

Thomas Wenzlaff

Browserübergreifendes und dynamisches Webdesign

mit

Microsoft Silverlight (ehemals WPF/E)

- Übersicht für den Hobby-Programmierer



## Hinweise zum Dokument

**Autor:** Thomas Wenzlaff  
[www.twseiten.de](http://www.twseiten.de)

**Stand des Dokumentes:** Sonntag, 3. Februar 2008

### Rechtswahrung, Haftung und Verbesserungsvorschläge:

Das Dokument richtet sich an den Nutzerkreis der Hobby-Programmierer.

Die Informationen in diesem Produkt werden ohne Rücksicht auf einen eventuellen Patentschutz veröffentlicht. Warennamen, Hardware- und Softwarebezeichnungen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und gehören - wenn eingetragen - dem jeweiligen Eigentümer, oder sollten als solche betrachtet werden.

Der Autor kann für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

Für Verbesserungsvorschläge und Hinweise ist der Autor dankbar.

Berlin, Sonntag, 3. Februar 2008



## Inhaltsverzeichnis

<b>ÜBERSICHT ZU MICROSOFT SILVERLIGHT (EHEMALS WPF/E)</b>	<b>7</b>
Browserübergreifendes dynamisches Webdesign mit Microsoft Silverlight	8
Webseiten-Links (z.T. nur englisch-sprachige Inhalten)	10
<b>SILVERLIGHT CONTROL IM WEB-DOKUMENT ERZEUGEN</b>	<b>11</b>
Silverlight-Control-Objekt im Web-Dokument per HTML statisch erzeugen	11
Silverlight-Control-Objekt im Web-Dokument per JavaScript dynamisch erzeugen	12
HTML-Container (Host)	12
Helper Fles aus dem SDK zu Silverlight (CreateSilverlight.js und Silverlight.js)	13
<b>SILVERLIGHT CONTROL IM WEB-DOKUMENT PER JAVASCRIPT VERWALTEN</b>	<b>21</b>
Verfügbarkeit von Silverlight (Methode IsInstalled)	22
Control per ID verwalten - Beispiele	23
Control per Eigenschaft .settings verwalten - Beispiele	23
Control per Eigenschaft .content verwalten - Beispiele	23
<b>Control-Eigenschaften - Beispiele</b>	<b>24</b>
ActualHeight Property	24
ActualWidth Property	24
Background (Silverlight Control) Property	24
EnableFramerateCounter Property (nur IE)	25
EnableHtmlAccess Property	25
FullScreen Property	25
InitParams Property	26
IsLoaded Property	26
MaxFrameRate Property	26
OnError Event	26
OnFullScreenChange Event	27
OnLoad Event	28
OnResize Event	29
Source Property	29
Version Property	29
Windowless Property	29
<b>Control-Methoden - Beispiele</b>	<b>30</b>
AddEventListener Method	30
CreateObject Method	30
CreateFromXaml Method	30
CreateFromXamlDownloader Method	31
FindName Method	31
RemoveEventListener Method	31
<b>SILVERLIGHT-CONTROL UND EVENTS</b>	<b>33</b>
OnLoad Event -Beispiel	33
Eventtypen	34



<b>Eventhandler zuweisen</b>	<b>34</b>
Eventhandler zuweisen per XAML	34
Eventhandler zuweisen per JavaScript	34
<b>Parameter des Eventhandlers</b>	<b>35</b>
<b>Control-Event-Methoden - Beispiele</b>	<b>36</b>
AddEventListener Method	36
RemoveEventListener Method	36
<b>SILVERLIGHT APPLICATION SCRIPTING ERROR HANDLING</b>	<b>38</b>
<b>onError Event des Silverlight-Ccontrol</b>	<b>38</b>
<b>OnError Eventhandler des Silverlight-Control</b>	<b>38</b>
<b>ErrorEvent zu Aktionen (anhand von Methoden) des Silverlight-Objektes</b>	<b>39</b>
synchrone Aktion des Silverlight-Objektes	39
a-synchrone Aktion des Silverlight-Objektes	40
<b>Silverlight ErrorEvent-Objekte</b>	<b>40</b>
ErrorEventArgs Object	40
ErrorCode Property	40
ErrorMessage Property	41
ErrorType Property	41
ParserErrorEventArgs Object	41
CharPosition (ParserErrorEventArgs) Property	41
ErrorCode Property	41
ErrorMessage Property	42
ErrorType Property	42
LineNumber (ParserErrorEventArgs) Property	42
XamlFile Property	42
XmlAttribute Property	42
XmlElement Property	43
RuntimeErrorEventArgs Object	43
CharPosition (RuntimeErrorEventArgs) Property	43
ErrorCode Property	43
ErrorMessage Property	43
ErrorType Property	43
LineNumber (RuntimeErrorEventArgs) Property	44
MethodName Property	44
<b>SILVERLIGHT OBJEKTE VERWALTEN (REFERENZIERUNG UND MODIFIZIERUNG)</b>	<b>45</b>
<b>Silverlight JavaScript Debugging im Internet Explorer</b>	<b>45</b>
<b>Verwendung von DOM (Document Objekt Model)</b>	<b>45</b>
Document Object Model (DOM)	45
Silverlight Document Object Model (Silverlight-DOM)	45
Silverlight Objekte	45
Silverlight Methoden	48
Silverlight Eigenschaften (Properties)	49
Silverlight Ereignisse (Events)	53
Silverlight Enumerations	53
Referenzierung per JavaScript - Beispiele	55
Silverlight-DOM-Methoden - Beispiele	56
Add Method	56
Insert Method	57



Remove Method	57
RemoveAt Method	57
Clear Method	58

## **SILVERLIGHT OBJEKTE (AUSWAHL) 59**

<b>Silverlight DependencyObject Object (Abstract)</b>	<b>59</b>
GetValue Method	59
SetValue Method	59

### **Silverlight Control Object 60**

#### **Silverlight Control-Objekt Downloader 62**

Anzeige eines Fortschrittsbalkens des Downloads - Beispiel	62
Zugriff auf gedownloadete Daten (nach Ende des Downloads) - Beispiel	63
Abbruch des gestarteten Downloads - Beispiel	64
Daten in ZIP-Datei - Beispiel	64
Verwaltung von HTTP-Fehler während Download - Beispiel	64
Events - Übersicht	65
Completed Event	65
DownloadProgressChanged (Downloader) Event	65
Methoden - Übersicht	66
Abort Method	66
CreateObject Method	66
GetResponseText Method	66
Open Method	67
Send Method	67
SetSource Method (Methode des Objektes, das um Downloader-Objekt erweitert wurde)	67
Eigenschaften - Übersicht	68
DownloadProgress Property	68
Status Property	68
StatusText Property	68
URI Property	69

#### **Silverlight Control-Objekt MediaElement 70**

#### **Silverlight Control-Objekt TextBlock 72**

## **XAML-SCRIPT - ÜBERBLICK 74**

### **Deklaration von Objekten 74**

Syntax allgemein	74
Abhängigkeiten der Objekte bei Silverlight	74
Beispiele für Deklarationen für Silverlight	75
Beispiel für Ellipse, Rectangle und Line	75
Beispiel für Polyline und Polygon	75
Beispiel für Path	75
Beispiel für Ausmalen einer Ellipse	76
Beispiel für Ausmalen eines Rectangle	76
Beispiel für Ausmalen eines Rectangle per Image	77
Beispiele für Bild	77
Beispiel für TextBlock (Container für Text)	78
Beispiel für Font-Angaben im Textblock	78
Beispiel für verschiedene Fonts innerhalb eines Textblockes anhand Objekt Run	78
Beispiel für MediaElement	78
Beispiel für Steuerung von Sound	79
Beispiel für Objekt-Animation anhand dessen Erweiterung um Storyboard und Timeline	80
Beispiel für Mausevents	80
Beispiel für Event Loaded	80
Beispiel für Hyperlink	81



Beispiel für Button	81
Beispiel für Schieberegler (Slider)	82
<b>XAML-Daten referenzieren</b>	<b>84</b>
CreateFromXaml Methode und Beispiele	84
Referenzierung per JavaScript - Beispiele	85
<b>SUPPORT ZU SILVERLIGHT IM INTERNET - BEISPIELE</b>	<b>87</b>
<b>Weblinks</b>	<b>87</b>
<b>Beispiel zur Erzeugung eines Silverlight-Objektes MediaElement</b>	<b>88</b>
<b>Getting Started</b>	<b>88</b>
<b>Beispiel für erste Schritte mit Silverlight</b>	<b>95</b>
Erste Schritte per englischsprachigem Video von microsoft.com	95
Erste Schritte in Deutsch per Microsoft Expression Produkt Blend 2	95
<b>Beispiel von Microsoft zur Programmierung mit Silverlight</b>	<b>103</b>
Original-Quellcode von Microsoft	104
Analysierte und aufbereitete Quellcode	113
<b>Silverlight-Webseite silverlight.net (u.a. Download von Silverlight-Anwendungen)</b>	<b>146</b>
<b>Microsoft Silverlight-Webseite auf microsoft.com</b>	<b>154</b>



## Übersicht zu Microsoft Silverlight (ehemals WPF/E)

Microsoft hat im Laufe der Entwicklung von Windows Komponenten erzeugt, die eine dynamische Gestaltung von Projekten ermöglicht (z.B. on PowerPoint etc.). Komponenten wurden auch im Internet Explorer bereitgestellt: Der Browser selbst war Teil vom modular gehaltenen Windows.

Im Gegensatz zu Browserherstellern, die kein Betriebssystem entwickeln, sind diese Hersteller auf Schnittstellen der Plattform und auf Standards angewiesen. Auch im Rahmen des Konkurrenzkampfes wurde der künstliche Umstand erzwungen, dass Browser nicht Teil eines Betriebssystems sein sollen, damit Konkurrenzprodukte nutzbar sind. Nachteilig ist diese Zwangsregelung z.B. im Bereich der Europäischen Union: Microsoft muss sein modulares Windows dem Kunden tatsächlich modular anbieten - und von der Konkurrenz umworbenen Microsoft-Kunden gibt es reichlich (abgesehen von Linux-Plattformen), die Windows-Schnittstellen nutzen, fordern oder sogar einklagen wollen.

Microsoft setzt weiterhin auf Komponenten der hauseigenen Plattform. Microsoft baut Features in Windows ein, die auch andere Hardware benötigen (z.B. Digitales RechteManagement bei Medien (DRM)) oder mal eben ein anderes Treibermodell. Anwender müssen sich mit Sicherheitsproblemen von Windows-Komponenten herumschlagen, die von Microsoft verändert werden. Microsoft macht es sich selbst schwer und fährt dafür gute Renditen ein, denn der Kunde macht z.T. den Trott brav mit (Abhängigkeiten von Microsoft-Produkten).

Renditeabsichten sind auch die Motivation von Microsoft, o.g. schwierigen Situationen abzumildern. Des Kaisers neue Kleider ist Marketing: Produkte von Microsoft verbinden mit innovativen Komponenten - und natürlich mit Windows-Komponenten. Dabei setzt Microsoft auf zahlende Kunden, die Investitionen in für Windows benötigte Hard- und Softwarekomponenten tätigen. Und hier liegt der Haken - am Beispiel von Microsoft Silverlight, das Windows ab XP SP2 oder Mac-OS benötigt:

Das, was Silverlight bietet, war z.T. schon in Windows bis XP SP1 implementiert und wurde von Microsoft z.T. lahmgelegt: Z.B. Direct X-Komponenten wie DirectDraw oder Direct3D, die per Javascript-Schnittstelle (und diversen kostenlosen SDK's) im Webdesign mit dem Internet Explorer zugänglich waren - bis zu deren künstlichen Ende namens Windows XP SP2 oder Sicherheitspatches.

Silverlight ist nicht wirklich ein neuartiges Produkt, dafür aber bereinigend z.B. im Bereich der microsoft-eigenen Windows-Features.

Silverlight schlägt in die Schwachstellenkerbe von Browserherstellern ein, die Erweiterung gern Dritten überlassen z.B. im Bereich Multimedia. Diese Browserhersteller predigen dabei Standards (für z.B. HTML) als Kompatibilitätsargument der Browser (z.B. per Gecko-DOM) und wissen ganz genau, dass Standards gezielt Bereiche ausschließen und daher nur durch kompatible Plugins von Dritten erweiterbar sind. Silverlight ist so ein Plugin für z.B. Firefox-Browser, wobei der Internet Explorer in Windows zusätzlich von nicht abgeschalteten Windows-Komponenten profitieren kann, die z.B. auch Standards erweitern. (Auch wenn die Fan-Welt der Nicht-Internet-Explorer-Browser riesig ist: Dynamische Medienwiedergabe im Internet Explorer ist seit langem einfach und prinzipiell möglich: Erweitertes HTML-DOM per BGSOUND. Das kompensieren auch Plugins von Drittherstellern nicht.)

Silverlight ersetzt abgeschaltete Windows-Features, die einst im Webdesign relevant waren (u.a. per JavaScript), und schlägt damit eine Brücke zwischen divergenten Windows-Versionen (z.B. Win XP SP1 zu Win Vista), damit der Kunde auch zum neueren Windows wechselt. Dabei wird es dem Kunden nicht zwingend leicht gemacht: Der Programmierungsaufwand unter Silverlight ist nicht ganz unerheblich - Nutzung eines XML-Dialektes und dessen (abstrakte) Anforderungen. JavaScript-Kenntnisse sind zwingend. Kenntnisse des HTML-DOM ebenso (Silverlight hat ein eigenes XML-DOM per XAML).

Silverlight entfällt ersatzlos für User, die die Laufzeitumgebung (Plugin) von Silverlight nicht installieren (wollen). Es gibt nicht mal eine Standardvariante von Silverlight in Windows für den Internet Explorer. Wer also dynamisches Webdesign mit Silverlight macht, muss wissen, ob er User der Webseite haben wird oder nicht (Vertrauen der Kunden in Microsoft-Produkte). Programmierer, die Webdesign unter Nutzung von einst nicht abgeschalteten Windows-Komponenten vollzogen haben und nach deren Abschaltung Ersatz programmieren mussten, werden sich genau überlegen, wieder umzusteigen - diesmal auf Silverlight (also eventuell wieder den Ereignissen hinterher programmieren zu müssen). Die fehlende Abwärtskompatibilität von Windows (auch zwischen abschaltenden Sicherheitspatches innerhalb einer Windows-Version) ist ein Aufwandsrisiko.



## **Browserübergreifendes dynamisches Webdesign mit Microsoft Silverlight**

Silverlight benötigt Windows ab XP SP2

Apple Mac-OS

unterstützt Browser

Microsoft Internet Explorer 6, (Silverlight als Active-X-Control von Microsoft)

Windows Internet Explorer 7, (Silverlight als Active-X-Control von Microsoft)

Mozilla Firefox 1.5.0.8 (Silverlight als Plugin von Microsoft)

Firefox 2.0.x (Silverlight als Plugin von Microsoft)

Silverlight benötigt eine Laufzeitumgebung - passend zum Betriebssystem und Browser. Diese Umgebung stellt Microsoft (als Browser-Plugin) bereit - falls (einmal) nicht, entfällt (endet) damit die Nutzung und Programmierung mit Silverlight.

Silverlight-Versionen sind nicht unbedingt parallel installierbar.

Silverlight ist mit Schnittstellen wie Ajax, ASP.Net-Ajax, JavaScript Python und Ruby programmierbar.

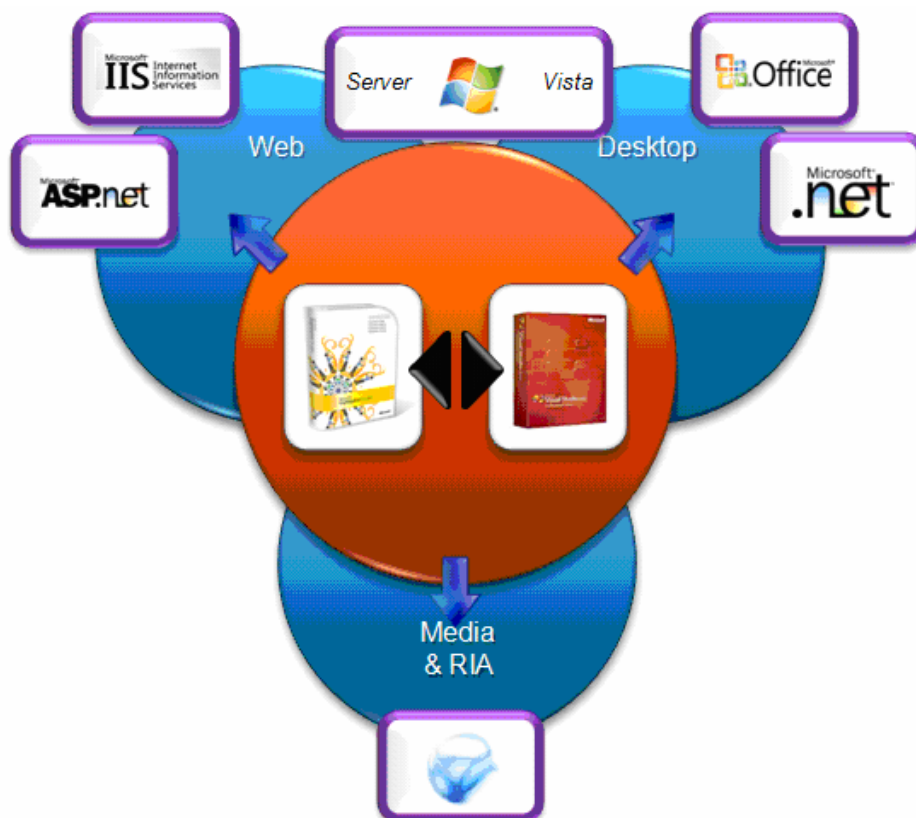
Silverlight benutzt XAML um sichtbare User Interface-Elemente (UI elements) zu erzeugen (wobei anhand Silverlight-DOM, HTML-DOM und z.B. Javascript zugegriffen werden kann).

Silverlight unterliegt permanenter Veränderung: Der Programmierer muss die Änderungen kennen.

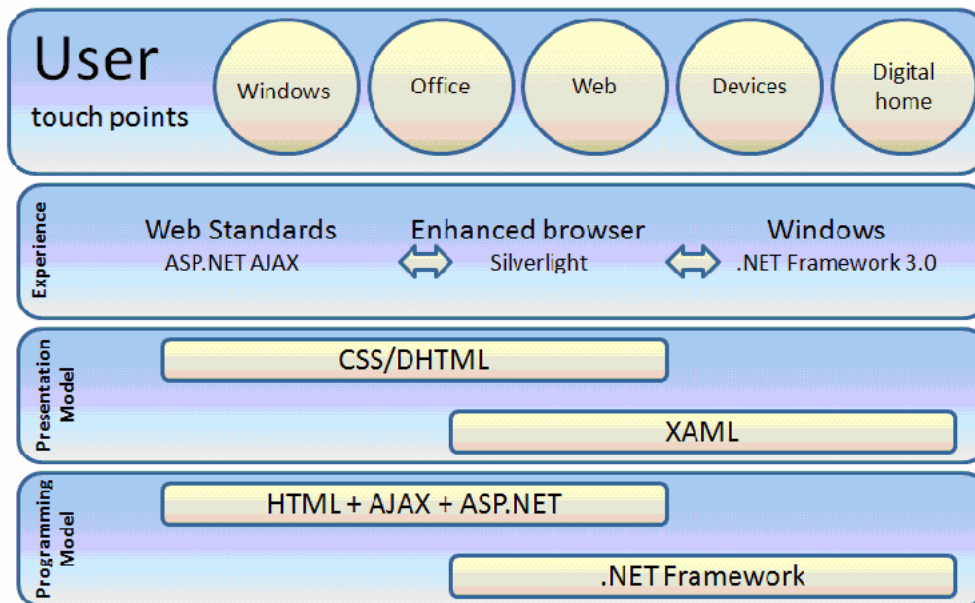
Bsp.: z.Z. werden als numerischen Werte als Single-Precision-Values gespeichert, so dass Werte mit höherer Genauigkeit automatisch abgeschnitten oder gerundet werden.

Silverlight hat ein versionspezifisches Software Developer Kit, das u.a. komplette Beispiel-JS-Dateien (JavaScript-Bibliotheken als Helper Files) enthält, die die Programmierung erleichtern (Microsoft empfiehlt sogar die Nutzung der Helper Files).

Microsoft setzt mit Silverlight laut <http://msdn2.microsoft.com/en-us/library/bb428859.aspx> auf folgenden Kontext:

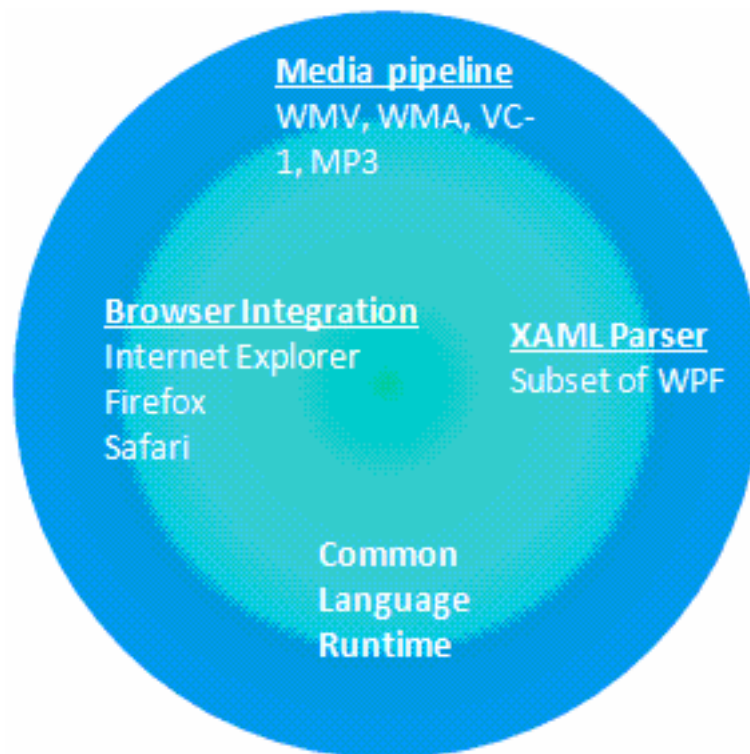






Die oberste Ebene ist die des Microsoft-Kunden, der Microsoft-Produkte nutzt. Natürlich trifft es nicht zu, dass Windows gleichberechtigt ist zu den Anwendungen, die in Windows eingebettet sind.

Die 2. Ebene von oben suggeriert eine Trennung von der obersten - dem ist natürlich nicht so. Auch alle anderen Ebenen sind reine Aufzählungen.



Common Language Runtime als Laufzeitumgebung, die für Browser-Plugin und Scriptmaschine Daten wie Media im Browser verfügbar macht



XAML-Daten und Javascript können inline (im Quellcode) oder extern als Dateiform vorliegen. Zusätzlich sind gepackte Daten möglich, die XAML und Daten wie Bilder, Fonts und Medien enthalten (Zordnung Daten zu XAML erfolgt z.B. per Javascript anhand Silverlight-DOM).

### **Webseiten-Links (z.T. nur englisch-sprachige Inhalte)**

Alle Links können sich ändern - besonders die auf Microsoft-Server !

<a href="http://silverlight.net/Default.aspx">http://silverlight.net/Default.aspx</a>	Silverlight im Internet
<a href="http://silverlight.net/community/communitygallery.aspx">http://silverlight.net/community/communitygallery.aspx</a>	Silverlight Community - Beispiele
<a href="http://www.microsoft.com/silverlight/default01.aspx">http://www.microsoft.com/silverlight/default01.aspx</a>	Silverlight auf Microsoft
<a href="http://msdn2.microsoft.com/en-us/library/bb188743.aspx">http://msdn2.microsoft.com/en-us/library/bb188743.aspx</a>	Silverlight auf MSDN Library
<a href="http://msdn2.microsoft.com/en-us/library/bb428859.aspx">http://msdn2.microsoft.com/en-us/library/bb428859.aspx</a>	Silverlight Architecture Overview
<a href="http://msdn2.microsoft.com/en-us/library/bb412393.aspx">http://msdn2.microsoft.com/en-us/library/bb412393.aspx</a>	Silverlight Scripting Application Notes
<a href="http://msdn2.microsoft.com/en-us/library/bb404710.aspx">http://msdn2.microsoft.com/en-us/library/bb404710.aspx</a>	Silverlight Development with JavaScript
<a href="http://msdn2.microsoft.com/en-us/library/bb188567.aspx">http://msdn2.microsoft.com/en-us/library/bb188567.aspx</a>	Silverlight Reference
<a href="http://msdn2.microsoft.com/en-us/library/bb412392.aspx">http://msdn2.microsoft.com/en-us/library/bb412392.aspx</a>	XAML-Überblick
<a href="http://msdn2.microsoft.com/en-us/silverlight/default.aspx">http://msdn2.microsoft.com/en-us/silverlight/default.aspx</a>	Silverlight auf Microsoft
<a href="http://ajax.asp.net/Default.aspx">http://ajax.asp.net/Default.aspx</a>	Microsoft ASP.Net Ajax



## Silverlight Control im Web-Dokument erzeugen

<http://msdn2.microsoft.com/en-us/library/bb412394.aspx>

### Silverlight-Control-Objekt im Web-Dokument per HTML statisch erzeugen

per OBJECT-Tag oder EMBED-Tag (EMBED bei Nicht-IE-Browsern)

Die programmtechnischen Daten (Content) des Silverlight-Controls werden per XAML-Script implementiert.

Beispiel für Internet Explorer

```
<OBJECT      TYPE="application/ag-plugin"
              ID="myControl"
              WIDTH="400"
              HEIGHT="100"

>
  <PARAM NAME="background" VALUE="#ffebcd"></PARAM>
  <PARAM NAME="enableFramerateCounter" VALUE="true"></PARAM>
  <PARAM NAME="enableHtmlAccess" VALUE="true"></PARAM>
  <PARAM NAME="initParams" VALUE="paramValue1,paramValue2"></PARAM>
  <PARAM NAME="maxFrameRate" VALUE="30"></PARAM>
  <PARAM NAME="onError" VALUE="myErrorHandler"></PARAM>
  <PARAM NAME="onLoad" VALUE="onLoad"></PARAM>
  <PARAM NAME="source" VALUE="HelloWorld.xaml"></PARAM>
  <PARAM NAME="windowless" VALUE="true"></PARAM>
</object>
```

OBJECT-Eigenschaften:

type	MIME-Typ des Plugin
id	Kennung des Objektes im HTML-DOM
width	Breite, integer, Standardeinheit ist Pixel
height	Höhe, integer, Standardeinheit ist Pixel

PARAMs:

<b>background</b>	Hintergrundfarbe String oder hexa, Standardwert ist Leerkette weiss
<b>enableFramerateCounter</b>	boolean true für Anzeige der aktuellen Framerate in der Statusleiste des Browsers Standard ist false
<b>enableHtmlAccess</b>	boolean true für Zugriff vom Silverlight-Control auf das DOM Standard ist true
<b>initParams</b>	String, optionale user-definierte Parameter
<b>maxFrameRate</b>	integer maximale Anzahl der pro Sekunde zu erzeugenden Frames Standard ist 24 (PAL ist 25)
<b>onError</b>	String Bezeichner der JavaScript-Funktion, die Fehler behandelt Standardbezeichner ist 'default_error_handler' beachte Eventverwaltung von Silverlight
<b>onLoad</b>	String Bezeichner der JavaScript-Funktion, die Event OnLoad behandelt Standard ist null also Leerkette
<b>source</b>	String Daten in XAML-Scriptform Hinweis: Alle zu rendernden Daten müssen XAML sein
<b>windowless</b>	boolean true für fensterloses Control Standard ist false



Zugriff per JavaScript: `var control = document.getElementById("myControl");`  
 oder ID-Attributwert direkt verwenden

Beispiel für Firefox und Macintosh:

```
<EMBED TYPE="application/ag-plugin"
        ID="myControl"
        WIDTH="400"
        HEIGHT="100"
        background="#ffebed"
        enableFramerateCounter="true"
        enableHtmlAccess="true"
        initParams="paramValue1, paramValue2"
        maxFrameRate="30"
        OnError="myErrorHandler"
        OnLoad="onLoad"
        source="HelloWorld.xaml" windowless="false">
</EMBED>
```

Params: siehe Beispiel für IE

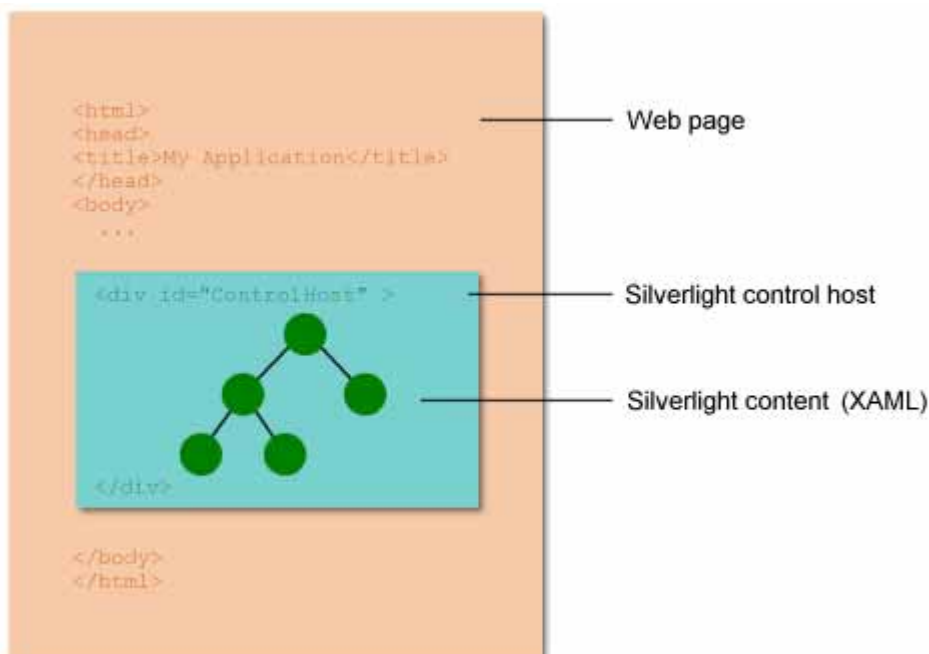
Zugriff per JavaScript: `var control = document.getElementById("myControl");`

## **Silverlight-Control-Objekt im Web-Dokument per JavaScript dynamisch erzeugen**

### **HTML-Container (Host)**

Das Steuerungselement (Control) von Silverlight dient zur Web-Programmierung anhand des Silverlight DOM. Der Host, also dasjenige Element der Webseite, das ein Silverlight-Control erhält, wird per HTML-DOM implementiert.

Die programmtechnischen Daten (Content) des Silverlight-Controls werden per XAML-Script implementiert.



Host ist der Container-DIV mit ID. Der gerenderte Inhalt des DIV stammt aus XAML-Script-Daten.

Beispiel: `var control = document.getElementById("ControlHost");`  
`control.style.visibility='hidden';`  
`ControlHost.style.visibility='hidden';` // nur IE



## Helper Files aus dem SDK zu Silverlight (CreateSilverlight.js und Silverlight.js)

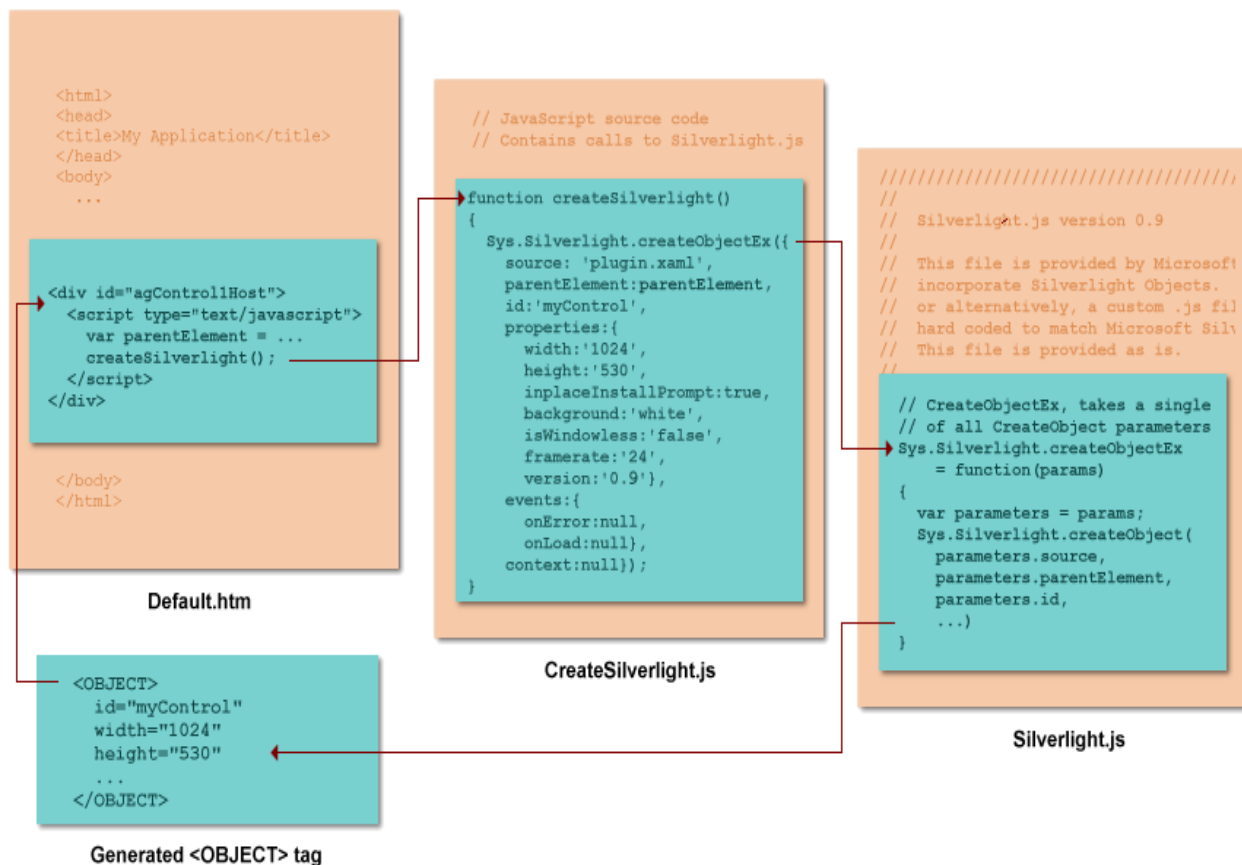
Das SDK von Microsoft zu Silverlight enthält vordefinierte Script-Dateien (Helper-Files)

CreateSilverlight.js Erzeugung Control

Silverlight.js Initialisierung des erzeugten Control

Es wird von Microsoft empfohlen, diese Script-Dateien angepasst zu nutzen.

Nachfolgend ein Beispiel:



Default.html bzw. default.aspx etc.

CreateSilverlight.js

Silverlights.js

lädt die JS-Dateien

enthält den Host, in dem das Control dynamisch erzeugt wird

enthält Methode CreateSilverlight, die Methode aus Silverlights.js aufruft  
JS-Datei muss manuell angepasst werden, also z.B.

kopieren nach SampleCreateSilverlight.js

und diese JS-Datei verwenden

enthält Methoden

CreateObject

CreateObjectEx

wobei diese Methoden u.a. Version von Silverlight prüfen

Control dynamisch im Host erzeugen

Beispiel für Default.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" xml:lang="en">
<head>
<title>A Sample HTML page</title>
<script type="text/javascript" src="Silverlight.js"></script>
<!-- aus dem SDK -->
<script type="text/javascript" src="SampleCreateSilverlight.js"></script>
<!-- angepasste JS-Datei aus dem SDK -->
</head>
<body>
```



```

<!-- Where the Silverlight ActiveX control will go-->
<div id="mySilverlightControlHost">
</div>
<script type="text/javascript">

    // Retrieve the div element you created in the previous step.
    var parentElement = document.getElementById("mySilverlightControlHost");
    createMySilverlightControl();

</script>

</body>
</html>

```

### Beispiel für CreateSilverlight.js bzw. Sample CreateSilverlight.js

Dort anpassen der Funktion `createMySilverlightControl`

Hinweis: Das Datenformat der Kodierung ist JSON (siehe dazu weiter unten)

```

function createMySilverlightControl()
{
    Sys.Silverlight.createObject(
        "myxaml.xaml",           // Source property value.
        parentElement,           // DOM reference to hosting DIV tag.
        "mySilverlightControl",  // Unique control ID value.
        {                         // Control properties.
            width: '300',         // Width of rectangular region of
                                // control in pixels.
            height: '300',        // Height of rectangular region of
                                // control in pixels.
            inplaceInstallPrompt: false, // Determines whether to display
                                // in-place install prompt if
                                // invalid version detected.
            background: '#D6D6D6', // Background color of control.
            isWindowless: 'false', // Determines whether to display control
                                // in Windowless mode.
            framerate: '24',       // MaxFrameRate property value.
            version: '0.9'         // Control version to use.
        },
        {
            onError: null,         // OnError property value --
                                // event handler function name.
            onLoad: null          // OnLoad property value --
                                // event handler function name.
        },
        null);                   // Context value -- event handler
                                // function name.
}

```

### Beispiel für XAML-Daten als Datei Daten.xaml

```

<Canvas Width="300" Height="300"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

    <Ellipse Height="200" Width="200" Canvas.Left="30" Canvas.Top="30"
        Stroke="Black" StrokeThickness="10" Fill="LightBlue"/>

</Canvas>

```

### Beispiel für Silverlight.js aus dem SDK (nicht anzupassen, inhaltlich abhängig von Version von Silverlight)

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//
// Silverlight.js    version 0.9
//
// This file is provided by Microsoft as a helper file for websites that
// incorporate Silverlight Objects. It must be used in conjunction with createSilverlight.js,
// or alternatively, a custom .js file specific to your site. The 0.9 version of this file is
// hard coded to match Microsoft Silverlight v1.0 Beta, which exposes 0.9 as its version number.
// This file is provided as is.

