



Artikel om Silverlight

Av Mikael Peterson, Agero

2008-01-23

Bakgrund

För er som inte känner till något alls om Silverlight eller vill ha en mer ingående bakgrund erbjuder Wikipedia en artikel med en utmärkt sammanfattning på <http://en.wikipedia.org/wiki/Silverlight>. Kort kan man säga att Silverlight är ett insticksprogram som finns till de flesta populära webbläsare. Det består av en grafikrenderingsmotor som är en delmängd av Windows Presentation Foundation (WPF) som finns i senaste versionen av .NET. Beroende på version av Silverlight styrs denna sedan genom antingen vanlig JavaScript eller "managed code" och en delmängd av .NET Framework. Silverlight kan kommunicera både med webbservern och med klienten (webbsidan).

Versioner

Det finns två olika versioner av Silverlight; version 1.0 och version 1.1.

Silverlight 1.0 är släppt för några månader sedan och innehåller grafikrenderingsmotorn som är en delmängd av WPF. Denna styrs helt med JavaScript. Min egen tolkning av denna release är att Microsoft kände sig pressade att få ut något fort för att kunna börja etablera varumärket Silverlight och släppte då version 1.0 utan stöd för managed code.

Silverlight 1.1 planeras att släppas sommaren 2008 och innehåller i den senaste alpha-releasen exakt samma grafikrenderingsmotor som version 1.0, men utöver det en delmängd av .NET Framework (något större än Compact Framework) och stöd för managed code i valfritt .NET-språk. I version 1.1 kompilerar man koden till en DLL, vilket innebär hastighetsökningar vid exekvering på uppemot 200 gånger jämfört med version 1.0 (som alltid körs som JavaScript).

Jag vill påstå (och detta kommer att bli tydligare ju längre du läser) att man helt kan strunta i Silverlight 1.0 om man inte har mycket speciella behov. Vänta istället in den slutliga versionen av Silverlight 1.1.

Kompabilitet

Silverlight finns för Windows och Mac och ska snart finnas även för Linux. Linuxstödet utvecklas inte av Microsoft självt vilket gör mig något skeptisk. Vi får se hur det går med det. För mer detaljerad beskrivning av kompabiliteten se <http://en.wikipedia.org/wiki/Silverlight#Compatibility>.

Det finns inget stöd för mobila enheter.

Miljöer

Utveckling sker lämpligast i Visual Studio 2008 samt i Expressionprodukterna. Alla dessa finns idag endast som pre-release. En icke obetydlig detalj är att man för närvarande inte får upp IntelliSense för JavaScript i Visual Studio 2008. Eftersom JavaScript inte kompileras innebär det i praktiken att det blir väldigt mycket testning för att fånga upp alla fel som man har skrivit. Man ska inte underskatta hur mycket fel det blir! Att utveckla i denna miljö känns som en nostalgitripp till 90-talet. Det pratas om en smartare IntelliSense som ska kunna hantera icke typad JavaScript på ett intelligent sätt, eftersom det ändå måste skapas något liknande för LINQ. Jag är dock tveksam till att man kommer att lyckas med det fullt ut för JavaScript och detta är en anledning till att tänka sig för innan man startar ett stort projekt i Silverlight 1.0.

Konkurrerande teknologier

Många talar om Silverlight som en utmanare till Macromedia Flash, vilket i viss mån stämmer. Dock kommer Flash förmodligen att vara mer spritt under ytterligare något (många?) år, vilket gör att publika sidor på Internet förmodligen kommer att behöva använda Flash så långt som möjligt. Microsoft har hittills sagt att de inte kommer att distribuera Silverlight via Windows Update. Vi får dock se hur långt det löftet håller om nedladdningarna inte tar fart genom genuin efterfrågan.

När det gäller Flash-Silverlight-jämförelsen måste man också komma ihåg att Silverlight kommer att vara en mycket modernare teknologi vad gäller allt från hanteringen av gränssnittet till server- och klientintegration. Vid utveckling av applikationer med tyngre logik tenderar förmodligen valet att falla på Silverlight i högre utsträckning.

Vidare finns en konkurrerande teknologi från Microsoft självt, nämligen WPF-applikationer. En kompilerad WPF-applikation kan köras direkt i webbläsaren på alla datorer med .NET 3.0. I praktiken begränsas då klienterna till Windows-installationer, eftersom det är den enda plattformen som .NET 3.0 finns på. Silverlight däremot, finns, som sagt, även för Mac OS och snart också för Linux. WPF-applikationer kommer ofta bort lite i diskussionerna, men har man en kontrollerad klientmiljö (exempelvis ett företags intranät) så kan det vara ett fullgott alternativ med lite mer muskler än både Flash och Silverlight.

