişebilirsiniz. Küçük bir örnek ile bunun nasıl yapılacağına değinelim:

```
using System.IO.IsolatedStorage;
```

```
using System.IO;
```

```
private void btnKaydet_Click(object sender, RoutedEventArgs e)
```

```
{
```

```
    IsolatedStorageFile yeniDepo = IsolatedStorageFile.GetUserStoreForApplication();
```

```
    yeniDepo.CreateDirectory("metinDosyaları");
```

```
    StreamWriter yazici = new StreamWriter(new
```

```
IsolatedStorageFileStream("metinDosyaları\\yeniDosya.txt", FileMode.OpenOrCreate, yeniDepo));
```

```
    yazici.WriteLine(TextBox1.Text);
```

```
    yazici.Close();
```

```
}
```

```
private void btnGoster_Click(object sender, RoutedEventArgs e)
```

```
{
```

```
    IsolatedStorageFile kayitliDepo = IsolatedStorageFile.GetUserStoreForApplication();
```

```
    StreamReader okuyucu = new StreamReader(new
```

```
IsolatedStorageFileStream("metinDosyaları\\yeniDosya.txt", FileMode.Open, kayitliDepo));
```

```
    string veri = okuyucu.ReadLine();
```

```
    Label1.Text = veri;
```

```
    okuyucu.Close();
```

```
}
```

Bu örnekte IsolatedStorage içerisinde .txt uzantılı bir dosya oluşturulur. Yeni bir dosya oluşturmak için IsolatedStorageFile class'ının **GetUserStoreForApplication** metodu kullanılarak depolama için alan elde edilir ve *yeniDepo* isimli nesneye eşitlenir. *yeniDepo* nesnesinde **CreateDirectory** metodu ile *metinDosyolari* adlı bir klasör oluşturulur. Sonra **StreamWriter** ile oluşturduğunuz klasör içine yeniDosya.txt isminde bir metin dosyası eklenerek, içine TextBox1'in Text özelliğindeki veri yazılır. Yazılmış olan yeniDosya.txt isimli dosyanın içindeki veriyi okumak için veri yazarken kullanılmış olan **GetUserStoreForApplication** metodu ile depolama alanı elde edilir ve kayitliDepo isimli nesneye eşitlenir. Yazma işlemi yapılan dosyanın adresi **StreamReader**'a parametre olarak verilerek bu dosya elde edilir. StreamReader'ın **ReadLine** metodu ile dosya içindeki ilk satır okunarak veri ismindeki string değişkene eşitlenir. Dosyanın tamamının okunması için StreamReader'ın **ReadToEnd** metodu kullanılabilir.