```



```
// © 2007 Microsoft Corporation. All Rights Reserved.  
//  
// This file is licensed as part of the Silverlight 1.0 SDK, for details look here:  
http://go.microsoft.com/fwlink/?LinkID=89144&clid=0x409  
//  
/////////////////////////////////////  
  
if (!window.Sys)  
{  
    window.Sys = { };  
}  
  
if (!window.Sys.Silverlight)  
{  
    window.Sys.Silverlight = { };  
}  
  
/////////////////////////////////////////  
// isInstalled, checks to see if the correct version is installed  
/////////////////////////////////////////  
Sys.Silverlight.isInstalled = function(version)  
{  
  
    var uaString = navigator.userAgent;  
    var reqVersionArray = version.split(".");  
    reqMajorVer = (reqVersionArray[0] != null) ? reqVersionArray[0] : 0;  
    reqMinorVer = (reqVersionArray[1] != null) ? reqVersionArray[1] : 9;  
    reqBuildVer = (reqVersionArray[2] != null) ? reqVersionArray[2] : 0;  
  
function detectAgControlVersion()  
{  
    agVersion = -1;  
  
    if ((navigator.plugins != null) && (navigator.plugins.length > 0))  
    {  
        if (document.getElementById && !document.all && navigator.plugins["WPFfe Plug-In"] )  
        {  
            if (navigator.userAgent.indexOf("Firefox") != -1)  
            {  
                var tmpAgObjectTag = '<OBJECTid="tmpSilverlightVersion" width="1"  
height="1" type="application/ag-plugin"/>';  
                range = document.createRange();  
                range.selectNode(document.body);  
                range.setStartBefore(document.body);  
                tmpAgControlDiv = document.createElement('DIV');  
                document.body.appendChild(tmpAgControlDiv);  
                tmpAgControlDiv.innerHTML=tmpAgObjectTag;  
                agVersionElement=document.getElementById("tmpSilverlightVersion");  
                agVersion=agVersionElement.settings.version;  
                document.body.removeChild(tmpAgControlDiv);  
            }  
            else  
            {  
                agVersion = navigator.plugins["WPFfe Plug-In"].description;  
            }  
        }  
        else if ((navigator.userAgent.indexOf('Windows') != -1) &&  
(navigator.appVersion.indexOf('MSIE') != -1) )  
        {  
            try  
            {  
                var AgControl = new ActiveXObject("AgControl.AgControl");  
                agVersion = AgControl.settings.version;  
                AgControl = null;  
            }  
            catch (e)  
            {  
                agVersion = -1;  
            }  
        }  
    }  
    return agVersion;  
}  
  
var versionStr = detectAgControlVersion();  
if (versionStr == -1 )  
{  
    return false;
```





```

    }
    else if (versionStr != 0)
    {
        versionArray = versionStr.split(".");

        var versionMajor = versionArray[0];
        var versionMinor = versionArray[1];
        var versionBuild = versionArray[2];

        if (versionMajor > parseFloat(reqMajorVer))
        {
            return true;
        }
        else if (versionMajor == parseFloat(reqMajorVer))
        {
            if (versionMinor > parseFloat(reqMinorVer))
            {
                return true;
            }
            else if (versionMinor == parseFloat(reqMinorVer))
            {
                if (versionBuild >= parseFloat(reqBuildVer))
                {
                    return true;
                }
            }
        }
        return false;
    }
}

// Silverlight event instance counter for memory mgt
Sys.Silverlight._counterL = 0;

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
// createObject(); Params:
// parentElement of type Element, the parent element of the Silverlight Control
// source of type String
// id of type string
// properties of type String, object literal notation { name:value, name:value, name:value},
//     current properties are: width, height, background, framerate, isWindowless, enableHtmlAccess,
inplaceInstallPrompt: all are of type string
// events of type String, object literal notation { name:value, name:value, name:value},
//     current events are onLoad onError, both are type string
// initParams of type Object or object literal notation { name:value, name:value, name:value}
// userContext of type Object
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Sys.Silverlight.createObject = function(source, parentElement, id, properties, events, initParams,
userContext)
{
    var slPluginHelper = new Object();
    var slProperties = properties;
    var slEvents = events;
    slPluginHelper.source = source;
    slPluginHelper.parentElement = parentElement;
    slPluginHelper.id = id;
    slPluginHelper.width = slProperties.width;
    slPluginHelper.height = slProperties.height;
    slPluginHelper.background = slProperties.background;
    slPluginHelper.isWindowless = slProperties.isWindowless;
    slPluginHelper.framerate = slProperties.framerate;
    slPluginHelper.ignoreBrowserVer = slProperties.ignoreBrowserVer;
    slPluginHelper.inplaceInstallPrompt = slProperties.inplaceInstallPrompt;
    slPluginHelper.enableHtmlAccess = slProperties.enableHtmlAccess;
    slPluginHelper.initParams = initParams;

    //memory management for onLoad event
    if (slEvents.onLoad)
    {
        var uniqueID = '_sl' + (Sys.Silverlight._counterL++);
        slPluginHelper.loadedHandlerName = 'javascript:' + uniqueID;

        function _dispose()
        {
            if (window.detachEvent)
            {
                window.detachEvent('onunload', _dispose);
            }
        }
    }
}

```





```

    }
    else
    {
        window.removeEventListener('unload', _dispose, false);
    }
    window[uniqueID] = null;
}

function _loadedHandler(sender)
{
    slEvents.onLoad(document.getElementById(slPluginHelper.id), userContext, sender);
    _dispose();
}

window[uniqueID] = _loadedHandler;
if (window.attachEvent)
{
    window.attachEvent('onunload', _dispose);
}
else
{
    window.addEventListener('unload', _dispose, false);
}
}

//set error handler
if (!slEvents.onError)
{
    slPluginHelper.onError = "default_error_handler";
}
else
{
    slPluginHelper.onError = slEvents.onError;
}

var slPluginHTML = "";

//direct download pointer
var directDownload;

if (navigator.userAgent.indexOf('Windows') != -1)
{
    directDownload = "http://go.microsoft.com/fwlink/?LinkID=86008";
}

else if (navigator.userAgent.indexOf('PPC Mac OS X') != -1)
{
    directDownload = "http://go.microsoft.com/fwlink/?LinkID=87380";
}

else if (navigator.userAgent.indexOf('Intel Mac OS X') != -1)
{
    directDownload = "http://go.microsoft.com/fwlink/?LinkID=87384";
}

//point to correct image/landing page for Alpha (0.95.x) and Beta (0.90.x)
var inDirectDownloadPage, inDirectDownloadImage;

var curVer = slProperties.version.split(".");
majorVer = curVer[0];
minorVer = curVer[1];

//if Alpha, disallow inplaceInstall
if (minorVer == "95")
{
    slPluginHelper.inplaceInstallPrompt = false;
    inDirectDownloadPage = "http://go.microsoft.com/fwlink/?LinkID=88363";
    inDirectDownloadImage = "http://go.microsoft.com/fwlink/?LinkID=88365";
}
else
{
    inDirectDownloadPage = "http://go.microsoft.com/fwlink/?LinkID=86009";
    inDirectDownloadImage = "http://go.microsoft.com/fwlink/?LinkID=87023";
}

}

// text for Silverlight image link, used for non-inplaceInstallPrompt and unsupported browser

var silverlightLink = '<div style="width: 205px; height: 67px; background-color: #FFFFFF"><a
href="'+inDirectDownloadPage+'"></a></div>';

```



```

// detect supported browser version & that the correct version of WPF/e is installed, else
display install

if (browserIsSupportedVersion(slPluginHelper))
{
    if (Sys.Silverlight.isInstalled(slProperties.version))
    {
        slPluginHTML = buildHTML(slPluginHelper);
    }
    else if (!slPluginHelper.inplaceInstallPrompt)
    {
        slPluginHTML = silverlightLink;
    }
    else //inPlaceInstallPrompt
    {
        slPluginHTML += '<div style="width: 205px; height: 101px background-color: #FFFFFF;"><a
href="'+directDownload+'"><img style="border:0";
SRC="http://go.microsoft.com/fwlink/?LinkID=87024"></a>';
        slPluginHTML += '<div style="margin-top: -60px;text-align: center;color: #FFFFFF; font-
size: 10px;font-family: Arial ">By clicking <b>Get Microsoft Silverlight</b> you accept the ';
        slPluginHTML += '<a href="http://go.microsoft.com/fwlink/?LinkID=87025" style="text-
decoration: underline;color: #FFFFFF;">Silverlight license agreement.</a></div>';
        slPluginHTML += '<div style="margin-top: 8px;text-align: center;color: #FFFFFF; font-
family: Arial; font-size: 10px;">Silverlight updates automatically, <a
href="http://go.microsoft.com/fwlink/?LinkID=87026" style="text-decoration: underline;color:
#FFFFFF;">learn more.</a></div></div>';

    }
    else
    {
        slPluginHTML = silverlightLink;
    }
    // insert the HTML into the requested host element or return <object> tag.
    if(parentElement != null)
    {
        parentElement.innerHTML = slPluginHTML;
    }
    else
    {
        return slPluginHTML;
    }
}

////////////////////////////////////
// detect to see if this is a supported browser version
////////////////////////////////////
function browserIsSupportedVersion(slPluginHelper)
{
    var supportedBrowser = true;

    if (slPluginHelper.ignoreBrowserVer == true)
    {
        return supportedBrowser;
    }
    else
    {
        var supportedBrowser = false;

        // detection for Internet Explorer 6.0+, 32-bit only
        if (navigator.userAgent.indexOf('MSIE') != -1)
        {
            if (navigator.userAgent.indexOf('Win64') == -1)
            {
                var tempVersion = navigator.userAgent.split("MSIE");
                browserMajorVersion = parseInt(tempVersion[1]);
                if (browserMajorVersion >= 6.0)
                {
                    supportedBrowser = true;
                }
            }
        }
    }
}
// detection for Firefox 1.5+ and 2.0

```



```

else if (navigator.userAgent.indexOf("Firefox") != -1)
{
    var tempVersion = navigator.userAgent.split("Firefox/");
    tempVersion = tempVersion[1].split(".");
    browserMajorVersion = parseFloat(tempVersion[0]);
    browserMinorVersion = parseFloat(tempVersion[1]);

    if (browserMajorVersion >= 2)
    {
        supportedBrowser = true;
    }
    else
    {
        if ((browserMinorVersion >= 5))
        {
            supportedBrowser = true;
        }
    }
}
else if (navigator.userAgent.indexOf("Safari") != -1)
{
    supportedBrowser = true;
}

return supportedBrowser;
}

////////////////////////////////////
//
// create HTML that instantiates the control
//
////////////////////////////////////
function buildHTML(slPluginHelper)
{
    var slPluginHTML = '<OBJECT type="application/ag-plugin" id="'+slPluginHelper.id+'"'
width="'+slPluginHelper.width+'"' height="'+slPluginHelper.height+'"' >';

    if (slPluginHelper.source != null)
    {
        slPluginHTML += ' <PARAM NAME="source" VALUE="'+slPluginHelper.source+'"' />';
    }
    if (slPluginHelper.framerate != null)
    {
        slPluginHTML += ' <PARAM NAME="maxFramerate" VALUE="'+slPluginHelper.framerate+'"' />';
    }

    slPluginHTML += ' <PARAM NAME="onError" VALUE="'+slPluginHelper.onError+'"' />';

    if (slPluginHelper.background != null)
    {
        slPluginHTML += ' <PARAM NAME="background" VALUE="'+slPluginHelper.background+'"' />';
    }
    if (slPluginHelper.isWindowless != null)
    {
        slPluginHTML += ' <PARAM NAME="windowless" VALUE="'+slPluginHelper.isWindowless+'"' />';
    }
    if (slPluginHelper.initParams != null)
    {
        slPluginHTML += ' <PARAM NAME="initParams" VALUE="'+slPluginHelper.initParams+'"' />';
    }
    if (slPluginHelper.enableHtmlAccess != null)
    {
        slPluginHTML += ' <PARAM NAME="enableHtmlAccess" VALUE="'+slPluginHelper.enableHtmlAccess+'"'
/>';
    }
    if (slPluginHelper.loadedHandlerName != null)
    {
        slPluginHTML += ' <PARAM NAME="onLoad" VALUE="'+slPluginHelper.loadedHandlerName+'"' />';
    }

    slPluginHTML += '<\object>';

    if (navigator.userAgent.indexOf("Safari") != -1)
    {
        // disable Safari caching
        // for more information, see http://developer.apple.com/internet/safari/faq.html#anchor5
        slPluginHTML += "<iframe style='visibility:hidden;height:0;width:0' />";
    }
}

```



```

    }

    return slPluginHTML;
}

////////////////////////////////////
//
// Default error handling function to be used when a custom error handler is
// not present
//
////////////////////////////////////

function default_error_handler(sender, args)
{
    var iErrorCode;
    var errorType = args.ErrorType;

    iErrorCode = args.ErrorCode;

    var errMsg = "\nSilverlight error message      \n" ;

    errMsg += "ErrorCode: " + iErrorCode + "\n";

    errMsg += "ErrorType: " + errorType + "      \n";
    errMsg += "Message: " + args.ErrorMessage + "      \n";

    if (errorType == "ParserError")
    {
        errMsg += "XamlFile: " + args.xamlFile + "      \n";
        errMsg += "Line: " + args.lineNumber + "      \n";
        errMsg += "Position: " + args.charPosition + "      \n";
    }
    else if (errorType == "RuntimeError")
    {
        if (args.lineNumber != 0)
        {
            errMsg += "Line: " + args.lineNumber + "      \n";
            errMsg += "Position: " + args.charPosition + "      \n";
        }
        errMsg += "MethodName: " + args.methodName + "      \n";
    }

    alert(errMsg);
}

// createObjectEx, takes a single parameter of all createObject parameters enclosed in {}

Sys.Silverlight.createObjectEx = function(params)
{
    var parameters = params;
    var html = Sys.Silverlight.createObject(parameters.source, parameters.parentElement,
parameters.id, parameters.properties, parameters.events, parameters.initParams, parameters.context);
    if (parameters.parentElement == null)
    {
        return html;
    }
}

```



Beispiel für JSON-Datenformat: TXT-Datei enthält Daten im JSON-Format

Script-Beispiele:

```
var json_String1= ["eins", "zwei", "drei"];
var Json_Feld=eval('(' + json_String1 + ')');
// Referenz auf Array erzeugen
//       Feldelement ist String
var json_String2= '{"Kette": "test", "Routine": function() {alert(this.Kette);}}';
var Json_Objekt=eval('(' + json_String2 + ')');
// Referenz auf ein benutzerdefiniertes Objekt erzeugen
//       mit Eigenschaft 'Kette' die den Wert 'test' hat
//       mit Methode 'Routine' mit dem Funktionsrumpf
//       alert(this.Kette);
Json_Objekt.Routine();
// entspricht alert(Kette); also Anzeige von 'test'
//       nur wegen this-Zeiger wird auch die Eigenschaft Kette
//       referenziert

var json_String3= '[ {"e1": "e1_w1", "e2": "e2_w1"},'
+ ' {"e1": "e1_w2", "e2": "e2_w2"},'
+ ' {"e1": "e1_w3", "e2": "e2_w3"}'
+ ']';
var Json_Objekt=eval('(' + json_String3 + ')');
// Referenz auf Array erzeugen
//       Feld hat 3 Objekte mit je 2 Eigenschaften
alert(Json_Objekt[0].Eigenschaft2);
// zeigt 'e2_w1' an
```

Daten der Script-Daten als TXT-Daten: Jeder JSON-String liegt in einer eigenständigen TXT-Datei.

```
Script var json_String1= ["eins", "zwei", "drei"];
Text      ["eins", "zwei", "drei"]
```

```
Script var json_String3= '[ {"e1": "e1_w1", "e2": "e2_w1"},'
+ ' {"e1": "e1_w2", "e2": "e2_w2"},'
+ ' {"e1": "e1_w3", "e2": "e2_w3"}'
+ ']';
Text [ {"e1": "e1_w1", "e2": "e2_w1"}, {"e1": "e1_w2", "e2": "e2_w2"}, {"e1": "e1_w3", "e2": "e2_w3"} ]
```

## Silverlight Control im Web-Dokument per JavaScript verwalten

Die Verwaltung differiert z.T. zwischen Browsern bei gleichem Betriebssystem

Beispiel 1:

```
<html>
<head>
  <script type="text/javascript" src="CreateSilverlight.js"></script>
  <script type="text/javascript" src="Silverlight.js"></script>
  <!-- Define Loaded event handler for TextBlock. -->
  <script type="text/javascript"> function setDate(sender, eventArgs) { sender.text = Date(); } </script>
</head>
<body bgcolor="Teal">
  <!-- Define XAML content. -->
  <script type="text/xml" id="xamlContent"><?xml version="1.0"?>
    <Canvas xmlns="http://schemas.microsoft.com/client/2007" Background="Wheat">
      <TextBlock Canvas.Left="20" FontSize="24" Loaded="setDate" />
    </Canvas>
  </script>
  <div id="controlHost" >
    <script type="text/javascript">
      var parentElement = document.getElementById("controlHost");
      createSilverlightEx();
```



```

        </script>
    </div>
</body>
</html>

```

Beispiel 2:

```

<div id="controlHost" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_1";
        var context = "context_1";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>
<div id="controlHost2" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost2");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_2";
        var context = "context_2";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>

function createSilverlight()
{ Sys.Silverlight.createObject( "plugin.xml",           // Source property value. parentElement,
                                // DOM reference to hosting DIV tag. name,
                                // Unique control id value.

                                { // Control properties.
                                    width:'600',           // Width of rectangular region of control in pixels.
                                    height:'200',          // Height of rectangular region of control in pixels.
                                    version:'0.9'          // Control version to use.
                                },
                                { onError:null,           // Use default error handler.
                                  onLoad:onLoad          // OnLoad event handler that can be used for multiple
                                                         // controls.
                                },
                                null,                    // InitParams property value set to null.
                                context);                // Unique context id.
}

function onLoad(control, userContext, sender)
{ alert(control.id + " : " + userContext + " : " + sender.toString()); }

// control      Object
//              Identifies the Silverlight control that invoked the event.
//userContext    String
//              Refers to the unique identifier for the control.
//              sender  Object
//              Refers to root element of the control.

```

### Verfügbarkeit von Silverlight (Methode IsInstalled)

nur wenn Control mit CreateSilverlight.js und Silverlights.js erzeugt wird: Die beiden JS-Dateien müssen als Script eingebunden sein.



Silverlight.js file enthält die Prüfung der Silverlight-Runtime-Version ermöglicht die Nutzung von IsInstalled.

```
<div id="controlHost" >
<script type="text/javascript">
    // Determine whether the specified Silverlight runtime version is installed.
    var version = "0.9.0";
    var isInstalled = Sys.Silverlight.isInstalled(version);
    // Create the control if the specified Silverlight runtime version is installed.
    if (isInstalled)
    {
        var parentElement = document.getElementById("controlHost");
        createSilverlight();
    }
    else { alert("Silverlight runtime version " + version + " not installed."); }
</script>
</div>
```

### **Control per ID verwalten - Beispiele**

control ist der Zeiger auf das Control

CreateObject	control.createObject(objectType) Silverlight-Objekt erzeugen und Zeiger des Objektes liefern
InitParams	control.initParams userspezifische Parameter der Initialisierung des Controls setzen
IsLoaded	control.isLoaded Lade-Status des Control liefern
OnError	control.onError = "eventhandlerFunction" Runtime-Error-Handler zuweisen
Source	control.source XAML-Datenquelle festlegen (Daten die zu rendern sind)

### **Control per Eigenschaft .settings verwalten - Beispiele**

Background	control.settings.background Backgroundfarbe festlegen
EnableFramerateCounter	control.settings.enableFramerateCounter Anzeige des aktuellen Framerate (Frames pro Sekunde) in der Statuszeile des Browser ein- bzw. ausschalten
EnableRedrawRegions	control.settings.enableRedrawRegions Bereiche anzeigen in denen Frame gezeichnet werden
EnableHtmlAccess	control.settings.enableHtmlAccess Zugriff des Control auf DOM ein-bzw. ausschalten
MaxFrameRate	control.settings.maxFrameRate maximale Anzahl der renderbaren Frames pro Sekunde
Version	control.settings.version Version des Control liefern
Windowless	control.settings.windowless Fensterlosigkeit des Control ein- bzw. ausschalten

### **Control per Eigenschaft .content verwalten - Beispiele**

ActualHeight	control.content.actualHeight aktuelle Höhe des Anzeigebereiches des Controls
ActualWidth	control.content.actualWidth aktuelle Breite des Anzeigebereiches des Control
CreateFromXaml	control.content.createFromXaml(xamlContent) XAML-Content dynamisch erzeugen
CreateFromXamlDownloader	control.content.createFromXamlDownloader(xamlContent, part) XAML-Content mit Download-Featuresw dynamisch erzeugen
FindName	control.content.findName(objectName) Zeiger eines Silverlight-Objektes anhand des Attributs x:Name liefern
FullScreen	control.content.fullScreen Fullscreen der Anzeig ein- bzw. ausschalten



OnFullScreenChange control.content.onFullScreenChange = "eventhandlerFunction"  
 Eventhandler zur Fullscreen-Anzeige einbinden

OnResize control.content.onResize = "eventhandlerFunction"  
 Eventhandler zum Resize der Dimension des Control einbinden

## Control-Eigenschaften - Beispiele

### ActualHeight Property

Specifies the height of the rendering area of the Silverlight control.

HTML	You cannot use this property in HTML.
Scripting	value = silverlightObject.content.actualHeight

Property Value

Integer

The height of the rendering area of the Silverlight control in pixels, not device-independent units.

This property is read-only. The default value is the height of the Silverlight control, as specified by the OBJECT tag.

### ActualWidth Property

Specifies the width of the rendering area of the Silverlight control.

HTML	You cannot use this property in HTML.
Scripting	value = silverlightObject.content.actualWidth

Property Value

Integer

The width of the rendering area of the Silverlight control in pixels, not device-independent units.

This property is read-only. The default value is the height of the Silverlight control, as specified by the OBJECT tag.

### Background (Silverlight Control) Property

Specifies the background color of the rectangular region that display XAML content.

HTML	<OBJECTBackground="colorValue" .../>
Scripting	colorValue = silverlightObject.settings.background silverlightObject.settings.background = colorValue

Property Value

colorValue

Specifies the background color value as a string, which can either be a named color value, or an 8-bit or 16-bit color value, with or without alpha transparency:

Type	Example	Description
Color Name	"Teal"	A named color value. For a listing of color names, see the Remarks section in Color.
8 Bit RGB	"#RGB"	An 8-bit RGB value.
8 Bit RGB + Alpha	"#ARGB"	An 8-bit RGB value with alpha transparency.
16 Bit RGB	"#RRGGBB"	A 16-bit RGB value. For a listing of hexadecimal equivalents for color names, see the Remarks section in Color.
16 Bit RGB + Alpha	"#AARRGGBB"	A 16-bit RGB value with alpha transparency. For a listing of hexadecimal equivalents for color names, see the Remarks section in Color.

This property is read/write. The default value is null, which is equivalent to the color white. For more information on how this property is set during control initialization, see Using CreateSilverlight.js and Silverlight.js.





**EnableFramerateCounter Property (nur IE)**

Gets or sets a value that determines whether to display the current framerate in the hosting browser's status bar.

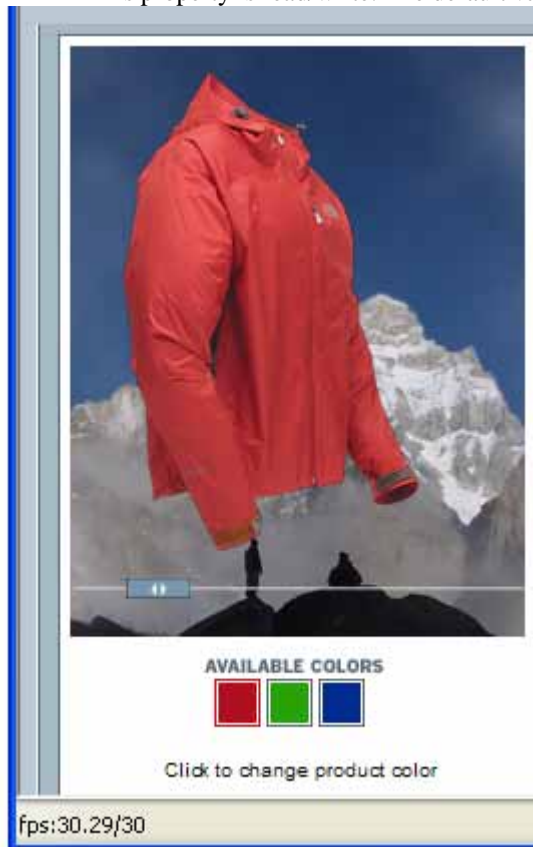
HTML	<OBJECTEnableFramerateCounter=booleanValue .../>
Scripting	booleanValue = silverlightObject.settings.enableFramerateCounter

**Property Value**

Boolean

Determines whether to show the frames-per-second (fps) of the current rendered Silverlight content in the hosting browser's status bar.

This property is read/write. The default value is false.

**EnableHtmlAccess Property**

Determines whether the hosted content in the Silverlight control has access to the browser Document Object Model (DOM).

HTML	<OBJECTEnableHtmlAccess=booleanValue .../>
Scripting	booleanValue = silverlightObject.settings.enableHtmlAccess

**Property Value**

Boolean

Determines whether the hosted content has access to the browser DOM.

This property is read/write. The default value is true.

**FullScreen Property**

Determines whether the Silverlight control displays as a full-screen control.

HTML	You cannot use this property in HTML.
Scripting	value = silverlightObject.content.fullScreen silverlightObject.content.fullScreen = value

**Property Value**

Boolean

Determines whether the Silverlight control displays as a full-screen control or embedded control.



This property is read/write. The default value is false.

### InitParams Property

Specifies the optional set of user-defined initialization parameters.

HTML	<OBJECTInitParams=params .../>
Scripting	params = silverlightObject.initParams

Property Value

String

Specifies a string that represents a set of user-defined initialization parameters. The string format of the params value is user-defined.

This property is read/write. The default value is null. The InitParams property cannot be set after control initialization.

### IsLoaded Property

Gets a value that indicates whether the Silverlight control is loaded.

HTML	You cannot use this property in HTML.
Scripting	value = silverlightObject.isLoaded

Property Value

Boolean

true if the Silverlight control is loaded; otherwise, false.

This property is read-only. The default value is false.

### MaxFrameRate Property

Specifies the maximum number of frames to render per second.

HTML	<OBJECTMaxFrameRate="frameRate" .../>
Scripting	frameRate = silverlightObject.settings.maxFrameRate silverlightObject.settings.maxFrameRate = frameRate

Property Value

frameRate

n integer value that specifies the maximum number of frames to render per second.

This property is read/write. The default value is 24.

The maximum frame rate is 64. Any value that is set to greater than 64 is considered to be set to 64.

### OnError Event

Occurs when the Silverlight control generates a runtime error.

HTML	<OBJECTOnError='functionName' .../>
Scripting	functionName = silverlightObject.onError silverlightObject.onError = functionName

Event Handler Parameters

sender	Object Identifies the Silverlight control that invoked the event.
errorEventArgs	Object Identifies the error and provides information about its source location. For more information, see ErrorEventArgs.

Remarks

The OnError event allows you to specify a user-defined JavaScript error handling function that is invoked when an error is generated in the Silverlight runtime components. The default value is null. If you are using the JavaScript helper file, Silverlight.js, and do not set the OnError property to an error handling function, the default error handling function, default\_error\_handler, is used. For more information on setting this event during control initialization, see Using CreateSilverlight.js and Silverlight.js.

<div id="controlHost" >



```

<script type="text/javascript">
    // Create a variable that references the HTML element that hosts the control.
    var parentElement = document.getElementById("controlHost");
    // Create variables that contain unique identifiers for the control.
    var name = "myControl_1";
    var context = "context_1";
    // Initialize and create the control.
    createSilverlight();
</script>
</div>
<div id="controlHost2" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost2");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_2";
        var context = "context_2";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>

function createSilverlight()
{ Sys.Silverlight.createObject( "plugin.xaml",           // Source property value. parentElement,
                                                                    // DOM reference to hosting DIV tag. name,
                                                                    // Unique control id value.

    { // Control properties.
        width:'600',           // Width of rectangular region of control in pixels.
        height:'200',          // Height of rectangular region of control in pixels.
        version:'0.9'          // Control version to use.
    },
    { onError:null,             // Use default error handler.
      onLoad:onLoad            // OnLoad event handler that can be used for multiple
                                // controls.
    },
    null,                       // InitParams property value set to null.
    context);                   // Unique context id.
}

function onLoad(control, userContext, sender)
{ alert(control.id + " : " + userContext + " : " + sender.toString()); }

// control      Object
//              Identifies the Silverlight control that invoked the event.
//userContext    String
//              Refers to the unique identifier for the control.
//              sender  Object
//              Refers to root element of the control.

```

## OnFullScreenChange Event

Occurs whenever the FullScreen property of the Silverlight control changes.

HTML	You cannot use this property in HTML.
Scripting	silverlightObject.content.onFullScreenChange = "eventhandlerFunction"

### Event Handler Parameters

sender	Object Identifies the object that invoked the event.
eventArgs	Object This parameter is always set to null.



## Event Information

The OnFullScreenChange event occurs whenever the Silverlight control changes between embedded mode and full-screen mode:

embedded mode --control displays within the Web browser.

full-screen mode--control displays on top of all other applications.

**OnLoad Event**

Occurs when the Silverlight control has loaded.

HTML	<OBJECTOnLoad='eventhandlerFunction' .../>
Scripting	Cannot be used in scripting.

## Event Handler Parameters

sender	Object Identifies the Silverlight control that invoked the event.
eventArgs	Object This parameter is always set to null.

## Event Information

The OnLoad event occurs after the XAML content in the Silverlight control has completely loaded. The Onload event cannot be set after control initialization.

```

<div id="controlHost" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_1";
        var context = "context_1";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>
<div id="controlHost2" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost2");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_2";
        var context = "context_2";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>

function createSilverlight()
{ Sys.Silverlight.createObject( "plugin.xaml",
                                // Source property value. parentElement,
                                // DOM reference to hosting DIV tag. name,
                                // Unique control id value.

                                { // Control properties.
                                  width:'600',
                                  height:'200',
                                  version:'0.9'
                                },
                                { onError:null,
                                  onLoad:onLoad
                                },
                                // Use default error handler.
                                // OnLoad event handler that can be used for multiple
                                // controls.

                                },
                                // InitParams property value set to null.
                                // Unique context id.
                                null,
                                context);
}

```



```
function onLoad(control, userContext, sender)
{ alert(control.id + " : " + userContext + " : " + sender.toString()); }

// control      Object
//              Identifies the Silverlight control that invoked the event.
//userContext   String
//              Refers to the unique identifier for the control.
//              sender  Object
//              Refers to root element of the control.
```

## OnResize Event

Occurs whenever the ActualHeight or ActualWidth property of the Silverlight control changes.

HTML	You cannot use this property in HTML.
Scripting	silverlightObject.content.onResize = "eventhandlerFunction"

Event Handler Parameters

sender	Object Identifies the object that invoked the event.
--------	---

Event Information

The OnResize event occurs whenever the Silverlight control changes its ActualHeight or ActualWidth property when in embedded mode. When the Silverlight control is in full-screen mode, the OnResize event does not occur:  
 embedded mode --control displays within the Web browser.  
 full-screen mode--control displays on top of all other applications.

## Source Property

Specifies the XAML content to render.

HTML	<OBJECTSource="[#]sourceReference" .../>
Scripting	sourceReference = silverlightObject.source silverlightObject.source = [#]sourceReference

Property Value

sourceReference  
 Loads the contents of a XAML file or XAML scripting tag into the Silverlight control.  
 This property is read/write. The default value is null.

## Version Property

Gets the version of the Silverlight control.

HTML	You cannot use this property in HTML.
Scripting	value = silverlightObject.settings.version

Property Value

String  
 A string that represents the version of the Silverlight control.  
 This property is read-only.  
 The format of the Version string contains four integer values that are separated by dots, for example, "0.9.0.0".  
 The sequence of integers represent the following version values:  
 major - the major number  
 minor - the minor number  
 build - the build number  
 revision - the revision number

## Windowless Property

Determines whether the Silverlight control displays as a window-less control.

HTML	<OBJECTWindowless=booleanValue .../>
------	--------------------------------------



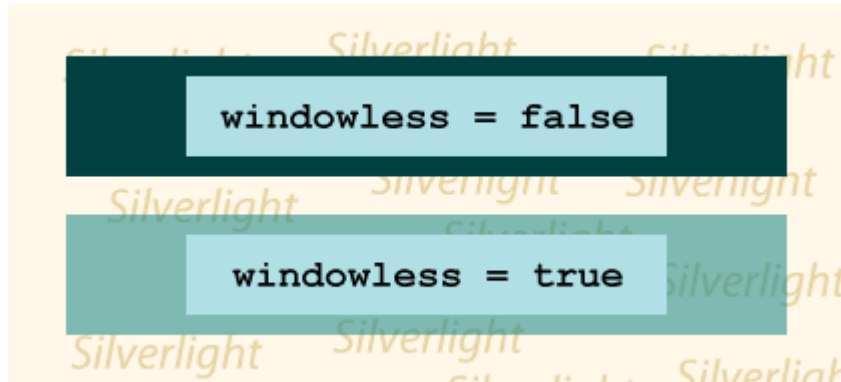
Scripting	<code>booleanValue = silverlightObject.settings.windowless</code>
-----------	---

**Property Value**

Boolean

Determines whether the Silverlight control displays as a window-less or windowed control.

This property is read/write. The default value is false. The Windowless property cannot be set after initialization.

**Control-Methoden - Beispiele****AddEventListener Method**

Registers an event listener on the event target object.

XAML	You cannot use methods in XAML.
Scripting	<code>object.addEventListener(eventType, functionName)</code>

**Parameters**

eventType	String The type of event, such as "KeyDown". An exception is thrown if eventType is invalid.
functionName	String The name of the event handler function.

**Return Value**

None

**Remarks**

The AddEventListener method allows you to registers an event listener on a UIElement-derived object, such as a Canvas or Image. The event handler function that is defined as the event listener receives a notification when the specified eventType occurs.

**Applies To**

Canvas, Ellipse, Glyphs, Image, Line, LineBreak, MediaElement, Path, Polygon, Polyline, Rectangle, Run, Shape, TextBlock, UIElement

**CreateObject Method**

Creates a specified object and returns a reference to it.

XAML	You cannot use methods in XAML.
Scripting	<code>silverlightObject.createObject(objectType)</code>

**Parameters**

objectType	String The type of object to create.
------------	---

**Return Value**

An object reference if the object was successfully created.

**CreateFromXaml Method**

Creates XAML content dynamically.

XAML	You cannot use methods in XAML.
------	---------------------------------



Scripting	silverlightObject.content.createFromXaml(xamlContent, nameScope)
-----------	--

## Parameters

xamlContent	String The XAML content to add to the existing Silverlight object hierarchy.
nameScope	Boolean Determines whether to create x:Name references in XAML content that do not conflict with other named elements. The nameScope parameter is optional, and its value defaults to false if not specified.

## Return Value

An object reference if the XAML content was successfully created; otherwise, returns null.

**CreateFromXamlDownloader Method**

Creates XAML content dynamically using downloader content.

XAML	You cannot use methods in XAML.
------	---------------------------------

Scripting	silverlightObject.content.createFromXamlDownloader(downloader, part)
-----------	--

## Parameters

downloader	Downloader The object containing the downloaded content.
part	String The name of the specific part of the downloaded content package. When the downloaded content package is a Zip file, part refers to the contents of a filename within downloader. If the downloaded content does not represent packaged content, set part to an empty string.

