Silverlight 4

پژوهش و نگارش: وحید نصیری
Vahid_Nasiri@Yahoo.Com
بهار ۱۳۸۹

فهرست مطالب

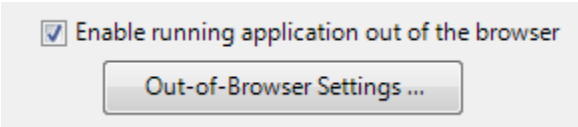
فصل ۲۰ – بررسی اجرای خارج از مرورگر برنامه‌های Silverlight

مقدمه

در این فصل قصد داریم با ریز تنظیمات برنامه‌ها و پروژه‌هایی از Silverlight که قرار است در خارج از مرورگر اجرا شوند، آشنا شویم. به این نوع برنامه‌ها اصطلاحا Out of browser applications و یا به صورت مخفف OOB نیز گفته می‌شود. این ویژگی، امکان تهیه‌ی برنامه‌های Desktop قابل اجرای بر روی چندین سیستم عامل مختلف مانند ویندوز، Mac و Linux را فراهم می‌سازد و از نگارش سوم این مجموعه به آن اضافه شده است. علاوه بر آن یک سری از ویژگی‌های جدید Silverlight 4 نیز صرفا منحصر به این حالت اجرا می‌باشند.

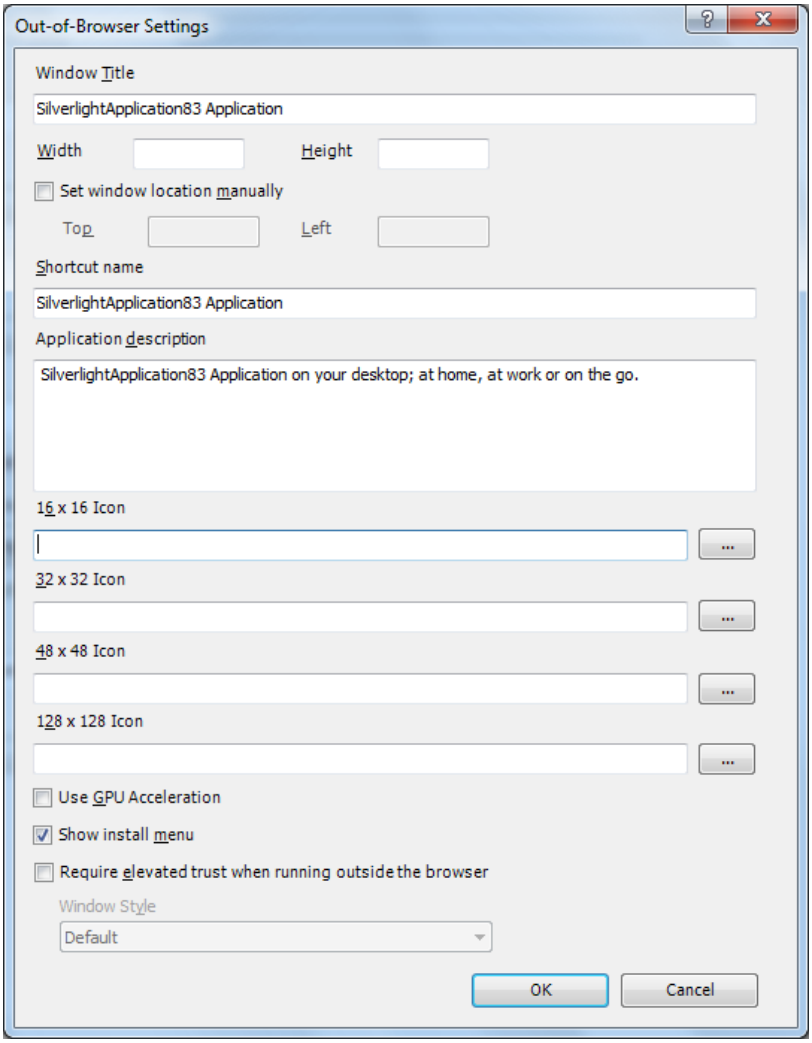
تنظیمات لازم جهت اجرای یک پروژه‌ی Silverlight در خارج از مرورگر

برای این منظور ابتدا به خواص پروژه‌ی جاری در VS.NET مراجعه کرده و به برگه‌ی Silverlight آن مراجعه کنید. سپس گزینه‌ی فعال سازی اجرای برنامه در خارج از مرورگر را انتخاب نمائید (شکل ۱).



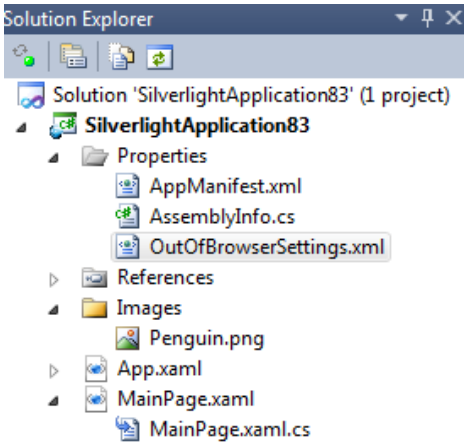
شکل ۱- فعال سازی قابلیت اجرای برنامه در خارج از مرورگر

علاوه بر آن در این حالت اگر بر روی دکمه‌ی Out-Of-Browser Settings... کلیک نمائید، همانند شکل ۲ امکان تنظیم عنوان پنجره‌ی اصلی برنامه، طول و عرض آن، آیکون برنامه (حتما باید از نوع PNG باشد)، نام میانبر آن و توضیحاتی در مورد عملکرد برنامه ، وجود دارد.



