



Silverlight 4

فهرست مطالب

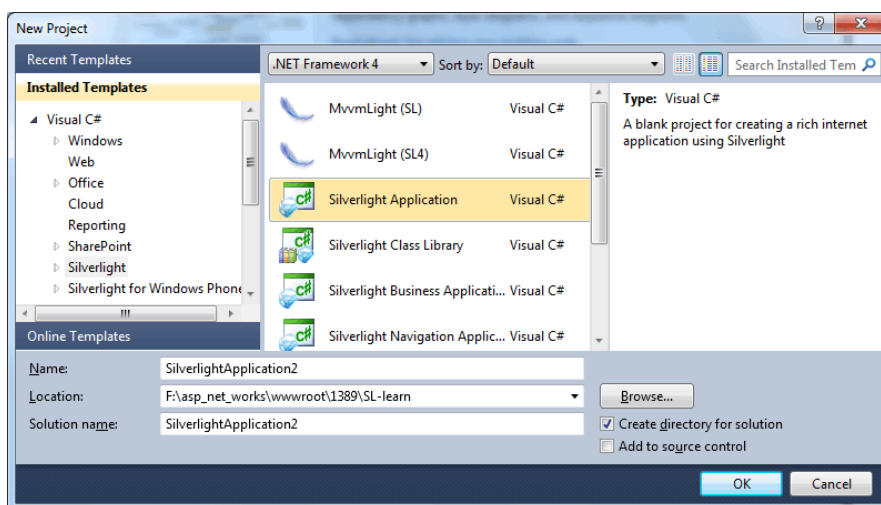
فصل ۲ – بررسی ساختار یک پروژه‌ی Silverlight در VS.NET	۱۴
آشنایی با خواص مقدماتی یک پروژه‌ی Silverlight	۱۴
آشنایی با فرآیندهای رخ داده در حین Compile یک برنامه‌ی Silverlight	۱۸
نکته‌ای مهم جهت انتخاب Build type صحیح در برنامه‌های Silverlight	۲۱
نکاتی در مورد توزیع برنامه‌های Silverlight	۲۱
آشنایی با نحوه‌ی پردازش و طول عمر یک برنامه‌ی Silverlight	۲۳
نکته: سفارشی سازی صفحه‌ی نصب و به روز رسانی افزونه‌ی Silverlight	۲۵

چاپ عمومی غیر رایگان این مطالب بدون مجوز کتبی از طرف نویسنده به هر نحوی غیرمجاز است.
انتشار این مطالب بر روی اینترنت و یا استفاده از آن به صورت مستقیم و یا غیر مستقیم در نشریات الکترونیکی با ذکر مأخذ بلا مانع است.

فصل ۲ - بررسی ساختار یک پروژهی Silverlight در VS.NET

آشنایی با خواص مقدماتی یک پروژهی Silverlight

یک پروژهی جدید Silverlight را در VS.NET 2010 (**File→New→Project...**) آغاز نمائید (شکل ۱). پس از انتخاب گزینهی Silverlight application و انتخاب نامی مناسب برای آن و سپس کلیک بر روی دکمه‌ی OK، بلافاصله صفحه‌ای ظاهر خواهد شد (شکل ۲) که توسط آن می‌توان نگارش Silverlight مورد استفاده و همچنین نحوه‌ی نمایش آن را مشخص ساخت.

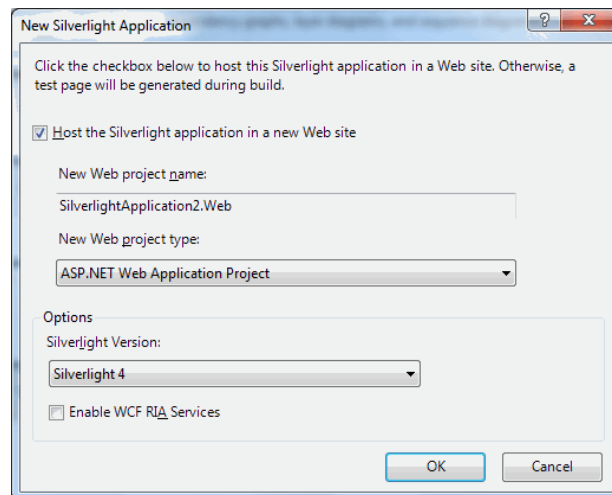


شکل ۱- آغاز یک پروژهی جدید Silverlight در VS.NET 2010.

در اینجا یا می‌توان یک website از نوع ASP.NET را جهت ارائه و نمایش نهایی برنامه Silverlight تهیه شده معرفی نمود و یا جهت آزمایش، هیچ نوع میزبانی را برای آن انتخاب ننمود که در این حالت، در حین اجرای برنامه Silverlight توسط VS.NET، یک صفحه‌ی HTML ساده جهت میزبانی این برنامه به صورت خودکار تولید می‌گردد.