## Return Value

An object reference if the XAML content was successfully created; otherwise, returns null.

**FindName Method**

Gets any object in the Silverlight object hierarchy by referencing the object's x:Name attribute value.

XAML	You cannot use methods in XAML.
------	---------------------------------

Scripting	value = silverlightObject.content.findName(objectName)
-----------	--

## Parameters

objectName	String The name of the object to get.
------------	--

## Return Value

A reference to the specified object if the object was successfully found; otherwise, returns null.

**RemoveEventListener Method**

Unregisters an event listener on the event target object.

XAML	You cannot use methods in XAML.
------	---------------------------------

Scripting	object.removeEventListener(eventType, functionName)
-----------	---

## Parameters

eventType	String The type of event, such as "KeyDown". An exception is thrown if eventType is invalid.
functionName	String The name of the event handler function.

## Return Value

None

## Remarks

Use the RemoveEventListener method to remove a functionName that has been added using the AddEventListener method.

Note The Silverlight implementation of RemoveEventListener is different from the HTML Document Object Model (DOM) version of the same name. The DOM version provides an additional third parameter that



enables event capture.

If multiple identical event listeners are registered on the same event target with the same parameters, the duplicate instances are discarded. They do not cause the event listener to be called twice, and since they are discarded, they do not need to be removed with the RemoveEventListener method.

Applies To

Canvas, Ellipse, Glyphs, Image, Line, LineBreak, MediaElement, Path, Polygon, Polyline, Rectangle, Run, Shape, TextBlock, UIElement





## Silverlight-Control und Events

<http://msdn2.microsoft.com/en-us/library/bb412396.aspx>

### OnLoad Event -Beispiel

Occurs when the Silverlight control has loaded.

HTML	<OBJECTOnLoad='eventhandlerFunction' .../>
Scripting	Cannot be used in scripting.

#### Event Handler Parameters

sender	Object Identifies the Silverlight control that invoked the event.
eventArgs	Object This parameter is always set to null.

#### Event Information

The OnLoad event occurs after the XAML content in the Silverlight control has completely loaded. The Onload event cannot be set after control initialization.

Beispiel:

```
<div id="controlHost" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_1";
        var context = "context_1";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>
<div id="controlHost2" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost2");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_2";
        var context = "context_2";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>

function createSilverlight()
{ Sys.Silverlight.createObject( "plugin.xaml",
                                // Source property value. parentElement,
                                // DOM reference to hosting DIV tag. name,
                                // Unique control id value.

                                { // Control properties.
                                  width:'600',
                                  height:'200',
                                  version:'0.9'
                                },
                                { onError:null,
                                  onLoad:onLoad
                                },
                                null,
                                context);
                                // Width of rectangular region of control in pixels.
                                // Height of rectangular region of control in pixels.
                                // Control version to use.

                                // Use default error handler.
                                // OnLoad event handler that can be used for multiple
                                // controls.

                                // InitParams property value set to null.
                                // Unique context id.
```



```

    }

    function onLoad(control, userContext, sender)
    { alert(control.id + " : " + userContext + " : " + sender.toString()); }

    // control      Object
    //              Identifies the Silverlight control that invoked the event.
    // userContext   String
    //              Refers to the unique identifier for the control.
    // sender        Object
    //              Refers to root element of the control.

```

Bubble wird unterstützt (Event im DOM nach oben zum jeweiligen Elternobjekt reichen).

## Eventtypen

Event Type	Events
Control support	OnError
	OnFullScreenChange
	OnLoad
	OnResize
Downloader support	Completed (Downloader)
	DownloadProgressChanged (Downloader)
Focus	GotFocus
	Loaded
	LostFocus
Keyboard input	KeyDown
	KeyUp
Media	BufferingProgressChanged
	Completed (Storyboard)
	CurrentStateChanged
	DownloadProgressChanged (MediaElement)
	MarkerReached
	MediaEnded
	MediaFailed
	MediaOpened
Mouse input	MouseEnter
	MouseLeave
	MouseLeftButtonDown
	MouseLeftButtonUp

## Eventhandler zuweisen

### Eventhandler zuweisen per XAML

Beispiel:

```

<Canvas xmlns="http://schemas.microsoft.com/client/2007"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Loaded="onLoaded"
>
    <TextBlock x:Name="myTextBlock" />
</Canvas>

```

Loaded-Attribut: keine Parameter im Eventhandler angebbbar !

### Eventhandler zuweisen per JavaScript

Beispiel:

```

<div id="controlHost" >

```



```

<script type="text/javascript">
    // Create a variable that references the HTML element that hosts the control.
    var parentElement = document.getElementById("controlHost");
    // Create variables that contain unique identifiers for the control.
    var name = "myControl_1";
    var context = "context_1";
    // Initialize and create the control.
    createSilverlight();
</script>
</div>
<div id="controlHost2" >
    <script type="text/javascript">
        // Create a variable that references the HTML element that hosts the control.
        var parentElement = document.getElementById("controlHost2");
        // Create variables that contain unique identifiers for the control.
        var name = "myControl_2";
        var context = "context_2";
        // Initialize and create the control.
        createSilverlight();
    </script>
</div>

function createSilverlight()
{ Sys.Silverlight.createObject( "plugin.xaml",           // Source property value. parentElement,
                                                                    // DOM reference to hosting DIV tag. name,
                                                                    // Unique control id value.

    { // Control properties.
        width:'600',           // Width of rectangular region of control in pixels.
        height:'200',          // Height of rectangular region of control in pixels.
        version:'0.9'          // Control version to use.
    },
    { onError:null,             // Use default error handler.
      onLoad:onLoad            // OnLoad event handler that can be used for multiple
                                // controls.
    },
    null,                       // InitParams property value set to null.
    context);                   // Unique context id.
}

function onLoad(control, userContext, sender)
{ alert(control.id + " : " + userContext + " : " + sender.toString()); }

// control      Object
//               Identifies the Silverlight control that invoked the event.
// userContext   String
//               Refers to the unique identifier for the control.
// sender        Object
//               Refers to root element of the control.

```

### Parameter des Eventhandlers

XAML: Eventhandler, der per Attribut Loaded eingebunden wird, kann keine Parameterliste haben

Parameterliste nur kodieren wenn Eventhandler versorgt werden soll

kodierte Parameterliste umfasst 2 Parameter

sender	Zeiger auf Silverlight-Objekt, das das Event erzeugt
eventArgs	Werteliste oder null

Beispiel 1:

```

function onLoaded(sender, eventArgs)
{ // Set the TextBlock to display the current date and time.

```



```
sender.findName("myTextBlock").text = Date();
}
```

Beispiel 2:

```
function onLoaded(sender, eventArgs)
{ // Set the TextBlock to display the current date and time, and set the FontFamily and FontSize properties.
  var textblock = sender.findName("myTextBlock");
  textblock.Text = Date();
  textblock.FontFamily = "Comic Sans MS";
  textblock.FontSize = "24";
}
```

Beispiel 3:

```
function onLoaded(sender, eventArgs)
{ textBlock = sender.findName("Status");
  textBlock.addEventListener("MouseEnter", "onMouseEnter");
  textBlock.addEventListener("MouseLeave", "onMouseLeave");
}
```

Beispiel 4:

```
function removeEvents()
{ textBlock.removeEventListener("MouseEnter", "onMouseEnter");
  textBlock.removeEventListener("MouseLeave", "onMouseLeave");
}
```

## Control-Event-Methoden - Beispiele

### AddEventListener Method

Registers an event listener on the event target object.

XAML	You cannot use methods in XAML.
Scripting	object.addEventListener(eventType, functionName)

Parameters

eventType	String The type of event, such as "KeyDown". An exception is thrown if eventType is invalid.
functionName	String The name of the event handler function.

Return Value

None

Remarks

The AddEventListener method allows you to registers an event listener on a UIElement-derived object, such as a Canvas or Image. The event handler function that is defined as the event listener receives a notification when the specified eventType occurs.

Applies To

Canvas, Ellipse, Glyphs, Image, Line, LineBreak, MediaElement, Path, Polygon, Polyline, Rectangle, Run, Shape, TextBlock, UIElement

### RemoveEventListener Method

Unregisters an event listener on the event target object.

XAML	You cannot use methods in XAML.
Scripting	object.removeEventListener(eventType, functionName)

Parameters

eventType	String
-----------	--------



	The type of event, such as "KeyDown". An exception is thrown if eventType is invalid.
functionName	String The name of the event handler function.

Return Value  
None

#### Remarks

Use the RemoveEventListener method to remove a functionName that has been added using the AddEventListener method.

Note The Silverlight implementation of RemoveEventListener is different from the HTML Document Object Model (DOM) version of the same name. The DOM version provides an additional third parameter that enables event capture.

If multiple identical event listeners are registered on the same event target with the same parameters, the duplicate instances are discarded. They do not cause the event listener to be called twice, and since they are discarded, they do not need to be removed with the RemoveEventListener method.

#### Applies To

Canvas, Ellipse, Glyphs, Image, Line, LineBreak, MediaElement, Path, Polygon, Polyline, Rectangle, Run, Shape, TextBlock, UIElement



## Silverlight Application Scripting Error Handling

<http://msdn2.microsoft.com/en-us/library/bb412395.aspx>

### **onError Event des Silverlight-Ccontrol**

synchrone und asynchrone Eventbehandlung möglich (per JavaScript z.B. innerhalb try-catch-Block)

### **OnError Eventhandler des Silverlight-Control**

Es existiert ein Standardhandler, wenn die JavaScript Helper Files CreateSilverlight.js und Silverlight.js aus dem SDK benutzt werden. Der Standard-Handler ist in Silverlight.js deklariert und verarbeitet auch Fehler aus XAML parsing, Fehler während Laden, Runtime-Fehler.

Der Eventhandler wird anhand Eigenschaft OnError des Silverlight-Control zugewiesen (siehe Eventbehandlung). Ansonsten gilt: 2. Parameter des Handlers ist eine Instanz einer Argumentenliste aus folgenden Objekttypen:

ErrorEventArgs	für Fehlermeldungen zum Event
ParserErrorEventArgs	für Parserfehler
RuntimeErrorEventArgs	für Runtime-Fehler

ErrorEventArgs umfasst folgende Parameter

errorMessage	Fehlermeldung
errorType	Typ des Fehlers (Wert des Fehlertypes ist vom Typ ErrorType enumeration)
errorCode	Fehlercode

ParserErrorEventArgs umfasst folgende Parameter

charPosition	Position des 1. Zeichens	der Fehlerstelle im Quelltext
lineNumber	Nummer der Zeile	der Fehlerstelle im Quelltext
xamlFile	Dateiname der XAML-Datei als Quelltext,	der den Fehler verursacht hat
xmlAttribute	XML-Attribut im Quelltext,	das den Fehler verursacht hat
xmlElement	XML-Element im Quelltext,	das den Fehler verursacht hat

RuntimeErrorEventArgs umfasst folgende Parameter

charPosition	Position des 1. Zeichens	der Fehlerstelle im Quelltext
lineNumber	Nummer der Zeile	der Fehlerstelle im Quelltext
methodName	Bezeichner der Methode im Quelltext,	die den Fehler verursacht hat

Beispiel 1:

```

Sys.Silverlight.createObject(    "ErrorEvent.xaml",
                                pe,
                                "AgControl1",
                                {
                                    width:'1024',
                                    height:'530',
                                    inplaceInstallPrompt:false,
                                    background:'white',
                                    isWindowless:'false',
                                    framerate:'24',
                                    version:'0.8'
                                },
                                {onError:"OnErrorEventHandler",
                                  // Set the onError Handler
                                  onLoad:null
                                },
                                null
                                );

```

Beispiel 2:

```
function OnOnErrorEventHandler(sender, errorArgs)
```



```

{
    // The error message to display.
    var errorMsg = "Silverlight Error: \n\n";
    // Error information common to all errors.
    errorMsg += "Error Type: "
               + errorArgs.errorType
               + "\n";
    errorMsg += "Error Message: "
               + errorArgs.errorMessage
               + "\n";
    errorMsg += "Error Code: "
               + errorArgs.errorCode
               + "\n";
    // Determine the type of error and add specific error information.
    switch(errorArgs.errorType)
    {
        case "RuntimeError": // Display properties specific to RuntimeErrorEventArgs.
            if (errorArgs.lineNumber != 0)
            {
                errorMsg += "Line: "
                           + errorArgs.lineNumber
                           + "\n";
                errorMsg += "Position: "
                           + errorArgs.charPosition
                           + "\n";
            }
            errorMsg += "MethodName: "
                       + errorArgs.methodName
                       + "\n";
            break;
        case "ParserError": // Display properties specific to ParserErrorEventArgs.
            errorMsg += "Xaml File: "
                       + errorArgs.xamlFile
                       + "\n";
            errorMsg += "Xml Element: "
                       + errorArgs.xmlElement
                       + "\n";
            errorMsg += "Xml Attribute: "
                       + errorArgs.xmlAttribute
                       + "\n";
            errorMsg += "Line: "
                       + errorArgs.lineNumber
                       + "\n";
            errorMsg += "Position: "
                       + errorArgs.charPosition
                       + "\n";
            break;
        default: break;
    }
    // Display the error message.
    alert(errorMsg);
}

```

## **ErrorEvent zu Aktionen (anhand von Methoden) des Silverlight-Objektes**

### **synchrone Aktion des Silverlight-Objektes**

Wird ein synchrones Event als Aktion des Silverlight-Objektes erzeugt, dann wird ein Fehler der Aktion wie folgt abgefangen - Zitat Microsoft:

If a synchronous method call fails and the method call is in a try/catch block, an error object, described in ECMA-262, is passed to the catch block and an onError event is not raised. Since an onError event is not raised, the onError event handler will not be called. If a synchronous method call fails and the method call is not inside a try/catch block, an onError event is raised and routed to the onError handler of the Silverlight Control. The ErrorType in the event arguments sent to the onError handler will be set to RuntimeError.



The only Silverlight specific error information returned to the catch block via the error object is the ErrorMessage and on Internet Explorer only, the ErrorCode. The message property on the error object is set to the error message.'

Try-Catch wird z.Z. nicht unterstützt von Safari/Mac. Daher sollte onError für alle synchrone Aktionen benutzt werden.

Beispiel:

```
// Get the MediaElement object.
var player = sender.findName("MediaPlayer");
try
{
    // Attempt to start the media.
    player.play();
}
catch(errorObj)
{
    // Perform necessary error handling.
    // Display the message of the Error Object.
    alert(errorObj.message);
}
```

### a-synchrone Aktion des Silverlight-Objektes

Der Eventhandler ist der Silverlight-Control-Eigenschaft zum asynchronen Event zuzuweisen. Wird kein Handler zugewiesen, dann wird im Fehlerfall onError-Event verwendet.

Beispiel:

XAML-Daten:

```
<MediaElement x:Name="MediaPlayer" MediaFailed="MediaFailedHandler" />
```

JavaScript:

```
function MediaFailedHandler(sender, args)
{
    // Create basic error message.
    var errorMsg = "\n Media Error Message \n" ;
    // Add Media information.
    errorMsg += "MediaElement Name: "
               + sender.Name
               + "\n";
    errorMsg += "Media File Name: "
               + sender.Source
               + "\n";
    // Display error information.
    alert(errorMsg);
}
```

## Silverlight ErrorEvent-Objekte

### ErrorEventArgs Object

Provides data for the onError event.

XAML	You cannot use this object in XAML.
Scripting	Cannot be created in script.

Properties	ErrorCode ErrorMessage, ErrorType
Methods	None
Events	None

### ErrorCode Property

Gets the error code associated with the error.

XAML	This property cannot be used in XAML
------	--------------------------------------





Scripting	value = object.errorCode
-----------	--------------------------

Property Value

integer

The error code of the event.

This property is read-only.

### **ErrorMessage Property**

Gets the error description associated with the error.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.errorMessage
-----------	-----------------------------

Property Value

string

The error message describing the error.

This property is read-only.

### **ErrorType Property**

Gets the ErrorType of the error associated with the event.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.errorType
-----------	--------------------------

Property Value

ErrorType

The type of the error that occurred, expressed as a value of the ErrorType enumeration.

This property is read-only.

### **ParserErrorEventArgs Object**

Provides data for XAML parser error events.

XAML	You cannot use this object in XAML.
------	-------------------------------------

Scripting	Cannot be created in script.
-----------	------------------------------

Properties

CharPosition ,  
 ErrorCode,  
 ErrorMessage,  
 ErrorType,  
 LineNumber,  
 XmlFile,  
 XmlAttribute,  
 XmlElement

Methods

None

Events

None

### **CharPosition (ParserErrorEventArgs) Property**

The character position within the line that the parser error occurred on.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.charPosition
-----------	-----------------------------

Property Value

integer

The character position the parser error occurred on.

This property is read/write.

### **ErrorCode Property**

Gets the error code associated with the error.

XAML	This property cannot be used in XAML
------	--------------------------------------



Scripting	value = object.errorCode
-----------	--------------------------

Property Value

integer

The error code of the event.

This property is read-only.

### ***ErrorMessage Property***

Gets the error description associated with the error.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.errorMessage
-----------	-----------------------------

Property Value

string

The error message describing the error.

This property is read-only.

### ***ErrorType Property***

Gets the ErrorType of the error associated with the event.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.errorType
-----------	--------------------------

Property Value

ErrorType

The type of the error that occurred, expressed as a value of the ErrorType enumeration.

This property is read-only.

### ***LineNumber (ParserEventArgs) Property***

Gets the line number at which the parser error occurred.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.lineNumber
-----------	---------------------------

Property Value

integer

The number of the line the parser error occurred on.

This property is read-only. The default value is 0.

### ***XamlFile Property***

Gets the name of the Xaml file that the parser error occurred in.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.xmlFile
-----------	------------------------

Property Value

String

The name of the Xaml file that the parser error occurred in.

This property is read-only.

### ***XmlAttribute Property***

Gets the xml attribute that the parser error occurred in.

XAML	This property cannot be used in XAML
------	--------------------------------------

Scripting	value = object.xmlAttribute
-----------	-----------------------------

Property Value

string

The xml attribute that the parser occurred in.

This property is read-only.



**XmlElement Property**

Gets the xml element that the parser error occurred in.

XAML	This property cannot be used in XAML
Scripting	value = object.xmlElement

Property Value

string

The xml element that the parser error occurred in.

This property is read-only.

**RuntimeErrorEventArgs Object**

Provides data for runtime error events.

XAML	You cannot use this object in XAML.
Scripting	Cannot be created in script.

Properties

CharPosition

ErrorCode

ErrorMessage

ErrorType

LineNumber

MethodName

Methods

None

Events

None

**CharPosition (RuntimeErrorEventArgs) Property**

The character position within the line that the runtime error occurred on.

XAML	This property cannot be used in XAML
Scripting	value = object.charPosition

Property Value

integer

The character position the runtime error occurred on.

This property is read/write.

**ErrorCode Property**

Gets the error code associated with the error.

XAML	This property cannot be used in XAML
Scripting	value = object.errorCode

Property Value

integer

The error code of the event.

This property is read-only.

**ErrorMessage Property**

Gets the error description associated with the error.

XAML	This property cannot be used in XAML
Scripting	value = object.errorMessage

Property Value

string

The error message describing the error.

This property is read-only.

**ErrorType Property**

Gets the ErrorType of the error associated with the event.



XAML	This property cannot be used in XAML
Scripting	value = object.errorType

Property Value

ErrorType

The type of the error that occurred, expressed as a value of the ErrorType enumeration.

This property is read-only.

### ***LineNumber (RuntimeErrorEventArgs) Property***

Gets the line number at which the runtime error occurred.

XAML	This property cannot be used in XAML
Scripting	value = object.lineNumber

Property Value

integer

The number of the line the runtime error occurred on.

This property is read-only. The default value is 0.

### ***MethodName Property***

Gets the name of the method associated with the runtime error.

XAML	This property cannot be used in XAML
Scripting	value = object.methodName

Property Value

String

The name of the method.

This property is read-only.



## Silverlight Objekte verwalten (Referenzierung und Modifizierung)

<http://msdn2.microsoft.com/en-us/library/bb188573.aspx>

### **Silverlight JavaScript Debugging im Internet Explorer**

Abhängig mit welcher Software Debugging durchgeführt wird, muss im Internet Explorer unter Internet Optionen - Erweitert das Scriptdebugging des IE berücksichtigt werden.

Der Debugger muss neben HTML-DOM auch das Silverlight-DOM kennen (HTML-DOM je nach Browser, Silverlight-DOM je nach Plugin).

### **Verwendung von DOM (Document Objekt Model)**

Es muss das HTML-DOM verwendet werden, das der Browser kennt. Z.B. kann im Internet Explorer der Wert eines ID-Attributes als Zeiger verwendet werden - andere Browser können den Zeiger nur anhand getElementById() ermitteln (IE unterstützt auch getElementById() da eine DOM-Funktion ist).

Es muss das Silverlight-DOM verwendet werden, das der Browser per Plugin von Microsoft erhalten hat. Dabei kann die Implementation des DOM browserspezifisch sein.

### **Document Object Model (DOM)**

<http://msdn2.microsoft.com/en-us/library/bb412399.aspx>

Document Object Model (DOM) ist die plattformneutrale Programmierungsschnittstelle anhand einer baumähnlichen Knotenstruktur aus Objektes und deren Eigenschaften, Methoden und Ereignissen. Diese Struktur kann anhand einer Programmiersprache (z.B. JavaScript) plattformbezogen verwendet werden (wegen der Struktur ist Programmierung möglich, wobei eine Laufzeitumgebung der Plattform benötigt wird z.B. eine Scriptmaschine mit Interpreter). DOM kann standardisiert sein. Tatsächlich weichen DOM vom Standard ab - z.B. Microsoft DHTML-DOM. Bekannte DOM-Anbieter sind u.a. Gecko (Netscape, Mozilla, Firefox, Opera (developer.mozilla.org)) und Microsoft. Auch wenn Prototyping eines DOM möglich ist, sind neuartige Browseraktionen nur durch Kombination der per DOM anhand von Objekten vordefinierten Browseraktion erzeugbar. Auch deshalb unterscheiden sich DOM (z.B. microsoftseigene DHTML-DOM-Abänderung BGSOUND als Objekt, das Sound auf einfache Weise erzeugt ohne spezielle Plugins und deren Eigenheiten - in Kombination mit mehreren BGSOUND-Objekten möglich).

### **Silverlight Document Object Model (Silverlight-DOM)**

Dieses DOM stamm von Microsoft und wird per Plugins in die Plattformen Windows und Macintosh integriert (Linux z.Z. nicht verfügbar). Ohne Nutzung des plattform- und browserspezifischen Plugins keine Nutzung des DOM. Das DOM ist u.a. per JavaScript nutzbar.

Nachfolgend eine Übersicht zum DOM, dessen Weiterentwicklung je Auffassungen von Microsoft andauert. Dabei gilt:

- Es gibt nur vordefinierte Objekte.
- Objekte haben vordefinierte Methoden, Eigenschaften und Events etc..
- Die Methode eines Objektes symbolisiert eine vordefinierte Aktion des Objektes.
- Die Erzeugung neuer Objekte und Aktionen ist nur anhand vordefinierter möglich, da letztere auch eine Aktion im Browser verursachen.
- DOM kann browserspezifisch implementiert sein (je nach Auffassung von Microsoft):
  - beim Internet Explorer per Active-X-Control von Windows
  - ansonsten per Plugin (Windows, Mac-OS, kein Linux) für bestimmte Browser.

### **Silverlight Objekte**

<http://msdn2.microsoft.com/en-us/library/bb412366.aspx>

Objekte werden zwingend mit XAML-Script deklariert.  
Silverlight befindet sich in permanenter Veränderung - also auch das Silverlight-DOM.

wichtige Objekte sind z.B.



Canvas	Elternbereich in dem andere Objekte als Kinder erzeugt werden können (relativ zum Bereich), Canvas-Eigenschaften sind in anderen Objekten als deren Eigenschaften hinterlegt.
Downloader	implementiert Downloadfähigkeiten in Objekt (außer Downloader selbst)
Ellipse	rendert (zeichnet) eine Ellipse
Glyphs	Satz von Glyphen erzeugen für Anzeige von fixem Text
Image	rendert ein Bild
ImageBrush	Bildbereich ausmalen
Line	Linie zwischen 2 Punkten zeichnen
LineBreak	Zeilenumbruch in einem Textblock-Objekt
MediaElement	Mediendaten anzeigen (Audio, Video, Sprachausgabe per Audio und nicht per Microsoft Text To Speech)
Path	verbundene Linien und Kurven als Pfad anzeigen, der nicht am Pfadbeginn endet
Polygon	verbundene Linien und Kurven als Pfad anzeigen, der am Pfadbeginn endet
Polyline	verbundene gerade Linien anzeigen
Rectangle	Polygon als Rechteck anzeigen
Run	ausgewählte Sektion eines Textes (formatiert oder unformatierter Text)
Shape	abstraktes Objekt für Objekte mit Kanten wie Ellipse, Polygon, and Rectangle
TextBlock	abstraktes Objekt als Basis-Klasse für Elemente mit Kanten wie Ellipse, Polygon und Rectangle
UIElement	abstraktes Objekt für alle anzeigbaren (renderbaren) Objekte

Animation  
ArcSegment  
BeginStoryboard  
BezierSegment  
Boolean  
Brush  
Canvas  
Collection  
Color  
ColorAnimation  
ColorAnimationUsingKeyFrames  
ColorKeyFrame  
ColorKeyFrameCollection  
DependencyObject  
DiscreteColorKeyFrame  
DiscreteDoubleKeyFrame  
DiscretePointKeyFrame  
Double  
DoubleAnimation  
DoubleAnimationUsingKeyFrames  
DoubleCollection  
DoubleKeyFrame  
DoubleKeyFrameCollection  
Downloader  
DrawingAttributes  
Duration  
Ellipse  
EllipseGeometry  
EventArgs  
EventTrigger  
FontFamily  
Geometry  
GeometryCollection  
GeometryGroup  
Glyphs  
GradientBrush  
GradientStop  
GradientStopCollection



Image  
ImageBrush  
InkPresenter  
KeyboardEventArgs  
KeyFrame  
KeySpline  
KeyTime  
Line  
LinearColorKeyFrame  
LinearDoubleKeyFrame  
LinearGradientBrush  
LinearPointKeyFrame  
LineBreak  
LineGeometry  
LineSegment  
Matrix  
MatrixTransform  
MediaAttribute  
MediaAttributeCollection  
MediaElement  
MouseEventArgs  
ParserErrorEventArgs  
Path  
PathFigure  
PathFigureCollection  
PathGeometry  
PathSegment  
PathSegmentCollection  
Point  
PointAnimation  
PointAnimationUsingKeyFrames  
PointCollection  
PointKeyFrame  
PointKeyFrameCollection  
PolyBezierSegment  
Polygon  
Polyline  
PolyLineSegment  
PolyQuadraticBezierSegment  
QuadraticBezierSegment  
RadialGradientBrush  
Rect  
Rectangle  
RectangleGeometry  
RepeatBehavior  
ResourceCollection  
RotateTransform  
Run  
RuntimeErrorEventArgs  
ScaleTransform  
Shape  
SilverlightControl  
Size  
SkewTransform  
SolidColorBrush  
SplineColorKeyFrame  
SplineDoubleKeyFrame  
SplinePointKeyFrame  
Storyboard  
Stretch  
Stroke



StrokeCollection  
StylusInfo  
StylusPoint  
StylusPointCollection  
TextBlock  
Timeline  
TimeLineCollection  
TimelineGroup  
TimelineMarker  
TimeLineMarkerCollection  
TimeLineMarkerEventArgs  
TimeSpan  
Transform  
TransformCollection  
TransformGroup  
TranslateTransform  
TriggerAction  
TriggerActionCollection  
TriggerCollection  
UIElement  
UIElementCollection  
VideoBrush

### ***Silverlight Methoden***

Silverlight befindet sich in permanenter Veränderung - also auch das Silverlight-DOM.

Abort  
Add  
AddEventListener  
AddStylusPoints  
Begin  
CaptureMouse  
Clear  
CreateFromXAML  
CreateFromXAMLDownloader  
CreateObject  
FindName (Silverlight Control)  
FindName (UIElement)  
GetBounds (Stroke)  
GetBounds (StrokeCollection)  
GetHost  
GetItem  
GetItemByName  
GetParent  
GetPosition  
GetResponseText  
GetStylusInfo  
GetStylusPoints  
GetValue  
HitTest (Stroke)  
HitTest (StrokeCollection)  
Insert  
Open  
Pause (MediaElement)  
Pause (Storyboard)  
Play  
ReleaseMouseCapture  
Remove  
RemoveAt  
RemoveEventListener  
Resume





Send  
SetFontSource  
SetSource  
SetValue  
Stop (MediaElement)  
Stop (Storyboard)

### ***Silverlight Eigenschaften (Properties)***

Silverlight befindet sich in permanenter Veränderung - also auch das Silverlight-DOM.

Actions  
ActualHeight (Silverlight Control)  
ActualHeight (TextBlock)  
ActualWidth (Silverlight Control)  
ActualWidth (TextBlock)  
AlignmentX  
AlignmentY  
Angle  
AngleX  
AngleY  
Attributes  
AutoPlay  
AutoReverse  
Background (Silverlight Control)  
Background (UIElement)  
Balance  
BeginTime  
BufferingProgress  
BufferingTime  
By (Animation)  
By (ColorAnimation)  
By (DoubleAnimation)  
By (PointAnimation)  
Canvas.Left  
Canvas.Top  
Canvas.ZIndex  
Center (EllipseGeometry)  
Center (RadialGradientBrush)  
CenterX (RotateTransform)  
CenterX (ScaleTransform)  
CenterX (SkewTransform)  
CenterY (RotateTransform)  
CenterY (ScaleTransform)  
CenterY (SkewTransform)  
CharPosition (ParserEventArgs)  
CharPosition (RuntimeErrorEventArgs)  
Children (GeometryGroup)  
Children (Storyboard)  
Children (TimelineGroup)  
Children (TransformGroup)  
Children (UIElement)  
Clip  
Color (DrawingAttributes)  
Color (GradientStop)  
Color (SolidColorBrush)  
ColorInterpolationMode  
Count  
Ctrl (KeyboardEventArgs)  
Ctrl (MouseEventArgs)  
CurrentState  
Cursor



Data  
DeviceType  
DownloadProgress (Downloader)  
DownloadProgress (MediaElement)  
DownloadProgress (UIElement)  
DrawingAttributes  
Duration  
EnableFramerateCounter  
EnableHtmlAccess  
EnableRedrawRegions  
EndPoint (LinearGradientBrush)  
EndPoint (LineGeometry)  
ErrorCode  
ErrorMessage  
ErrorType  
Figures  
Fill (Glyphs)  
Fill (Shape)  
FillBehavior  
FillRule (PathGeometry)  
FillRule (Polygon)  
FillRule (Polyline)  
FontFamily  
FontRenderingEmSize  
FontSize  
FontStretch  
FontStyle  
FontUri  
FontWeight  
Foreground  
From (Animation)  
From (ColorAnimation)  
From (DoubleAnimation)  
From (PointAnimation)  
FullScreen  
GradientOrigin  
GradientStops  
Height (DrawingAttributes)  
Height (Size)  
Height (UIElement)  
ImageSource  
Indices  
InitParams  
IsClosed  
IsFilled  
IsHitTestVisible  
IsInverted  
IsLargeArc  
IsLoaded  
IsMuted  
Key  
KeyFrames (ColorAnimationUsingKeyFrames)  
KeyFrames DoubleAnimationUsingKeyFrames)  
KeyFrames PointAnimationUsingKeyFrames)  
KeySpline (SplineColorKeyFrame)  
KeySpline (SplineDoubleKeyFrame)  
KeySpline (SplinePointKeyFrame)  
KeyTime  
LineNumber (ParserErrorEventArgs)  
LineNumber (RuntimeErrorEventArgs)  
M11



M12  
N21  
M22  
MappingMode  
Marker  
Markers  
MaxFrameRate  
MethodName  
Name  
NaturalDuration  
NaturalVideoHeight  
NaturalVideoWidth  
Offset  
OffsetX.htm  
Offsety.htm  
Opacity (Brush)  
Opacity (UIElement)  
OpacityMask  
OriginX  
OriginY  
OutlineColor  
PlatformKeyCode  
Point (ArcSegment)  
Point (LineSegment)  
Point1 (BezierSegment)  
Point1 (QuadraticBezierSegment)  
Point2 (BezierSegment)  
Point2 (QuadraticBezierSegment)  
Point3  
Points (PolyBezierSegment)  
Points (Polygon)  
Points (Polyline)  
Points (PolyLineSegment)  
Points (PolyQuadraticBezierSegment)  
Position  
PressureFactor  
RadiusX (EllipseGeometry)  
RadiusX (RadialGradientBrush)  
RadiusX (Rectangle)  
RadiusX (RectangleGeometry)  
RadiusY (EllipseGeometry)  
RadiusY (RadialGradientBrush)  
RadiusY (Rectangle)  
RadiusY (RectangleGeometry)  
Rect  
RelativeTransform  
RenderTransform  
RenderTransformOrigin  
RepeatBehavior  
Resources  
RotationAngle  
RoutedEvent  
ScaleX  
ScaleY  
Segments  
Shift (KeyboardEventArgs)  
Shift (MouseEventArgs)  
Size  
Source (Image)  
Source (MediaElement)  
Source (Silverlight Control)



SourceName  
SpeedRatio  
SpreadMethod  
StartPoint (LinearGradientBrush)  
StartPoint (LineGeometry)  
StartPoint (PathFigure)  
Status  
StatusText  
Storyboard  
Stretch (Image)  
Stretch (ImageBrush)  
Stretch (MediaElement)  
Stretch (Shape)  
Stretch (VideoBrush)  
Stroke  
StrokeDashArray  
StrokeDashCap  
StrokeDashOffset  
StrokeEndLineCap  
StrokeLineJoin  
StrokeMiterLimit  
Strokes  
StrokeStartLineCap  
StrokeThickness  
StyleSimulations  
StylusPoints  
SweepDirection  
TargetName  
TargetProperty  
Text (TextElement)  
Text (TimelineMarker)  
TextDecorations  
TextWrapping  
Time  
To (Animation)  
To (ColorAnimation)  
To (DoubleAnimation)  
To (PointAnimation)  
Transform (Brush)  
Transform (Geometry)  
Transform (MatrixTransform)  
Triggers  
Type (TimelineMarker)  
UnicodeString  
URI  
Value (ColorKeyFrame)  
Value (DoubleKeyFrame)  
Value (KeyFrame)  
Value (MediaAttribute)  
Value (PointKeyFrame)  
Version  
Visibility  
Volume  
Width (DrawingAttributes)  
Width (Size)  
Width (UIElement)  
Windowless  
X (Point)  
X (StylusPoint)  
X (TranslateTransform)  
X1



X2  
XamlFile  
XmlAttribute  
XmlElement  
Y (Point)  
Y (StylusPoint)  
Y (TranslateTransform)  
Y1  
Y2

### ***Silverlight Ereignisse (Events)***

Silverlight befindet sich in permanenter Veränderung - also auch das Silverlight-DOM.

BufferingProgressChanged  
Completed (Downloader)  
Completed (Storyboard)  
CurrentStateChanged  
DownloadProgressChanged (Downloader)  
DownloadProgressChanged (MediaElement)  
GotFocus  
KeyDown  
KeyUp  
Loaded  
LostFocus  
MarkerReached  
MediaEnded  
MediaFailed  
MediaOpened  
MouseEnter  
MouseLeave  
MouseLeftButtonDown  
MouseLeftButtonUp  
MouseMove  
OnError  
OnFullScreenChange  
OnLoad  
OnResize

### ***Silverlight Enumerations***

Silverlight befindet sich in permanenter Veränderung - also auch das Silverlight-DOM.