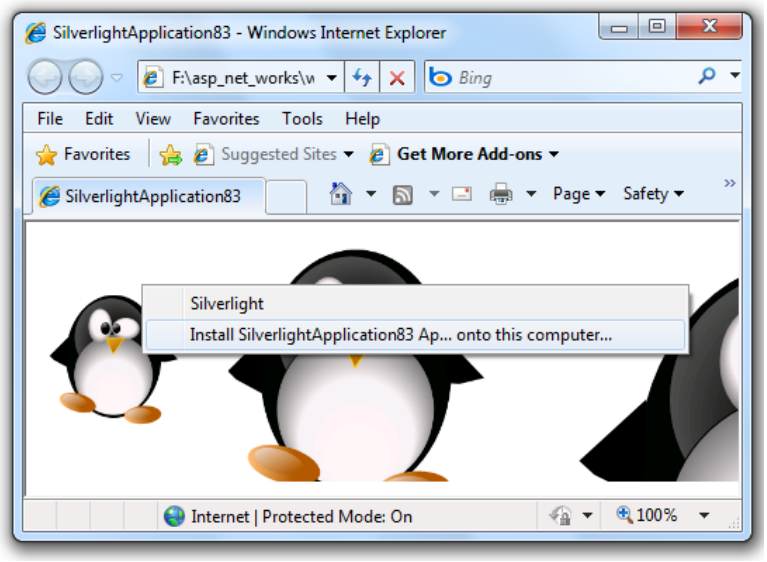
شکل ۲– تنظیمات مخصوص یک برنامه‌ی قابل اجرا در خارج از مرورگر

این تنظیمات در فایلی به نام Properties\OutOfBrowserSettings.xml ذخیره خواهند شد (شکل ۳) و به سادگی قابل ویرایش هستند.



شکل ۳– محل قرارگیری فایل تنظیمات OOB .

اکنون برنامه را یکبار اجرا نموده و بر روی صفحه کلیک راست نمائید (شکل ۴). از طریق منوی ظاهر شده می‌توان برنامه را پس از اخذ مجوز از کاربر بر روی سیستم نصب نمود.



شکل ۴– نحوه‌ی نصب برنامه‌های OOB .

پس از اجرای برنامه در حالت خارج از مرورگر، اگر کاربر بر روی صفحه‌ی برنامه کلیک راست کند، منوی نمایش داده شده امکان حذف برنامه را نیز مهیا خواهد نمود.

توسط کد نویسی نیز می‌توان نسبت به نصب یک برنامه‌ی Silverlight جهت اجرا در خارج از مرورگر اقدام کرد:

C#

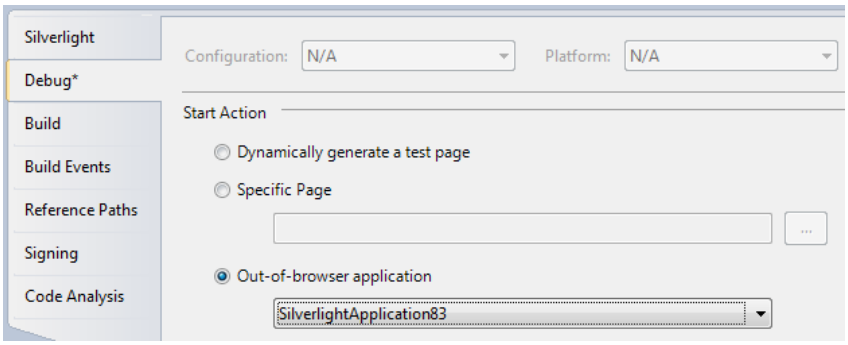
```
private void btnInstall_Click(object sender, RoutedEventArgs e)
{
   if (!Application.Current.IsRunningOutOfBrowser)
```

```
        {
    if (Application.Current.InstallState == InstallState.NotInstalled)
        Application.Current.Install();
  }
}
```

پس از نصب برنامه، امکان دسترسی به آن از طریق میانبر قرار گرفته بر روی Desktop و یا مراجعه به منوی Start → All programs میسر خواهد بود.

نکته

اگر علاقمند هستید که این نوع برنامه‌ها در VS.NET ، از همان ابتدای کار در حالت اجرای در خارج از مرورگر جهت Debug ساده‌تر آغاز شوند، به قسمت خواص پروژه، برگه‌ی Debug آن مراجعه کرده و گزینه‌ی مرتبط را مطابق شکل ۵ انتخاب نمائید.



شکل ۵– آغاز برنامه در حالت خارج از مرورگر جهت Debug در VS.NET

مطلع سازی کاربران از وجود نگارش جدیدی از برنامه

در زمان هر بار اجرای برنامه‌های Silverlight در حالت خارج از مرورگر، برنامه اگر اتصال به شبکه را تشخیص دهد، آخرین نسخه‌ی موجود بر روی Server را بررسی کرده (ETag دریافتی از Server بررسی می‌گردد) و در صورت لزوم نسبت به دریافت و نصب خودکار آن اقدام خواهد شد (auto update). در این حالت نگارش جدید برنامه در اجرای بعدی آن توسط کاربر به صورت خودکار در دسترس او می‌باشد. علاوه بر آن می‌توان این بررسی و دریافت را توسط کد نویسی نیز انجام داد:

C#

```
using System;
using System.Windows;

namespace SilverlightApplication83
```

```
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            if (Application.Current.IsRunningOutOfBrowser)
            {
                Application.Current.CheckAndDownloadUpdateCompleted +=
                    CheckAndDownloadUpdateCompleted;
                Application.Current.CheckAndDownloadUpdateAsync();
            }
        }

        private void CheckAndDownloadUpdateCompleted(object sender,
            CheckAndDownloadUpdateCompletedEventArgs e)
        {
            if (e.Error == null && e.UpdateAvailable)
            {
                MessageBox.Show("An update has been downloaded. " +
                    "Restart the application to run the new version.");
            }
            else if (e.Error != null &&
                e.Error is PlatformNotSupportedException)
            {
                MessageBox.Show("An application update is available, " +
                    "but it requires a new version of Silverlight. " +
                    "Visit the application home page to upgrade.");
            }
            else
            {
                MessageBox.Show("There is no update available.");
            }
        }
    }
}
```

در اینجا توسط متد CheckAndDownloadUpdateAsync وضعیت نگارش موجود بر روی سرور بررسی شده و در صورت لزوم ، به روز رسانی مرتبط، به شکلی غیرهمزمان دریافت می‌گردد. پایان عملیات با بررسی رخداد CheckAndDownloadUpdateCompleted قابل تشخیص است. در روال رخدادگردان آن اگر خاصیت UpdateAvailable نتیجه‌ی کار به true تنظیم شده باشد، تنها کافی است پیغامی را جهت اجرای مجدد برنامه به

کاربر نشان دهیم . اگر در حالت بروز خطایی، این خطا از نوع PlatformNotSupportedException باشد، به این معنا است که کاربر نیاز دارد نگارش جدیدی از افزونه‌ی Silverlight را نصب کند.