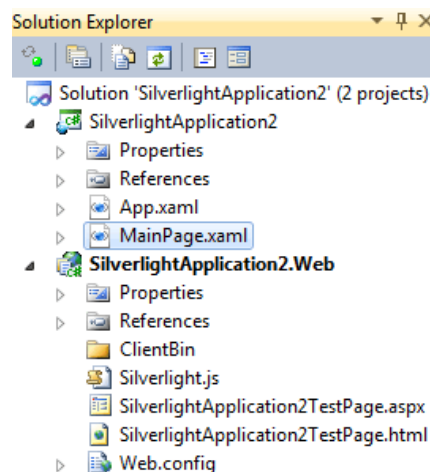
شکل ۳ ساختار ابتدایی پروژهی تولید شده را نمایش می‌دهد که از یک پروژهی اصلی Silverlight و یک پروژهی میزبان ASP.NET تشکیل شده است. پروژه‌ای که با قلم ضخیم (bold) در قسمت Solution explorer

نمایش داده می‌شود، پروژه‌ی آغازین برنامه خواهد بود و اگر دکمه‌ی F5 را جهت اجرای برنامه بفشاریم، این پروژه آغاز کننده‌ی پروسه‌ی دیباگ برنامه می‌باشد.



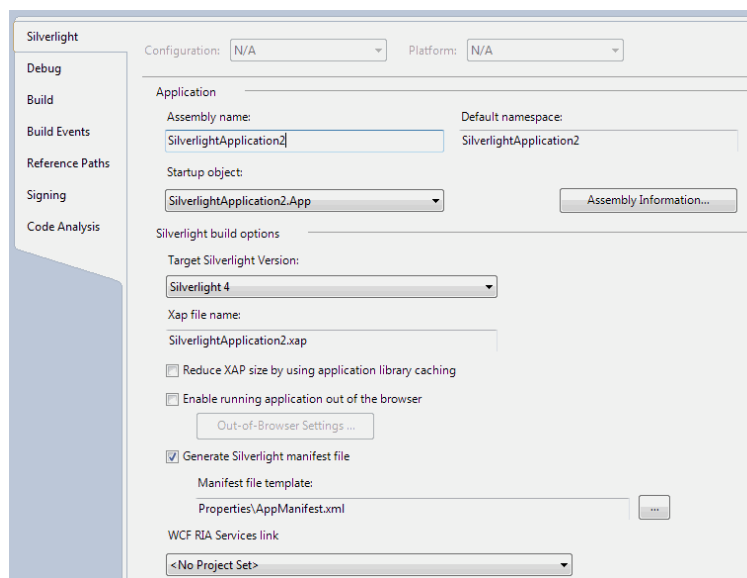
شکل ۲- انتخاب نگارش و میزبانی جهت نمایش برنامه Silverlight

جهت تغییر پروژه‌ی آغاز کننده تنها کافی است بر روی پروژه‌ی مورد نظر در Solution explorer کلیک راست نموده و گزینه‌ی Set as Startup project را انتخاب نمود (که البته در اینجا نیازی به انجام اینکار نیست).

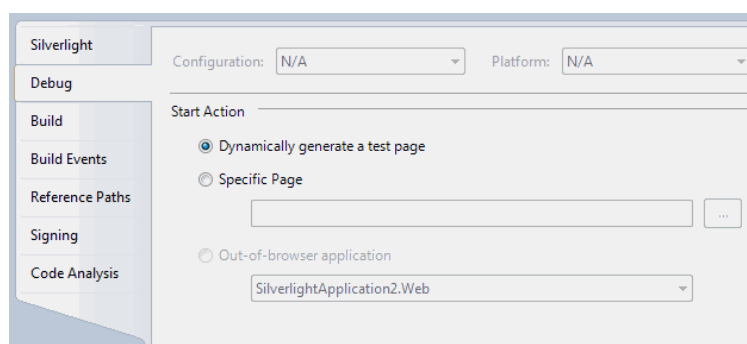


شکل ۳- ساختار ابتدایی یک پروژه Silverlight به همراه میزبانی از نوع یک برنامه‌ی ASP.NET

در ادامه لازم است تا با دو برگه‌ی اول خواص یک پروژه‌ی Silverlight آشنا گردیم (شکل‌های ۴ و ۵). برای این منظور در قسمت Solution explorer در VS.NET دوبار بر روی قسمت Properties پروژه‌ی Silverlight کلیک نمائید تا برگه‌های خواص آن نمایان گردند. همچنین انتخاب پروژه، کلیک راست و سپس انتخاب گزینه‌ی Properties نیز به همین صفحه منتهی می‌گردد.