AlignmentX  
AlignmentY  
BrushMappingMode  
ColorInterpolationMode  
DeviceType  
ErrorType  
FillBehavior  
FillRule  
FontStretch  
FontStyle  
FontWeight  
GradientSpreadMethod  
Key  
MediaState  
MouseCursor  
PenLineCap  
PenLineJoin  
SpreadMethod  
Stretch  
StyleSimulations



SweepDirection  
TextDecorations  
TextWrapping  
TileMode  
Visibility



**Referenzierung per JavaScript - Beispiele**

anhand HTML-DOM

übliche Funktionen und Collectionen z.B. getElementById()

anhand Silverlight-DOM

passend zum Objekt

Beispiel: XAML-Daten adressieren

```
<!-- Visual progress indicator -->
<Canvas Canvas.Top="70">
    <Rectangle      x:Name="progressRectangle"
                    Canvas.Left="20"
                    Height="10"
                    Width="0"
                    Fill="Maroon"
    />
    <Rectangle      Canvas.Top ="-1"
                    Canvas.Left="19"
                    Height="12"
                    Width="202"
                    StrokeThickness="1"
                    Stroke="Black"
    />
    <TextBlock      x:Name="progressText"
                    Canvas.Top ="-4"
                    Canvas.Left="230"
                    Text="0%" FontSize="12"
    />
</Canvas>
```

```
Methode var zeiger=findName(Name_laut_x:Name);
           //      liefert Zeiger
           //      Anzeige des Typs des Objektes per alert(zeiger.toString());
```

Beispiel: Wert einer Objekt-Eigenschaft (Properties) adressieren

per Methoden GetValue() für Wert lesen

SetValue() für Wert schreiben

per objekt\_zeiger.eigenschaft für Lesen bzw. Schreiben (objekt\_zeiger wird z.B. per  
getElementById() ermittelt)

Lesen:

```
var opacity = objekt_zeiger.getValue("opacity");
opacity = objekt_zeiger.opacity;           // per Zeiger
alert("Opacity = " + opacity);
```

Schreiben:

```
objekt_zeiger.setValue("Canvas.Top", 40);
objekt_zeiger ["Canvas.Top"] = 40;        // per Zeiger
```

Beispiel: Objekt hinzufügen

per Methode Add()

```
var textBlock;
function onLoaded(sender, eventArgs)
{
    // Retrieve the id of the control.
    var control = sender.getHost();
    // Define and create a XAML fragment.
    var xamlFragment = '<TextBlock Canvas.Top="200"'
        + ' Opacity=".5" Text="Click for more info" />';
```



```

        textBlock = control.content.createFromXaml(xamlFragment);
        // Add the TextBlock to the root Canvas object.
        sender.children.add(textBlock);
    }

    // Toggle the Opacity property for the TextBlock.
    function onToggle()
    {
        if (textBlock.opacity)
        { textBlock.opacity = 0; }
        else { textBlock.opacity = 1; }
    }

    // Add three overlapping Rectangle objects.
    function AddItems(rootCanvas)
    {
        var control = rootCanvas.getHost();
        var xamlFragment = '<Rectangle Fill="Maroon" Canvas.Top="20" Canvas.Left="20"'
            + ' Height="100" Width="100" />';
        var rectangle_1 = control.content.createFromXaml(xamlFragment);
        rootCanvas.children.add(rectangle_1);    // an das Ende der Kinderkette anhängen
        var xamlFragment = '<Rectangle Fill="LightBlue" Canvas.Top="40" Canvas.Left="40"'
            + ' Height="100" Width="100" />';
        var rectangle_2 = control.content.createFromXaml(xamlFragment);
        rootCanvas.children.add(rectangle_2);    // an das Ende der Kinderkette anhängen
        var xamlFragment = '<Rectangle Fill="Teal" Canvas.Top="60" Canvas.Left="60"'
            + ' Height="100" Width="100" />';
        var rectangle_3 = control.content.createFromXaml(xamlFragment);
        rootCanvas.children.add(rectangle_3);    // an das Ende der Kinderkette anhängen

        // das zuletzt erzeugte Objekt wird als oberstes gerendert (Lage auf der Z-Achse)
        // wenn ZIndex-Eigenschaft verfügbar, so Lage auf Z-Achse einstellbar
    }

```

Insert-Methode fügt in die Kette der Kinder ein

Kinder sind per RemoveXX-Methoden und Clear-Methode entfernbar.

## Silverlight-DOM-Methoden - Beispiele

### Add Method

Adds an object to the end of the collection.

XAML	You cannot use methods in XAML.
Scripting	object.add(value)

#### Parameters

value	Object The object to add to the collection.
-------	--

#### Return Value

Returns the zero-based index of the inserted object in the collection if successful; otherwise, returns null.

#### Applies To

ColorKeyFrameCollection  
 DoubleKeyFrameCollection  
 GradientStopCollection  
 PathFigureCollection  
 PathSegmentCollection  
 PointKeyFrameCollection  
 TriggerActionCollection  
 TriggerCollection  
 UIElementCollection





**Insert Method**

Inserts an object into the collection at the specified index.

XAML	You cannot use methods in XAML.
Scripting	object.insert(index,value)

**Parameters**

index	Integer The index at which the object should be inserted.
value	Object The object to insert into the collection.

**Return Value**

None

**Applies To**

ColorKeyFrameCollection,  
DoubleKeyFrameCollection,  
GradientStopCollection,  
PathFigureCollection,  
PathSegmentCollection,  
PointKeyFrameCollection,  
TriggerActionCollection,  
TriggerCollection,  
UIElementCollection

**Remove Method**

Removes the specified object from the collection.

XAML	You cannot use methods in XAML.
Scripting	object.remove(value)

**Parameters**

value	Object The object to remove from the collection.
-------	---

**Return Value**

Boolean

true if value was removed from the collection; otherwise, false.

**Applies To**

ColorKeyFrameCollection,  
DoubleKeyFrameCollection,  
GradientStopCollection,  
PathFigureCollection,  
PathSegmentCollection,  
PointKeyFrameCollection,  
TriggerActionCollection,  
TriggerCollection,  
UIElementCollection

**RemoveAt Method**

Removes the object at the specified index.

XAML	You cannot use methods in XAML.
Scripting	value = object.removeAt(index)

**Parameters**

index	Object The zero-based index of the object to remove.
-------	---

**Return Value**

A reference to the removed item from the collection if successful; otherwise, returns null.

**Applies To**

ColorKeyFrameCollection,  
DoubleKeyFrameCollection,  
GradientStopCollection,  
PathFigureCollection,  
PathSegmentCollection,  
PointKeyFrameCollection,  
TriggerActionCollection,  
TriggerCollection,  
UIElementCollection

### **Clear Method**

Removes all objects from the collection.

XAML	You cannot use methods in XAML.
Scripting	object.clear()

#### Return Value

Boolean

true if all the objects were removed from the collection; otherwise, false.

#### Applies To

ColorKeyFrameCollection,  
DoubleKeyFrameCollection,  
GradientStopCollection,  
PathFigureCollection,  
PathSegmentCollection,  
PointKeyFrameCollection,  
TriggerActionCollection,  
TriggerCollection,  
UIElementCollection



## Silverlight Objekte (Auswahl)

Silverlight befindet sich in permanenter Veränderung - also auch das Silverlight-DOM.

### ***Silverlight DependencyObject Object (Abstract)***

Provides Silverlight property system services.

XAML	Cannot be used in XAML.
Scripting	Cannot instantiate an abstract object.
Properties	Name
Methods	GetValue SetValue

#### **GetValue Method**

Gets the value of the specified property.

XAML	You cannot use methods in XAML.
Scripting	value = object.GetValue(propertyName)

Parameters

propertyName	Object The property to get the value of.
--------------	---

Return Value

Object  
The value of the specified property.

#### **SetValue Method**

Sets the specified property or event to the specified value.

XAML	You cannot use methods in XAML.
Scripting	object.SetValue(propertyName, value)

Parameters

propertyName	Object The name of the dependency property to set.
value	Object The value to set the property to.

Return Value

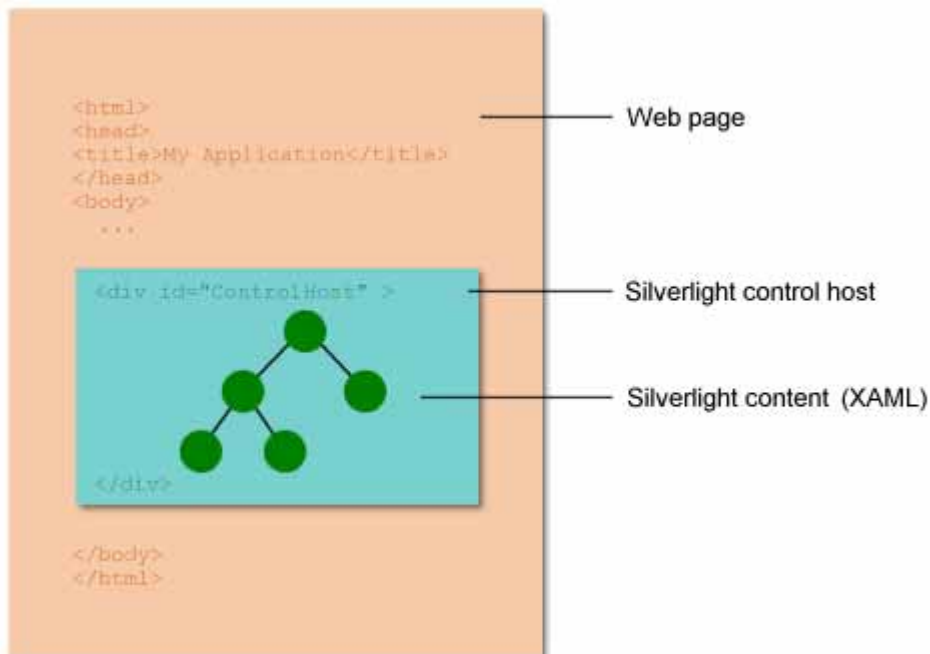
Returns value if successful; otherwise, returns null.



**Silverlight Control Object**

Represents the Silverlight control hosted in the browser.

HTML	<OBJECTid=silverlightObjectName ... />
Scripting	This object cannot be created using scripting.
Properties	ActualHeight ActualWidth Background EnableFramerateCounter EnableHtmlAccess EnableRedrawRegions FullScreen InitParams IsLoaded MaxFrameRate Source Version Windowless
Methods	CreateFromXaml CreateFromXamlDownloader CreateObject FindName
Events	OnError OnFullScreenChange OnLoad OnResize



**Silverlight Control-Object Canvas**

<http://msdn2.microsoft.com/en-us/library/bb188312.aspx>

Diese Objekt ist z.T. Basis-Objekt für andere Objekte: Canvas-Eigenschaften tauchen als Eigenschaften andere Objekte auf.

Defines an area within which you can explicitly position child elements by using coordinates relative to the Canvas area.

XAML	<Canvas .../>
Scripting	To create an object using scripting, see CreateFromXAML.
Properties	Background Canvas.Left Canvas.Top Canvas.ZIndex Children Clip Cursor Height IsHitTestVisible Name Opacity OpacityMask RenderTransform RenderTransformOrigin Resources Triggers Visibility Width
Methods	AddEventListener CaptureMouse FindName GetHost GetParent GetValue ReleaseMouseCapture RemoveEventListener SetValue
Events	GotFocus KeyDown KeyUp Loaded LostFocus MouseEnter MouseLeave MouseLeftButtonDown MouseLeftButtonUp MouseMove



**Silverlight Control-Objekt Donwloader**

<http://msdn2.microsoft.com/en-us/library/bb232870.aspx>

Represents the set of download functionality for a Silverlight control.

HTML	You cannot use this object in HTML.	
Scripting	To create an object using scripting, see CreateObject.	
Properties	DownloadProgress Status StatusText URI	
Methods	Abort GetResponseText Open Send	
Events	Completed	Occurs when the state of the download request has changed.
	DownloadProgressChanged	Occurs while content is being downloaded during a download request.

Das Objekt dient zur Implementierung der Downloadfähigkeit eines anderen Objektes. Daten wie XAML, Script, Media, Bilder, ZIP als Download Packages) etc. können geladen werden, wobei der Download komplett überwacht wird: Es wird dabei die Technik des XMLHttpRequests verwendet (im Internet Explorer ab Version 7 als HTML-DOM-Element verfügbar, davor als Active-X-Control): Synchroner und asynchroner.

Beispiel:

```
// Event handler for initializing and executing a download request.
function onMouseLeftButtonUp(sender, eventArgs)
{
    // Retrieve a reference to the control.
    var control = sender.getHost();
    // Create a Downloader object.
    var downloader = control.createObject("downloader");
    // Add DownloadProgressChanged and Completed events.
    downloader.addEventListener("downloadProgressChanged",
        "onDownloadProgressChanged");
    downloader.addEventListener("completed", "onCompleted");
    // Initialize the Downloader request.
    downloader.open("GET", "promo.png", true);
    // Execute the Downloader request.
    downloader.send();
}
```

**Anzeige eines Fortschrittsbalkens des Downloads - Beispiel**

Die Anzeige muss programmiert werden.

XAML-Daten

```
<!-- Visual progress indicator -->
<Canvas Canvas.Top="70">
    <Rectangle
        x:Name="progressRectangle"
        Canvas.Left="20"
        Height="10"
        Width="0"
        Fill="Maroon"
    />
    <Rectangle
        Canvas.Top ="-1"
        Canvas.Left="19"
        Height="12"
        Width="202"
        StrokeThickness="1"
        Stroke="Black"
```



```

/>
<TextBlock      x:Name="progressText"
                Canvas.Top ="-4"
                Canvas.Left="230"
                Text="0%"
                FontSize="12"
            />
</Canvas>

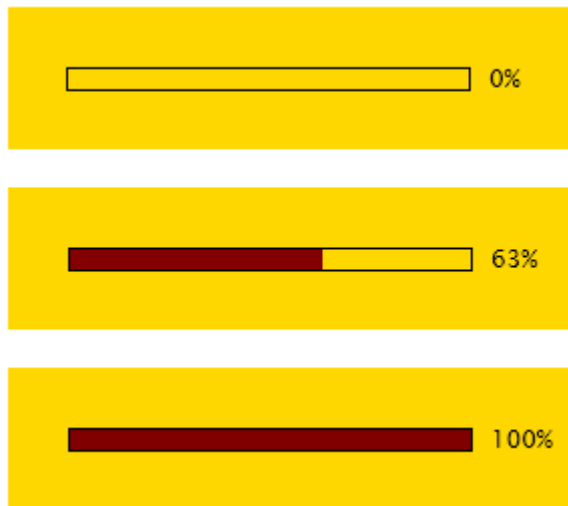
```

Eventhandler:

```

// Event handler for updating visual progress indicator
function onDownloadProgressChanged(sender, eventArgs)
{
    // Calculate the downloaded percentage.
    var percentage = Math.floor(sender.downloadProgress * 100);
    // Update the Rectangle and TextBlock objects of the visual progress indicator.
    progressText.text = percentage + "%";
    progressRectangle.width = percentage * 2;
}

```



### Zugriff auf gedownloadete Daten (nach Ende des Downloads) - Beispiel

Download von JavaScript-Datei button.js  
 XAML-Script-Fragment das einen Button definiert (Variante 1)  
 XAML-Datei button.xaml die einen Button definiert (Variante 2)

```

// Event handler for the Completed event.
function onCompleted(sender, eventArgs)
{
    // If HTTP status code indicates "OK", determine action based on URI property.
    if ((sender.status == 200) && (sender.uri == "OK_button.js"))
    {
        // Evaluate JavaScript code -- causes code to be memory-resident.
        eval(sender.getResponseText());
    }
    if ((sender.status == 200) && (sender.uri == "OK_button.xaml"))
    {
        var control = sender.getHost();
        var rootCanvas = sender.findName("rootCanvas");

        // Variante 1 Retrieve downloaded XAML content.
        var xamlFragment = sender.getResponseText();
        // and create an object reference.
        var button = control.content.createFromXaml(xamlFragment);
        // and add downloaded XAML content to the control.
    }
}

```



```

        rootCanvas.children.add(button);

        // Variante 2    Retrieve downloaded XAML content and create an object
        //                  reference.
        var xamlFragment =
            control.content.createFromXamlDownloader(sender, "");
        //                  and add downloaded XAML content to the control.
        root.children.add(xamlFragment);
    }
}

```

### Abbruch des gestarteten Downloads - Beispiel

per Methode Abort() nach erfolgtem Senden per Methode Send()

Abort() setzt folgende Eigenschaften zurück

DownloadProgress	auf 0
Status	auf 0
StatusText	auf Leerkette

```

// Cancel download on mouse click.
function onLeftMouseButtonUp(sender, eventArgs)
{ // Cancel Downloader activity and restore state. downloader.abort(); }

```

### Daten in ZIP-Datei - Beispiel

XAML-Daten die das Layout mehrerer Objekte mit je eigenem x:Name beschreiben

z.B. XAML-Datei Zipp\_Package.xaml

ZIP-gepackte Daten der per XAML beschriebenen Objekte: Pro Beschreibung 1 Datei im ZIP-Bestand

z.B. Picture1.png für XAML-Beschreibung mit x:Name "Zipp\_Package\_Part1"

XAML-Daten und Daten der beschriebenen Objekte liegen gemeinsam in der ZIP-Datei

z.B. XAML-Datei Zipp\_Package.xaml und Picture1.png.

Für den Zugriff muss aber die ZIP-Datei entpackt sein also wie bei Einzeldownloads.

Die logische Zuordnung der XAML-Beschreibung zur Datei erfolgt per Methode setSource(), die die Eigenschaft Source manipuliert.

```

function onDownloadCompleted(sender, eventArgs)
{
    // Determine whether the download was successful.
    if (currentDownloadProgress != 1)
    {
        alert("Failed to succesfully download zip file");
        return;
    }
    var control = sender.getHost();
    // Retrieve the XAML content from the downloaded XAML-package file.
    var Zipp_PackageXaml = sender.getResponseText("Zipp_Package.xaml");
    // Create the objects from the XAML content.
    var Zipp_Package = control.content.createFromXaml(Zipp_PackageXaml);
    // insert downloaded XAML content to the control.
    sender.findName("rootCanvas ").children.insert(0, Zipp_Package);
    // sender.findName liefert Zeiger auf Root des Canvas
    // sender.findName(..).children liefert Zeiger auf children-Collection der Root
    // sender.findName(..).children.insert() fügt Objekt in die Root ein
    // Retrieve a reference to the Image object representing by x:Name "Zipp_Package_Part1"
    var Zipp_Package_Part1_Image = sender.findName("Zipp_Package_Part1");
    // set logical XAML-content to physical file-content
    Zipp_Package_Part1_Image.setSource(sender, "Picture1.png");
}

```

### Verwaltung von HTTP-Fehler während Download - Beispiel

ähnelt der vom XMLHttpRequest-Objekt





z.B. per Eigenschaften  
 Status  
 StatusText

Beispiele für HTTP-Status:

Status code	Status text	Description
200	OK	erfolgreicher Zugriff auf HTTP-Server
403	Forbidden	HTTP-Server verbietet sich den Zugriff
404	Not Found	URI des Zugriffs auf HTTP-Server nicht findbar

// Event handler for the Completed event.

Beispiel für Statusprüfung

```
function onCompleted(sender, eventArgs)
{
    // Determine whether the download was successful.
    if (sender.status == 200)
    { // Retrieve downloaded content. }
    else { // Perform error handling actions. }
}
```

## Events - Übersicht

### Completed Event

Occurs when the state of the download request has changed.

HTML	You cannot use this property in HTML.
Scripting	downloaderObject.addEventListener("completed", "eventhandlerFunction")

Event Handler Parameters

sender	Object Identifies the object that invoked the event.
eventArgs	Object This parameter is always set to null.

Event Information

The Completed event occurs after the download request has been executed by using the Send method of the Downloader object. The Status and StatusText properties reflect the HTTP status code values for the completed download request.

Remarks

The Completed event occurs when the state of the download request has changed. The following JavaScript example shows how to define a Completed event handler function that accesses the downloaded content:

### DownloadProgressChanged (Downloader) Event

Occurs while content is being downloaded during a download request.

HTML	You cannot use this property in HTML.
Scripting	downloaderObject.addEventListener("downloadProgressChanged", "eventhandlerFunction")

Event Handler Parameters

sender	Object Identifies the object that invoked the event.
eventArgs	Object This parameter is always set to null.

Event Information

The DownloadProgressChanged event can be used monitor the progress of a download request. The DownloadProgressChanged event occurs whenever the percentage of total content downloaded increases by 0.05 or more, or reaches 1.0. The Completed event occurs when the state of the download request has changed.



## Methoden - Übersicht

### **Abort Method**

Cancels the downloader request and resets properties to their initial state.

XAML	You cannot use methods in XAML.
Scripting	downloaderObject.abort()

#### Parameters

objectType	String The type of object to create.
------------	---

#### Return Value

None.

#### Remarks

The Abort method cancels the downloader request that has been executed by using the Send method of the Downloader object. In addition, the Abort method resets the following Downloader properties to their initial state:

Property name	Initial state
DownloadProgress	0
Status	0
StatusText	Empty string

### **CreateObject Method**

Creates a specified object and returns a reference to it.

XAML	You cannot use methods in XAML.
Scripting	silverlightObject.createObject(objectType)

#### Parameters

objectType	String The type of object to create.
------------	---

#### Return Value

An object reference if the object was successfully created.

#### Remarks

The Downloader object is the only object that can be specified as the objectType parameter for the CreateObject method. If you specify an invalid objectType parameter, the CreateObject method throws an exception. The following JavaScript example shows how to invoke the CreateObject method:

### **GetResponseText Method**

Gets a string representation of the downloaded data.

HTML	You cannot use this property in HTML.
Scripting	value = downloaderObject.GetResponseText(part)

#### Parameters

part	String The name of the specific part of the downloaded content package. When the downloaded content package is a Zip file, part refers to the contents of a filename within the downloaded content. If the downloaded content does not represent packaged content, set part to an empty string.
------	--

#### Return Value

String representation of the downloaded data.

#### Remarks

When a Downloader object request is successful, the GetResponseText method returns a representation of the downloaded content as a string value. The return value from GetResponseText can represent XAML content, JavaScript content, or a media asset, such as an image.

Note If you are downloading media assets for use by the Source property of Image and MediaElement objects,



you should use the SetSource method, as this is a more efficient way to transfer large-sized media assets.

### Open Method

Initializes the download request.

XAML	You cannot use methods in XAML.
Scripting	downloaderObject.Open(verb, uri, async)

#### Parameters

verb	String The type of download action. The only supported action is "GET".
uri	String The Uniform Resource Identifier (URI) of the content to perform the download action.
async	Boolean Determines whether to perform the download action asynchronously. The async parameter is optional and defaults to true if not specified.

#### Return Value

None.

#### Remarks

The Open method initializes the Downloader object by specifying the action to perform, the content to perform the action on, and whether to perform the action asynchronously.

### Send Method

Executes a download request.

XAML	You cannot use methods in XAML.
Scripting	downloaderObject.send()

#### Parameters

None

#### Return Value

None.

#### Remarks

The Send method executes the request to download data using the Downloader object functionality. The Send method executes either synchronously or asynchronously, depending on the value of the async parameter of the corresponding Open method invocation.

### SetSource Method (Methode des Objektes, das um Downloader-Objekt erweitert wurde)

Sets the source value of an object with downloaded content.

XAML	You cannot use methods in XAML.
Scripting	object.setSource(downloader, part)

#### Parameters

downloader	Downloader The object containing the downloaded content.
part	String The name of the specific part of the downloaded content package. When the downloaded content package is a Zip file, part refers to the contents of a filename within downloader. If the downloaded content does not represent packaged content, set part to an empty string.

#### Return Value

None

#### Remarks

The SetSource method can be used as a way of setting downloaded media content to the Source property of the Image and MediaElement objects, or the ImageSource property of the ImageBrush object. The first parameter, downloader, identifies the Downloader object representing the downloaded content. The second parameter, part, identifies the specific part to retrieve within the downloaded content. If the downloaded content does not represent packaged content, such as a Zip file, part must be set to an empty string.



Applies To

Image,  
ImageBrush,  
MediaElement

## Eigenschaften - Übersicht

### **DownloadProgress Property**

Gets a value that indicates the percentage of downloaded content.

HTML	You cannot use this property in HTML.
Scripting	value = downloaderObject.downloadProgress

Property Value

Double

Gets a value between 0 and 1 inclusive that represents the percentage amount of total content downloaded.

This property is read-only. The default value is 0.

Remarks

Once you have invoked the Send method on the Downloader object, the DownloadProgress property allows you to determine the percentage of total content downloaded. The return value is expressed as double value between 0.0 and 1.0 inclusive, with 0.0 representing no content downloaded, and 1.0 representing all content downloaded.

The DownloadProgressChanged event can be used monitor the progress of a download request. The DownloadProgressChanged event occurs whenever the percentage of total content downloaded increases by 0.05 or more, or reaches 1.0.

### **Status Property**

Gets the current status of the Downloader object request.

HTML	You cannot use this property in HTML.
Scripting	value = downloaderObject.status

Property Value

Integer

Gets the HTTP status code value that represents the current state of the Downloader object request.

This property is read-only. The default value is 0.

Remarks

The HTTP status code reflects the current state of the Downloader request as specified by the verb parameter in the Open method. For example, if the verb parameter is "GET" and the resulting download was successful, the value of Status is 200, and the value of StatusText is "OK".

The following table lists some of the more commonly received HTTP status codes. For more information, see HTTP Status Codes.

Status code	Status text	Description
200	OK	The request completed successfully.
403	Forbidden	The server understood the request, but is refusing to fulfill it.
404	Not Found	The server has not found anything matching the request URI.

### **StatusText Property**

Gets the current status of the Downloader object request.

HTML	You cannot use this property in HTML.
Scripting	value = downloaderObject.statusText

Property Value



String

Gets a string value that corresponds to the Status property value of a Downloader object request.

This property is read-only. The default value is an empty string.

Remarks

The HTTP status code reflects the current state of the Downloader request as specified by the verb parameter in the Open method. For example, if the verb parameter is "GET" and the resulting download was successful, the value of Status is 200, and the value of StatusText is "OK".

The following table lists some of the more commonly received HTTP status codes. For more information, see HTTP Status Codes.

Status code	Status text	Description
200	OK	The request completed successfully.
403	Forbidden	The server understood the request, but is refusing to fulfill it.
404	Not Found	The server has not found anything matching the request URI.

### URI Property

Gets the Uniform Resource Identifier (URI) of the download request.

HTML	You cannot use this property in HTML.
Scripting	value = downloaderObject.uri

Property Value

String

Gets a string value that corresponds to the URI of the Downloader object request.

This property is read-only. The default value is an empty string.

Remarks

When you invoke the Send method of the Downloader object, you specify a uri parameter that represents the data to be downloaded. The URI property corresponds to the value of the uri parameter of the Send method.

The URI property allows you to differentiate downloaded content when you use the share DownloadProgressChanged or Completed event handler functions for multiple Downloader object requests.

Note The URI value must reference a file that comes from the same site of origin as the instantiated Silverlight control. In addition, the URI file reference must be relative to the HTML file that defines the Silverlight control.



**Silverlight Control-Objekt MediaElement**

<http://msdn2.microsoft.com/en-us/library/bb188356.aspx>

Represents an object that contains audio, video, or both.

XAML	<MediaElement .../>
Scripting	To create an object using scripting, see CreateFromXAML.

Properties	Attributes, AutoPlay, Balance, BufferingProgress, BufferingTime, Canvas.Left, Canvas.Top, Canvas.ZIndex, Children, Clip, CurrentState, Cursor, DownloadProgress, Height, IsMuted, Markers, NaturalDuration, NaturalVideoHeight, NaturalVideoWidth, Opacity, OpacityMask, Position, RenderTransform, RenderTransformOrigin, Resources, Source, Stretch, Triggers, Volume, Width
Methods	AddEventListener, CaptureMouse, FindName, GetHost, GetParent, GetValue, Pause, Play, ReleaseMouseCapture, RemoveEventListener, SetValue, Stop
Events	BufferingProgressChanged, CurrentStateChanged, DownloadProgressChanged, Loaded, MarkerReached, MediaEnded, MediaFailed, MediaOpened, MouseEnter, MouseLeave, MouseLeftButtonDown,



MouseLeftButtonUp,  
MouseMove

Remarks

By default, the media defined by Source plays immediately after the MediaElement has loaded. To suppress the media from automatically starting, set the AutoPlay property to false.

Beispiel:

```
<MediaElement x:Name="media"
              Source="xbox.wmv"
              CurrentStateChanged="media_state_changed"
              Width="300"
              Height="300"
/>
```



**Silverlight Control-Objekt TextBlock**

<http://msdn2.microsoft.com/en-us/library/bb188394.aspx>

Represents a lightweight object for displaying single-line and multi-line multi-formatted text.

XAML	<TextBlock .../>
Scripting	To create an object using scripting, see CreateFromXAML.
Properties	ActualHeight, ActualWidth, Canvas.Left, Canvas.Top, Canvas.ZIndex, Clip, Cursor, FontFamily, FontSize, FontStretch, FontStyle, FontWeight, Foreground, Height, Name, Opacity, OpacityMask, RenderTransform, RenderTransformOrigin, Resources, Text, TextDecorations, TextWrapping, Triggers, Visibility, Width
Methods	AddEventListener, CaptureMouse, FindName, GetHost, GetParent, GetValue, ReleaseMouseCapture, RemoveEventListener, SetFontSource, SetValue
Events	Loaded, MouseEnter, MouseLeave, MouseLeftButtonDown, MouseLeftButtonUp, MouseMove

**Remarks**

The TextBlock object is the primary element for displaying text in Silverlight applications.

TextBlock rendering with default font properties

There are several interesting items to note about the display of the rendered text:

The default value of the FontSize property of the rendered TextBlock is 14.666 pixels, which is exactly 11 points.

The default value of the FontFamily property of the rendered TextBlock is "Lucida Sans Unicode, Lucida Grande".

No font files on the hosting Web server, or residing in the same directory location as the XAML content, are required to enable the default font.

Note: Any leading or trailing whitespace is not preserved when setting the Text property.





The Run and LineBreak objects can also be used to render text. The Run object is a text element that represents a discrete section of formatted or unformatted text. The LineBreak object represents an explicit new line in a TextBlock.

Beispiel 1 mit Text-Attribut:

```
<Canvas xmlns="http://schemas.microsoft.com/client/2007">
  <TextBlock
    Text="Hello, world!"
  />
</Canvas>
```

Beispiel 2 ohne Text-Attribut:

```
<Canvas xmlns="http://schemas.microsoft.com/client/2007">
  <TextBlock>
    Hello, world!
  </TextBlock>
</Canvas>
```



## XAML-Script - Überblick

### XAML-Scriptsprache

wird verwendet, um Elemente des User Interface (z.B. von Silverlight) zu beschreiben z.B. in  
Funktion (Aktion des Elementes)  
Eigenschaften (z.B. Daten, Datenbezüge, Layoutangaben)  
Events (Ereignisse der Aktion)

Die Beschreibungen werden von der Laufzeitumgebung der Scriptmaschine interpretiert und  
z.T. im User Interface angezeigt (gerendert) und in andere Aktionen des User Interface  
umgewandelt.

Die Beschreibungen müssen nicht portiert werden, da nur die Laufzeitumgebung passend zum  
Betriebssystem und zur Software sein muss, die das User Interface erzeugen.

ist keine Neuerung vielmehr technisches Hilfsmittel, Softwarekomponenten einfacher programmierbar zu  
machen oder alte Softwarekomponenten zu ersetzen (Script als Mittel zum Zweck).

kann nur insoweit vom Programmierer als akzeptabel gehalten werden, wie die Anwenderfreundlichkeit  
und die Nachfrage nach User Interfaces bestehen (inklusive Konkurrenz zu bestehenden  
Systemen z.B. Action Script von Adobe Flash (ehemals Macromedia). Im Falle von Microsoft  
werden sogar abgeschaltete Komponenten des Betriebssystems ersetzt (DirectX mit DirectDraw  
etc.), so dass ein (prinzipiell) abgeändertes Betriebssystem entsteht.).

bedeutet einigen Aufwand an Pflege und Erstellung (dafür sind Text-Dateien gut packbar), so dass der  
Eindruck einer Literaten-Script-Sprache entsteht, die zwar sehr gut (vordefinierte) Objekte und deren  
Abhängigkeiten beschreiben kann, aber enormen Code benötigt (inklusive Tippelaufwand).

## Deklaration von Objekten

### Syntax allgemein

Objekte werden wie folgt deklariert:

```
<objectName>  
</objectName>
```

Es gibt die Möglichkeit von /> als Ende-Tag

Bsp.: 

```
<objectName>  
    <objectName.property>  
        <propertyValue ... />  
    </objectName.property>  
</objectName>
```

## Abhängigkeiten der Objekte bei Silverlight

für Silverlight gibt es diverse vordefinierte Objekte im Silverlight-DOM

Objekte benötigen teilweise untereinander:

z.B. Canvas-Objekt ist Elternobjekt als Container für Kindobjekte.

Canvas-Kinder (Shapes-Elemente) zum Zeichnen und Malen per Vektorgraphik  
sind z.B.

Ellipse  
Rectangle  
Line  
Polyline  
Polygone  
Path

Das Ausmalen dieser Objekte wird anhand deren Erweiterung durchb  
Brush-Objekte implementiert:

SolidColorBrush  
LinearGradientBrush  
RadialGradientBrush



ImageBrush  
VideoBrush (siehe MediaElement)

z.B. <Canvas> <Rectangle> </Rectangle> </Canvas>

```
z.B. <Canvas>
      <Rectangle Width="100" Height="100">
        <Rectangle.Fill>
          <SolidColorBrush />
        </Rectangle.Fill>
      </Rectangle>
    </Canvas>
```

Es wird vererbt, so dass diverse Objekte kombinierbar sind zu komplexeren Objekten.

Es gibt rein abstracte Objekte nur zum Zweck der Vererbung.

## Beispiele für Deklarationen für Silverlight

In nachfolgenden XAML-Beispielen werden JavaScript-Funktionen der Eventhandler z.T. nicht angegeben, so dass diese Beispiele nur exemplarischer Art sind !

### Beispiel für Ellipse, Rectangle und Line

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Ellipse Height="200" Width="200" Canvas.Left="30" Canvas.Top="30"
    Stroke="Black" StrokeThickness="10" Fill="SlateBlue"/>
  <Rectangle Height="100" Width="100" Canvas.Left="5" Canvas.Top="5"
    Stroke="Black" StrokeThickness="10" Fill="SlateBlue"/>
  <Line X1="280" Y1="10" X2="10" Y2="280"
    Stroke="black" StrokeThickness="5"/>
</Canvas>
```

### Beispiel für Polyline und Polygon

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Polyline Points="150, 150 150, 250 250, 250 250, 150"
    Stroke="Black" StrokeThickness="10"/>
  <Polygon Points="10,10 10,110 110,110 110,10"
    Stroke="Black" StrokeThickness="10" Fill="LightBlue"/>
</Canvas>
```

### Beispiel für Path

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Path Data="M0,0 L11.5,0 11.5,30 5.75,40 0,30z"
    Stroke="Black" Fill="SlateBlue"
    Canvas.Left="10" Canvas.Top="10" />

  <Path Data="M 10,100 C 10,300 300,-200 250,100z"
    Stroke="Red" Fill="Orange"
    Canvas.Left="10" Canvas.Top="10" />
```



```
<Path Data="M 0,200 L100,200 50,50z"
  Stroke="Black" Fill="Gray"
  Canvas.Left="150" Canvas.Top="70" />
```

```
</Canvas>
```

### Beispiel für Ausmalen einer Ellipse

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <Ellipse Height="90" Width="90" Canvas.Left="10" Canvas.Top="10"
    Fill="black"/> <!-- SolidColorBrush by color name. -->

  <Ellipse Height="90" Width="90" Canvas.Left="110" Canvas.Top="10"
    Fill="#000000"/> <!-- SolidColorBrush by 6-digit hexadecimal notation. -->

  <Ellipse Height="90" Width="90" Canvas.Left="10" Canvas.Top="110"
    Fill= "#ff000000"/> <!-- SolidColorBrush by 8-digit hexadecimal notation. -->

  <Ellipse Height="90" Width="90" Canvas.Left="110" Canvas.Top="110">
    <Ellipse.Fill>

      <!-- SolidColorBrush by object element syntax. -->
      <SolidColorBrush Color="Black"/>
    </Ellipse.Fill>
  </Ellipse>
</Canvas>
```

### Beispiel für Ausmalen eines Rectangle

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <!-- Linear gradients -->
  <Rectangle Width="140" Height="70" Canvas.Left="10" Canvas.Top="10">
    <Rectangle.Fill>
      <LinearGradientBrush>
        <GradientStop Color="Yellow" Offset="0.0" />
        <GradientStop Color="Red" Offset="0.25" />
        <GradientStop Color="Blue" Offset="0.75" />
        <GradientStop Color="LimeGreen" Offset="1.0" />
      </LinearGradientBrush>
    </Rectangle.Fill>
  </Rectangle>

  <Rectangle Width="140" Height="70" Canvas.Left="155" Canvas.Top="10">
    <Rectangle.Fill>
      <LinearGradientBrush StartPoint="0,0" EndPoint="1,0">
        <GradientStop Color="Yellow" Offset="0.0" />
        <GradientStop Color="Red" Offset="0.25" />
        <GradientStop Color="Blue" Offset="0.75" />
        <GradientStop Color="LimeGreen" Offset="1.0" />
      </LinearGradientBrush>
    </Rectangle.Fill>
  </Rectangle>

  <!-- Radial gradients -->
  <Rectangle Width="140" Height="70" Canvas.Left="10" Canvas.Top="110">
```



```

<Rectangle.Fill>
  <RadialGradientBrush>
    <GradientStop Color="Yellow" Offset="0.0" />
    <GradientStop Color="Red" Offset="0.25" />
    <GradientStop Color="Blue" Offset="0.75" />
    <GradientStop Color="LimeGreen" Offset="1.0" />
  </RadialGradientBrush>
</Rectangle.Fill>
</Rectangle>

<Rectangle Width="140" Height="70" Canvas.Left="155" Canvas.Top="110">
  <Rectangle.Fill>
    <RadialGradientBrush GradientOrigin="1,0">
      <GradientStop Color="Yellow" Offset="0.0" />
      <GradientStop Color="Red" Offset="0.25" />
      <GradientStop Color="Blue" Offset="0.75" />
      <GradientStop Color="LimeGreen" Offset="1.0" />
    </RadialGradientBrush>
  </Rectangle.Fill>
</Rectangle>
</Canvas>

```

**Beispiel für Ausmalen eines Rectangle per Image**

```

<Canvas Width="300" Height="300"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Background="White">

  <Rectangle Height="180" Width="90" Canvas.Left="10" Canvas.Top="10"
    Stroke="Black" StrokeThickness="1">
    <Rectangle.Fill>
      <ImageBrush ImageSource="star.png"/>
    </Rectangle.Fill>
  </Rectangle>

  <Rectangle Height="180" Width="90" Canvas.Left="110" Canvas.Top="10"
    Stroke="Black" StrokeThickness="1">
    <Rectangle.Fill>
      <ImageBrush ImageSource="star.png" Stretch="Uniform"/>
    </Rectangle.Fill>
  </Rectangle>
</Canvas>

```

**Beispiele für Bild**

```

<Canvas Width="300" Height="300"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Image Source="star.png" />
</Canvas>

<Canvas Width="300" Height="300"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Background="White">
  <Image Source="star.png" Stretch="None"
    Height="100" Width="200" Canvas.Left="100" />

  <Image Source="star.png" Stretch="Fill"

```



```
Height="100" Width="200" Canvas.Top="100" Canvas.Left="100" />

<Image Source="star.png" Stretch="Uniform"
Height="100" Width="200" Canvas.Top="200" Canvas.Left="100" />

d</Canvas>
```

### **Beispiel für TextBlock (Container für Text)**

```
<Canvas
xmlns="http://schemas.microsoft.com/client/2007"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<TextBlock></TextBlock>
</Canvas>
```

### **Beispiel für Font-Angaben im Textblock**

FontSize: The font size, in pixels.  
FontStyle: The font style. Choices are Normal, Italic, and Oblique.  
FontWeight: The font weight. Choices are Thin, ExtraLight, Light, Normal, Medium, SemiBold, Bold, ExtraBold, Black, ExtraBlack.  
FontFamily: The font typeface family name.  
Foreground: The Brush that paints the text inside the TextBlock. You can use a solid color, gradient, or an image.