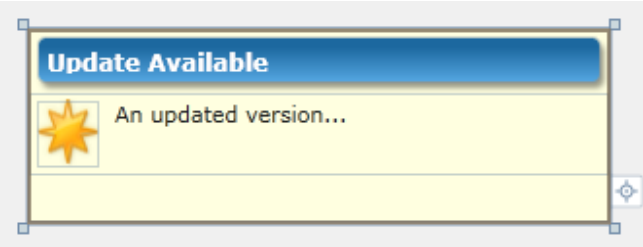
هیچ راهی برای بستن خودکار برنامه و مجبور ساختن کاربر به این کار نیست (البته برنامه‌های OOB با سطح دسترسی بالا (که در ادامه توضیح داده خواهند شد) امکان فراخوانی متد Application.Current.MainWindow.Close() را دارند). بنابراین می‌توان صفحه را غیرفعال کرد و پیغام مناسبی را به او جهت به روز رسانی برنامه نمایش داد (برای مثال استفاده از toast window که در ادامه توضیح داده خواهد شد).

در ادامه به یک سری از ویژگی‌های مختص برنامه‌های قابل اجرا در حالت خارج از مرورگر خواهیم پرداخت که از تازه‌های Silverlight 4 به شمار می‌روند و در حالت اجرای درون مرورگر در دسترس نیستند.

آشنایی با Toasts

قابلیتی به نام Toasts به Silverlight 4 اضافه شده است که عملکرد آن شبیه به نمایش پیغام‌های دریافت ایمیل جدید در برنامه‌ی معروف Outlook است. این پنجره‌ی کوچک که جهت اطلاع رسانی پیغام‌های مهم کاربرد دارد فقط در سمت راست و پایین صفحه نمایش داده خواهد شد (به صورت خودکار و غیر قابل تغییر (به دلایل امنیتی))، حداکثر به اندازه‌ی 400×100 Pixels محدود بوده و دارای حاشیه‌ی خاصی نمی‌باشد. این قابلیت برای برنامه‌های اجرا شده درون مرورگر مهیا نیست.

برای آشنایی با این قابلیت یک برنامه‌ی جدید را آغاز نموده و تنظیمات OOB آن‌را مطابق قسمت‌های قبل انجام دهید. سپس یک User control جدید را به پروژه به نام MyUpdateNotificationWindow.xaml اضافه نمائید (شکل ۶). کدهای XAML این صفحه را در ادامه مشاهده خواهید نمود.



شکل ۶– نمایی از پنجره‌ی اطلاع رسانی در حال طراحی

MyUpdateNotificationWindow.xaml

```
<UserControl x:Class="SilverlightApplication83.MyUpdateNotificationWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Height="100" Width="300">
    <Grid x:Name="LayoutRoot" Background="White">
        <Border x:Name="Frame" Width="300" Height="100"
                        Background="LightYellow">
```

```
            <StackPanel Orientation="Vertical">
                <Border Width="290" Height="24"
                        CornerRadius="4" Margin="2,4,2,4">
                    <Border.Background>
                        <LinearGradientBrush StartPoint="0.5,0.0"
                        EndPoint="0.5,1.0">
                         <GradientStop Offset="0.2" Color="#FF1C68A0" />
                         <GradientStop Offset="1.0" Color="#FF54A7E2" />
                        </LinearGradientBrush>
                    </Border.Background>
                    <Border.Effect>
                        <DropShadowEffect BlurRadius="4" ShadowDepth="4"
                        Opacity="0.4" />
                    </Border.Effect>
                    <TextBlock Text="Update Available" FontSize="12"
                    FontWeight="Bold" Foreground="White" Margin="4" />
                </Border>
                <StackPanel Orientation="Horizontal">
                    <Image Source="Images/burst.png" Width="32"
                       Height="34"
                    Stretch="Fill" Margin="4" VerticalAlignment="Top" />
                    <TextBlock Width="240" Text="An updated version..."
                    FontSize="11" Foreground="#FF202020"
                     TextWrapping="Wrap"
                    Margin="4" />
                </StackPanel>
            </StackPanel>
        </Border>
    </Grid>
</UserControl>
```

در این مثال اگر دقت کرده باشید طول و عرض صفحه دقیقا ذکر شده‌اند؛ زیرا از این مقادیر در کدهای نمایش صفحه‌ی اطلاع رسانی استفاده خواهیم کرد.