شکل ۴- برگه‌ی تنظیمات پروژه‌ی Silverlight



شکل ۵- مشخص سازی نحوه‌ی اجرا و میزبانی برنامه‌ی Silverlight در حین Debug

در برگه‌ی Silverlight (شکل ۴) خواصی مانند نام اسمبلی برنامه، فضای نام پیش فرض، پروژه‌ی آغاز کننده‌ی برنامه و یا شماره‌ی نگارش Silverlight مورد استفاده را می‌توان تغییر داد. مهمترین قسمت آن گزینه‌ی XAP file name است. فایل‌های نهایی کامپایل شده‌ی یک برنامه‌ی Silverlight به همراه تمامی منابع مورد استفاده مانند تصاویر و غیره با فرمت XAP (Zap تلفظ می‌شود) بسته بندی و ارائه می‌شوند. این قالب در حقیقت یک نوع فایل Zip است که تنها پسوند آن به XAP تغییر داده شده است.

در برگه‌ی Debug (شکل ۵)، نحوه‌ی آغاز و میزبانی برنامه‌ی Silverlight جهت Debug در VS.NET مشخص می‌گردد. گزینه dynamically generate a test page که گزینه‌ی پیش فرض نیز می‌باشد، به این معنا است که اگر Website میزبان برنامه‌ی Silverlight را در حین ایجاد اولیه‌ی پروژه انتخاب نکنیم، VS.NET در حین Debug برنامه، به صورت پویا صفحه‌ای را جهت میزبانی برنامه ایجاد خواهد کرد. یا توسط گزینه‌ی دوم

می‌توان صفحه‌ای سفارشی را برای این منظور معرفی کرد و یا کلا می‌توان برنامه را به شکل برنامه‌ای با قابلیت اجرای در خارج از مرورگر معرفی نمود.

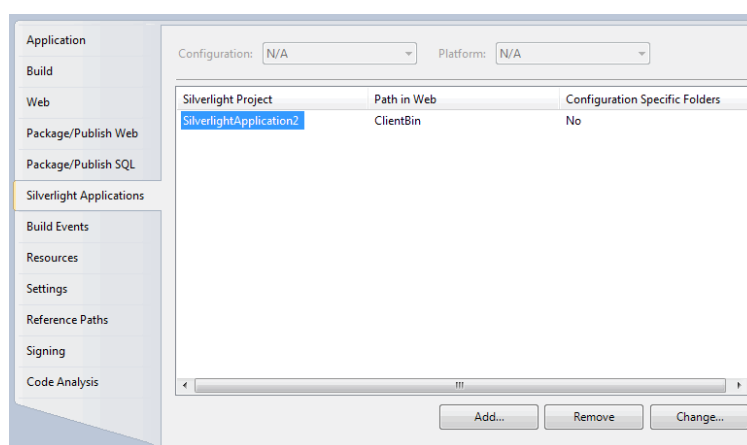
فایل استاندارد دیگری که در حین ایجاد یک پروژه‌ی Silverlight به برنامه اضافه می‌شود، App.xaml نام دارد. این فایل و سورس کد متناظر با آن وظیفه‌ی مدیریت رویدادهایی را در سطح کل برنامه بر عهده دارند؛ همانند آغاز و پایان آن و همچنین امکان معرفی تنظیمات عمومی برنامه. برای مثال در قطعه کد ذیل در روال رویداد گردان Application_Startup، وهله‌ای (Instance) از کلاس MainPage متعلق به سورس کد فایل MainPage.xaml برنامه، ایجاد شده و به عنوان صفحه‌ی آغازین برنامه Silverlight معرفی می‌گردد:

C#

```
private void Application_Startup(object sender, StartupEventArgs e)
{
    this.RootVisual = new MainPage();
}
```

صفحات یک برنامه‌ی Silverlight از نوع UserControl هستند (منهای صفحاتی که جهت مباحث Navigation در طی فصل‌های آتی معرفی خواهند شد). بنابراین اگر نیاز به تغییر این صفحه‌ی آغازین وجود داشت، با استفاده از منوی پروژه، گزینه‌ی Add new item، یک Silverlight UserControl جدید را به پروژه اضافه کرده و سپس وهله‌ای از آن را به خاصیت RootVisual ذکر شده انتساب دهید.

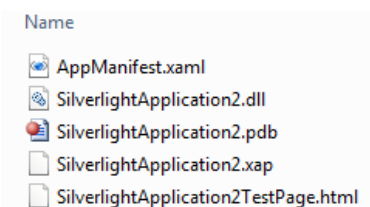
در ادامه به خواص پروژه‌ی Website برنامه مراجعه نمائید (دوبار کلیک بر روی گزینه‌ی Properties در Solution explorer) و سپس برگه‌ی Silverlight Applications را باز نمائید (شکل ۶). در اینجا می‌توان چندین پروژه‌ی Silverlight را در یک پروژه‌ی ASP.NET، میزبانی و معرفی نمود. به صورت پیش فرض اولین پروژه‌ی Silverlight اضافه شده، در این برگه به صورت خودکار قید گردیده است. با این تنظیم، خروجی نهایی برنامه‌ی Silverlight در پوشه‌ی ClientBin تعریف شده به صورت خودکار کپی خواهد شد.