Teknik

XAML

Grafikrenderingsmotorn visar grafikelement som kan genereras antingen med kod eller genom markup-språket XAML. XAML är det nya språket som används i WPF och påminner lite om hur man ritat kontroller på en aspx-sida (eller HTML-sida för de som inte lever i Microsoft-världen). I nuvarande version av grafikrenderingsmotorn, som delas av både version 1.0 och 1.1, tillhandahålls endast en delmängd av de XAML-konstruktioner som finns tillgängliga i WPF. Det talas löst om att låta "Silverlight-XAML" närma sig "WPF-XAML" mer igen i nästa betaversion. Det är nog en relativt öppen fråga, dock.

.NET Framework

Precis som Microsoft har valt att klippa ner XAML i Silverlight har man också klippt ner .NET Framework som distribueras med Silverlight 1.1. Den självklara anledningen är att ett för stort installationsarkiv skulle minska incitamentet för miljoner slutanvändare att ladda ner och installera insticksprogrammet i sin webbläsare. Just nu verkar det som att installationsarkivet för Silverlight 1.1 landar på ca 4 MB. Som jag förstår det är denna version av ramverket något större än Compact Framework för mobila enheter.

Expression Blend

Expression Blend är ett verktyg för att skapa XAML (objekt och animeringar) grafiskt. Detta verktyg fungerar mycket bra och det gick väldigt snabbt att sätta sig in i. Jag hade inga problem att på några minuter skapa mycket snygga effekter och animationer ("storyboards") som jag sedan kunde anropa från händelsestyrd kod. Tanken från Microsoft sida är att de som jobbar med grafisk design ska kunna använda Expression Blend och sedan skicka XAML till utvecklarna som bygger på med gränssnittslogiken och affärslogiken.

Visual Studio 2008

Visual Studio 2008 pre-release har IntelliSense-stöd för XAML och Silverlightkonstruktioner. Det integrerar mycket väl med Expression Blend och ofta jobbar man med båda verktygen öppna samtidigt mot samma projekt. Båda varnar när det andra verktyget modifierat en fil och man får då möjlighet att synkronisera projektet. Detta fungerade helt smärtfritt för mig.

Nuvarande version av Visual Studio 2008 har inte ordentligt stöd för automatisk konstruktion av händelsehantering, men det finns ingen anledning att tro att händelsehanteringsstödet inte kommer att vara lika bra som för WinForms vid slutlig release.

Document Object Model

I Silverlight 1.1 kan du nå Document Object Model (DOM), för att manipulera element i HTML-sidan, genom objektet `HtmlPage.Document`. Detta tycks fungera helt smärtfritt. Man får dock tänka på, att trots att man sitter i managed code, så är man ute på lite hal is vad gäller webbläsarkompatibilitet. DOM kan ju skilja sig något från webbläsare till webbläsare.

Man kan för övrigt också nå query-strängar, kakor, navigation, mm från managed code.

Isolated storage

Isolated storage är ett utrymme som utvecklaren når från managed code på klienten som kan användas för att spara data mellan sessioner. I dagsläget begränsas detta till 1 MB per klient. Utvecklaren kan av säkerhetsskäl naturligtvis inte nå den ordinarie filstrukturen på klienten, så isolated storage kan vara ett sätt att lagra enklare användaruppgifter, "high score", etc. Användaren kan dock när som helst välja att tömma sitt isolated storage.

Lager

Det går att skapa enklare händelsehantering direkt i XAML, men jag tycker en bra princip är att skapa grafik och animeringar i XAML och sedan knyta upp händelsehantering och gränssnittslogik i code behind. Affärslogiken placerar man lämpligast på webbservern och anropar via webservice.

Av säkerhetsskäl kan dock Silverlight endast anropa den server som den distribueras från. Vill man nå annat material på Internet får man gå via en webservice på den egna servern.

Slutligen

Silverlight gör sig nog bäst där man behöver ett snyggt gränssnitt eller en speciell datavisualisering som inte går att göra med DHTML eller där Flash inte integrerar väl med befintlig infrastruktur på serversidan.

Grafikrenderingsmotorn är imponerande. Allt är kantutjämnat och flyter mycket snyggt. Filmvisning är inga problem och stödet för hantering av strömmande media är gediget. Det tar inte många minuter att göra en enkel YouTube-liknande klient. Det finns också ett fullskärmsläge vilket ger ytterligare användningsområden.

Den stora frågan som jag ser det är om Silverlight 1.1 kommer att finnas installerat på tillräckligt många datorer den 1 januari 2009 för att få det genomslag som Microsoft hoppas på. Målsättningen för dem är säkert att få en minst lika stor spridning som Flash. Det återstår att se om de lyckas.

Jag anser som jag sa redan inledningsvis att man kan hoppa över version 1.0 och titta direkt på version 1.1. När Silverlight 1.1 släpps så tycker jag att det är ett mycket bra alternativ när man vill att applikationen skall se ut som i Windows fast i webbläsaren. Har man en kontrollerad Windows baserad klientmiljö kan man emellertid lika gärna köra WPF-applikationer – som kan göra mycket mer – rakt i webbläsaren.