اکنون در صفحه‌ی اصلی برنامه برای نمایش آن یک دکمه‌ی ساده را قرار داده و کدهای روال رخدادگردان آن‌را به شرح ذیل تغییر خواهیم داد:

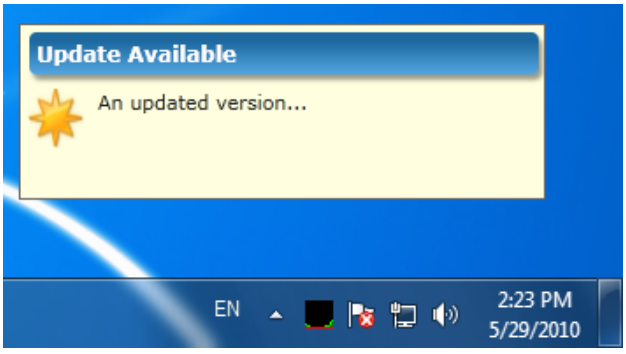
MainPage.xaml.cs

```
private NotificationWindow _notifyWin;
private void btnToasts_Click(object sender, RoutedEventArgs e)
{
     if (!Application.Current.IsRunningOutOfBrowser)
           return;

     if (_notifyWin == null)
         _notifyWin = new NotificationWindow();

     if(_notifyWin.Visibility== Visibility.Visible)
```

```
        {
      _notifyWin.Close();
    }

    var content = new MyUpdateNotificationWindow();
    _notifyWin.Width = content.Width;
    _notifyWin.Height = content.Height;
     _notifyWin.Content = content;
     _notifyWin.Show(5000);
}
```

توضیحات:

ابتدا یک شیء جدید از نوع استاندارد NotificationWindow را جهت نمایش پنجره‌ی اطلاع رسانی ایجاد خواهیم نمود. سپس بررسی خواهیم نمود که آیا برنامه در حالت خارج از مرورگر در حال اجرا است یا خیر. در ادامه اگر برنامه در خارج از مرورگر در حال اجرا بود، یک وهله‌ی جدید از شیء NotificationWindow را در صورت Null بودن متغیر آن ایجاد خواهیم کرد. سپس بررسی خواهیم نمود که آیا هم اکنون همان پنجره در حال نمایش است؟ (علت تعریف این متغیر در سطح کلاس) اگر بلی، این پنجره را بسته و در ادامه یک وهله‌ی جدید از user control حاوی تعاریف رابط کاربری این پنجره را ایجاد خواهیم نمود. طول و عرض این user control به طول و عرض شیء NotificationWindow نسبت داده شده و همچنین خاصیت content آن نیز با محتوای وهله‌ای از این user control مقدار دهی می‌گردد. در آخر به کمک متد Show ، این پنجره را برای مدت زمان معینی (به میلی ثانیه) نمایش خواهیم داد (شکل ۷).



شکل ۷– نمایی از محل نمایش از پیش تعیین شده‌ی NotificationWindow .

معرفی کنترل WebBrowser

کنترل WebBrowser برای نمایش صفحات HTML، درون یک برنامه‌ی Silverlight کاربرد دارد. این امکان نیز در Silverlight 4 تنها جهت برنامه‌هایی که خارج از مرورگر اجرا می‌شوند، مهیا شده است. مثالی از نحوه‌ی تعریف آن را در کدهای XAML یک صفحه و سپس نمایش یک محتوای HTML را در آن، توسط کدهای برنامه در ادامه ملاحظه خواهید نمود:

XAML

```
<WebBrowser x:Name="MyBrowserControl" Width="100" Height="100" />
```

C#

```
MyBrowserControl.NavigateToString(
 "<div style='color:red;width:100;height:100'><b>Test!</b></div>");
```

این کنترل دارای خواص و متدهای ذیل می‌باشد:

- Source : این خاصیت بیانگر آدرس صفحه‌ای است که نمایش داده خواهد شد (و تنها در کدهای برنامه باید تنظیم شود).
- Navigate : متدی است جهت نمایش یک صفحه از طریق کدهای برنامه و معادل با تنظیم خاصیت Source می‌باشد.
- NavigateToString : همانند مثال قبل از آن جهت نمایش یک متن رشته‌ای حاوی عناصر HTML استفاده خواهد شد.

لازم به ذکر است اگر برنامه‌ی OOB ، در حالت دسترسی بالا (که در ادامه توضیح داده خواهد شد) اجرا نگردد، تنها مجاز به نمایش صفحاتی از همان Domain که برنامه اصلی در آن قرار گرفته است، می‌باشید.

معرفی WebBrowserBrush

همزمان با افزودن کنترل WebBrowser به Silverlight 4 ، WebBrowserBrush نیز جهت پوشاندن کنترل‌های UI با محتوای HTML مهیا شده است. لطفا به مثالی در این زمینه دقت بفرمائید:

MainPage.xaml

```
<UserControl x:Class="SilverlightApplication84.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc=
    "http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <StackPanel>
            <WebBrowser Name="demoWB"
                        Width="200" Height="100"
                        Visibility="Collapsed" />
```

```
            <Button Content="New Content" Click="ButtonNewCnt_Click" />
        <Button Content="Refresh" Click="Button_Click" />
        <Rectangle Height="100" Width="200">
            <Rectangle.Fill>
                <WebBrowserBrush SourceName="demoWB"
                        x:Name="demoBrush" />
            </Rectangle.Fill>
        </Rectangle>
    </StackPanel>
</Grid>
</UserControl>
```

باید دقت داشت که اگر کنترل WebBrowser به صفحه‌ی دیگری تنظیم شد، نیاز است تا متد Redraw مرتبط با WebBrowserBrush یکبار فراخوانی گردد تا محتوای آن بر اساس اطلاعات جدید کنترل WebBrowser به روز گردد:

MainPage.xaml.cs

```
using System.Windows;

namespace SilverlightApplication84
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();
            this.Loaded += MainPage_Loaded;
        }

        private void MainPage_Loaded(object sender, RoutedEventArgs e)
        {
            demoWB.NavigateToString(
                "<div style='color:red;width:100;height:100'><b>Test!</b></div>");
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            demoBrush.Redraw();
        }

       private void ButtonNewCnt_Click(object sender, RoutedEventArgs e)
       {
            demoWB.NavigateToString(
                "<div style='color:green;width:100;height:100'><b>Test2!</b></div>");
        }
    }
}
```

در این مثال محتوای کنترل WebBrowser بر روی شیء Rectangle نقاشی شده است. اگر بر روی دکمه‌ی محتوای جدید کلیک نمائید، تاثیر آن‌را مشاهده نخواهید نمود زیرا تنها پس از کلیک بر روی دکمه‌ی Refresh، متد Redraw فراخوانی می‌گردد.