شکل ۶- تنظیمات برنامه Silverlight در میزبان متناظر ASP.NET آن.

آشنایی با فرآیندهای رخ داده در حین Compile یک برنامه‌ی Silverlight

به ازای هربار Compile یک پروژه Silverlight، حداقل ۵ فایل در پوشه‌ی Bin برنامه تشکیل می‌شوند (شکل ۷).



شکل ۷- نمایی از فایل‌های تشکیل شده پس از Compile پروژه

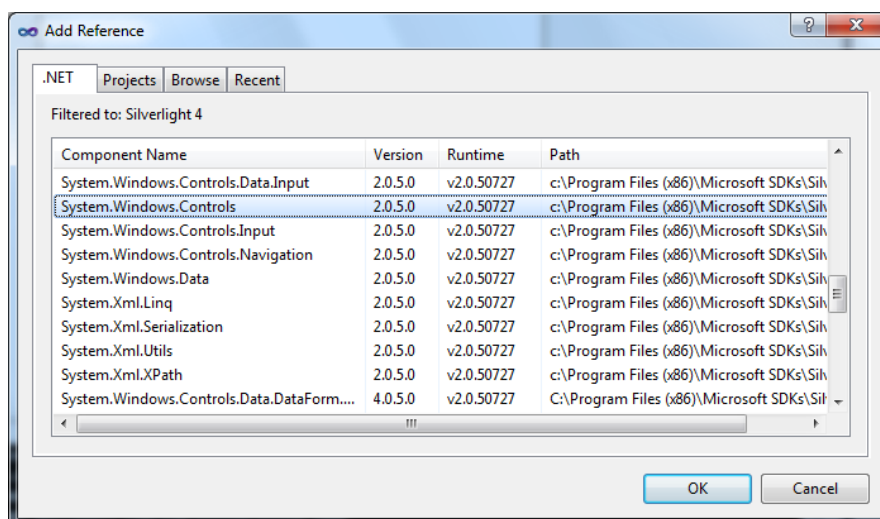
فایل AppManifest.xaml شامل تعاریف وابستگی‌های برنامه است. توسط این فایل مشخص می‌شود که چه تعداد فایل dll به همراه پروژه است و باید بارگذاری شوند. همچنین شماره‌ی نگارش Silverlight مورد استفاده و نام کلاس App برنامه جهت مشخص شدن نقطه‌ی آغازین کار نیز در آن قید می‌گردند. فایل SilverlightApplication2.dll شامل کدهای Compile شده‌ی برنامه به همراه تمامی فایل‌های Xaml آن است. به این صورت نیازی به ارائه‌ی فایل‌های متعدد Xaml برنامه نخواهد بود. فایل SilverlightApplication2.pdb برای نگهداری Debug symbols و بالارفتن کیفیت Debug برنامه در VS.NET تولید می‌شود. تنها فایلی که از یک پروژه Silverlight نیاز است ارائه گردد، فایل SilverlightApplication2.xap می‌باشد. برای مشاهده‌ی محتویات آن، پسوند XAP را به zip تغییر دهید. پس از بررسی محتوای آن، تنها به دو فایل dll و AppManifest بسته بندی شده با فرمت XAP خواهیم رسید. همانطور که پیشتر نیز ذکر شد، فایل SilverlightApplication2TestPage.html جهت برنامه‌هایی که میزبانی را برای نمایش برنامه‌ی Silverlight انتخاب نکرده‌اند ایجاد می‌شود.

باید در نظر داشت که اگر برنامه‌ی Silverlight ما ارجاعی را به سایر اسمبلی‌های ارائه نشده توسط نصاب افزونه‌ی Silverlight داشته باشد، این اسمبلی‌ها به فایل XAP برنامه جهت دریافت آن توسط کاربران اضافه می‌گردد. برای نمونه، اسمبلی استاندارد زیر را به پروژه اضافه کنید (توسط منوی پروژه، گزینه‌ی Add reference، شکل ۸):

System.Windows.Controls

این اسمبلی که توسط SDK مربوط به Silverlight ارائه می‌شود شامل یک سری کنترل جانبی مانند TreeView و غیره است.