```
<Canvas
xmlns="http://schemas.microsoft.com/client/2007"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<TextBlock FontSize="40"
FontFamily="Georgia"
FontStyle="Italic" FontWeight="Bold"
Canvas.Top="20" Canvas.Left="20">

<TextBlock.Foreground>
<LinearGradientBrush>
<GradientStop Color="SlateBlue" Offset="0.0" />
<GradientStop Color="Black" Offset="1.0" />
</LinearGradientBrush>
</TextBlock.Foreground>
</TextBlock>
</Canvas>
```

### **Beispiel für verschiedene Fonts innerhalb eines Textblockes anhand Objekt Run**

```
<Canvas
xmlns="http://schemas.microsoft.com/client/2007"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<TextBlock>
<Run FontSize="30"></Run>
</TextBlock>
</Canvas>
```

### **Beispiel für MediaElement**

```
<Canvas Width="300" Height="300"
xmlns="http://schemas.microsoft.com/client/2007"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
<MediaElement Source="thebutterflyandthebear.wmv" Width="300" Height="300" />

<Ellipse Height="200" Width="200" Canvas.Left="30" Canvas.Top="30"
```



```

Stroke="Black" StrokeThickness="10" Fill="SlateBlue"
Opacity="0.6" />
</Canvas>

```

### Beispiel für Steuerung von Sound

```

<Canvas Width="300" Height="300"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <MediaElement x:Name="media"
    Source="thebutterflyandthebear.wmv"
    Width="300" Height="300" />

  <!-- Stops media playback.-->
  <Canvas MouseLeftButtonDown="media_stop"
    Canvas.Left="10" Canvas.Top="265">
    <Rectangle Stroke="Black"
      Height="30" Width="55" RadiusX="5" RadiusY="5">
      <Rectangle.Fill>
        <RadialGradientBrush GradientOrigin="0.75,0.25">
          <GradientStop Color="Orange" Offset="0.0" />
          <GradientStop Color="Red" Offset="1.0" />
        </RadialGradientBrush>
      </Rectangle.Fill>
    </Rectangle>
    <TextBlock Canvas.Left="5" Canvas.Top="5"></TextBlock>
  </Canvas>

  <!-- Pauses media playback. -->
  <Canvas MouseLeftButtonDown="media_pause"
    Canvas.Left="70" Canvas.Top="265">
    <Rectangle Stroke="Black"
      Height="30" Width="55" RadiusX="5" RadiusY="5">
      <Rectangle.Fill>
        <RadialGradientBrush GradientOrigin="0.75,0.25">
          <GradientStop Color="Yellow" Offset="0.0" />
          <GradientStop Color="Orange" Offset="1.0" />
        </RadialGradientBrush>
      </Rectangle.Fill>
    </Rectangle>
    <TextBlock Canvas.Left="5" Canvas.Top="5"></TextBlock>
  </Canvas>

  <!-- Begins media playback. -->
  <Canvas MouseLeftButtonDown="media_begin"
    Canvas.Left="130" Canvas.Top="265">
    <Rectangle Stroke="Black" RadiusX="5" RadiusY="5"
      Height="30" Width="55">
      <Rectangle.Fill>
        <RadialGradientBrush GradientOrigin="0.75,0.25">
          <GradientStop Color="LimeGreen" Offset="0.0" />
          <GradientStop Color="Green" Offset="1.0" />
        </RadialGradientBrush>
      </Rectangle.Fill>
    </Rectangle>
    <TextBlock Canvas.Left="5" Canvas.Top="5"></TextBlock>
  </Canvas>
</Canvas>

```



**Beispiel für Objekt-Animation anhand dessen Erweiterung um Storyboard und Timeline**

zu animierendes Objekt mit x:Name "ellipse"

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Ellipse x:Name="ellipse"
    Height="20" Width="20" Canvas.Left="30" Canvas.Top="30"
    Fill="black" />
</Canvas>
```

Eventsteuerung

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Canvas.Triggers>
    <EventTrigger RoutedEvent="Canvas.Loaded">
      <EventTrigger.Actions>
        <BeginStoryboard>
          <Storyboard>
            <DoubleAnimation
              Storyboard.TargetName="ellipse"
              Storyboard.TargetProperty="(Canvas.Left)"
              To="300" Duration="0:0:1" />
          </Storyboard>
        </BeginStoryboard>
      </EventTrigger.Actions>
    </EventTrigger>
  </Canvas.Triggers>
  <Ellipse x:Name="ellipse"
    Height="20" Width="20" Canvas.Left="30" Canvas.Top="30"
    Fill="black"/>
</Canvas>
```

**Beispiel für Mausevents**

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <Ellipse x:Name="e1" MouseMove="e1Move"
    MouseEnter="e1Enter" MouseLeave="e1Leave"
    MouseLeftButtonDown="e1Down" MouseLeftButtonUp="e1Up"
    Height="100" Width="100" Canvas.Left="80" Canvas.Top="30"
    Stroke="Black" StrokeThickness="10" Fill="LightBlue"/>
</Canvas>
```

**Beispiel für Event Loaded**

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <Ellipse Loaded="ellipse_loaded"
    Height="200" Width="200"
    Canvas.Left="30" Canvas.Top="30"
    Stroke="Black" StrokeThickness="10" Fill="LightBlue"/>
</Canvas>
```





**Beispiel für Hyperlink**

```

<Canvas Width="300" Height="300"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">

  <!-- Hyperlink -->
  <Canvas Width="90" Height="30" Canvas.Left="20" Canvas.Top="20"
    Background="transparent"
    Cursor="Hand"
    MouseEnter="hyperlink_MouseEnter"
    MouseLeave="hyperlink_MouseLeave"
    MouseLeftButtonDown="hyperlink_MouseLeftButtonDown">
    <TextBlock Text="hyperlink" Foreground="Blue"/>
    <Line Stroke="blue" StrokeThickness="1" X1="0" Y1="20" X2="65" Y2="20"
      x:Name="hyperlink_line" Opacity="0"/>
  </Canvas>
</Canvas>

function hyperlink_MouseLeftButtonDown(sender, args) {
  window.location = "default.aspx";
}

function hyperlink_MouseEnter(sender,args)
{
  sender.findName("hyperlink_line").opacity = 1;
}

function hyperlink_MouseLeave(sender,args)
{
  sender.findName("hyperlink_line").opacity = 0;
}

```

**Beispiel für Button**

```

<Canvas Width="300" Height="300"
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <!-- Button -->
  <Canvas
    x:Name="button"
    Canvas.Top="10" Canvas.Left="20"
    MouseLeftButtonDown="button_MouseLeftButtonDown"
    MouseLeftButtonUp="button_MouseLeftButtonUp"
    MouseEnter="button_MouseEnter"
    MouseLeave="button_MouseLeave">
    <Canvas.RenderTransform>
      <TransformGroup>
        <TranslateTransform x:Name="button_transform" X="0" Y="0"/>
      </TransformGroup>
    </Canvas.RenderTransform>
    <Rectangle Stroke="#FF000000" Fill="sc#1, 0.8123474, 0.8123474, 0.8123474"
      Width="128.8" Height="56" x:Name="button_rectangle"/>
    <Rectangle Stroke="sc#1, 0.912730157, 0.37122494, 0.17111966" StrokeThickness="5"
      Width="126.8" Height="54" Canvas.Left="1" Canvas.Top="1"
      Opacity="0"
      x:Name="button_highlight"/>
    <TextBlock Text="Press me!" FontSize="20" Canvas.Left="22" Canvas.Top="12"/>
  </Canvas>
</Canvas>

```



```

var mouseOver = false;
var pressed = false;

function button_MouseLeftButtonDown(sender,args) {
    sender.captureMouse();
    mouseOver = true;
    pressed = true;
    updateVisuals(sender);
}

function button_MouseLeftButtonUp(sender,args) {
    sender.releaseMouseCapture();
    pressed = false;

    updateVisuals(sender);

    if (mouseOver) {
        alert("you pressed the button!");
    }
}

function button_MouseEnter(sender,args) {
    mouseOver = true;
    updateVisuals(sender);
}

function button_MouseLeave(sender,args) {
    mouseOver = false;
    updateVisuals(sender);
}

function updateVisuals(sender) {
    //background
    if (pressed && mouseOver) {
        sender.findName("button_rectangle").fill = "sc#1, 0.548430264, 0.5354195, 0.5354195";
        var transform = sender.findName("button_transform");
        transform.x = 2;
        transform.y = 2;
    } else {
        sender.findName("button_rectangle").fill = "sc#1, 0.8123474, 0.8123474, 0.8123474";
        var transform = sender.findName("button_transform");
        transform.x = 0;
        transform.y = 0;
    }

    // highlight
    if (mouseOver || pressed) {
        sender.findName("button_highlight").opacity = 1;
    } else {
        sender.findName("button_highlight").opacity = 0;
    }
}

```

### **Beispiel für Schieberegler (Slider)**

```

<Canvas Width="300" Height="300"
    xmlns="http://schemas.microsoft.com/client/2007"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    Loaded="slider_Loaded">
    <!-- Slider -->
    <Canvas x:Name="slider"
        Canvas.Top="50" Canvas.Left="20" Width="200" Height="45"

```



```

Background="transparent">

<Line x:Name="slider_line"
  Stroke="black" StrokeThickness="1"
  X1="0" Y1="25" X2="200" Y2="25" />

<Rectangle
  Fill="Transparent"
  Width="200" Height="45"
  MouseLeftButtonDown="slider_MouseLeftButtonDown" />

<Path x:Name="slider_thumb" Stroke="#FF000000"
  Fill="sc#1, 0.548430264, 0.5354195, 0.5354195"
  Data="M0,0 L11.5,0 11.5,30 5.75,40 0,30z"
  MouseLeftButtonUp="slider_thumb_MouseLeftButtonUp"
  MouseMove="slider_thumb_MouseMove"
  MouseLeftButtonDown="slider_thumb_MouseLeftButtonDown" />
</Canvas>
</Canvas>

var mouseDownPosition = 0;
var mouseDownValue = -1;
var thumbCenter = 5.75;

function slider_Loaded(sender, args) {
  slider_SetValue(sender, 0);
}

function slider_MouseLeftButtonDown(sender, args) {

  var coordinate = args.getPosition(null).x;
  var slider = sender.findName("slider");
  coordinate -= slider["Canvas.Left"];
  slider_SetValue(slider, coordinate - thumbCenter);
}

function slider_thumb_MouseLeftButtonDown(sender, args) {
  var slider = sender.findName("slider");
  sender.captureMouse();
  mouseDownValue = slider_GetValue(slider);
  mouseDownPosition = args.getPosition(null).x;
}

function slider_thumb_MouseLeftButtonUp(sender, args) {
  var slider = sender.findName("slider");
  slider.releaseMouseCapture();
  mouseDownValue = -1;
}

function slider_thumb_MouseMove(sender, args) {
  var slider = sender.findName("slider");
  if (mouseDownValue != -1) {
    var newValue = mouseDownValue + (args.getPosition(null).x - mouseDownPosition);
    slider_SetValue(slider, newValue);
  }
}

function slider_GetValue(sender) {
  var thumb = sender.findName("slider_thumb");
  return thumb["Canvas.Left"];
}

```



```
function slider_SetValue(sender, newValue) {

    if (newValue > sender.width ) {
        newValue = sender.width;
        mouseDownValue = -1;
    }
    if (newValue < - thumbCenter) {
        newValue = - thumbCenter;
        mouseDownValue = -1;
    }
    var thumb = sender.findName("slider_thumb");

    thumb["Canvas.Left"] = newValue;
}
```

## XAML-Daten referenzieren

### CreateFromXaml Methode und Beispiele

Creates XAML content dynamically.

XAML	You cannot use methods in XAML.
Scripting	silverlightObject.content.createFromXaml(xamlContent, nameScope)

#### Parameters

xamlContent	String The XAML content to add to the existing Silverlight object hierarchy.
nameScope	Boolean Determines whether to create x:Name references in XAML content that do not conflict with other named elements. The nameScope parameter is optional, and its value defaults to false if not specified.

#### Return Value

An object reference if the XAML content was successfully created; otherwise, returns null.

#### Remarks

The CreateFromXaml method allows you to create XAML content dynamically, which you can then add to the existing Silverlight object hierarchy. You can create a single Silverlight object, such as a TextBlock, or an entire tree of Silverlight objects. For more information, see Using the CreateFromXaml Method.

Zur Erzeugung eines XAML-Contents muss die Objektabhängigkeit beachtet werden:

z.B. benötigen diverse Objekte das Canvas-Objekt als Container.

Die Erzeugung des Contents kann eventuelle nur per Eventhandler erfolgen

z.B. bei Verwendung des Downloader-Objektes

z.B. wenn Erzeugung aktionsabhängig ist z.B. durch ein Mausklick aktiviert wird.

Die Erzeugung bedeutet:

XAML-Daten bereitstellen zum Auslesen

ausgelesene XAML-Daten benutzen um Objekt zu erzeugen im DOM anhand DOM-Funktion

Beispiel: genau 1 XAML-Objekt erzeugen und damit Objekt in Silverlight erzeugen

```
// MouseLeftButtonUp event handler for the root Canvas object.
```

```
function onMouseLeftButtonUp(sender, eventArgs)
```

```
{ // Zeiger auf den Container (Host) holen
```

```
var control = sender.getHost();
```

```
// XAML-Daten bereitstellen
```

```
var xamlFragment = '<TextBlock Canvas.Top="200" Text="Click for more info..." />';
```

```
// XAML-Datenobjekt erzeugen
```



```

textBlock = control.content.createFromXaml(xamlFragment);

// Objekt im DOM anhand XAML-Datenobjekt erzeugen: hier per anhängen an die
// vorhandenen Kinderstruktur per Collection children des Containers control
sender.children.add(textBlock);
}

```

Beispiel: Teile von XAML-Daten verwenden

Teil der XAML-Daten (Abschnitt) wird per Attribut x:Name markierbar, also ein ID im Scriptformat. Das ID kann per Methode zum Objekt zu einem Zeiger umgewandelt werden.

```

// Define a XAML fragment and create it.
var xamlFragment = '<TextBlock xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" ';
xamlFragment += 'x:Name="myCanvas" Canvas.Top="200" Text="Click for more info" />';
textBlock = control.content.createFromXaml(xamlFragment);

```

## Referenzierung per JavaScript - Beispiele

anhand HTML-DOM

übliche Funktionen und Collectionen z.B. getElementById()

anhand Silverlight-DOM

passend zum Objekt

Beispiel: XAML-Daten adressieren

```

<!-- Visual progress indicator -->
<Canvas Canvas.Top="70">
    <Rectangle      x:Name="progressRectangle"
                    Canvas.Left="20"
                    Height="10"
                    Width="0"
                    Fill="Maroon"

    />
    <Rectangle      Canvas.Top ="-1"
                    Canvas.Left="19"
                    Height="12"
                    Width="202"
                    StrokeThickness="1"
                    Stroke="Black"

    />
    <TextBlock      x:Name="progressText"
                    Canvas.Top ="-4"
                    Canvas.Left="230"
                    Text="0%" FontSize="12"

    />
</Canvas>

```

```

Methode var zeiger=findName(Name_laut_x:Name);
//      liefert Zeiger
//      Anzeige des Typs des Objektes per alert(zeiger.toString());

```

Beispiel: Wert einer Objekt-Eigenschaft (Properties) adressieren

per Methoden    GetValue() für Wert lesen

                 SetValue() für Wert schreiben

per objekt\_zeiger.eigenschaft für Lesen bzw. Schreiben (objekt\_zeiger wird z.B. per  
getElementById() ermittelt)

Lesen:

```

var opacity = objekt_zeiger.getValue("opacity");
opacity = objekt_zeiger.opacity;           // per Zeiger
alert("Opacity = " + opacity);

```



Schreiben:

```
objekt_zeiger.setValue("Canvas.Top", 40);
objekt_zeiger ["Canvas.Top"] = 40;      // per Zeiger
```

Beispiel: Objekt hinzufügen  
per Methode Add()

```
var textBlock;
function onLoaded(sender, eventArgs)
{
    // Retrieve the id of the control.
    var control = sender.getHost();
    // Define and create a XAML fragment.
    var xamlFragment = '<TextBlock Canvas.Top="200"'
        + ' Opacity=".5" Text="Click for more info" />';
    textBlock = control.content.createFromXaml(xamlFragment);
    // Add the TextBlock to the root Canvas object.
    sender.children.add(textBlock);
}

// Toggle the Opacity property for the TextBlock.
function onToggle()
{
    if (textBlock.opacity)
    { textBlock.opacity = 0; }
    else { textBlock.opacity = 1; }
}

// Add three overlapping Rectangle objects.
function AddItems(rootCanvas)
{
    var control = rootCanvas.getHost();
    var xamlFragment = '<Rectangle Fill="Maroon" Canvas.Top="20" Canvas.Left="20"'
        + ' Height="100" Width="100" />';
    var rectangle_1 = control.content.createFromXaml(xamlFragment);
    rootCanvas.children.add(rectangle_1);    // an das Ende der Kinderkette anhängen
    var xamlFragment = '<Rectangle Fill="LightBlue" Canvas.Top="40" Canvas.Left="40"'
        + ' Height="100" Width="100" />';
    var rectangle_2 = control.content.createFromXaml(xamlFragment);
    rootCanvas.children.add(rectangle_2);    // an das Ende der Kinderkette anhängen
    var xamlFragment = '<Rectangle Fill="Teal" Canvas.Top="60" Canvas.Left="60" '
        + ' Height="100" Width="100" />';
    var rectangle_3 = control.content.createFromXaml(xamlFragment);
    rootCanvas.children.add(rectangle_3);    // an das Ende der Kinderkette anhängen

    // das zuletzt erzeugte Objekt wird als oberstes gerendert (Lage auf der Z-Achse)
    // wenn ZIndex-Eigenschaft verfügbar, so Lage auf Z-Achse einstellbar
}
}
```

Insert-Methode fügt in die Kette der Kinder ein

Kinder sind per RemoveXX-Methoden und Clear-Methode entfernbar.



## Support zu Silverlight im Internet - Beispiele

### Weblinks

Alle Links können sich ändern - besonders die auf Microsoft-Server !

<a href="http://silverlight.net/Default.aspx">http://silverlight.net/Default.aspx</a>	Silverlight im Internet
<a href="http://silverlight.net/community/communitygallery.aspx">http://silverlight.net/community/communitygallery.aspx</a>	Silverlight Community - Beispiele
<a href="http://www.microsoft.com/silverlight/default01.aspx">http://www.microsoft.com/silverlight/default01.aspx</a>	Silverlight auf Microsoft
<a href="http://msdn2.microsoft.com/en-us/library/bb188743.aspx">http://msdn2.microsoft.com/en-us/library/bb188743.aspx</a>	Silverlight auf MSDN Library
<a href="http://msdn2.microsoft.com/en-us/library/bb428859.aspx">http://msdn2.microsoft.com/en-us/library/bb428859.aspx</a>	Silverlight Architecture Overview
<a href="http://msdn2.microsoft.com/en-us/library/bb412393.aspx">http://msdn2.microsoft.com/en-us/library/bb412393.aspx</a>	Silverlight Scripting Application Notes
<a href="http://msdn2.microsoft.com/en-us/library/bb404710.aspx">http://msdn2.microsoft.com/en-us/library/bb404710.aspx</a>	Silverlight Development with JavaScript
<a href="http://msdn2.microsoft.com/en-us/library/bb188567.aspx">http://msdn2.microsoft.com/en-us/library/bb188567.aspx</a>	Silverlight Reference
<a href="http://msdn2.microsoft.com/en-us/library/bb412392.aspx">http://msdn2.microsoft.com/en-us/library/bb412392.aspx</a>	XAML-Überblick
<a href="http://msdn2.microsoft.com/en-us/silverlight/default.aspx">http://msdn2.microsoft.com/en-us/silverlight/default.aspx</a>	Silverlight auf Microsoft
<a href="http://ajax.asp.net/Default.aspx">http://ajax.asp.net/Default.aspx</a>	Microsoft ASP.Net Ajax



## **Beispiel zur Erzeugung eines Silverlight-Objektes MediaElement**

Hinweis zu nachfolgendem Link, der im IE 7 unter Win XP SP2 gerendert wurde:

Die Webseite ist im Layout Querformat.

Kopieren der gesamten Webseite in Word von Office XP zerstört das Layout der Texte - besonders der Quelltexte. Es sind daher pro Quelltext je 1 Link verfügbar, der den Quelltext ohne Layoutänderung in die Windows-Zwischenablage kopieren lässt, aus der der User dann manuell in Word einfügen kann.

Die Bilder wurden von Microsoft als z.T. unleserlich offeriert: Man muss diese klicken, um lesbare Bilder zu erhalten. Z.B. ist Figure 1 als lesbares Bild erreichbar unter [http://msdn2.microsoft.com/en-us/library/bb428858.crtembsilverlight011\(en-us,msdn.10\).gif](http://msdn2.microsoft.com/en-us/library/bb428858.crtembsilverlight011(en-us,msdn.10).gif)

Wegen obigen Mängeln musste die nachfolgende Darstellung der Webseite angepasst werden.

<http://msdn2.microsoft.com/en-us/library/bb428858.aspx>

### **Embeddable Media Playback with Silverlight**

Laurence Moroney  
Microsoft Corporation

Updated: May 2007

Applies to:

Microsoft Silverlight  
Microsoft Visual Studio 2005

**Summary:** Learn how to use Silverlight to build a service that can embed video on a blog (or any other Web page), where all you need is an iFrame to the URL that contains your service. (6 printed pages)

### **Contents**

Introduction

Getting Started

#### **Introduction**

One of the really neat things about Silverlight is that it isn't a closed development and runtime environment. Because you can implement your UI in XAML and control it from JavaScript, you can build services that deliver a Silverlight front end. In this example, I'll take you through the steps of building a service that can be used to embed video on a blog (or any other Web page), where all you need is an iFrame to the URL that contains your service.

**Note** For this article, we will be using Visual Web Developer Express. This software is a free download from the [Visual Web Developer site](#).

### **Getting Started**

First off, you will create a new Web site using Visual Web Developer.

To create a new Web site

Open Visual Web Designer.

On the File menu, click New Web Site.

In the New Web Site dialog box that will then appear, select ASP.NET Web Site.





Name the Web site VideoService by typing that at the end of the location path, as shown in Figure 1.

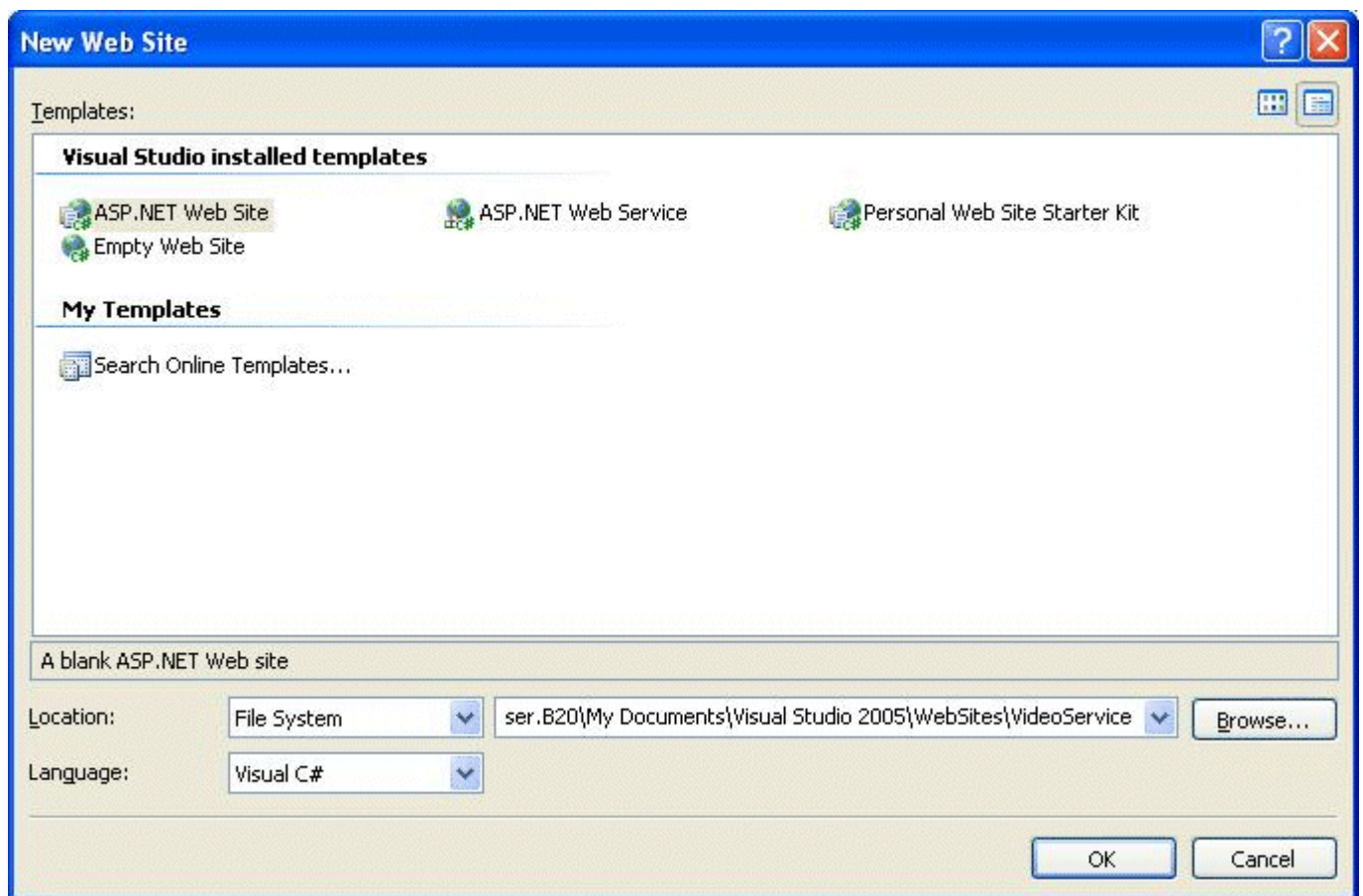


Figure 1. Creating a new Web site in Visual Web Developer (Click on the picture for a larger image)

You will now have a new Web site containing a page called Default.aspx. We'll edit that page in a moment, but first add a couple of new directories to your Web site: "js," and "xaml."

To add two directories named "js" and "xaml" to Solution Explorer

Right-click the project in Solution Explorer, and click New Folder.

Name the folder xaml.

Repeat Steps 1 and 2 to create a new folder named js.

When you are done, Solution Explorer will look like Figure 2.



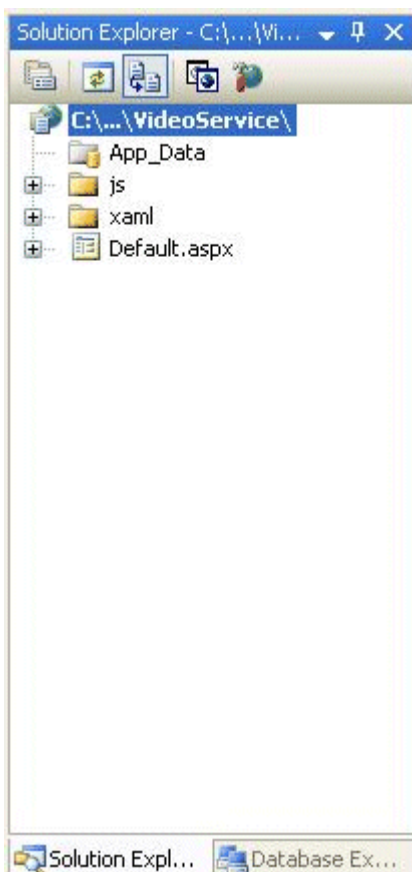


Figure 2. Your solution, with "js" and "xaml" directories added

Next, you will create your Media Player XAML.

To create a Media Player XAML

Right-click the XAML directory you just created and select Add New Item.

In the ensuing dialog box, select XML File.

Name the file videoplayer.xaml, as shown in Figure 3.



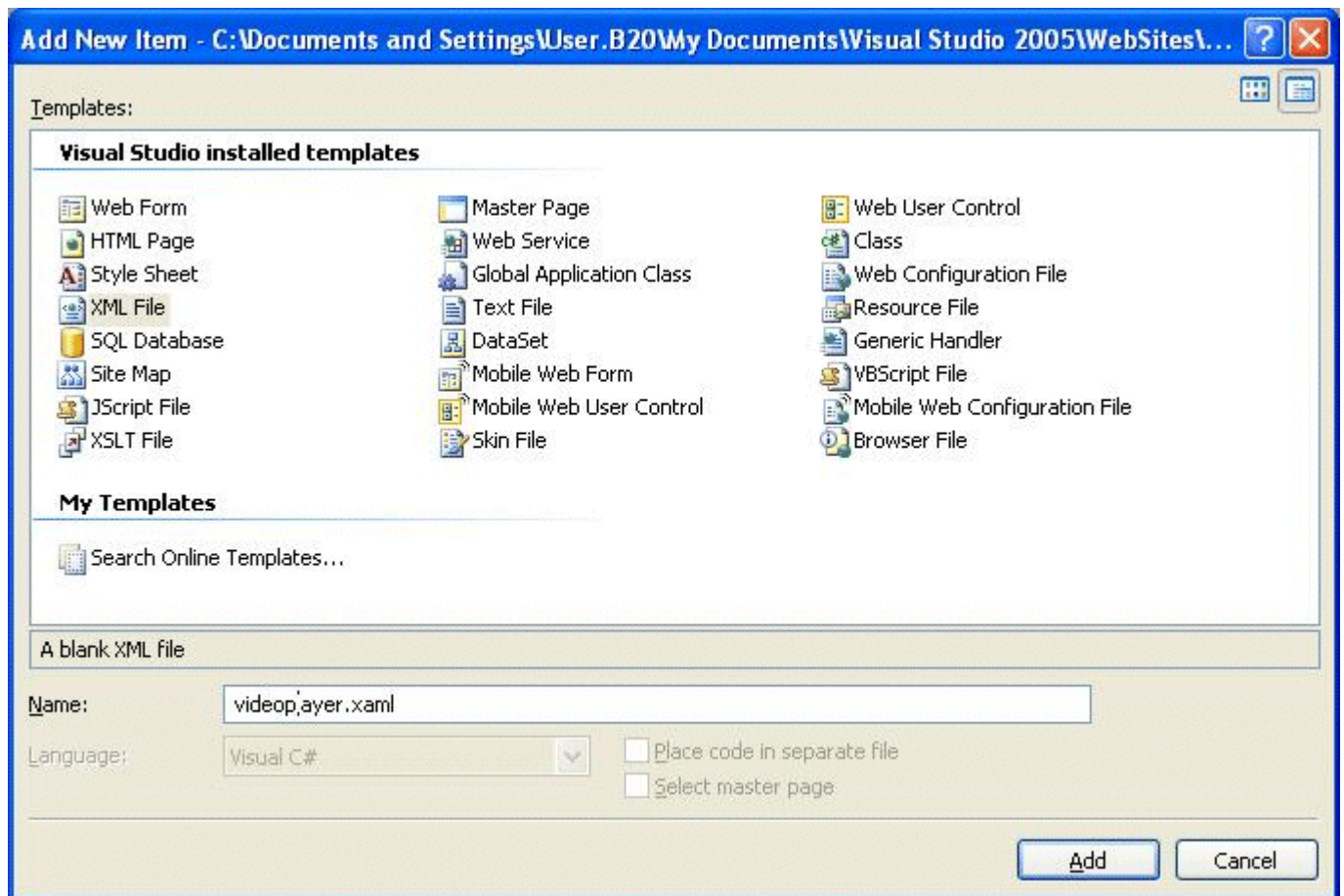


Figure 3. Creating the XAML file (Click on the picture for a larger image)

Replace the contents of this file with the following XAML:

<Canvas

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

x:Name="root"

Loaded="javascript:handleLoad">

<Canvas x:Name="VideoLayer">

<MediaElement x:Name="VidElement" Canvas.Top="0"

Canvas.Left="0" Height="100" Width="100">

</MediaElement>

</Canvas>

</Canvas>

This creates a media element that will be used to render the video. XAML is the core technology that is used to define Silverlight user experiences. For more on XAML, browse the [Windows Presentation Foundation \(WPF\)](#) section of the MSDN Library.

**Note** Not all XAML tags that are supported in the WPF object model are supported on Silverlight.



Now you will need the Silverlight loader scripts. The first of these, `Silverlight.js`, is available as part of the Silverlight SDK: <http://www.silverlight.net>. Put this script in the `js` folder you created earlier.

You will also need the `createSilverlight.js` script. This handles instantiation of Silverlight for this application. Create a new JavaScript file in the `js` folder, and name it `createSilverlight.js`.

Fill the `createSilverlight.js` with the following code:

```
function createSilverlight()
{
    Sys.Silverlight.createObject("xaml/videoplayer.xaml",
        slControlHost, "slControl1",
        {width:'232', height:'240',
        inplaceInstallPrompt:false, background:'white',
        isWindowless:'true', framerate:'24', version:'0.8'},

        {onError:null, onLoad:null},
        null);
}
```

This instantiates the Silverlight control when it is called, rendering the XAML stored in `videoplayer.xaml` that you created earlier. For the full set of properties check out the Silverlight documentation on <http://www.silverlight.net>.

For the final step, open the `Default.aspx` page and replace the code in it with this code.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="_Default" %>

<html >

<head>

    <title>Video Server Sample</title>

</head>

<script src="js/silverlight.js" type="text/javascript"></script>
<script src="js/createsilverlight.js" type="text/javascript"></script>
<script type="text/javascript">
function handleLoad() {
    var slC = document.getElementById("slControl1");
    var elMedia = slC.content.findName("VidElement");

    elMedia.Width="<%
    if (Request.Params["width"] == null)
        Response.Write("425");
    %>";
}
```



```

else

    Response.Write(Request.Params["width"]);

%>";

elMedia.Height="<%

if (Request.Params["height"] == null)

    Response.Write("350");

else

    Response.Write(Request.Params["height"]);

%>";

elMedia.source = "<% Response.Write(Request.Params["src"]); %>";

elMedia.play();
}
</script>
</head>

<body bgcolor="black">

    <div id='slControlHost'>

        <script type='text/javascript'>

            createSilverlight();

        </script>

    </div>

</body>
</html>

```

This code implements a page (Default.aspx), to which you can pass three parameters:

**Src**, which contains the URL of the video that you want to play back.

**Width**, which contains the desired width of the video element.

**Height**, which contains the desired height of the video element.

If you look back at the earlier XAML file, you will see the `Loaded="javascript:handleLoad` attribute. This specifies that when the XAML is loaded and rendered, the named JavaScript function (in this case, `handleLoad`) should be called.