کنترل محل قرارگیری پنجره‌ی اصلی برنامه در خارج از مرورگر

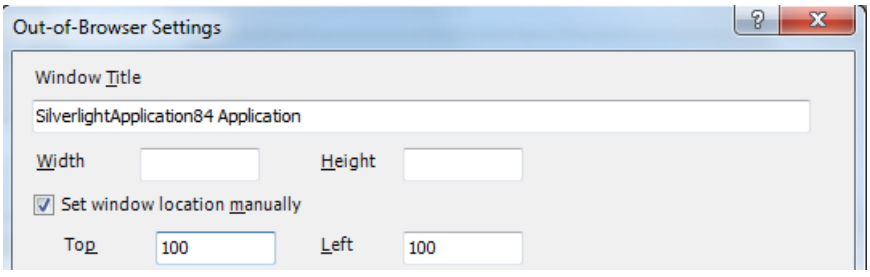
به صورت پیش فرض محل قرارگیری پنجره‌ی اصلی برنامه در خارج از مرورگر به صورت خودکار در وسط صفحه‌ی نمایش تنظیم می‌گردد. اگر علاقمند باشید که این مورد را به صورت دستی تنظیم کنید، دو راه حل برای آن وجود دارد:

۱. مراجعه به خواص پروژه، مراجعه به تنظیمات OOB (شکل ۲) و انتخاب گزینه‌ی تنظیم دستی موقعیت پنجره (شکل ۸).
۲. انجام مرحله یک و سپس کد نویسی در فایل App.xaml.cs به شرح کدهای ذیل.

App.xaml.cs

```
private void Application_Startup(object sender, StartupEventArgs e)
{
    this.RootVisual = new MainPage();
    this.MainWindow.Left = 1;
    this.MainWindow.TopMost = true;
    this.MainWindow.WindowState = WindowState.Normal;
}
```

در این مثال به کمک شیء MainWindow می‌توان موقعیت قرارگیری برنامه در صفحه و همچنین خواص TopMost (آیا بالاتر از تمام برنامه‌های دیگر قرار گیرد) و وضعیت آغازین پنجره‌ی اصلی برنامه را مشخص نمود.

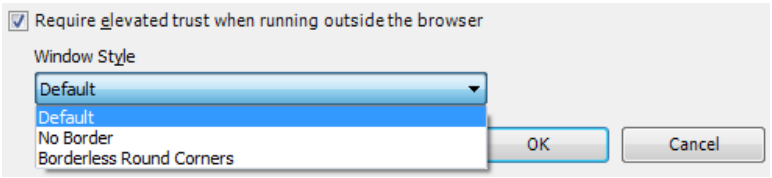


شکل ۸– تنظیم موقعیت آغازین پنجره برنامه در خارج از مرورگر به صورت دستی

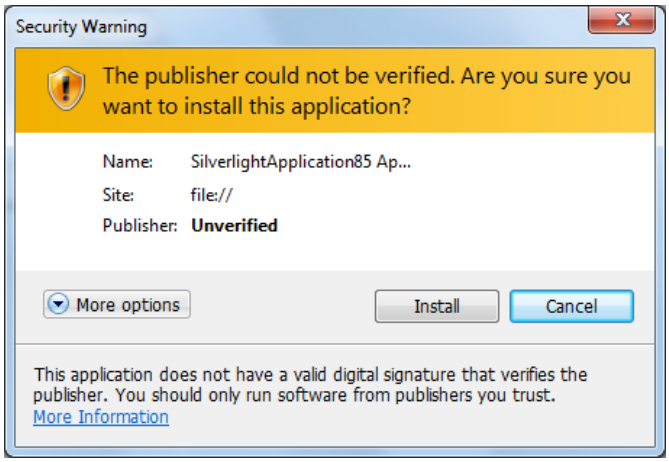
درخواست سطح دسترسی بالاتر

به صورت پیش فرض سطح دسترسی برنامه‌های OOB نیز همانند حالت اجرا درون مرورگر است. اما در Silverlight 4 صرفا جهت برنامه‌های OOB ، امکان درخواست دسترسی بیشتر نیز وجود دارد؛ اما این دسترسی بیشتر به معنای دسترسی مدیریتی نیست. تنها یک سری از ویژگی‌های خاص و کاملا معین در اختیار برنامه قرار خواهند گرفت:

- امکان دسترسی به اطلاعات سایر Domains بدون نیاز به فایل clientaccesspolicy.xml .
- مسیر کامل فایل‌های مورد استفاده توسط کلاس‌های Open/SaveFileDialog در دسترس خواهند بود.
- ورود به حالت تمام صفحه یا گشودن صفحات Open/SaveFileDialog نیاز به اقدام کاربر نداشته و به صورت مستقیم میسر است.
- محدودیت‌های کار با صفحه کلید در حالت تمام صفحه، برداشته می‌شود.
- تنها امکان دسترسی به پوشه‌های My Documents، My Music و امثال آن وجود خواهد داشت (بدون نیاز به کلاس‌های Open/SaveFileDialog).
- امکان دسترسی به COM interop (در ویندوز البته).



شکل ۹– تنظیمات درخواست دسترسی بیشتر یک برنامه‌ی OOB .



شکل ۱۰– صفحه‌ی نصب برنامه‌ی OOB در حالت درخواست دسترسی بالا.

برای این منظور باید در صفحه‌ی تنظیمات پروژه، قسمت تنظیمات برنامه‌ی OOB ، گزینه‌ی مرتبط را انتخاب نمود (شکل ۹). در این حالت زمانیکه بر روی صفحه کلیک راست کرده و گزینه‌ی Install را انتخاب می‌کنیم، با تصویر متفاوتی نسبت به قبل روبرو خواهیم شد (شکل ۱۰). زیرا این نوع برنامه‌ها به سطح دسترسی بیشتری نیاز دارند.