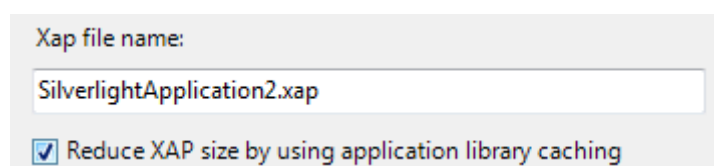
اکنون اگر برنامه را مجدداً Compile نموده و سپس محتویات فایل XAP تولید شده را در پوشه‌ی bin برنامه مورد بررسی قرار دهیم، شاهد خواهیم بود که فایل System.Windows.Controls.dll نیز در این بسته قرار گرفته است. با اسمبلی‌های پروژه‌هایی از نوع کتابخانه که توسط برنامه نویس ممکن است به پروژه اضافه شوند نیز به همین شکل برخورد خواهد شد.



شکل ۸- افزودن ارجاعی به اسمبلی استاندارد System.Windows.Controls

با توجه به اینکه این dll های جانبی هربار باید توسط کاربر دریافت شوند، امکانات caching آنها نیز پیش بینی شده است. برای این منظور به خواص پروژه، برگه‌ی Silverlight مراجعه نموده و گزینه‌ی زیر را انتخاب نمایید (شکل ۹):

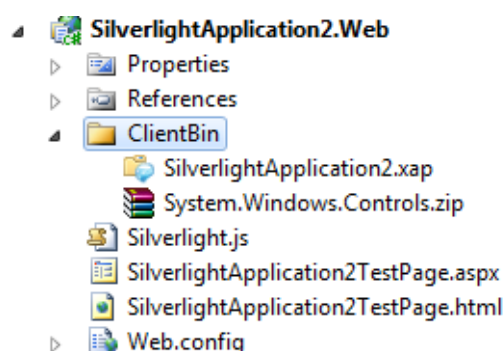
Reduce XAP size by using application library caching



شکل ۹- استفاده از قابلیت‌های caching جهت بارگذاری‌های بعدی سریعتر برنامه

اکنون اگر تنظیمات برنامه را ذخیره نموده و مجدداً پروژه را Compile نمائیم شاهد خواهیم بود که فایل System.Windows.Controls.zip در کنار فایل XAP برنامه تشکیل می‌گردد و فایل XAP برنامه دیگر شامل این dll های جانبی نخواهد بود. به این صورت کاربران در اولین بار مشاهده‌ی سایت، فایل‌های zip را دریافت نموده و مرورگر نسبت به عملیات caching آنها اقدام خواهد نمود که سبب بارگذاری سریعتر برنامه در دفعات آتی مشاهده‌ی آن توسط کاربران می‌گردد.

این فایل‌های نهایی تولید شده را در پوشه‌ی ClientBin میزبان ASP.NET برنامه نیز می‌توان مشاهده نمود (شکل ۱۰).



شکل ۱۰- نمایی از محل قرارگیری فایل‌های Silverlight در میزبان ASP.NET

لازم به ذکر است که اگر کاربر cache مرورگر خود را تخلیه کند، این فایل‌های cache شده نیز حذف خواهند شد. هر بار که کاربر برنامه‌ی Silverlight مورد نظر را اجرا می‌کند، افزونه‌ی Silverlight فایل‌های cache شده را با نمونه‌های موجود بر روی سرور مقایسه کرده و در صورتیکه تغییری در سرور مشاهده شود، نسبت به دریافت نمونه‌های به روز شده اقدام خواهد کرد. همچنین فایل‌های cache شده در دسترس سایر برنامه‌های Silverlight نیز خواهند بود.

نکته‌ای مهم جهت انتخاب Build type صحیح در برنامه‌های Silverlight

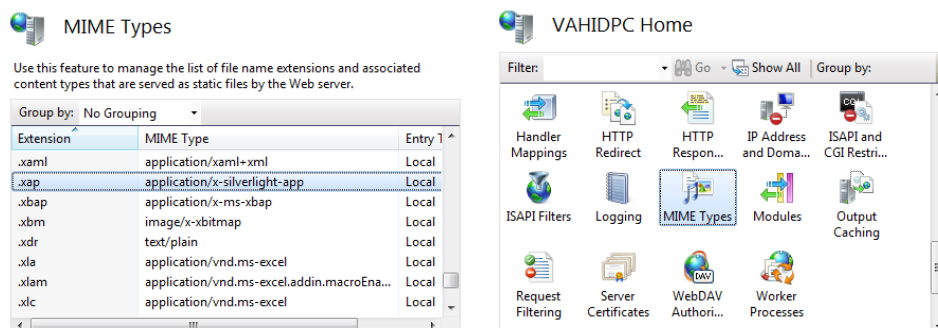
یکی از مواردی که بسیاری از برنامه نویس‌ها هنگام ارائه‌ی برنامه‌های خود رعایت نمی‌کنند، تفاوت قائل نشدن بین حالت debug و release در زمان کامپایل پروژه، برای ارائه نهایی است. هنگام استفاده از حالت release، گزینه‌های بهینه سازی کامپایلر فعال شده و همچنین debug symbols از اسمبلی نهایی تولید شده حذف می‌گردند (بنابراین حجم اسمبلی نهایی نیز کمتر خواهد شد). لازم به ذکر است در حالت release، میزان مصرف حافظه برنامه تولید شده نیز کمتر از حالت debug خواهد بود. گاهی از اوقات سرعت اجرای این دو حالت تا چندین برابر در بعضی از الگوریتم‌ها می‌توانند متفاوت باشند. مطابق مستندات موجود، وجود debug symbols هیچگونه تاثیری بر روی کارایی یک برنامه دات نت ندارند. لازم به ذکر است که عمده بهینه‌سازی‌ها در دات نت توسط JIT compiler صورت می‌گیرد (تا ۹۹ درصد) و نه توسط کامپایلر زبان مورد استفاده (به همین جهت است که عده‌ای اعتقاد دارند در نهایت و هنگام اجرا، تفاوتی مابین زبان‌های مختلف دات نت وجود نخواهد داشت).