So, what does `handleLoad` do? First of all, it creates a reference to the Silverlight component that JavaScript can recognize. It does this by using the standard JavaScript methodology to fish a reference to an element out of the HTML DOM. The Silverlight control was named `slControl1` (check back to `CreateSilverlight.js` and you'll see this), and this name is used. A reference to it is now controllable by using the `slC` JavaScript var. Once you have this element, you can use the `findName` function on its



content to inspect its XAML and get a reference to an element within the XAML. We do this to get a reference to the MediaElement that we created in the XAML (and called VidElement).

```
var sIC = document.getElementById("slControl1");
```

```
var elMedia = sIC.content.findName("VidElement");
```

Next, we just fish the parameters off the request line using some C#, and assign them to the Width, Height, and Source properties of the media element, respectively.

```
elMedia.Width="<%
```

```
    if (Request.Params["width"] == null)
```

```
        Response.Write("425");
```

```
    else
```

```
        Response.Write(Request.Params["width"]);
```

```
%>";
```

```
elMedia.Height="<%
```

```
    if (Request.Params["height"] == null)
```

```
        Response.Write("350");
```

```
    else
```

```
        Response.Write(Request.Params["height"]);
```

```
%>";
```

```
elMedia.source = "<% Response.Write(Request.Params["src"]); %>";
```

```
elMedia.play();
```

Now all you have to do is execute your application and pass it some parameters. As an example, you can do the following:

```
http://localhost:XXXX/VideoService/Default.aspx?src=
```

```
http://download.microsoft.com/download/d/e/2/de2bec9c-4ba1-406e-8029-
```

```
5c4767dca3eb/WPFE_Getting_Started_2MB_Ch9.wmv&height=200&width=200
```

Where XXXX is a port that will be assigned by the built-in Web server at run time.

And when you run your page, you will have a video player! The magic of this is that you can deploy this to a Web server, and then, from your blog, you can <IFrame> into this URL to embed video!



## **Beispiel für erste Schritte mit Silverlight**

### **Erste Schritte per englischsprachigem Video von microsoft.com**

[http://download.microsoft.com/download/5/9/8/5985f834-b3ea-459b-98ba-0aede5bc653a/ScuttGu\\_Silverlight\\_Demo.wmv](http://download.microsoft.com/download/5/9/8/5985f834-b3ea-459b-98ba-0aede5bc653a/ScuttGu_Silverlight_Demo.wmv)

Die Url ist 1 Zeile !

Das Video zeigt bei permanent verzerrtem Ton die Ersten Schritte bei Nutzung von Visual Studio. Der Audioinhalt ist zusätzlich mit Dialekt, selbstgefällig und ohne Punkt und Komma (abschreckende Laiendarstellung).

### **Erste Schritte in Deutsch per Microsoft Expression Produkt Blend 2**

<http://www.microsoft.com/germany/msdn/library/net/wpf/ErsteSchritteMitSilverlight.aspx?mfr=true>

#### **Erste Schritte mit Silverlight**

Veröffentlicht: 07. Mai 2007

Von Laurence Moroney

**Dieses Whitepaper bietet einen Überblick über Silverlight (ehemals „WPF/E“) und darüber, wie es sich in die Entwicklung der nächsten Generation von Webanwendungen einfügt.**

**Auf dieser Seite**

↓ [Was ist Silverlight?](#)

↓ [Die Geschichte der Webentwicklung: Übergang zu Web.Next](#)

↓ [Erstellen einer einfachen Silverlight-Anwendung](#)

↓ [Erstellen der Benutzeroberfläche für einen Videoplayer](#)

↓ [Erläuterungen zum Silverlight-Aufruf](#)

↓ [Schlussbemerkung](#)

### **Was ist Silverlight?**

Silverlight (ehemals „WPF/E“) ist eine neue Webpräsentationstechnologie, die auf verschiedenen Plattformen ausgeführt werden kann. Es erlaubt die Erstellung aufwändiger, visuell beeindruckender, interaktiver Funktionen, die überall ausgeführt werden können: innerhalb von Browsern und auf verschiedenen Geräten und Desktopbetriebssystemen (wie z. B. Apple Macintosh). In Übereinstimmung mit WPF (Windows Presentation Foundation), der Präsentationstechnologie in Microsoft .NET Framework 3.0, bildet XAML (Extensible Application Markup Language) die Grundlage für die Silverlight-Präsentationsfunktionen.

Dieses Whitepaper bietet eine schrittweise Einführung in die Grundlagen von Silverlight und die vielfältigen Microsoft-Tools zum Erstellen aufwändiger grafischer Websites, einschließlich Microsoft Expression Blend, Microsoft Visual Studio 2005 und XAML. Zunächst ein kurzer Überblick über die Hintergründe der Entwicklung von Silverlight und seine Einordnung in Bezug auf andere Entwicklungen.

↕ [Zum Seitenanfang](#)

### **Die Geschichte der Webentwicklung: Übergang zu Web.Next**





Als das World Wide Web von Tim Berners-Lee am CERN erfunden wurde, entwickelte er es als netzwerkbasiertes System, das die Speicherung und Verknüpfung statischer Dokumente ermöglichen sollte. Im Verlauf der Jahre kamen weitere Neuerungen hinzu. Der nächste logische Schritt bestand darin, „aktive“ Dokumente mit zeit- oder benutzerspezifischen Informationen dann zu erstellen, wenn sie angefordert wurden. Dies wurde durch Technologien wie z. B. CGI ermöglicht. Mit der Zeit wurde die Möglichkeit, Dokumente im Web zu generieren, sehr wichtig, und die Technologie wurde mittels CGI, Java, ASP und schließlich ASP.NET weiterentwickelt.

ASP.NET war ein Meilenstein, da es Entwicklern ermöglichte, auf Grundlage des Paradigmas der Serverentwicklung und mithilfe von Best-of-Breed-Tools der Visual Studio-Produktfamilie schnell hochwertige Webanwendungen zu entwickeln.

Als ein großes Problem für Webanwendungen erwies sich die Benutzerfreundlichkeit. Technische Einschränkungen verhinderten, dass Webanwendungen die gleiche Benutzerfreundlichkeit wie Clientanwendungen mit lokalen Daten boten.

Das **XMLHttpRequest**-Objekt, das von Microsoft als Teil von Internet Explorer 5 im Jahr 2000 veröffentlicht wurde, wurde die Grundlage des asynchronen JavaScript und der XML (AJAX)-Technologie. Dadurch konnten Webanwendungen dynamischer auf Benutzereingaben reagieren, indem kleinere Teile einer Webseite aktualisiert wurden, ohne den Inhalt vollständig neu zu laden. Innovative Lösungen die auf AJAX aufbauten, wie z. B. Windows Live Local-Karten, ließen Webanwendungen einen weiteren Schritt in Richtung einer clientähnlichen Funktionalität für Benutzer machen.

Silverlight ist der nächste Schritt in der Entwicklung einer möglichst vielfältigen Funktionalität für Benutzer, die Anwendungsentwickler und Designer ihren Kunden bieten können. Dadurch sind Designer in der Lage, ihre Kreativität zu entfalten und ihre Arbeit in einem Format zu speichern, das im Web direkt funktioniert. In der Vergangenheit hätte ein Designer eine Website und die Funktionalität für Benutzer entworfen, indem er Tools verwendet, die vielfältige Ergebnisse liefern. Um sie aber bereitstellen zu können, hätte der Entwickler seine Ergebnisse an die Einschränkungen der Webplattform anpassen müssen. Im Silverlight-Modell können Designer die gewünschte Funktionalität für Benutzer erstellen und sie in XAML ausdrücken. Dieses XAML kann dann durch einen Entwickler mithilfe der Silverlight-Laufzeitumgebung direkt in eine Webseite eingefügt werden. Deshalb können beide besser als je zuvor zusammenarbeiten, um eine reichhaltige Clientfunktionalität für Benutzer bereitzustellen.

XAML ist XML und somit textbasiert. Es ist eine Firewall-freundliche und leicht zu prüfende Beschreibung reichhaltiger Inhalte. Obgleich es mit anderen Technologien (z. B. Java Applets, ActiveX und Flash) möglich ist, reichhaltigere Inhalte als DHTML, CSS und JavaScript bereitzustellen, senden diese Technologien ausnahmslos binären Inhalt an den Browser. Dies macht es schwierig, die Sicherheit zu überwachen – ganz zu schweigen davon, dass eine Aktualisierung schwierig ist, da jede Änderung bedeutet, dass die gesamte Anwendung neu installiert werden muss. Das ist nicht besonders benutzerfreundlich und kann zur Stagnation bei Seitenaufrufen führen. Wenn Silverlight verwendet und eine Änderung am reichhaltigen Inhalt notwendig wird, wird serverseitig eine neue XAML-Datei generiert. Wenn der Benutzer das nächste Mal die Seite besucht, wird dieses XAML heruntergeladen, und die Funktionalität ohne Neuinstallation aktualisiert.

Das Herzstück von Silverlight ist das Browsererweiterungsmodul. Dieses rendert das XAML und zeigt die entstandene Grafik im Browser an. Das Browsererweiterungsmodul steht als kleiner Download bereit (unter 2 MB), der bei Bedarf installiert werden kann, wenn der Benutzer eine Website aufruft, die Silverlight-Inhalt enthält. Dieses Modul macht das zugrunde liegende Framework des XAML für JavaScript-Entwickler verfügbar. Dadurch wird Interaktion mit dem Inhalt auf Seitenebene möglich. Deshalb kann der Entwickler zum Beispiel Ereignishandler





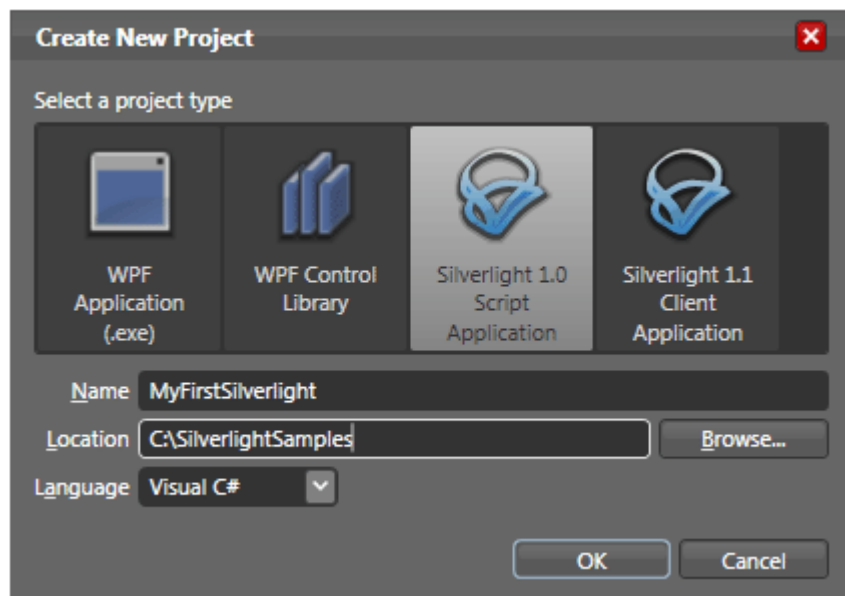
schreiben oder den XAML-Seiteninhalt mittels JavaScript-Code ändern.

Als Nächstes können Sie selbst aktiv werden und einen Blick auf unser erstes Silverlight-Projekt werfen.

[⚡ Zum Seitenanfang](#)

### Erstellen einer einfachen Silverlight-Anwendung

Zunächst einige Informationen zur Software „Microsoft Expression Blend“ und zum Erstellen einer einfachen Anwendung in XAML für Silverlight. Um eine Silverlight-Anwendung in Blend zu erstellen, wählen Sie **File** (Datei) -> **New Project** (Neues Projekt). Das Dialogfeld „New Project“ wird geöffnet. Siehe Abbildung 1.



**Abbildung 1: Erstellen eines neuen Silverlight-Projekts mit Blend**

Klicken Sie auf **OK**, um ein neues Projekt zu erstellen. Dieses Projekt wird eine Standard-HTML-Seite, JavaScript-Codebehind für diese Seite, ein XAML-Dokument, JavaScript-Codebehind für das XAML-Dokument und Silverlight.js enthalten.

Silverlight.js enthält den Code zum Herunterladen und Instanzieren des Silverlight-Steuerelements. Dieses wird Ihnen als Teil des Silverlight-SDKs zur Verfügung gestellt.

Default.html ist eine Standard-HTML-Webseite. Sie enthält drei JavaScript-Skriptverweise, die auf Silverlight.js, Default.html.js (enthält den anwendungsspezifischen Code zum Instanzieren von Silverlight) und Scene.xaml.js (enthält die Ereignishandler für im XAML definierte Anwendungsereignisse) verweisen.

Auf diese Weise wird die Seiten- (default.html) von der Instanzierungslogik (default.html.js) sowie der Entwurfs- (Scene.xaml) vom Ereigniscode (Scene.xaml.js) getrennt. Lassen Sie uns nun eine einfache Anwendung erstellen.

[⚡ Zum Seitenanfang](#)

### Erstellen der Benutzeroberfläche für einen Videoplayer

Fügen Sie Ihrem Projekt eine Videodatei hinzu. Klicken Sie dazu mit der rechten Maustaste auf die Projektdatei im Fenster „Project Files“ (Projektdateien) rechts oben auf dem Bildschirm, und wählen Sie **Add Existing Item...** (Vorhandenes Element hinzufügen) aus.

Wenn Sie eine WMV Datei auswählen und Ihrem Projekt hinzufügen, wird die Datei im Projektexplorer angezeigt, und Ihrer Szene wird ein Medienelement hinzugefügt.



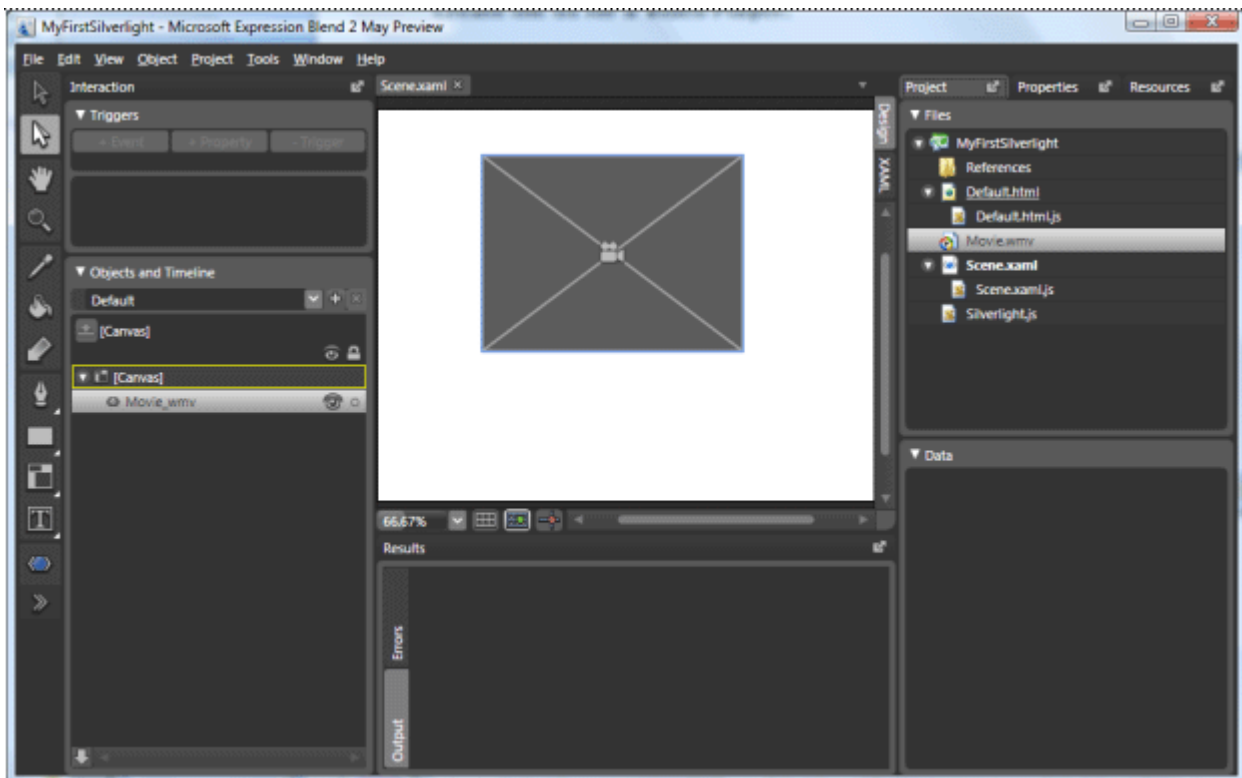
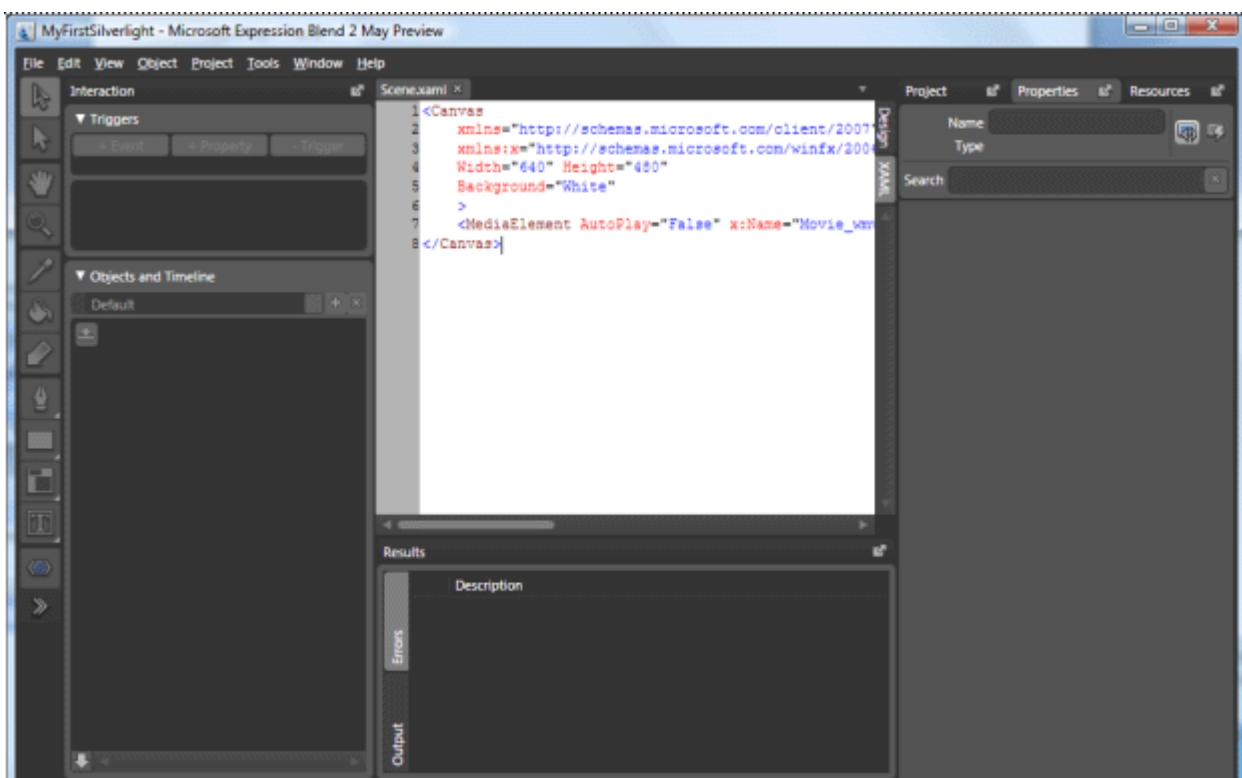


Abbildung 2: Hinzufügen eines Medienelements zur XAML-Szene

Wenn Sie Ihr Projekt jetzt ausführen, wird der Browser gestartet, und Ihr Video wird abgespielt.

Wenn Sie die automatische Wiedergabe des Videos beenden möchten, bearbeiten Sie das XAML. Auf der rechten Seite des XAML-Designers werden zwei Registerkarten angezeigt: **Design** und **XAML**. Wählen Sie die Registerkarte „XAML“ aus, um den XAML-Editor zu öffnen (siehe Abbildung 3.) Bearbeiten Sie den XAML-Text für **MediaElement**, um ein Attribut **AutoPlay=False** hinzuzufügen.



**Abbildung 3: Bearbeiten des XAML im XAML-Editor**

Wenn Sie die Anwendung jetzt ausführen, rendert der Silverlight-Inhalt den ersten Frame Ihres Videos, spielt ihn jedoch nicht ab.

**Hinzufügen von Steuerelementen zu Ihrem Videoplayer**

Fügen Sie der Anwendung zwei Textblöcke, hinzu, und geben Sie den Text **Play** und **Stop** sowie die Namen **txtPlay** und **txtStop** an. Das XAML sieht nun etwa folgendermaßen aus:

```
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/wpf/2006/xaml"
  Width="640" Height="480"
  Background="White">

  <MediaElement AutoPlay="False"
    x:Name="Movie_wmv"
    Width="320"
    Height="240"
    Canvas.Left="128"
    Canvas.Top="56"
    Source="Movie.wmv"
    Stretch="Fill"/>

  <TextBlock x:Name="txtPlay"
    Width="72"
    Height="24"
    Canvas.Left="136"
    Canvas.Top="336"
    Text="Play"
    TextWrapping="Wrap"/>

  <TextBlock x:Name="txtStop"
    Width="80"
    Height="24"
    Canvas.Left="136"
    Canvas.Top="368"
    Text="Stop"
```



```
TextWrapping="Wrap"/>

</Canvas>
```

**Fügen Sie anschließend dem XAML die Ereignishandlerdeklaration für die Textblöcke hinzu. Deklarieren Sie hierzu den Handler für einen Mausklick mithilfe des Attributs **MouseLeftButtonDown**.** Fügen Sie dem Textblock **txtPlay** einen Ereignishandler für **DoPlay** und dem Textblock **txtStop** einen Ereignishandler für **DoStop** hinzu. Das XAML sieht nun folgendermaßen aus:

```
<TextBlock x:Name="txtPlay" Width="72" Height="24" Canvas.Left="136"

    Canvas.Top="336" Text="Play" TextWrapping="Wrap"

    MouseLeftButtonDown="javascript: DoPlay"/>

<TextBlock x:Name="txtStop" Width="80" Height="24" Canvas.Left="136"

    Canvas.Top="368" Text="Stop" TextWrapping="Wrap"

    MouseLeftButtonDown="javascript: DoStop"/>
```

**Wenn der Benutzer auf einen der Textblöcke klickt, wird ein Ereignis ausgelöst, das Sie von einer JavaScript-Funktion aus abfangen und verarbeiten können.**

### Verarbeiten der Ereignisse in JavaScript

Mithilfe der Vorlage wurde eine Scene.xaml.js erstellt, mit der Benutzerereignisse in JavaScript aufgezeichnet und verarbeitet werden können. Die Ereignishandler **DoPlay** und **DoStop** wurden innerhalb des XAML angegeben. Deshalb sollten Sie sie hier Ihren Funktionscode implementieren. Der erforderliche Code sieht folgendermaßen aus:

```
function DoPlay(sender, eventArgs)
{
    var theHost = document.getElementById("SilverlightControl");
    var theMedia = theHost.content.findName("Movie_wmv");
    theMedia.Play();
}

function DoStop(sender, eventArgs)
{
    var theHost = document.getElementById("SilverlightControl");
    var theMedia = theHost.content.findName("Movie_wmv");
    theMedia.Stop();
}
```

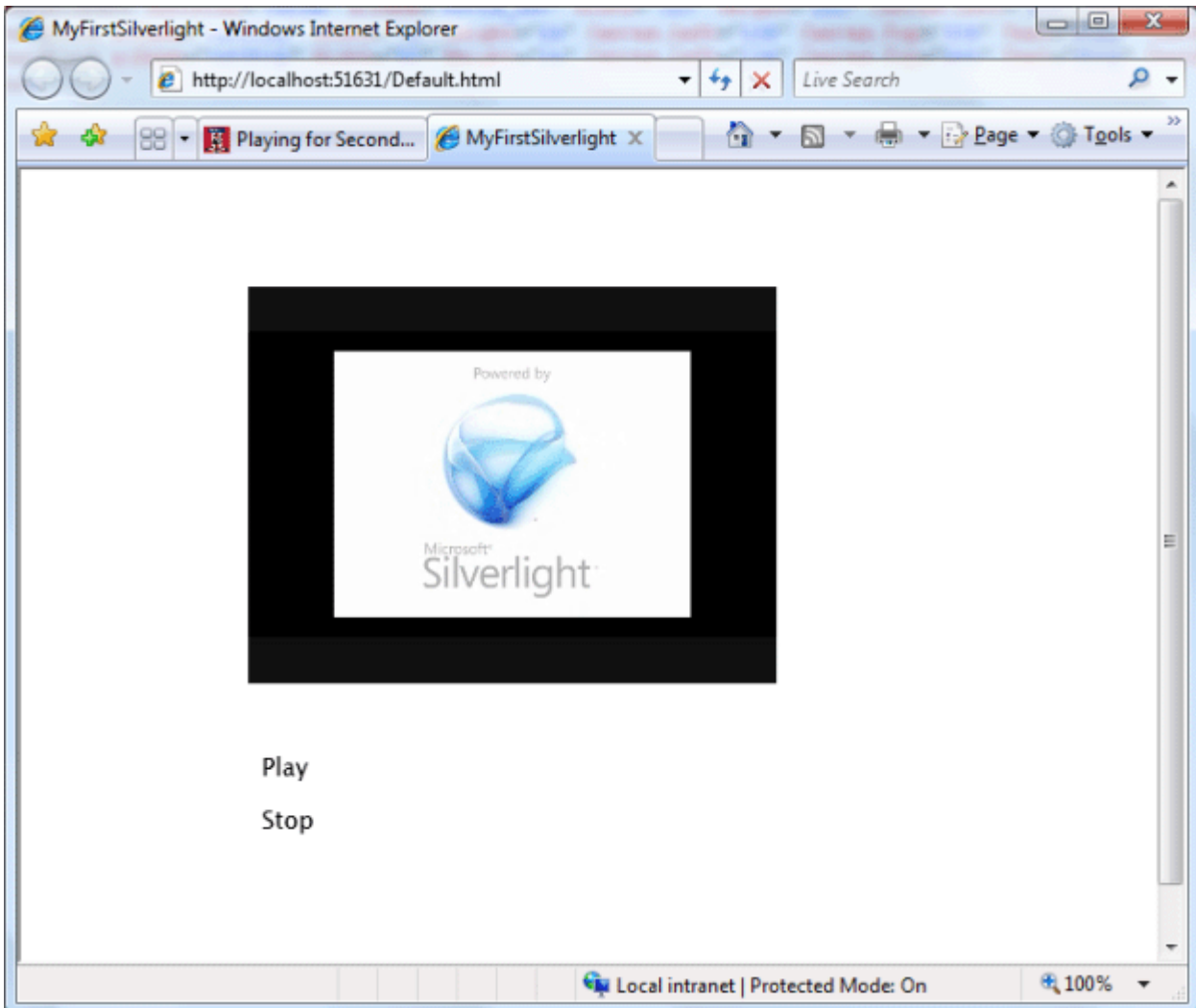
In diesem Fall trägt das Silverlight-Steuerelement den Namen „SilverlightControl“, und es wird mit der JavaScript-Variablen „theHost“ darauf verwiesen. Diese wird anschließend verwendet, um das **Media**-Element zu finden, das in diesem Fall den Namen „Movie\_wmv“ trägt. Als Sie dem Projekt den Film hinzugefügt haben, wurde dieses Medienelement für Sie erstellt und der Name des Films



als Grundlage für den Namen verwendet. Wenn Ihr Film den Namen „Movie.wmv“ hat, entsteht daraus der Name „Movie\_wmv“. Wenn Sie einen anderen Film verwenden, ändert sich der Name des Steuerelements entsprechend.

Das Medienelement verfügt über **Play**- und **Stop**-Methoden, mit denen die Wiedergabe des Mediums gestartet oder beendet werden kann.

Da nun ein Verweis auf das Medienelement vorhanden ist, können Sie diese Methoden aufrufen, um den Film anzuhalten oder zu starten. Betrachten Sie hierzu Abbildung 4.



**Abbildung 4: Ausführen der Anwendung**

Sie haben jetzt Ihre erste Silverlight-Anwendung erstellt! Weitere Ressourcen zu Silverlight finden Sie im neuen [Silverlight-Entwicklercenter](http://www.silverlight.net/) und unter <http://www.silverlight.net/>.

[^Zum Seitenanfang](#)

### Erläuterungen zum Silverlight-Aufruf

Die HTML-Seite ruft **createSilverlight()** auf. Dieses Element ist in der Codebehind-Seite Default.html.js enthalten.

```
Sys.Silverlight.createObjectEx({
    source: "Scene.xaml",
    parentElement: document.getElementById("SilverlightControlHost"),
```



```

    id: "SilverlightControl",
    properties: {
        width: "100%",
        height: "100%",
        version: "0.9"
    },
    events: {
        onLoad: Sys.Silverlight.createDelegate(scene, scene.handleLoad)
    }
});

```

Hierfür ist eine Reihe von Eigenschaften erforderlich, einschließlich der Eigenschaften, die zum Definieren des zu rendernden XAML und des Erscheinungsbilds des Silverlight-Steuerelements verwendet werden, sowie der **onLoad**- und **onError**-Ereignishandler.

Die Eigenschaft **source**: wird verwendet, um das XAML zu definieren, das das Silverlight-Steuerelement rendern soll. Dies kann eine externe Datei (wie in diesem Fall) oder ein benanntes **<script>**-Tag auf der Seite sein, das XAML enthält.

Wenn Sie ein Silverlight-Steuerelement auf einer Seite platzieren, sollte es innerhalb eines benannten **<DIV>**-Elements platziert werden. Die Eigenschaft **parentElement**: sollte mit dem Namen dieses **<DIV>**-Elements belegt werden.

Die ID des Steuerelements wird über die Eigenschaft **id**: angegeben.

Die physischen Attribute des Steuerelements, z. B. Höhe, Breite und Version, werden über ein Array festgelegt, das in die Eigenschaft **properties**: geladen wird. Den kompletten Eigenschaftensatz finden Sie in der [Dokumentation zum Silverlight-SDK](#).

[↑ Zum Seitenanfang](#)

### Schlussbemerkung

In diesem Whitepaper wurde Ihnen ein Überblick über Silverlight geboten. Sie haben erfahren, wie Silverlight sich in die Entwicklung der nächsten Generation von Webanwendungen einfügt. Sie haben gesehen, wie XAML verwendet wird, um die Spezifikationen des Designers mit den Tools des Entwicklers und der Bereitstellung für den Benutzer zu verbinden. Sie haben Expression Blend kennengelernt und gesehen, wie es verwendet wird, um Benutzeroberflächen für Ihre Webseite zu definieren, und wie diese Benutzeroberflächen mit JavaScript programmiert werden können.

Wir haben in diesem Artikel nur die Oberfläche von dem gestreift, was mit Silverlight alles möglich ist. Diese Technologie bietet eine reichhaltige Funktionalität, mit der Sie die Webanwendungen von morgen bereits heute erstellen können. Viel Spaß!



### **Beispiel von Microsoft zur Programmierung mit Silverlight**

Microsoft hat DirectX-Komponenten aus Windows dauerhaft entfernt und u.a. durch Silverlight ersetzt (analog dazu Adobe, das Macromedia und somit Flash (FLV, SWF etc.) gekauft und dafür das hauseigene SVG vom Markt genommen hat).

Nachfolgend der analysierte und aufbereitete Quelltext eines von Microsoft als Paradebeispiel veröffentlichten Silverlight-Demos "Grand Piano".

Die Analyse und Aufbereitung wurde notwendig, da gezeigt werden soll, dass die Programmierung mit Silverlight **extrem aufwendig** und relativ **kompliziert** ist. Im Gegensatz zum Quellcode von Microsoft zeigt der aufbereitete Quellcode die zwingend **dynamische Programmierung** von Silverlight: Reduzierung des Aufwandes der Verwaltung und Einführung der durchgehenden Modularität im Programmcode. - Mit anderen Worten: Vom Microsoft Quellcode und dessen Logik ist sehr wenig übrig geblieben: Programmierungsfehler von Microsoft wurde zugleich eliminiert.

Derjenige, der Silverlight als Software einsetzen will, muss zwingend Aufwand und Nutzen im System von Silverlight kalkulieren. Es könnte sein, dass Konkurrenzprodukte wie Flash oder SVG (beides von Adobe wobei SVG zugunsten Flash eingestellt wurde) wesentlich einfacher benutzbar sind. Mit anderen Worten: Silverlight ist **nicht** wärmstens zu empfehlen. - Wieso ein Rad mehrmals erfunden wird, ist neben der Prinzipienfrage eine Frage des Kommerz in der Softwarewelt (z.B. will Microsoft seine Server- und Programmierungssoftware mit Silverlight als Köder verkaufen).