مثالی در مورد نحوه‌ی دسترسی به پوشه‌ی ویژه‌ی MyDocuments

MainPage.xaml.cs

```
using System;
using System.IO;
using System.Windows;
namespace SilverlightApplication85
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();
            this.Loaded += MainPage_Loaded;
        }

        void MainPage_Loaded(object sender, RoutedEventArgs e)
        {
            if (Application.Current.HasElevatedPermissions)
            {
                string myDocuments = Environment.GetFolderPath(
                    Environment.SpecialFolder.MyDocuments);
                string filenamestr = "testfile.txt";
                string pathstr = Path.Combine(myDocuments, filenamestr);
                if (!File.Exists(pathstr))
                {
                    File.WriteAllText(pathstr,"Test!");
                    MessageBox.Show("Done!");
                }
            }

        }
    }
}
```

در این مثال ساده ابتدا توسط خاصیت HasElevatedPermissions بررسی خواهد شد که آیا دسترسی لازم مهیا است یا خیر. سپس مسیر پوشه‌ی MyDocuments بر روی سیستم دریافت شده و در آخر یک فایل متنی در آن نوشته خواهد شد.

تعامل با اشیاء COM

در برنامه‌های OOB با دسترسی بالا امکان تعامل با اشیاء COM نیز وجود دارد. برای مثال دسترسی به اشیاء COM مجموعه‌ی Microsoft Office . بدیهی است این مورد فقط در ویندوز پشتیبانی خواهد شد. در مثال بعد یک وهله از شیء برنامه‌ی Excel ایجاد شده و برگه‌ی فعال آن دریافت می‌گردد. سپس یک نمودار خالی به آن اضافه خواهد شد.

MainPage.xaml.cs

```
//using System.Runtime.InteropServices.Automation;

private void comInteroperability()
{
  if (!Application.Current.HasElevatedPermissions)
                return;
  dynamic excel = AutomationFactory.CreateObject("Excel.Application");
  excel.Visible = true;  // make it visible to the user.
  // add a workbook to the instance
  dynamic workbook = excel.workbooks;
  workbook.Add();
  dynamic sheet = excel.ActiveSheet; // get the active sheet

   // add a chart
   dynamic sheetShapes = sheet.Shapes;
   sheetShapes.AddChart(-4100, 200, 2, 400, 300);
 }
```

در این مثال از واژه‌ی کلیدی dynamic تعریف شده در C# 4.0 استفاده شده است. برای این منظور نیاز است تا ارجاعی را به اسمبلی استاندارد Microsoft.CSharp.dll به پروژه افزود.

کنترل رخداد بسته شدن پنجره‌ی اصلی برنامه

در Silverlight 4 امکان مدیریت رخداد بسته شدن پنچره‌ی اصلی برنامه در برنامه‌های OOB با سطح دسترسی بالا میسر شده است. لطفا به مثالی در این مورد دقت بفرمائید:

MainPage.xaml.cs

```
public MainPage()
{
  InitializeComponent();

   if(Application.Current.IsRunningOutOfBrowser &&
                Application.Current.HasElevatedPermissions)
```

```
   {
     Application.Current.MainWindow.Closing += MainWindow_Closing;
   }
 }

void MainWindow_Closing(object sender,
            System.ComponentModel.ClosingEventArgs e)
{
   if(e.IsCancelable)
   {
      e.Cancel = true;
   }
}
```

در اینجا ابتدا بررسی‌های لازم در مورد برنامه‌های OOB و همچنین سطح دسترسی مورد نیاز، انجام شده و در صورت برآورده شدن این شرایط، روال رخدادگردانی برای رخداد Closing شیء MainWindow برنامه جاری تعریف می‌گردد. در این روال اگر e.Cancel به true تنظیم شود، امکان بسته شدن برنامه وجود نخواهد داشت. عموما استفاده از رخداد Closing جهت پایان دادن مطمئن به اعمال درحال جریان، بسیار مفید است.

آشنایی با مفهوم CustomChrome

Chrome در اینجا به معنای حاشیه و دکمه‌های استانداردی است که توسط سیستم عامل به یک پنجره اعمال می‌شوند. در Silverlight 4 تنها برای برنامه‌های OOB با سطح دسترسی بالا امکان حذف حاشیه‌های پیش فرض سیستم عامل و سفارشی کردن آن وجود دارد. اگر به شکل ۹ دقت نمائید سه نوع Window style را می‌توان انتخاب نمود:

- حالت پیش فرض و استاندارد
- بدون حاشیه
- بدون حاشیه با گوشه‌های گرد

اگر نوع‌های بدون حاشیه را انتخاب کنید باید راهی را نیز جهت بستن پنجره‌ها در اختیار کاربر قرار دهید و اینکار به کمک متد Application.Current.**MainWindow**.Close() میسر خواهد شد. همچنین کلاس MainWindow حاوی متدهایی برای Drag فرم جاری برنامه نیز می‌باشد:

C#

```
void Chrome_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)
{
```

```
        if (Application.Current.IsRunningOutOfBrowser &&
             Application.Current.HasElevatedPermissions)
        Application.Current.MainWindow.DragMove();
}

void ResizeHandle_MouseLeftButtonDown(object sender,
            MouseButtonEventArgs e)
{
  if (Application.Current.IsRunningOutOfBrowser &&
         Application.Current.HasElevatedPermissions)
       Application.Current.MainWindow.DragResize(
                        WindowResizeEdge.BottomRight);
}

void CloseButton_Click(object sender, RoutedEventArgs e)
{
  if (Application.Current.IsRunningOutOfBrowser &&
      Application.Current.HasElevatedPermissions)
        Application.Current.MainWindow.Close();
}

void MaximizeButton_Click(object sender, RoutedEventArgs e)
{
  if (Application.Current.IsRunningOutOfBrowser &&
              Application.Current.HasElevatedPermissions)
  {
      if (Application.Current.MainWindow.WindowState ==
                WindowState.Normal)
            Application.Current.MainWindow.WindowState =
                        WindowState.Maximized;
      else
            Application.Current.MainWindow.WindowState =
                        WindowState.Normal;
  }
}

void MinimizeButton_Click(object sender, RoutedEventArgs e)
{
  if (Application.Current.IsRunningOutOfBrowser &&
                Application.Current.HasElevatedPermissions)
      Application.Current.MainWindow.WindowState =
                WindowState.Minimized;
}
```

در مثال فوق روال‌های رخداد گردان دکمه‌ها و یا نواحی از صفحه را ملاحظه می‌نمائید که توسط آن‌ها کار حرکت پنجره‌ی بدون حاشیه، تغییر اندازه‌ی آن، بستن پنجره، به حالت حداقل یا حداکثر درآوردن اندازه‌ی آن را در عمل نمایش می‌دهند.