نکاتی در مورد توزیع برنامه‌های Silverlight

با توجه به نکاتی که در قسمت آشنایی با فرآیندهای رخداده در حین Compile یک برنامه‌ی Silverlight ذکر گردید، جهت ارائه‌ی یک برنامه‌ی Silverlight توسط یک Web Server نیاز به توزیع فایل‌های XAP و ZIP آن پروژه به همراه میزبان آن‌ها (یک سایت ASP.NET یا صفحات HTML) می‌باشد. اگر از IIS نگارش ۷ به بعد که در ویندوزهای سرور ۲۰۰۸ به بعد موجود هستند استفاده گردد، هیچگونه مشکلی جهت کار با Silverlight نخواهد بود؛ زیرا MIME type متناظر با فایل‌های XAP در این Web Server به صورت پیش فرض تعریف شده است (شکل ۱۱) و امکان ارائه‌ی آن‌ها را دارد. در سایر Web servers نیاز است تا این MIME Type را به صورت دستی معرفی نمود.

در IIS7.x، MIME types، ذیل به صورت پیش فرض موجود هستند:

Extension	MIME Type
.xap	application/x-silverlight-app
.xaml	application/xaml+xml
.xbap	application/x-ms-xbap
.manifest	application/manifest
.application	application/x-ms-application
.deploy	application/octet-stream
.xps	application/vnd.ms-xpsdocument
.dll	application/x-msdownload

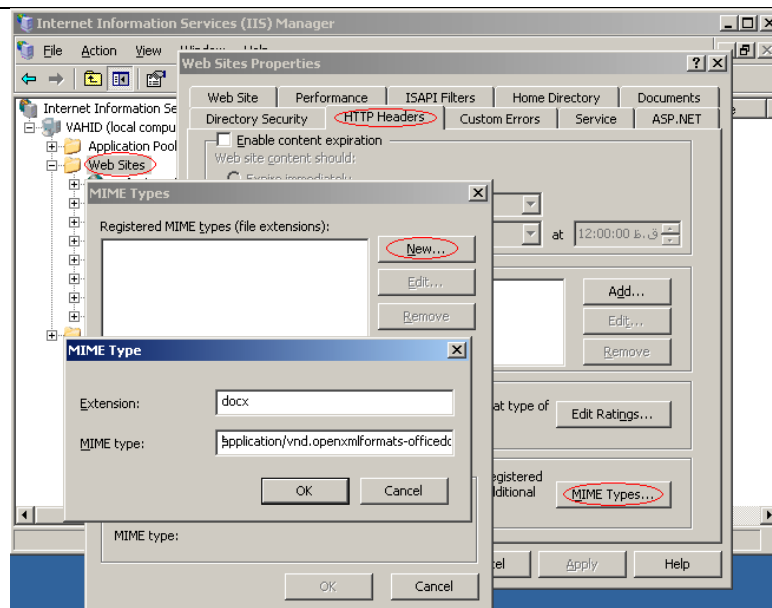


شکل ۱۱- نحوه‌ی دسترسی به MIME Types در IIS 7.x.

در IIS 6.0 ویندوز سرور ۲۰۰۳، تعدادی از این موارد تعریف نشده‌اند (با توجه به اینکه اولین نگارش Silverlight در سال ۲۰۰۷ ارائه شد، این مورد طبیعی است) که نحوه‌ی افزودن آن‌ها در ادامه ذکر خواهد شد (شکل ۱۲):

۱. بر روی نام سرور در IIS Manager کلیک راست نموده و گزینه‌ی خواص آن را انتخاب نمایید.
۲. سپس به برگه‌ی HTTP Headers مراجعه نموده و بر روی دکمه‌ی MIME Types کلیک کنید.
۳. در صفحه‌ی باز شده بر روی دکمه‌ی New کلیک کرده و موارد ذکر شده‌ی قبل را (جدول قبل) یکی پس از دیگری وارد نمایید.