## Original-Quellcode von Microsoft

Die Quellcode wurden in der Zeilenstruktur (Zeilenumbruch) unverändert übernommen (nur Schriftart und -größe wurden minimiert). Eine überlange Zeilen beginnt - automatisch umgebrochen - mit Beginn der Nachfolgezeile, die Teil ihrer Vorgängerzeile ist. Die Struktur der Quellcode geben ganz klare Hinweise auf Denkweisen und Fähigkeiten der Microsoft-Programmierer. Beim Lesen der Quellcode können die Haare zu Berge stehen (theoretisch müssten die Microsoft-Programmierer Kahlköpfe sein \*fg). **Ein Edelbeispiel für unglaublich schlechten Programmierungsstil ist die silverlight.js.**

### index.html

```
<!--
////////////////////////////////////
//
// default.html
//
//
// © 2007 Microsoft Corporation. All Rights Reserved.
//
// This file is licensed as part of the Silverlight 1.0 SDK, for details look
// here: http://go.microsoft.com/fwlink/?LinkID=89144&clcid=0x409
//
////////////////////////////////////
-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>Grand Piano</title>
  <script type="text/javascript" src="js/silverlight.js"></script>
  <script type="text/javascript" src="js/createSilverlight.js"></script>
  <script type="text/javascript" src="js/Piano.js"></script>

</head>
<body>
  <form>
    <div id="SilverlightControlHost" style="background:#FFFFFF">
      <script type="text/javascript">
        createSilverlight();
      </script>
    </div>
  </form>
</body>
</html>
```





**default.html.js (createsilverlight.js)**

```
////////////////////////////////////
//
// default.html.js
//
//
// © 2007 Microsoft Corporation. All Rights Reserved.
//
// This file is licensed as part of the Silverlight 1.0 SDK, for details look
// here: http://go.microsoft.com/fwlink/?LinkID=89144&clcid=0x409
//
////////////////////////////////////

function createSilverlight()
{

    var scene = new GrandPiano.Piano();
    Silverlight.createObjectEx({
        source: "xaml/GrandPiano.xaml",
        parentElement: document.getElementById("SilverlightControlHost"),
        id: "SilverlightControl",
        properties: {
            width: "1024",
            height: "530",
            background: 'white',
            isWindowless: 'false',
            framerate: '24',
            version: "0.9"
        },
        events: {
            onError: 'errorHandle', onLoad: Silverlight.createDelegate(scene, scene.handleLoad)
        }
    });
}

if (!window.Sys)
    window.Sys = {};

if (!window.Silverlight)
    window.Silverlight = {};

Silverlight.createDelegate = function(instance, method) {
    return function() {
        return method.apply(instance, arguments);
    }
}
```



**grandpiano.xaml.js (Piano.js)**

```

////////////////////////////////////
//
// grandpiano.xaml.js
//
//
// © 2007 Microsoft Corporation. All Rights Reserved.
//
// This file is licensed as part of the Silverlight 1.0 SDK, for details look
// here: http://go.microsoft.com/fwlink/?LinkID=89144&clcid=0x409
//
////////////////////////////////////
if (!window.GrandPiano)
    window.GrandPiano = {};

GrandPiano.Piano = function()
{
}

GrandPiano.Piano.prototype =
{
    handleLoad: function(control, userContext, rootElement)
    {
        this.control = control;

        //keyup and down should be attached to the root canvas
        this.rootElement = rootElement;
        this.rootElement.addEventListener("GotFocus", Silverlight.createDelegate(this, this.resetPiano));
        this.rootElement.addEventListener("KeyDown", Silverlight.createDelegate(this, this.handleKeyDown));
        this.rootElement.addEventListener("KeyUp", Silverlight.createDelegate(this, this.handleKeyUp));

        // Piano button event hookup: Find the button and then attach event handlers
        this.pathCanvas = rootElement.findName("pathCanvas");

        for(var i = 0; i<this.pathCanvas.Children.Count; ++i)
        {
            this.pathCanvas.Children.GetItem(i).addEventListener("MouseDown", Silverlight.createDelegate(this,
this.handleMouseDown));
            this.pathCanvas.Children.GetItem(i).addEventListener("MouseUp", Silverlight.createDelegate(this,
this.handleMouseUp));
        }
    },

    handleMouseDown: function(sender, eventArgs)
    {
        sender.captureMouse();
        var currentPianoKeyID = sender.name.substr(0, sender.name.indexOf("Path"));
        var currentImage = sender.findName("img" + currentPianoKeyID);
        currentImage.opacity = 1;
        var currentMediaElement = sender.findName(currentPianoKeyID);
        currentMediaElement.stop();
        currentMediaElement.play();
    },

    handleMouseUp: function(sender, eventArgs)
    {
        sender.releaseMouseCapture();
        var currentPianoKeyID = sender.name.substr(0, sender.name.indexOf("Path"));
        var currentImage = sender.findName("img" + currentPianoKeyID);
        currentImage.opacity = 0;
    },

    pressKey: function (currentPianoKeyID)
    {

```



```

var currentImage = this.control.content.findName("img" + currentPianoKeyID);
currentImage.opacity = 1;
var currentMediaElement = this.control.content.findName(currentPianoKeyID);
currentMediaElement.stop();
currentMediaElement.play();
},

depressKey: function (currentPianoKeyID)
{
    var currentImage = this.control.content.findName("img" + currentPianoKeyID);
    currentImage.opacity = 0;
},

handleKeyDown: function (sender, eventArgs)
{
    switch (eventArgs.Key)
    {
        case 55: this.pressKey("C"); break;
        case 48: this.pressKey("C2"); break;
        case 53: this.pressKey("D"); break;
        case 33: this.pressKey("D2"); break;
        case 32: this.pressKey("E"); break;
        case 51: this.pressKey("F"); break;
        case 36: this.pressKey("F2"); break;
        case 31: this.pressKey("G"); break;
        case 37: this.pressKey("G2"); break;
        case 43: this.pressKey("A"); break;
        case 39: this.pressKey("A2"); break;
        case 42: this.pressKey("B"); break;
    }
},

handleKeyUp: function (sender, eventArgs)
{
    switch (eventArgs.Key)
    {
        case 55: this.depressKey("C"); break;
        case 48: this.depressKey("C2"); break;
        case 53: this.depressKey("D"); break;
        case 33: this.depressKey("D2"); break;
        case 32: this.depressKey("E"); break;
        case 51: this.depressKey("F"); break;
        case 36: this.depressKey("F2"); break;
        case 31: this.depressKey("G"); break;
        case 37: this.depressKey("G2"); break;
        case 43: this.depressKey("A"); break;
        case 39: this.depressKey("A2"); break;
        case 42: this.depressKey("B"); break;
    }
},

resetPiano: function(sender, eventArgs)
{
    // In order to avoid problems when the control loses focus while a key is held down,
    // we use this function to reset the piano to its initial state (no key pressed). It is
    // called on the root canvas' GotFocus event.
    this.depressKey("C");
    this.depressKey("C2");
    this.depressKey("D");
    this.depressKey("D2");
    this.depressKey("E");
    this.depressKey("F");
    this.depressKey("F2");
    this.depressKey("G");
    this.depressKey("G2");
    this.depressKey("A");

```



```
this.depressKey("A2");  
this.depressKey("B");  
}  
}
```



**Silverright.js (silverlight.js)**

```

////////////////////////////////////
//
// Silverlight.js                      version 1.0
//
// This file is provided by Microsoft as a helper file for websites that
// incorporate Silverlight Objects. This file is provided under the Silverlight
// SDK 1.0 license available at http://go.microsoft.com/fwlink/?linkid=94240.
// You may not use or distribute this file or the code in this file except as
// expressly permitted under that license.
//
// Copyright (c) 2007 Microsoft Corporation. All rights reserved.
//
////////////////////////////////////

if(!window.Silverlight)window.Silverlight={};Silverlight._silverlightCount=0;Silverlight.ua=null;Silverlight.available=false;Silverlight.fw
linkRoot="http://go.microsoft.com/fwlink/?LinkID=";Silverlight.detectUserAgent=function(){var
a=window.navigator.userAgent;Silverlight.ua={OS:"Unsupported",Browser:"Unsupported"};if(a.indexOf("Windows
NT")>=0)Silverlight.ua.OS="Windows";else if(a.indexOf("PPC Mac OS X")>=0)Silverlight.ua.OS="MacPPC";else if(a.indexOf("Intel Mac
OS
X")>=0)Silverlight.ua.OS="MacIntel";if(Silverlight.ua.OS!="Unsupported"){if(a.indexOf("MSIE")>=0){if(navigator.userAgent.indexOf("Wi
n64")===-1){if(parseInt(a.split("MSIE")[1])>=6)Silverlight.ua.Browser="MSIE"}else if(a.indexOf("Firefox")>=0){var
b=a.split("Firefox/")[1].split(".");c=parseInt(b[0]);if(c>=2)Silverlight.ua.Browser="Firefox";else{var
d=parseInt(b[1]);if(c==1&&d>=5)Silverlight.ua.Browser="Firefox"}}}else
if(a.indexOf("Safari")>=0)Silverlight.ua.Browser="Safari"};Silverlight.detectUserAgent();Silverlight.isInstalled=function(d){var
c=false,a=null;try{var b=null;if(Silverlight.ua.Browser=="MSIE")b=new ActiveXObject("AgControl.AgControl");else
if(navigator.plugins["Silverlight Plug-
In"]){a=document.createElement("div");document.body.appendChild(a);if(Silverlight.ua.Browser=="Safari")a.innerHTML=<embed
type="application/x-silverlight" />;else a.innerHTML=<object type="application/x-silverlight" data="data;"
/>;b=a.childNodes[0]}document.body.innerHTML;if(b.IsVersionSupported(d))c=true;b=null;Silverlight.available=true}catch(e){c=false}if
(a)document.body.removeChild(a);return c};Silverlight.createObject=function(l,g,m,j,k,i,h){var
b={},a=j,c=k;a.source=l;b.parentElement=g;b.id=Silverlight.HtmlAttributeEncode(m);b.width=Silverlight.HtmlAttributeEncode(a.width);
b.height=Silverlight.HtmlAttributeEncode(a.height);b.ignoreBrowserVer=Boolean(a.ignoreBrowserVer);b.inplaceInstallPrompt=Boolea
n(a.inplaceInstallPrompt);var
e=a.version.split(".");b.shortVer=e[0]+ "."+e[1];b.version=a.version;a.initParams=i;a.windowless=a.isWindowless;a.maxFramerate=a.frame
rate;for(var d in c){if(c[d]&&d!="onLoad"&&d!="onError"){a[d]=c[d];c[d]=null}delete a.width;delete a.height;delete a.id;delete
a.onLoad;delete a.onError;delete a.ignoreBrowserVer;delete a.inplaceInstallPrompt;delete a.version;delete a.isWindowless;delete
a.framerate;delete a.data;delete
a.src;if(Silverlight.isInstalled(b.version)){if(Silverlight._silverlightCount==0){if(window.addEventListener)window.addEventListener("o
nunload",Silverlight.__cleanup,false);else window.attachEvent("onunload",Silverlight.__cleanup);var
f=Silverlight._silverlightCount++;a.onLoad="_slLoad"+f;a.onError="_slError"+f;window[a.onLoad]=function(a){if(c.onLoad)c.onLoad(
document.getElementById(b.id),h,a)};window[a.onError]=function(a,b){if(c.onError)c.onError(a,b);else
Silverlight.default_error_handler(a,b)};slPluginHTML=Silverlight.buildHTML(b,a)}else
slPluginHTML=Silverlight.buildPromptHTML(b);if(b.parentElement)b.parentElement.innerHTML=slPluginHTML;else return
slPluginHTML};Silverlight.supportedUserAgent=function(){var
a=Silverlight.ua,b=a.OS=="Unsupported"||a.Browser=="Unsupported"||a.OS=="Windows"&&a.Browser=="Safari"||a.OS.indexOf("Mac
")>=0&&a.Browser=="IE";return b};Silverlight.buildHTML=function(c,d){var
a=[],e,i,g,f,h;if(Silverlight.ua.Browser=="Safari"){a.push("<embed "e="";i="";g="";f="";h=" type="application/x-silverlight"/>+"<iframe
style='visibility:hidden;height:0;width:0'/>"}else{a.push("<object type="application/x-silverlight" data="data;"e="";i="<param
name="";g=" value="";f=""/>;h="</object>"}a.push(" id="+c.id+" width="+c.width+" height="+c.height+" 'e);for(var b in
d){if(d[b])a.push(i+Silverlight.HtmlAttributeEncode(b)+g+Silverlight.HtmlAttributeEncode(d[b])+f);a.push(h);return
a.join("");}Silverlight.default_error_handler=function(e,b){var d,c=b.ErrorType;d=b.ErrorCode;var a="\nSilverlight error message
\n";a+="ErrorCode: "+d+"\n";a+="ErrorType: "+c+"\n";a+="Message: "+b.ErrorMessage+"\n";if(c=="ParserError"){a+="XamlFile:
"+b.xamlFile+"\n";a+="Line: "+b.lineNumber+"\n";a+="Position: "+b.charPosition+"\n"}else
if(c=="RuntimeError"){if(b.lineNumber!=0){a+="Line: "+b.lineNumber+"\n";a+="Position: "+b.charPosition+"\n"}a+="MethodName:
"+b.methodName+"\n"}alert(a)};Silverlight.createObjectEx=function(b){var
a=b,c=Silverlight.createObject(a.source,a.parentElement,a.id,a.properties,a.events,a.initParams,a.context);if(a.parentElement==null)retur
n c;Silverlight.buildPromptHTML=function(l){var a=null,d=Silverlight.fwlinkRoot,c=Silverlight.ua.OS,b="92822",e,f="Get Microsoft
Silverlight",m="0x409";if(l.inplaceInstallPrompt){var n="98109";i;if(Silverlight.available){e="96189";i="96422"}else{e="96188";i="96422"}var
h="93481",g="93483";if(c=="Windows"){b="92799";h="92803";g="92805"}else if(c=="MacIntel"){b="92808";h="92804";g="92806"}else
if(c=="MacPPC"){b="92807";h="92815";g="92816"}var k="By clicking <b>Get Microsoft Silverlight</b> you accept the<br />a
title="Silverlight License Agreement" href="{2}" target="_top" style="text-decoration: underline; color: #96C5E1"><b>Silverlight license
agreement</b></a>;j="Silverlight updates automatically, <a title="Silverlight Privacy Statement" href="{3}" target="_top" style="text-
decoration: underline; color: #96C5E1"><b>learn more</b></a>;a=<table border="0" cellpadding="0" cellspacing="0"

```



```
width="206px"><tr><td></td></tr><tr><td style="width: 206px; margin: 0px; background: #FFFFFF;
color: #C7C7C7; text-align: left; border-left-style: solid; border-right-style: solid; padding-left: 6px; padding-right: 6px; padding-top:
3px; padding-bottom: 0px; border-width: 2px; border-color: #c7c7bd; font-family: Verdana; font-size: 55% ">'+k+'</td></tr><td></td></tr><tr><td style="width: 206px; margin: 0px; background: #D8EFF9; color: #C7C7C7;
text-align: left; border-left-style: solid; border-right-style: solid; padding-left: 6px; padding-right: 6px; padding-top: 0px; padding-
bottom: 2px; border-width: 2px; border-color: #c7c7bd; font-family: Verdana; font-size: 55% ">'+j+'</td></tr><td></td></tr></table>;a=a.replace("{2}",d+h);a=a.replace("{3}",d+g);a=a.replace("{4}",d+i);a=a.replace("{5}",d+n)}else{if(Silverlight.available
e="94377";else e="92801";if(c=="Windows")b="92800";else if(c=="MacIntel")b="92812";else if(c=="MacPPC")b="92811";a='<div
style="display:block; width: 205px; height: 67px;"></div>'}a=a.replace("{0}",b);a=a.replace("{1}",d+e+"&cid="+m);return
a);Silverlight.__cleanup=function(){for(var a=Silverlight._silverlightCount-1;a>=0;a--
){window["__slLoad"+a]=null;window["__slError"+a]=null;if(window.removeEventListener)window.removeEventListener("unload",Sil
verlight.__cleanup,false);else
window.detachEvent("onunload",Silverlight.__cleanup)};Silverlight.followFWLink=function(a){top.location=Silverlight.fwlinkRoot+Stri
ng(a)};Silverlight.HtmlAttributeEncode=function(c){var a,b="";if(c==null)return null;for(var
d=0;d<c.length;d++){a=c.charCodeAt(d);if(a>96&&a<123||a>64&&a<91||a>43&&a<58&&a!=47||a==95)b=b+String.fromCharCode(a);els
e b=b+"&#"+"a+";"}return b}
```



**grandpiano.xaml (GrandPiano.xaml)**

```

<!--
////////////////////////////////////
//
// grandpiano.xaml
//
//
// Â© 2007 Microsoft Corporation. All Rights Reserved.
//
// This file is licensed as part of the Silverlight 1.0 SDK, for details look
// here: http://go.microsoft.com/fwlink/?LinkID=89144&clcid=0x409
//
////////////////////////////////////
-->
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  >

  <Image Source="assets/images/bg.jpg" Height="500" Width="1024"/>

  <Image x:Name="imgC" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/C.jpg" Opacity="0"/>
  <Image x:Name="imgC2" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/C2.jpg" Opacity="0"/>
  <Image x:Name="imgD" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/D.jpg" Opacity="0"/>
  <Image x:Name="imgD2" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/D2.jpg" Opacity="0"/>
  <Image x:Name="imgE" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/E.jpg" Opacity="0"/>
  <Image x:Name="imgF" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/F.jpg" Opacity="0"/>
  <Image x:Name="imgF2" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/F2.jpg" Opacity="0"/>
  <Image x:Name="imgG" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/G.jpg" Opacity="0"/>
  <Image x:Name="imgG2" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/G2.jpg" Opacity="0"/>
  <Image x:Name="imgA" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/A.jpg" Opacity="0"/>
  <Image x:Name="imgA2" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/A2.jpg" Opacity="0"/>
  <Image x:Name="imgB" Height="500" Width="1024" Canvas.Top="0" Source="assets/images/B.jpg" Opacity="0"/>

  <MediaElement x:Name="C" AutoPlay="false" Source="assets/media/C.wma"/>
  <MediaElement x:Name="C2" AutoPlay="false" Source="assets/media/C2.wma"/>
  <MediaElement x:Name="D" AutoPlay="false" Source="assets/media/D.wma"/>
  <MediaElement x:Name="D2" AutoPlay="false" Source="assets/media/D2.wma"/>
  <MediaElement x:Name="E" AutoPlay="false" Source="assets/media/E.wma"/>
  <MediaElement x:Name="F" AutoPlay="false" Source="assets/media/F.wma"/>
  <MediaElement x:Name="F2" AutoPlay="false" Source="assets/media/F2.wma"/>
  <MediaElement x:Name="G" AutoPlay="false" Source="assets/media/G.wma"/>
  <MediaElement x:Name="G2" AutoPlay="false" Source="assets/media/G2.wma"/>
  <MediaElement x:Name="A" AutoPlay="false" Source="assets/media/A.wma"/>
  <MediaElement x:Name="A2" AutoPlay="false" Source="assets/media/A2.wma"/>
  <MediaElement x:Name="B" AutoPlay="false" Source="assets/media/B.wma"/>

  <Canvas x:Name="pathCanvas">
    <Path x:Name="CPath" Opacity="0.01" Fill="#FFFFFFFF" Stretch="Fill" Stroke="#FF000000" Width="161.666656494141"
      Height="272.333312988281" Canvas.Left="135.5" Canvas.Top="138.83332824707" Data="M272.66667,140 L136,330 147.33333,336
      162.66667,410.66667 247.33333,410 234.66667,330.66667 278,248 254,242.66667 258.66667,200.66667 296.66667,139.33333 272.66667,140" />
    <Path x:Name="C2Path" Opacity="0.01" Fill="#FFFFFFFF" Stretch="Fill" Stroke="#FF000000" Width="91" Height="136.333319306374"
      Canvas.Left="263.5" Canvas.Top="105.500005066396" Data="M325.33333,106.66667 L264,202.66667 266,241.33333 307.33333,238.66667
      354,138 351.33333,106.66666 326.66667,106" />
    <Path x:Name="DPath" Opacity="0.01" Fill="#FFFFFFFF" Stretch="Fill" Stroke="#FF000000" Width="131" Height="274.333297729492"
      Canvas.Left="247.499999999999" Canvas.Top="137.499984741212" Data="M358,142.66667 L311.33333,243.33333 288.66667,246
      248,330.66667 258,338 266.66667,411.33333 354,410.66665 347.33333,330.66668 374,246.66668 350,243.33336 352.66667,202.66669
      378,138.00005 360.66667,140.00005" />
    <Path x:Name="D2Path" Opacity="0.01" Fill="#FFFFFFFF" Stretch="Fill" Stroke="#FF000000" Width="70.3333129882813"
      Height="134.333319425583" Canvas.Left="356.83334350586" Canvas.Top="107.500003457069" Data="M394.66667,108.66667
      L357.33333,204 359.33333,241.33333 398,240.66667 426.66667,134.66667 425.33333,108" />
    <Path x:Name="EPath" Opacity="0.01" Fill="#FFFFFFFF" Stretch="Fill" Stroke="#FF000000" Width="117.666656494141"
      Height="274.333343505859" Canvas.Left="358.833343505859" Canvas.Top="137.5" Data="M431.33333,140 L403.33333,245.33333
      381.33333,248 359.33333,330 368.66667,339.33333 372,411.33333 458,411.33333 455.33333,331.33333 476,138 433.33333,138.66667" />
  </Canvas>
</Canvas>

```



```
<Path x:Name="FPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="99.6666564941406"
Height="273.666680037975" Canvas.Left="470.83334350586" Canvas.Top="138.833345949649" Data="M486.66667,140.66667
L471.33333,332.477.33333,334.476.66667,412.563.33333,411.33335.564,339.33334.570,328.560,246.66665.535.33333,243.33331.530,139.3333
486.66667,139.33329" />
<Path x:Name="F2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="51" Height="133.666656494141"
Canvas.Left="534.166687011719" Canvas.Top="108.16665649414" Data="M538.66667,109.33333 L534.66667,137.33333 542,241.33333
584,240.66666 584.66667,205.33334 568,108.66667 540.66667,108.66668" />
<Path x:Name="GPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="111" Height="274.333343505859"
Canvas.Left="570.166687011719" Canvas.Top="137.5" Data="M570.66667,248 L582.66667,336.582,411.33333.669.33333,410.66667
672.66667,332.66667.680.66667,330.655.33333,245.33333.631.33333,244.601.33333,138.578.66667,138.594.66667,244.66667.571.33333,248" />
<Path x:Name="G2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="76.3333129882813"
Height="131.666656494141" Canvas.Left="603.5" Canvas.Top="110.166664123534" Data="M612,111.33333 L604,136.66667.636,238.66667
636.66667,241.33333.679.33333,241.33333.679.33333,206.66667.641.33333,111.33334.612.66667,110.66668" />
<Path x:Name="APath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="138.333312988281"
Height="274.333343505859" Canvas.Left="655.5" Canvas.Top="137.5" Data="M665.33333,246 L694.66667,336.66667.686,411.33333
774.66667,410.785.33333,335.33333.793.33333,328.66667.750.66667,246.66667.722,243.33333.673.33333,138.656,138.66667
688.66667,216.66667.685.33333,245.33333.666,246" />
<Path x:Name="A2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="95" Height="131"
Canvas.Left="676.166687011719" Canvas.Top="110.833320617675" Data="M682,111.33333 L676.66667,138.66667.730.66667,240.66667
768.66667,241.33333.770.66667,204.00001.713.33333,111.33334.683.33333,111.33335" />
<Path x:Name="BPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="172.333312988281"
Height="275.666687011719" Canvas.Left="733.5" Canvas.Top="136.833320051431" Data="M758,247.33333 L806,333.33333.790,412.878,412
894.66667,334.905.33333,327.33333.764.66667,138.66667.734,137.33333.776.66667,207.33334.778.66667,246.66668.758.66667,248" />
</Canvas>

</Canvas>
```





**Analysierte und aufbereitete Quellcode****index.html**

```

<!--
////////////////////////////////////
//
// default.html
//
//
// © 2007 Microsoft Corporation. All Rights Reserved.
//
// This file is licensed as part of the Silverlight 1.0 SDK, for details look
// here: http://go.microsoft.com/fwlink/?LinkID=89144&clcid=0x409
//
////////////////////////////////////
-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
  <title>Grand Piano</title>
  <script type="text/javascript" src="GrandPiano.js"></script>
    <!-- Daten für Grand-Piano bereitstellen
        Routinen der Animation des Grand-Piano bereitstellen und instanzieren
    -->
  <script type="text/javascript" src="silverlight.js"></script>
    <!-- prüfen auf Möglichkeit der Nutzung von Silverlight u.a.
        ob Betriebssystem UND Browser von Silverlight unterstützt werden
        Silverlight-Verwaltung erzeugen
        Animation wird nicht automatisch gestartet (siehe unten)
    -->
  <script type="text/javascript">
function Y_MicrosoftGrandPianoAktivieren() // benötigt instanziierten DIV-Container (siehe unten)
{var X00=' nicht';
  var X01=' aktiv !';
  var X02=createSilverlight();              // Silverlight-Animation starten silverlight.js
                                           // liefert true, wenn Animation erfolgreich gestartet ist

  if(X02){X00=X01;}
  else{X00+=X01;}
  alert('Microsoft Grand-Piano ist' + X00);
}
</script>
</head>
<body>
<div id="SilverlightControlHost" style="background:#FFFFFF"></div>
    <!-- DIV-Container des Silverlight-HTML-Elementes
        Änderung des ID auch in GrandPiano.js hinterlegen
    -->
  <script type="text/javascript">
Y_MicrosoftGrandPianoAktivieren();          // benötigt DIV-Container
</script>
</body>
</html>

```



**GrandPiano.js**

```
// #####
//
//      GrandPiano.js
//
//      muss vor silverlight.js geladen werden
//
//      anzupassende JavaScript-Daten   zum Grand Piano
//      anzupassende JavaScript-Routinen zum Grand-Piano
//
//      Hinweis: Es gibt noch XAML-Daten in der GrandPiano.xaml
//              Änderungen in der XAML-Datei können JavaScript-Daten beeinflussen (siehe unten)
//
//      JS-Datei wird von silverlight.js benötigt:
//      Es werden Variablen erzeugt, die ij silverlight.js gelesen werden:
//      X1_SilverlightControl_Version
//      X1_SilverlightControlHost_ID
//      X1_Daten_OBJECT_ElementBezeichner
//      X1_Daten_PARAM_ElementBezeichner
//      X1_Silverlight_DatenElementBezeichner
//      X1_Silverlight_DatenElementBezeichner_parentElement_Index
//      X1_Silverlight_DatenElementBezeichner_events_Index
//      X1_Silverlight_DatenElementBezeichner_onError_Index
//      X1_Silverlight_DatenElementBezeichner_onLoad_Index
//      X1_SilverlightDaten
//      X1_GrandPiano
//
//      aufbereitete JS-Datei anhand der Programcode der Microsoft-programmierten JS-Dateien
//      silverlight.js
//      createsilverlight.js
//
//      Aufbereitung
//      erfolgt damit      Quelltext lesbar wird
//                        Programmlogik erkennbar wird
//      mit Entfernung aller Logik- und Programmierungsfehler, die von Microsoft eingebaut
//      wurden
//
//      Sollte Der Internet Explorer mit Start der index.html regelmäßig einen Adressfehler bringen,
//      der auch die Adresse 0x00000000 umfassen kann, ist das Silverlight-ActiveX-Control
//      defekt und es muss neu installiert werden: Dazu lade man sich die aktuellen Updates
//      zum IE per http://msdn2.microsoft.com/en-us/library/bb188743.aspx
//      zu Silverlight per http://silverlight.net/Default.aspx
//      herunter und installiere diese. Man beachte Nicht-Kompatibilität von
//      Silverlight-Varianten (inkl. deren Beta-Versionen).
//
//      WARNUNG: Microsoft zeigt mit seiner JS-Datei, wie man NICHT programmieren sollte:
//      Es werden Daten und Programmcode munter gemixt, so dass der
//      Quellcode nur als wertlose Einzelfall-Lösung dient.
//      Silverlight wird datenseitig konsequent NICHT dynamisch programmiert:
//      Daten liegen teilweise als Programmcode-Teile vor und zwar derart
//      festkodiert, dass deren Kapselung eine völlig andere Programmlogik
//      erzwingt, die aber Voraussetzung für dynamische Webseiten mit
//      geringem Pflegeaufwand ist.
//
//      Microsoft zeigt mit der Einzelfall-Lösung, dass die Programmierung mit
//      Silverlight nicht jedermanns Sache bei Microsoft ist.
//      Es werden Konzeptfehler on Silverlight sichtbar: Daten und Programmcode
//      der Anwendung sind von denen, die das Silverlight-Laufzeitsystem benötigt,
//      nur schwer trennbar, so dass deren Verquickung IMMER eine Einzelfalllösung
//      darstellt, wenn nicht genügend Programmierungsaufwand betrieben wird.
//      Mit anderen Worten: Es muss zwingend geprüft werden, ob Trennung möglich
//      ist: Wenn nein, sollte das Projekt mit anderer Laufzeitumgebung als
//      Silverlight geschrieben werden: Einzellösungen sind meist supportaufwendig
//      und die Kinderkrankheiten des programmierten Silverlight-Konzeptes
//      sind genau genommen einfach nur Müll - und das als Aushängeschild per
//      Grand-Piano mit Microsoft-Signatur.
//      Faulheiten des Programmieres führt zu uneinheitlichem Quelltext als Einzellösung:
//      Die Verwendung von Variablenamen ist Elend pur und gleicht dem von Anfängern.
```



```
//          Wegen Faulheit werden beim Codieren werden Anweisungen verwendet, deren
//          Komplexität mehr Laufzeit kostet, als bei Verwendung einfacher
//          Anweisungen.
//          Vorgaben von Microsoft zur Quelltext-Optimierung im Sinne der
//          Script-Laufzeit-Optimierung werden konsequent ignoriert.
//          Die Aufarbeitung des Quellcode zeigt Ansätze, wo dynamische Programmierung
//          notwendig und sinnvoll ist, ersetzt aber die vom Microsoft-Programmierer
//          verwendete Logik nicht komplett (aus Mist lässt sich kein Gold machen),
//          da ansonsten die Anwendung komplett neu programmiert werden müsste.
//          Die Verwendung von HTTPRequest zur Daten- und Script-Programmcodes-Beschaffung
//          zur Laufzeit ist nicht eingebaut worden, empfiehlt sich aber.
//          Die Routinen sollten möglichst als eigenständiges System unabhängig von Zeigern von Silverlight
//          programmiert werden und über Silverlight-Variablen bzw. -Zeiger referenziert werden
//          (z.B. Zeiger auf Funktion kopieren nach Silverlight-Funktionsvariable).
//          Silverlight wird von Microsoft permanent verändert und
//          Schnittstellenkonsistenz ist Sache von Microsoft: Werden Schnittstellen verändert,
//          so sind wenigstens die Routinen als System unberührt und der Fehler ist schneller
//          findbar.
//          Microsoft will anhand Silverlight seine (sowas teureren) Developer-Produkte wie Visual
//          und Co. verkaufen (Testversionen sind zeitlich limitiert) und zugleich ASP
//          in die Adobe-Flash-Konkurrenz drücken - allerdings ist das mehr ein Eigentor.
//          Alternativen zur Silverlight-Technik gabs schon seit 1998 (also rechtzeitig VOR Adobe Flash
//          und als Betriebssystemkomponenten, auf die u.a. auch Office-Powerpoint aufbaute (bis hin zur
//          Timeline-Programmierung per HTML, JavaScript oder VBasic) in DirectX mit Direct3D und
//          DirectMedia (inklusive HTML- und Javascript-Schnittstelle + kostenlosem SDK für Web-Desing), die
//          aber inzwischen von Microsoft ersatzlos und gnadenlos entfernt wurden (u.a. mit der Begründung,
//          dass neueres Windows neuere und sichere Schnittstellen benötigt). Die Abschaltung ist allerdings
//          ganz klar ein Analogon zum Wechsel des Treibersystems zwischen Win XP SP1 und 2 sowie erneut
//          von Win XP zu Win Vista (verbunden mit Folgekosten (abgesehen von Hardwarekosten): User, die
//          weiterhin Schnittstellen für Webseiten mit Animation auf DirectX-Basis nutzen wollen, werden
//          gnadenlos vom Markt gefegt: Sie müssen umstellen (und so auch Developer-Produkte kaufen) oder
//          zur Konkurrenz wechseln, denn ein höherer Programmierungsaufwand lohnt nur bei dauerhafteren
//          Schnittstellensystemen).
//          DirectX wurde gezielt nicht mehr abwärtskompatibel gehalten, da zusätzlich Net-Framework
//          gedrückt werden soll (Pendant zu Sun Java): Silverlight + DirectX mit Net-Framework sollen
//          zusammenarbeiten - möglichst mit ASP und das mit Browsern wie Safari oder Firefox. (Net-
//          Framework ist in Versionen parallel installierbar, Silverlight jedoch nicht).
//
// Sämtliche Entwicklungstools der Express-Serie (Blend, Media etc.) gibt es
// als Trial für einen begrenzten Laufzeitraum und sind ansonsten zu kaufen.
//
// Microsoft Expression Blend Free Trial
//
// We are pleased to present a trial version of Microsoft® Expression Blend™.
// This fully functioning version will expire 60 days after installation.
// This trial can be unlocked when you receive your purchased copy of the
// product by entering the product key found on the box.
//
// Die Vollversion kostet 600 US-Dollar
//
// Microsoft Expression Studio (Free evaluation is available for download)
//
// ist als Trial in Komponenten zerlegt, die verschiedene Laufzeiten haben
// z.B.
//
// Microsoft Expression Blend Free Trial
//
// We are pleased to present a trial version of Microsoft® Expression Blend™.
// This fully functioning version will expire 60 days after installation.
// This trial can be unlocked when you receive your purchased copy of the
// product by entering the product key found on the box.
//
// Whether creating standards-based Web user interfaces by using Microsoft ASP.NET,
// building cross-platform rich interactive applications with Silverlight,
// or creating ultimate Windows Vista experiences with Windows Presentation
```



```
//
//      Foundation (WPF), Expression tools enable designers to fully apply their
//      creativity while collaborating with developers using Microsoft Visual Studio.
//      Expression Studio includes the following products:
//      Expression Web for building standards-based Web sites.
//      Expression Design for professional illustration and graphic design.
//      Expression Blend for creating engaging Web-connected experiences for Microsoft Windows.
//      This special preview version is compatible with creating content for Silverlight 1.0
//      Release Candidate and Silverlight 1.1 Alpha.
//      Expression Media for assets management as well as media encoding.
//      This version also includes the Expression Encoder tool for transcoding video content.
//
//
// Windows Movie Maker
//
//
//      Import audio and video to your computer from a digital video camera, and then use the
//      imported content in your movies. You can also import existing audio, video, or still
//      pictures into Windows Movie Maker to use in the movies you create. After editing the audio
//      and video content in Windows Movie Maker, which can include adding titles, transitions,
//      or effects, you can then publish your final movie and share it with your friends and family.
//      Windows Movie Maker is also available for Microsoft Windows XP.
//
//
// #####

// ++++++
//
//      anzupassende Daten
//
// ++++++

// +++++ XAML-Datenquelle ++++++
// Bild- und Sound sind paarweise verfügbar und haben im Paar IDENTISCHEN Präfix, der
//      als PianoTastenID verwendet wird
//      als Suffix in x:Name von Elementen des Canvas laut X1_XAML_pathCanvas in der XAML-Datei verwendet wird
//      als Präfix in x:Name von Image und MediaElement in der XAML-Datei verwendet wird
//      Für alle Elemente eines Canvas sollte der Wert von x:Name einmalig sein (nicht mehrfach auftreten).
var X1_SilverlightControl_XAML_Source="GrandPiano.xml";

// +++++ Silverlight-Eigenschaften als HTML-Objekt (OBJECT bzw. EMBED) ++++++
var X1_SilverlightControl_Breite="1024";
var X1_SilverlightControl_Hoehe="530";
var X1_SilverlightControl_Background_Farbe="white";
var X1_SilverlightControl_Fensterrahmen="false";
var X1_SilverlightControl_maxFramerate="25";

// +++++ Version ++++++
var X1_SilverlightControl_Version="0.9";           // wird von silverlight.js gelesen

// +++++ ID des Containers (Eltern) des Silverlight-HTML-Objektes laut index.html
var X1_SilverlightControlHost_ID="SilverlightControlHost"; // ID des DIV-Container des Silverlight-Objektes
// Container in index.html erzeugt
// wird von silverlight.js gelesen

// +++++ ID des Silverlight-HTML-Objektes (OBJECT bzw. EMBED) ++++++
var X1_SilverlightControl_ID="SilverlightControl_ID"; // frei wählbar

// +++++ Datenstruktur zur Versorgung des Silverlight-HTML-Objektes mit Daten ++++++
// Silverlight wird mit einer Datenstruktur versorgt.
// Silverlight wird per HTML-Objekt erzeugt: OBJECT bzw. EMBED
// Damit müssen Elemente, deren Werte, das HTML-Objekt versorgen, im Bezeichner konform mit den Attributen
//      zum HTML-Objekt sein.
// OBJECT erhält zusätzlich PARAM, so dass Struktur-Elemente PARAM versorgen müssen
```



```
// ----- Elemente der Datenstruktur: Bezeichner -----

var X1_Silverlight_DatenElementeBezeichner=new Array // indexsynchron zu X1_Silverlight_DatenElementeWerte
// wird von silverlight.js gelesen
// Feld der Bezeichner u.a. von Attributen des OBJECT-Tags
// des PARAM-Tag
// Bezeichner darf nicht doppelt auftauchen
// nicht verwendbare Bezeichner --> siehe Pflicht-Elemente
// Bezeichner für Events
// Es gibt die Eventstruktur als Pflicht-Element, das weitere Pflichtelemente umfassen kann (siehe dort)
// Zusätzliche zu den Pflichtelementen) benötigte Bezeichner für Eventsstruktur-Elemente
// müssen später per Punktnotation verwendet werden (wie die Pflichtelemente in der Eventstruktur)
// Bsp.: Eventstruktur events
// Pflichtelement onload also Adressierung per events.onload
// Hinweis: Anstelle Punktnotation ist auch Index-Adressierung per Array möglich, wobei der Index
// eine Stringvariable ist, die den Bezeichner des Elementes enthält, und das Feldelement
// den Wert des Elementes.
// Bsp.: Datenstruktur.souce identisch mit Index='source'; Datenstruktur[Index]
// Jeder Punkt zeigt einen Unter-Index an
// Bsp.: Datenstruktur.properties.width identisch mit
// Datenstruktur['properties']['width']
// Kleinschreibung zulässig für Attribute in HTML-Tags
('source', // 0 PARAM-Attribut für OBJECT
'id', // 1 ID-Attribut des OBJECT-Tags
'width', // 2 WIDTH-Attribut des OBJECT-Tags
'height', // 3 HEIGHT-Attribut des OBJECT-Tags
'background', // 4 PARAM-Attribut für OBJECT
'windowless', // 5 PARAM-Attribut für OBJECT
'maxFramerate' // 6 PARAM-Attribut für OBJECT
);

var X1_Silverlight_DatenElementeBezeichner_souce_Index=0; // 'source'
var X1_Silverlight_DatenElementeBezeichner_id_Index=1; // 'id'
var X1_Silverlight_DatenElementeBezeichner_width_Index=2; // 'width'
var X1_Silverlight_DatenElementeBezeichner_height_Index=3; // 'height'
var X1_Silverlight_DatenElementeBezeichner_background_Index=4; // 'background'
var X1_Silverlight_DatenElementeBezeichner_windowless_Index=5; // 'windowless'
var X1_Silverlight_DatenElementeBezeichner_maxFramerate_Index=6; // 'maxFramerate'

// ----- Elemente der Datenstruktur: Werte -----

var X1_Silverlight_DatenElementeWerte=new Array // indexsynchron zu X1_Silverlight_DatenElementeBezeichner
(X1_SilverlightControl_XAML_Source, // source
X1_SilverlightControl_ID, // id
X1_SilverlightControl_Breite, // width
X1_SilverlightControl_Hoehe, // height
X1_SilverlightControl_Background_Farbe, // background
X1_SilverlightControl_Fensterrahmen, // windowless
X1_SilverlightControl_maxFramerate // maxFramerate
);

// ----- Datenelemente aus X1_Silverlight_DatenElementeBezeichner für Erzeugung des
// des Silverlight-HTML-Objektes als OBJECT
var X1_Daten_OBJECT_ElementBezeichner=new Array // wird von silverlight.js gelesen
// Datenlemente werden in X_Daten_OBJECT hinterlegt
// Reihenfolge egal
(X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_id_Index], // 'id'
X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_width_Index], // 'width'
X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_height_Index] // 'height'
);

// ----- Datenelemente aus X1_Silverlight_DatenElementeBezeichner für Erzeugung der PARAM im OBJECT
// des Silverlight-HTML-Objektes
var X1_Daten_PARAM_ElementBezeichner=new Array // wird von silverlight.js gelesen
// Datenelemente werden in X_Daten_PARAM hinterlegt
// Reihenfolge egal
```



```
(X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_soucre_Index], // 'source',
X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_background_Index], // 'background'
X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_windowless_Index], // 'windowless',
X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_maxFramerate_Index] // 'maxFramerate',
)
```

```
// +++++ Kennungen aus der XAML-Datei hier ebenfalls ablegen +++++
// Bild- und Sound sind paarweise verfügbar und haben im Paar IDENTISCHEN Präfix, der
// als PianoTastenID verwendet wird
// als Suffix in x:Name von Elementen des Canvas laut X1_XAML_pathCanvas in der XAML-Datei verwendet wird
// als Präfix in x:Name von Image und MediaElement in der XAML-Datei verwendet wird
// Für alle Elemente eines Canvas sollte der Wert von x:Name einmalig sein (nicht mehrfach auftreten).
```

```
var X1_XAML_pathCancas_xName_Suffix='Path';// laut XAML-Datei für element im Canvas laut X1_XAML_pathCanvas
var X1_XAML_pathCanvas='pathCanvas'; // laut XAML-Datei
var X1_XAML_IMG_xName_Praefix='img'; // laut XAML-Datei für Bild-Datei in x:Name
var X1_XAML_Sound_xName_Praefix='snd'; // laut XAML-Datei für Sound-Datei in x:Name
```

```
// +++++ Zuordnung Key-Code (Tasten-Code und NICHT Zeichencode) zu den Bild- und Sounddateien,
// die bei passendem Tastedruck aktiviert werden sollen.
// Dateinamen haben ZWINGEND (wegen XAML-Datei und Angaben dort) folgenden Aufbau
// Präfix Ton wie C2 oder B
// Suffix je nach Medienart z.B. JPG
// Ton wird musiktechnisch mit Buchstaben bezeichnet z.B. C oder C2 für Cis bzw. C#
// Der Buchstabe wird als PianoTastenID bezeichnet.
// Key-Code werden den Tonbuchstaben zugeordnet
```

```
var X1_DateiPraefix_BildUndSound=new Array(); // Zuordnung Key-Code zu PianotastenID
// Index ist Key-Code und nicht ASCII-Code
// Key-Code 55 ist ASCII '7'
```

```
// ----- allen Keycodes von 0 bis 255 den Standard-Buchstaben 'C' verpassen (PianoTastenID 'C')
for(var X1_Zahler=0;X1_Zahler<256;X1_Zahler++) // den Key-Code 0 bis 256
{X1_DateiPraefix_BildUndSound[X1_Zahler]='C';} // einen Standard 'C' zuordnen (PianoTastenID 'C')
```

```
// ----- Keycodes mit anderem ID belegen (einfach ausprobieren, je nach Geschmack)
X1_DateiPraefix_BildUndSound[55]='C'; // linkeste klickbare weisse Taste
X1_DateiPraefix_BildUndSound[48]='C2'; // C# oder D-es, linkeste klickbare schwarze Taste
X1_DateiPraefix_BildUndSound[53]='D'; // weisse Taste
X1_DateiPraefix_BildUndSound[33]='D2'; // D# oder E-es, schwarze Taste
X1_DateiPraefix_BildUndSound[32]='E'; // weisse Taste
X1_DateiPraefix_BildUndSound[51]='F'; // weisse Taste
X1_DateiPraefix_BildUndSound[36]='F2'; // F# oder G-es, schwarze Taste
X1_DateiPraefix_BildUndSound[31]='G'; // weisse Taste
X1_DateiPraefix_BildUndSound[37]='G2'; // G# oder A-es, schwarze Taste
X1_DateiPraefix_BildUndSound[43]='A'; // weisse Taste
X1_DateiPraefix_BildUndSound[39]='A2'; // Bb oder B-es, rechteste klickbare schwarze Taste
X1_DateiPraefix_BildUndSound[42]='B'; // rechteste klickbare weisse Taste
```

```
// +++++
//
// nicht zu verändernde interne Daten
//
// +++++
```

```
// +++++ Datenelemente laut X1_Silverlight_DatenElementeBezeichner: Werte zu weisen
var X1_SilverlightDaten=new Array(); // wird von silverlight.js gelesen
X1_Wert=X1_Silverlight_DatenElementeWerte.length;
var X_Zahler=0;
for(X_Zahler=0;X_Zahler<X1_Wert;X_Zahler++)
{X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X_Zahler]]=X1_Silverlight_DatenElementeWerte[X_Zahler];}
// Pflichtelemente werden automatisch initialisiert
```

```
// +++++ Pflicht-Elemente erzeugen und init
// sind mit Bezeichner fest kodiert in silverlight.js
// HTML-bezogene Daten wie onXXX-Events von HTML-Objekt
```





```

//
//                                     ID eines HTML-Objektes
//                                     nicht Events oder andere Daten des Silverlight-Objektes
//                                     (Silverlight-Events werden in
//                                     X1_GrandPiano.EventVerwaltungInstanzieren() verwaltet

// ----- Pflicht-Elemente anhängen an X1_Silverlight_DatenElementeBezeichner und Indexe merken
//                                     Indexe nicht als Feld sondern in Einzelvariablen speichern, deren Bezeichner besser erklärend
// sind.
// Reihenfolge egal
X1_Silverlight_DatenElementeBezeichner[X1_Wert]='parentElement';
//                                     // Kennung für Container (z.B. DIV-Container) des HTML-Objektes (OBJECT) von Silverlight
var X1_Silverlight_DatenElementeBezeichner_parentElement_Index=X1_Wert;           // wird von silverlight.js gelesen

X1_Wert++;X1_Silverlight_DatenElementeBezeichner[X1_Wert]='inplaceInstallPrompt'; // intern, boolean
var X1_Silverlight_DatenElementeBezeichner_inplaceInstallPrompt_Index=X1_Wert;

X1_Wert++;X1_Silverlight_DatenElementeBezeichner[X1_Wert]='context';           // intern Daten-Context, String
var X1_Silverlight_DatenElementeBezeichner_context=X1_Wert;

X1_Wert++;X1_Silverlight_DatenElementeBezeichner[X1_Wert]='events';           // Kennung für Eventstruktur von
//                                     // HTML-Events per PARAM
var X1_Silverlight_DatenElementeBezeichner_events_Index=X1_Wert;           // wird von silverlight.js gelesen

X1_Wert++;X1_Silverlight_DatenElementeBezeichner[X1_Wert]='onError';           // events.onError als HTML-Event
//                                     // per PARAM-Attribut
var X1_Silverlight_DatenElementeBezeichner_onError_Index=X1_Wert;           // wird von silverlight.js gelesen

X1_Wert++;X1_Silverlight_DatenElementeBezeichner[X1_Wert]='onLoad';           // events.onLoad als HTML-Event
//                                     // per PARAM-Attribut
var X1_Silverlight_DatenElementeBezeichner_onLoad_Index=X1_Wert;           // wird von silverlight.js gelesen

// +++++ Pflicht-Elemente init
X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_events_Index]]=new Array();

// Eventstruktur
// events.onError wird in Y_SilverlightVerwaltung_Aktivieren_InitUndStart() aus silverlight.js gefüllt
// events.onLoad wird in Y_SilverlightVerwaltung_Aktivieren_InitUndStart() aus silverlight.js gefüllt
X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_inplaceInstallPrompt_Index]]
=false;           // 'inplaceInstallPrompt'
// boolean
X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_context]]=";
// 'context'
// String

// ++++++
//
// anzupassende Routinen
//
// ++++++

// Bild- und Sound sind paarweise verfügbar und haben im Paar IDENTISCHEN Präfix, der
// als PianoTastenID verwendet wird
// als Suffix in x:Name von Elementen des Canvas laut X1_XAML_pathCanvas in der XAML-Datei verwendet wird
// als Präfix in x:Name von Image und MediaElement in der XAML-Datei verwendet wird
// Für alle Elemente eines Canvas sollte der Wert von x:Name einmalig sein (nicht mehrfach auftreten).

// +++++ Eventverwaltung-Instanz des Grand-Piano leer erzeugen
if(!window.X1_GrandPiano){window.X1_GrandPiano={};}

// +++++ Konstruktor-Instanz der Eventverwaltung des Grand-Piano leer erzeugen
X1_GrandPiano.EventVerwaltungInstanzieren=function({});

// +++++ Konstruktor-Instanz der Eventverwaltung des Grand-Piano deklarieren
// per Konstruktor erzeugte Eventverwaltung benötigt instanziiertes Silverlight-Objekt
// (siehe dazu createSilverlight() aus silverlight.js)