نصب بی سر و صدای برنامه‌های Silverlight

در Silverlight 4 امکان نصب کلیه برنامه‌های OOB (برنامه‌ی OOB معمولی و یا با سطح دسترسی بالا) بدون نیاز به مراجعه به Web site برنامه نیز میسر است. برای این منظور از برنامه‌ی sllauncher.exe می‌توان کمک گرفت:

```
"c:\Program Files\Microsoft Silverlight\sllauncher.exe"  /install:d:\Myfolder\MyApp.xap
/origin:http://www.mysite.com/slapps/MyApp.xap /shortcut:desktop+startmenu  /overwrite
```

دستور خط فرمان فوق برنامه‌ی MyApp.xap را بر اساس Web site مشخص شده نصب کرده و آیکون‌های میانبر دسترسی به آن‌را بر روی desktop و start menu قرار می‌دهد. این مورد برای توزیع برنامه‌ها از طریق CD/DVD بسیار مناسب است. بدیهی است در این حالت افزونه‌ی Silverlight باید نصب شده باشد و همچنین کاربر مورد نظر نیز دسترسی نصب برنامه‌ها را داشته باشد.

توضیحات مرتبط با پارامترهای دستور خط فرمان فوق:

- /install : مسیر فایل XAP را به این طریق باید مشخص ساخت. این مسیر می‌تواند یک مسیر به اشتراک گذاشته شده در شبکه، بر روی CD یا حالات دیگر باشد (الزامی).
- /origin : آدرس Web site برنامه XAP را مشخص می‌سازد و برای عملیات به روز رسانی خودکار مفید است. حتی اگر Web site ایی برای این منظور تدارک ندیده‌اید این پارامتر باید ذکر گردد و الزامی است.
- /shortcut : محل قرارگیری آیکون‌های میانبر دسترسی به برنامه را مشخص می‌سازند. بهترین حالت استفاده از desktop+startmenu می‌باشد (اختیاری).
- /overwrite : فایل‌های جدید را بر روی فایل‌های قدیمی موجود بازنویسی خواهد کرد. هر چند اختیاری است اما بهتر است استفاده گردد.

اگر صرفا علاقمند به اجرای یک برنامه‌ی OOB بدون نیاز به نصب آن هستید از دستور خط فرمان ذیل استفاده نمائید:

```
"%ProgramFiles%\Microsoft Silverlight\sllauncher.exe"  /emulate:d:\MyApp\Silverface.xap
/origin:"http://www.mysite.net/ClientBin/Silverface.xap" /overwrite
```

در اینجا یک پارامتر emulate بجای install قرار گرفته است و در حین اجرا وانمود خواهد شد که برنامه از آدرس ذکر شده توسط پارامتر origin دریافت شده است.

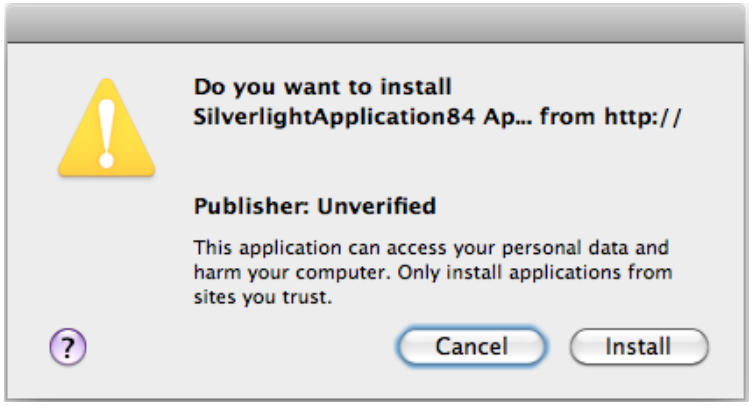
و برای عزل یک برنامه‌ی OOB از دستور خط فرمان زیر استفاده نمائید:

```
"%ProgramFiles%\Microsoft Silverlight\sllauncher.exe"  /uninstall /origin:"http://www.mysite.net/ClientBin/Silverface.xap"
```

همانطور که ملاحظه می‌نمائید پارامتر origin در اینجا بسیار مهم بوده و عملیات عزل برنامه بر این اساس صورت می‌گیرد.

افزودن امضای دیجیتال به فایل‌های XAP

اگر به شکل ۱۰ دقت نمائید، پیغام نصب برنامه‌های OOB با دسترسی بالا نسبت به برنامه‌های OOB معمولی که همان سطح دسترسی برنامه‌های داخل مرورگر به آن‌ها اعمال می‌شود، کاملا متفاوت است؛ نوار نارنجی رنگ بالای صفحه و منتشر کننده‌ی تعیین اعتبار نشده جزئی از این پیغام نصب هستند. علاوه بر آن این نوع برنامه‌ها(ی بدون امضای دیجیتال) از ویژگی به روز رسانی خودکار نیز محروم می‌باشند. شبیه به همین پیغام در سایر سیستم عامل‌ها نیز نمایش داده می‌شود (شکل ۱۱).



شکل ۱۱– پیغام نصب یک برنامه‌ی OOB با دسترسی بالا در MacOSX .