برای مثال در شکل ۱۲ جهت معرفی فایل‌های docx مجموعه‌ی Office 2007 نیز این عملیات انجام شده است؛ در غیر اینصورت کاربران در حین درخواست این نوع فایل‌ها با پیغام فایل مورد نظر بر روی سرور یافت نشد و یا پیغام "Could not download the Silverlight application. Check web server settings" مواجه خواهند شد. در مورد Silverlight یا نگاشت سایر پسوندهای جدید نیز باید به همین صورت عمل نمود.



شکل ۱۲- نحوه‌ی دسترسی به صفحه‌ی تعریف MIME Types در IIS 6.0.

در سایر Web Servers مانند Apache نیز باید به همین شکل عمل نمود و نسبت به تعریف MIME Types متناظر، اقدام کرد. برای مثال در cPanel یک هاست لینوکسی در قسمت Mime Types → Advanced می‌توان موارد ذکر شده را افزود. روش دیگر انجام اینکار ویرایش فایل htaccess می‌باشد؛ که برای نمونه خواهیم داشت (تمامی پسوندهای ذکر شده باید به روش ذیل معرفی شوند):

AddType application/x-silverlight-app .xap

و یا فایل mime.types وب سرور آپاچی را یافته و MIME Types لازم را اضافه نمائید.

در هاست‌های اشتراکی در صورت عدم همکاری مسئول‌های مرتبط نسبت به تعریف MIME Types ذکر شده، فایل‌های XAP را به ZIP تغییر پسوند داده و استفاده نمائید (با توجه به اینکه فایل‌های XAP در اصل همان قالب فایل‌های ZIP را دارند و MIME Type اینگونه فایل‌های ZIP در اکثر Web servers تعریف شده است).

آشنایی با نحوه‌ی پردازش و طول عمر یک برنامه‌ی Silverlight

زمانیکه یک صفحه‌ی حاوی محتوایی از نوع Silverlight از یک Web server درخواست می‌شود، وجود اشیاء Silverlight از طریق MIME Type معرفی شده و همچنین تگ Object متناظر با آن، توسط مرورگر وب تشخیص داده شده و افزونه‌ی Silverlight، بارگذاری می‌گردد. سپس مرورگر از Web server فایل XAP

برنامه را درخواست می‌کند. پس از دریافت فایل XAP ، محتویات آن گشوده شده و در ادامه فایل AppManifest.xml موجود (که نمونه‌ای از آن را در ادامه مشاهده خواهید کرد) خوانده و پردازش خواهد شد. در این فایل، همانطور که پیشتر نیز ذکر شد، نقطه‌ی آغازین برنامه، نگارش Silverlight مورد نیاز و همچنین dll ها و zip فایل‌های مورد نیاز جهت بارگذاری برنامه معرفی می‌شوند. پس از آن برنامه شروع به اجرا خواهد نمود و تنها در یکی از حالات ذیل، اجرای آن متوقف می‌گردد:

۱. خطایی مدیریت نشده در برنامه رخ دهد.
۲. کاربر به صفحه‌ای دیگر در سایت مراجعه کند.
۳. مرورگر بسته شود.

AppManifest.xml

```
<Deployment
  xmlns="http://schemas.microsoft.com/client/2007/deployment"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  EntryPointAssembly="SilverlightApplication2"
  EntryPointType="SilverlightApplication2.App"
  RuntimeVersion="4.0.50401.0">
  <Deployment.Parts>
    <AssemblyPart x:Name="SilverlightApplication2"
      Source="SilverlightApplication2.dll" />
  </Deployment.Parts>
  <Deployment.ExternalParts>
    <ExtensionPart Source="System.Windows.Controls.zip" />
  </Deployment.ExternalParts>
</Deployment>
```

بنابراین در اینجا برخلاف برنامه‌های بدون حالت ASP.NET ، با برنامه‌ای State full سروکار داریم. به این معنا که به محض نمایش آن‌ها، کلیه اطلاعات موجود در برنامه‌ی Silverlight به یکباره تخریب نخواهند شد؛ زیرا این برنامه‌ها در سمت کاربر اجرا می‌شوند (بر خلاف برنامه‌های سمت سرور ASP.NET) و مشکل کمبود منابع را به دلیل عدم تخریب اشیای مورد استفاده صفحات با تعداد کاربران بالا، برای Web server به همراه نخواهند داشت.