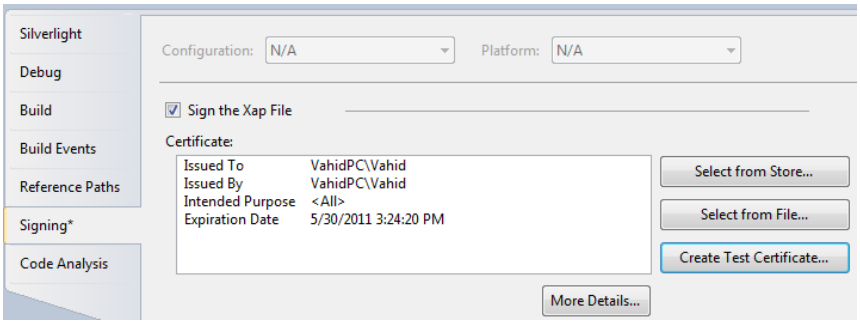
برای حذف این نوع پیغام‌ها نیاز است به فایل‌های XAP خود یک امضای دیجیتال را اضافه نمائیم. این نوع مجوزها (code-signing certificates) از شرکت‌هایی مانند VeriSign ، Thawte ، GoDaddy ، Comodo و غیره قابل تهیه است. شرکت‌های نامبرده شده جزو تامین کنندگان معتبر پیش فرض ویندوز شناخته می‌شوند و فایل‌های آن‌ها در بدو امر و بدون هیچگونه عملیات اضافی دیگری قابل استفاده هستند.

پس از اخذ فایل pfx خود، جهت یکپارچه کردن عملیات افزودن امضای دیجیتال به پروسه‌ی Build در VS.NET ، به خواص پروژه‌ی Silverlight خود مراجعه نموده و در برگه‌ی Build Events آن، دستورات ذیل را در قسمت Post-build event command line وارد نمائید:

```
"%ProgramFiles%\Microsoft SDKs\Windows\v7.0A\Bin\signtool.exe" sign /v /f
c:\users\vahid\documents\authenticode\vahid.pfx /p "MYPASSWORD" /t TIMESTAMP_URI_FROM_PROVIDER
$(TargetName).xap
```

این دستور از برنامه signtool که جزئی از SDK ویندوز است استفاده می‌کند.

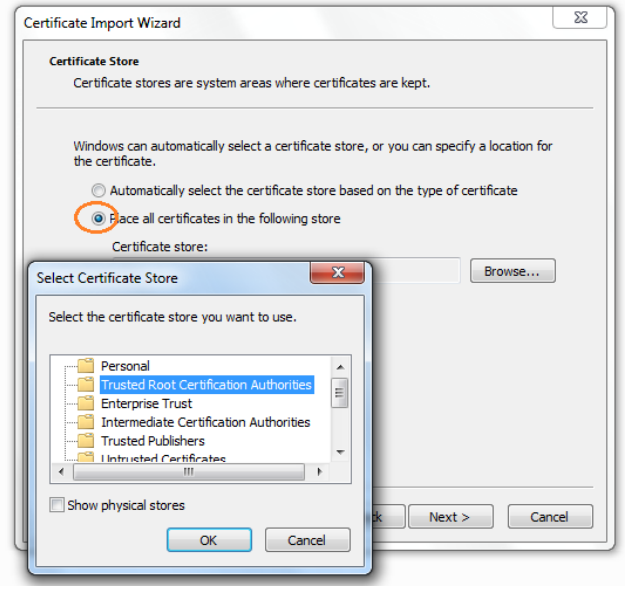
علاوه بر آن اگر علاقمند باشید که یک فایل pfx آزمایشی را تهیه نمائید، VS.NET این امکان را در قسمت خواص پروژه، برگه‌ی Signing فراهم آورده است (شکل ۱۲).



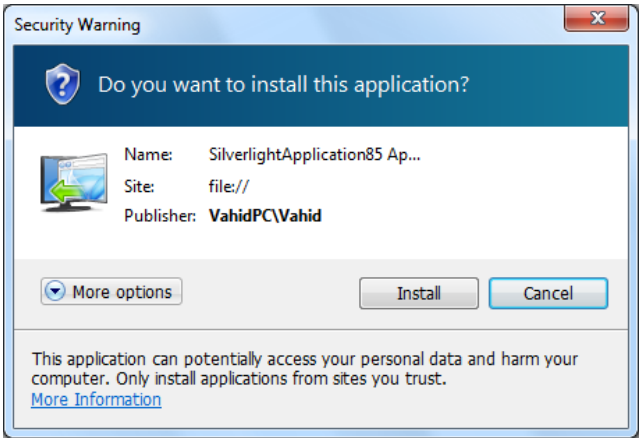
شکل ۱۲– ایجاد یک مجوز آزمایشی و موقت در VS.NET .

در این صفحه پس از انتخاب گزینه‌ی "Sign the XAP file" ، بر روی دکمه‌ی Create Test Certificate کلیک نمائید تا مجوز آزمایشی شما تولید شده و به پروژه اضافه گردد. همچنین اگر یک فایل pfx تهیه شده از منابع معتبر را نیز در اختیار دارید می‌توانید از دکمه‌ی Select from file استفاده نمائید.

پس از تولید فایل pfx خود، به Windows explorer مراجعه کرده و دوبار بر روی این فایل کلیک نمائید تا بتوان آن‌را به "Trusted Root Certification Authorities" ویندوز معرفی کرد (شکل ۱۳). اکنون اگر نسبت به نصب برنامه‌ی OOB نیازمند به سطح دسترسی بالا اقدام کنیم با صفحه‌ی زیبای شکل ۱۴ مواجه خواهیم شد.



شکل ۱۳– معرفی فایل pfx به "Trusted Root Certification Authorities" ویندوز



شکل ۱۴– نمایی از صفحه‌ی نصب یک برنامه‌ی OOB با دسترسی بالا دارای امضای دیجیتال