برای مشاهده‌ی تگ Object مربوط به معرفی برنامه‌ی Silverlight ، به فایل SilverlightApplication2TestPage.html در پروژه‌ی میزبان برنامه مراجعه نمایید؛ که نمونه‌ای از آن در ادامه معرفی شده است. توسط این تگ، مرورگر متوجه خواهد شد که باید افزونه‌ی Silverlight را بارگذاری کرده و سپس بر اساس مقدار پارامتر Source معرفی شده، فایل XAP متناظر را از Web server دریافت نماید. اگر خطایی مدیریت نشده در این برنامه رخ دهد، اطلاعات آن توسط تابع جاوا اسکریپتی onSilverlightError به کاربر نمایش داده خواهد شد. همچنین حداقل شماره‌ی نگارش افزونه‌ی Silverlight مورد نیاز در این تگ نیز

ذکر می‌گردد (minRuntimeVersion) و اگر پارامتر autoUpgrade به true تنظیم شده و مرورگر کاربر پیشتر این نگارش از افزونه‌ی Silverlight را دریافت نکرده باشد، پیغامی جهت معرفی محل دریافت نگارش به روز افزونه‌ی Silverlight به کاربر نمایش داده می‌شود.

SilverlightApplication2TestPage.html

```
<object
  data="data:application/x-silverlight-2,"
  type="application/x-silverlight-2"
  width="100%" height="100%">
  <param name="source" value="ClientBin/SilverlightApplication2.xap"/>
  <param name="onError" value="onSilverlightError" />
  <param name="background" value="white" />
  <param name="minRuntimeVersion" value="4.0.50401.0" />
  <param name="autoUpgrade" value="true" />
  <a href="http://go.microsoft.com/fwlink/?LinkId=149156&v=4.0.50401.0"
    style="text-decoration:none">
    
  </a>
</object>
```

نکته: سفارشی سازی صفحه‌ی نصب و به روز رسانی افزونه‌ی Silverlight

اگر علاقمند باشید که بجای صفحه و پیغام‌های پیش فرض نمایش دریافت افزونه‌ی Silverlight از سایت مایکروسافت، پیغام‌های سفارشی ویژه‌ای را نمایش دهید، به صورت زیر می‌توان عمل کرد. در ابتدا نیاز است جهت شبیه سازی آن دو پارامتر زیر را در تگ Object صفحه‌ی قبل کمی ویرایش نمود:

```
<param name="minRuntimeVersion" value="6.0.40624.0" />
<param name="autoUpgrade" value="false" />
```

با توجه به اینکه در حال حاضر هنوز نگارش ۶ ارائه نشده است و بر روی مرورگر کامپیوتر جاری ما نصب نیست، پیغام ارتقاء افزونه ظاهر خواهد شد اما چون مقدار پارامتر autoUpgrade را به false تنظیم کرده‌ایم، در مدت onSilverlightError، باید خطای شماره ۸۰۰۱ را که برای این منظور تدارک دیده شده است (به معنای upgrade required)، مدیریت نمود. کدهای پیاده سازی این عملیات را در ادامه ملاحظه خواهید کرد:

SilverlightApplication2TestPage.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>EndUserInstall</title>
```

```

<style type="text/css">
html, body {
    height: 100%;
    overflow: auto;
}
body {
    padding: 0;
    margin: 0;
}
#silverlightControlHost {
    height: 100%;
    text-align:center;
}
</style>
<script type="text/javascript" src="Silverlight.js"></script>
<script type="text/javascript">
    function onSilverlightError(sender, args) {
        // check for the incorrect version # error
        if (args.ErrorCode == 8001) {
            var msg =
                "<a href='http://go.microsoft.com/fwlink/?LinkId=108181'>"+
                "جهت دریافت افزونه‌ی سیلورلایت اینجا کلیک نمایید </a>";

            var hostContainer =
                document.getElementById("silverlightControlHost");
            hostContainer.innerHTML = msg;
        }
    }
</script>
</head>
<body background="gradient2.png" >
    <form id="form1" runat="server" style="height:100%">
        <div id="silverlightControlHost">
            <object
                data="data:application/x-silverlight-2,"
                type="application/x-silverlight-2"
                width="100%" height="100%">
                <param name="source"
                    value="ClientBin/SilverlightApplication2.xap" />
                <param name="onError" value="onSilverlightError" />
                <param name="background" value="white" />
                <param name="minRuntimeVersion" value="6.0.40624.0" />
                <param name="autoUpgrade" value="false" />
                <a href="http://go.microsoft.com/fwlink/?LinkId=149156&v=4.0.50401.0"
                    style="text-decoration:none">
                
            </a>
            </object>
        </div>
    </form>
</body>

```

```

        <iframe id="_sl_historyFrame"
            style="visibility:hidden;height:0px;width:0px;border:0px">
        </iframe></div>
    </form>
</body>
</html>

```