```



```

X1_GrandPiano.EventVerwaltungInstanzieren.prototype =
{
// #####
//
// Anpassungen der Eventhandler, anhand derer per Silverlight animiert wird
// Die Events sind KEINE HTML-Events sondern Events von Silverlight.
// Events können an die Events laut PARAM des Silverlight-HTML-Objektes als OBJECT
// nur soweit übergeben werden, wie Silverlight die PARAM verwaltet.
//
// Handler sind selbst Teil der Instanz Eventverwaltung und speichern ihre Daten dort auch
// (gekapselt, Nutzung des this-Zeigers auf Instanz der Eventverwaltung)
//
// Es ist zu beachten, dass die Eventverwaltung browserübergreifend ist, also keine
// typische Internet Explorer-Eventverwaltung.
//
// Die Bildanimation erfolgt per Eigenschaft .opacity eines Image-Elementes in der XAML-Datei
// Es wird durch Silverlight im Prinzip die folgende reine JavaScript-Variante gerendert:
// 2 IMG überlagern sich
// unterste Img enthält ungedrückte Taste und ist permanent sichtbar
// obere Img enthält gedrückte Taste und wird ein- oder ausgeblendet per visibility
// Empfehlung: Jedes Img in 1 DIV-Container und diese animieren,
// da IMG-Objekt nicht alle Eigenschaften wie DIV hat.
//
// oder 1 Img wird im .scr wechselseitig neu belegt
// Img1=new Image(...); Img1.scr='....';
// Img2=new Image(...); Img1.scr='....';
// Img_ID.scr=Img1.scr; bzw. img2.scr;
//
// Man beachte: Jede gedrückte Taste erscheint nur als einzeln gedrückt. Die Bilder sind immer
// über alle Tasten, so dass auch Schatteneffekte im Bild vorhanden sind, die
// NICHT durch Silverlight erzeugt werden: Silverlight animiert allerdings die Bilder
// per opacity.
// #####

// +++++ Überwachung der Events (lauschen / lauern auf Eventeintritt) einrichten und starten
Handler_SilverlightEvent_onLoad: function(control, userContext, rootElement)
{this.control=control;
this.rootElement = rootElement;
this.rootElement.addEventListener("GotFocus", Silverlight.createDelegate(this, this.Handler_SilverlightEvent_onGotFocus));
this.rootElement.addEventListener("KeyDown", Silverlight.createDelegate(this, this.Handler_SilverlightEvent_onKeyDown));
this.rootElement.addEventListener("KeyUp", Silverlight.createDelegate(this, this.Handler_SilverlightEvent_onKeyUp));

// Piano button event hookup: Find the button and then attach event handlers
this.pathCanvas = rootElement.findName(X1_XAML_pathCanvas); // "pathCanvas" laut XAML-Datei
for(var i=0; i<this.pathCanvas.Children.Count; ++i) // für alle Path-Elementen in der XAML-Datei
// die Maus-Eventhandler einbinden
{this.pathCanvas.Children.GetItem(i).addEventListener // Listener = lauschen (lauern) auf Eventeintritt
("MouseLeftButtonDown", // linke Maustaste
Silverlight.createDelegate(this, this.Handler_SilverlightEvent_onMouseDown)
);
this.pathCanvas.Children.GetItem(i).addEventListener // Listener = lauschen (lauern) auf Eventeintritt
("MouseLeftButtonUp", // linke Maustaste
Silverlight.createDelegate(this, this.Handler_SilverlightEvent_onMouseUp)
);
}
},

// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkannt MouseDown
Handler_SilverlightEvent_onMouseDown: function(sender, eventArgs)
{// +++++ aktuelles Mausevent des Senders holen
// sender ist ein Path-Element im Canvas mit x:Name laut X1_XAML_pathCanvas in der XAML-Datei
sender.captureMouse();
// +++++ PianoTastenID anhand x:Name des senders ermitteln
// Path-Elemente der XAML-Datei haben in x:Name
// am Ende die Konstante laut Konstante laut X1_XAML_pathCanvas_xName_Suffix
// und davor das PianoTastenID (welches auch identisch ist mit dem Präfix der Dateinamen der

```





```

//                                     zugehörigen Bild- und Sounddatei)
//
//      x:Name des Senders besteht aus      xxYYY
//
//                                     mit xx als PianoTastenID
//
//      YYY als Konstante laut X1_XAML_pathCancas_xName_Suffix
//
//      Es muss also xx als PianoTastenID isoliert werden
// ----- Zeiger auf x:Name-Attribut
var Sender_NameAttribut=sender.name;
// ----- im Wert des Attributes die Position von X1_XAML_pathCancas_xName_Suffix ermitteln
var EndePosition=Sender_NameAttribut.indexOf(X1_XAML_pathCancas_xName_Suffix);
// ----- PianoTastenID herauskopieren
var PianoTastenID = Sender_NameAttribut.substr(0, EndePosition);
// +++++ BildID des senders ermitteln
//      Image-Elemente der XAML-Datei haben im x:Name
//
//      zu Beginn eine Konstante laut X1_XAML_IMG_xName_Praefix
//
//      und danach das PianoTastenID (welches auch identisch ist mit dem Präfix der Dateinamen der
//
//      zugehörigen Bild- und Sounddatei)
//
//      x:Name des Image besteht aus      XXXyy
//
//      mit XXX als Konstante laut X1_XAML_IMG_xName_Praefix
//
//      yy als PianoTastenID
//
//      Es muss also yy als PianoTastenID isoliert werden
var BildID = sender.findName(X1_XAML_IMG_xName_Praefix + PianoTastenID);
// +++++ Bild-Opacity (Deckkraft) einstellen
BildID.opacity = 1;      // Bild einblenden also gedrückte Klaviertaste anzeigen als Silverlight-Animation
// +++++ SoundID des senders ermitteln
//      MediaElement-Elemente der XAML-Datei haben im x:Name
//
//      zu Beginn eine Konstante laut X1_XAML_Sound_xName_Praefix
//
//      und danach das PianoTastenID (welches auch identisch ist mit dem Präfix der Dateinamen der
//
//      zugehörigen Bild- und Sounddatei)
//
//      x:Name des Image besteht aus      XXXyy
//
//      mit XXX als Konstante laut X1_XAML_Sound_xName_Praefix
//
//      yy als PianoTastenID
//
//      Es muss also yy als PianoTastenID isoliert werden
var SoundID = sender.findName(X1_XAML_Sound_xName_Praefix + PianoTastenID);
// +++++ falls aktiv so Soundausgabe stoppen
SoundID.stop();
// +++++ und neu starten
SoundID.play();
},

// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkannttes MouseUp
Handler_SilverlightEvent_onMouseUp: function(sender, eventArgs)
{sender.releaseMouseCapture();
var PianoTastenID = sender.name.substr(0, sender.name.indexOf("Path"));
var BildID = sender.findName(X1_XAML_IMG_xName_Praefix + PianoTastenID);
BildID.opacity = 0;      // Bild ausblenden also ungedrückte Klaviertaste anzeigen als Silverlight-Animation
},

// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkannttes KeyPress
Handler_SilverlightEvent_onKeyPress: function (PianoTastenID)
{var BildID = this.control.content.findName(X1_XAML_IMG_xName_Praefix + PianoTastenID);
BildID.opacity = 1;      // Bild einblenden also gedrückte Klaviertaste anzeigen als Silverlight-Animation
var SoundID = this.control.content.findName(X1_XAML_Sound_xName_Praefix + PianoTastenID);
SoundID.stop();
SoundID.play();
},

// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkannttes KeyDePress
Handler_SilverlightEvent_onKeyDePress: function (PianoTastenID)
{var BildID = this.control.content.findName(X1_XAML_IMG_xName_Praefix + PianoTastenID);
BildID.opacity = 0;      // Bild ausblenden also ungedrückte Klaviertaste anzeigen als Silverlight-Animation
},

// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkannttes KeyDown
Handler_SilverlightEvent_onKeyDown: function (sender, eventArgs)
{this.Handler_SilverlightEvent_onKeyPress(X1_DateiPraefix_BildUndSound[eventArgs.Key]);},

```



```
// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkanntes KeyUp
Handler_SilverlightEvent_onKeyUp: function (sender, eventArgs)
{this.Handler_SilverlightEvent_onKeyDePress(X1_DateiPraefix_BildUndSound[eventArgs.Key]);},

// +++++ Reaktion auf ein vom Listener auf ein Path-Element aus XAML-Datei erkanntes gotFocus (entspricht onfocus im HTML)
Handler_SilverlightEvent_onGotFocus: function(sender, eventArgs)
{// In order to avoid problems when the control loses focus while a key is held down,
// we use this function to reset the piano to its initial state (no key pressed). It is
// called on the root canvas' GotFocus event.
var X_Zahler=0;
for(X_Zahler=0;X_Zahler<256;X_Zahler++)
{this.Handler_SilverlightEvent_onKeyDePress(X1_DateiPraefix_BildUndSound[X_Zahler]);}
}
}
```



**silverlight.js**

```
// #####
//
//      silverlight.js
//
//      benötigt bereits geladene GrandPiano.js
//
//      In dieser JS-Datei sind
//          nur Daten und Routinen kodiert, die
//              intern für Silverlight verwendet werden
//              folgende Daten aus GrandPiano.js lesend benutzen
//
//          X1_SilverlightControl_Version
//          X1_SilverlightControlHost_ID
//          X1_Daten_OBJECT_ElementBezeichner
//          X1_Daten_PARAM_ElementBezeichner
//          X1_Silverlight_DatenElementBezeichner
//          X1_Silverlight_DatenElementBezeichner_parentElement_Index
//          X1_Silverlight_DatenElementBezeichner_events_Index
//          X1_Silverlight_DatenElementBezeichner_onError_Index
//          X1_Silverlight_DatenElementBezeichner_onLoad_Index
//          X1_SilverlightDaten
//          X1_GrandPiano
//
//      KEINE Daten und Routinen des X1_GrandPiano kodiert (siehe dafür NUR GrandPiano.js)
//
//      aufbereitete JS-Datei anhand der Programcode der Microsoft-programmierten JS-Dateien
//          silverlight.js
//          createsilverlight.js
//
//      Aufbereitung
//          erfolgt damit      Quelltext lesbar wird
//                              Programmlogik erkennbar wird
//
//          mit Entfernung aller Logik- und Programmierungsfehler, die von Microsoft eingebaut
//          wurden
//
//      Sollte Der Internet Explorer mit Start der index.html regelmäßig einen Adressfehler bringen,
//      der auch die Adresse 0x00000000 umfassen kann, ist das Silverlight-ActiveX-Control
//      defekt und es muss neu installiert werden: Dazu lade man sich die aktuellen Updates
//          zum IE per http://msdn2.microsoft.com/en-us/library/bb188743.aspx
//          zu Silverlight per http://silverlight.net/Default.aspx
//          herunter und installiere diese. Man beachte Nicht-Kompatibilität von
//          Silverlight-Varianten (inkl. deren Beta-Versionen).
//
//      WARNUNG: Microsoft zeigt mit seiner JS-Datei, wie man NICHT programmieren sollte:
//          Es werden Daten und Programmcode munter gemixt, so dass der
//          Quellcode nur als wertlose Einzelfall-Lösung dient.
//          Silverlight wird datenseitig konsequent NICHT dynamisch programmiert:
//          Daten liegen teilweise als Programmcode-Teile vor und zwar derart
//          festkodiert, dass deren Kapselung eine völlig andere Programmlogik
//          erzwingt, die aber Voraussetzung für dynamische Webseiten mit
//          geringem Pflegeaufwand ist.
//
//          Microsoft zeigt mit der Einzelfall-Lösung, dass die Programmierung mit
//          Silverlight nicht jedermanns Sache bei Microsoft ist.
//
//          Es werden Konzeptfehler an Silverlight sichtbar: Daten und Programmcode
//          der Anwendung sind von denen, die das Silverlight-Laufzeitsystem benötigt,
//          nur schwer trennbar, so dass deren Verquickung IMMER eine Einzelfalllösung
//          darstellt, wenn nicht genügend Programmierungsaufwand betrieben wird.
//          Mit anderen Worten: Es muss zwingend geprüft werden, ob Trennung möglich
//          ist: Wenn nein, sollte das Projekt mit anderer Laufzeitumgebung als
//          Silverlight geschrieben werden: Einzellösungen sind meist supportaufwendig
//          und die Kinderkrankheiten des programmierten Silverlight-Konzeptes
//          sind genau genommen einfach nur Müll - und das als Aushängeschild per
//          Grand-Piano mit Microsoft-Signatur.
//
//          Faulheiten des Programmierers führt zu uneinheitlichem Quelltext als Einzellösung:
//          Die Verwendung von Variablennamen ist Elend pur und gleicht dem von Anfängern.
//          Wegen Faulheit werden beim Codieren werden Anweisungen verwendet, deren
//          Komplexität mehr Laufzeit kostet, als bei Verwendung einfacher
```



```
//
//                                Anweisungen.
//                                Vorgaben von Microsoft zur Quelltext-Optimierung im Sinne der
//                                Script-Laufzeit-Optimierung werden konsequent ignoriert.
//                                Die Aufarbeitung des Quellcode zeigt Ansätze, wo dynamische Programmierung
//                                notwendig und sinnvoll ist, ersetzt aber die vom Microsoft-Programmierer
//                                verwendete Logik nicht komplett (aus Mist lässt sich kein Gold machen),
//                                da ansonsten die Anwendung komplett neu programmiert werden müsste.
//                                Die Verwendung von HTTPRequest zur Daten- und Script-Programmcodes-Beschaffung
//                                zur Laufzeit ist nicht eingebaut worden, empfiehlt sich aber.
//                                Die Routinen sollten möglichst als eigenständiges System unabhängig von Zeigern von Silverlight
//                                programmiert werden und über Silverlight-Variablen bzw. -Zeiger referenziert werden
//                                (z.B. Zeiger auf Funktion kopieren nach Silverlight-Funktionsvariable).
//                                Silverlight wird von Microsoft permanent verändert und
//                                Schnittstellenkonsistenz ist Sache von Microsoft: Werden Schnittstellen verändert,
//                                so sind wenigstens die Routinen als System unberührt und der Fehler ist schneller
//                                findbar.
//                                Microsoft will anhand Silverlight seine (sowas teuren) Developer-Produkte wie Visual
//                                und Co. verkaufen (Testversionen sind zeitlich limitiert) und zugleich ASP
//                                in die Adobe-Flash-Konkurrenz drücken - allerdings ist das mehr ein Eigentor.
//                                Alternativen zur Silverlight-Technik gabs schon seit 1998 (also rechtzeitig VOR Adobe Flash
//                                und als Betriebssystemkomponenten, auf die u.a. auch Office-Powerpoint aufbaute (bis hin zur
//                                Timeline-Programmierung per HTML, JavaScript oder VBasic) in DirectX mit Direct3D und
//                                DirectMedia (inklusive HTML- und Javascript-Schnittstelle + kostenlosem SDK für Web-Desing), die
//                                aber inzwischen von Microsoft ersatzlos und gnadenlos entfernt wurden (u.a. mit der Begründung,
//                                dass neueres Windows neuere und sichere Schnittstellen benötigt). Die Abschaltung ist allerdings
//                                ganz klar ein Analogon zum Wechsel des Treibersystems zwischen Win XP SP1 und 2 sowie erneut
//                                von Win XP zu Win Vista (verbunden mit Folgekosten (abgesehen von Hardwarekosten): User, die
//                                weiterhin Schnittstellen für Webseiten mit Animation auf DirectX-Basis nutzen wollen, werden
//                                gnadenlos vom Markt gefegt: Sie müssen umstellen (und so auch Developer-Produkte kaufen) oder
//                                zur Konkurrenz wechseln, denn ein höherer Programmierungsaufwand lohnt nur bei dauerhafteren
//                                Schnittstellensystemen). DirectX wurde gezielt nicht mehr abwärtskompatibel gehalten, da zusätzlich
//                                Net-Framework durchgedrückt werden soll (Pendant zu Sun Java): Silverlight + DirectX mit Net-
//                                Framework sollen zusammenarbeiten - möglichst mit ASP und das mit Browsern wie Safari oder
//                                Firefox. (Net-Framework ist in Versionen parallel installierbar, Silverlight jedoch nicht).
//
// Sämtliche Entwicklungstools der Express-Serie (Blend, Media etc.) gibt es
// als Trial für einen begrenzten Laufzeitraum und sind ansonsten zu kaufen.
//
// Microsoft Expression Blend Free Trial
//
//                                We are pleased to present a trial version of Microsoft® Expression Blend™.
//                                This fully functioning version will expire 60 days after installation.
//                                This trial can be unlocked when you receive your purchased copy of the
//                                product by entering the product key found on the box.
//
//                                Die Vollversion kostet 600 US-Dollar
//
// Microsoft Expression Studio (Free evaluation is available for download)
//
// ist als Trial in Komponenten zerlegt, die verschiedene Laufzeiten haben
// z.B.
//
//                                Microsoft Expression Blend Free Trial
//
//                                We are pleased to present a trial version of Microsoft® Expression Blend™.
//                                This fully functioning version will expire 60 days after installation.
//                                This trial can be unlocked when you receive your purchased copy of the
//                                product by entering the product key found on the box.
//
// Whether creating standards-based Web user interfaces by using Microsoft ASP.NET,
// building cross-platform rich interactive applications with Silverlight,
// or creating ultimate Windows Vista experiences with Windows Presentation
// Foundation (WPF), Expression tools enable designers to fully apply their
// creativity while collaborating with developers using Microsoft Visual Studio.
```



```
//          Expression Studio includes the following products:
//          Expression Web for building standards-based Web sites.
//          Expression Design for professional illustration and graphic design.
//          Expression Blend for creating engaging Web-connected experiences for Microsoft Windows.
//          This special preview version is compatible with creating content for Silverlight 1.0
//          Release Candidate and Silverlight 1.1 Alpha.
//          Expression Media for assets management as well as media encoding.
//          This version also includes the Expression Encoder tool for transcoding video content.
//
//
// Windows Movie Maker
//
//          Import audio and video to your computer from a digital video camera, and then use the
//          imported content in your movies. You can also import existing audio, video, or still
//          pictures into Windows Movie Maker to use in the movies you create. After editing the audio
//          and video content in Windows Movie Maker, which can include adding titles, transitions,
//          or effects, you can then publish your final movie and share it with your friends and family.
//          Windows Movie Maker is also available for Microsoft Windows XP.
//
//
// #####
```

```
// ++++++
//          anzupassende GLOBALE Daten
// ++++++
```

```
// Es gibt noch Daten, die direkt in Routinen kodiert sind und dort im Kontext der Routine anzupassen sind
//          siehe Y_SilverlightOnlineInstallControl_HTMLCode_ermitteln()
//          Y_Silverlight_EventHandler_Standard()
```

```
var X_SilverlightDownloadUrl='http://go.microsoft.com/fwlink/?LinkID=';
```

```
var X_SilverlightMimeTypeAttribut=' TYPE="application/x-silverlight";
var X_Silverlight_Kennung_ActiveX='AgControl.AgControl';
var X_Silverlight_Kennung_PlugIn='Silverlight Plug-In';    // laut navigator.plugins
// nicht relevant für MS Internet Explorer, da der
//          ActiveX nutzt und kein Plugin
```

```
var X_SilverlightControl_IFrameTag='<IFRAME STYLE="visibility:hidden;height:0;width:0"></IFRAME>';
```

```
var X_SilverlightControl_EMBED_Tag1='<EMBED>';
var X_SilverlightControl_EMBED_Tag2='</EMBED>';
var X_SilverlightControl_EMBED_Tag3= X_SilverlightControl_EMBED_Tag1
                                     + X_SilverlightMimeTypeAttribut
                                     + X_SilverlightControl_EMBED_Tag2;
```

```
var X_SilverlightControl_OBJECT_Tag1='<OBJECT DATA="data:,"';
var X_SilverlightControl_OBJECT_Tag2='</OBJECT>';
var X_SilverlightControl_OBJECT_Tag3= X_SilverlightControl_OBJECT_Tag1
                                     + X_SilverlightMimeTypeAttribut
                                     + '>'
                                     + X_SilverlightControl_OBJECT_Tag2;
```

```
var X_SilverlightControl_PARAM_Tag1='<PARAM NAME="';
var X_SilverlightControl_PARAM_Tag2=' VALUE="';
var X_SilverlightControl_PARAM_Tag3=' "></PARAM>';
```

```
// +++++ Daten zu Betriebssystemen, die von Silverlight unterstützt werden ++++++
//          alle Datenfelder sind indexsynchron und dürfen nicht leer sein !
```

```
var X_KennzeichenFeld_SupportedBetriebssystem = new Array
```



```

// Kennung laut navigator.userAgent
// z.B. Microsoft® Internet Explorer 6 unter Windows XP liefert
//      "Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)"
// z.B. Microsoft® Internet Explorer 7 unter Windows XP SP2 liefert
//      "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .....)" wobei ..... hier weglassung bedeutet
//      "Windows NT", // Kennung bei Win32 und Win64
//      "PPC Mac OS X",
//      "Intel Mac OS X"
);

var X_KennzeichenFeld_EingeschraenktSupportedBetriebssystem = new Array
// Kennung laut navigator.userAgent
"Win64", // kein Win64
"", // es gibt keine Einschränkung
"", // es gibt keine Einschränkung
);

var X_TitelFeld_SupportedBetriebssystem = new Array
("Windows",
"MacPPC",
"MacIntel"
);

// +++++ Daten zu Browsern, die von Silverlight unterstützt werden +++++
//      alle Datenfelder sind indexsynchron und dürfen nicht leer sein !

var X_Index_MSIE_SupportedBrowser=0; // Index von Internet Explorer in nachfolgenden
Feldern
var X_Index_Firefox_SupportedBrowser=1; // Index von Firefox in nachfolgenden Feldern
var X_Index_Safari_SupportedBrowser=2; // Index von Safari in nachfolgenden Feldern
var X_Index_Opera_SupportedBrowser=3; // Index von Opera in nachfolgenden Feldern

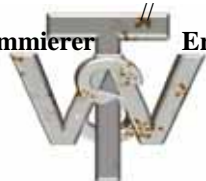
var X_Titel_SupportedBrowser = new Array
// Kennung laut navigator.userAgent aber ohne Sonderzeichen "/" (falls vorhanden)
"MSIE",
"Firefox",
"Safari",
"Opera"
);

var X_Kennung_SupportedBrowser = new Array
// Kennung laut navigator.userAgent mit Sonderzeichen "/" (falls vorhanden)
X_Titel_SupportedBrowser[X_Index_MSIE_SupportedBrowser],
X_Titel_SupportedBrowser[X_Index_Firefox_SupportedBrowser] + '/', // z.B. "Firefox/2.5"
X_Titel_SupportedBrowser[X_Index_Safari_SupportedBrowser],
X_Titel_SupportedBrowser[X_Index_Opera_SupportedBrowser] + '/' // z.B. "Opera/2.5"
);

var X_Hauptversion_SupportedBrowser = new Array // 0 bedeutet immer supported
// Kennung laut navigator.userAgent
6,
2, // Achtung: ab Firefox 1.5 ist zwar möglich,
//      aber geprüft erst ab 2.x
0,
99 // diese Versionsnummer gibt es niemals
//      damit ist der Browser nicht supported
);

var X_Unterversion_SupportedBrowser = new Array // 0 bedeutet immer supported
// Kennung laut navigator.userAgent
0,
0, // Achtung: ab Firefox 1.5 ist zwar möglich,
//      aber geprüft erst ab 2.x

```



```

0,
0
);

var X_SilverlightTestControl_Art_SupportedBrowser = new Array
(1, // ActiveXControl
0, // kein ActiveXControl
0, // kein ActiveXControl
0 // kein ActiveXControl
);

var X_SilverlightTestControl_HTMLCode_SupportedBrowser = new Array
(X_Silverlight_Kennung_ActiveX,
X_SilverlightControl_EMBED_Tag3,
X_SilverlightControl_OBJECT_Tag3,
X_SilverlightControl_EMBED_Tag3
);

var X_SilverlightTestControl_PluginActivexKennung_SupportedBrowser = new Array
(X_Silverlight_Kennung_ActiveX, // ActiveX und kein Plugin
X_Silverlight_Kennung_Plugin, // laut navigator.plugins
X_Silverlight_Kennung_Plugin, // laut navigator.plugins
X_Silverlight_Kennung_Plugin // laut navigator.plugins
);

// +++++ Kombinationen von Betriebssystem und Browser, die nicht erlaubt sind +++++
// nachfolgende Felder sind index-synchron
// können leer sein

var X_Index_Windows_SupportedBetriebssystem=0; // Index von Windows in nachfolgenden Feldern
var X_Index_MacPPC_SupportedBetriebssystem=1; // Index von MacPPC in nachfolgenden Feldern
var X_Index_MacIntel_SupportedBetriebssystem=2; // Index von MacIntel in nachfolgenden Feldern

var X_UnerlaubteKombination_TitelFeld_SupportedBetriebssystem=new Array
(X_Index_Windows_SupportedBetriebssystem, // Windows
X_Index_MacPPC_SupportedBetriebssystem, // MacPPC
X_Index_MacIntel_SupportedBetriebssystem // MacIntel
);

var X_UnerlaubteKombination_TitelFeld_SupportedBrowser=new Array
(X_Index_Safari_SupportedBrowser, // Safari
X_Index_MSIE_SupportedBrowser, // Internet Explorer
X_Index_MSIE_SupportedBrowser // Internet Explorer
);

// +++++ sonstiges +++++
var X_Kennzeichen_NichtSupported="Unsupported";
var X_VersionsNummerTrenner='.';
// Achtung: window.navigator.userAgent muss Zeichenkette enthalten in der Form "wxyz"
// wobei w laut X_Kennung_SupportedBrowser z.B. "Firefox/"
// x für Hauptversion als Ziffer z.B. "2" oder "20"
// y laut X_VersionsNummerTrenner z.B. "."
// z für Unterversion als Ziffer z.B. "5" oder "25"
// Die Syntax wird NUR dann geprüft, wenn erkannter Browser
// mit Kennung laut navigator.userAgent in X_Kennung_SupportedBrowser hinterlegt
ist
// und Hauptversionsnummer > 0 in X_Hauptversion_SupportedBrowser.
// Ein Browser, dessen Kennung laut navigator.userAgent nicht enthalten ist in
// in X_Kennung_SupportedBrowser
// gilt automatisch als nicht supported !

var X_WindowEventHandler_Load_Kennung="__slLoad";
var X_WindowEventHandler_Error_Kennung="__slError";
var X_FehlerKennung_ParseError="ParseError";
var X_FehlerKennung_RuntimeError="RuntimeError";

```





```

// Es gibt noch Daten, die direkt in Routinen kodiert sind und dort im Kontext der Routine anzupassen sind
//         siehe Y_SilverlightOnlineInstallControl_HTMLCode_ermitteln()
//         Y_Silverlight_EventHandler_Standard()

// ++++++ interne Daten, nicht ändern ++++++
//
// ++++++ HTML-Code zusammenbauen ++++++
//
// ++++++ HTML-Code des Silverlight-Control ermitteln ++++++
function Y_SilverlightControl_HTMLCode_ermitteln(X_Daten_OBJECT, X_Daten_PARAM)
// benötigt gefüllt X_SilverlightEinsetzbar true so Silverlight einsetzbar
//         Silverlight.ua
// Routine MUSS manuell aktualisiert werden bezüglich HTML-Code für Browser (siehe X_Flag)
{var X_Kette1="";
var X_Kette2="";
var X_Kette3="";
var X_Kette4="";
var X_Index;
var X_Flag=false;

// ++++++ prüfen ob Silverlight einsetzbar ist
if(X_SilverlightEinsetzbar)
// ++++++ ermitteln ob Safari benutzt wird
X_Flag=(Silverlight.ua.Browser == X_Titel_SupportedBrowser[X_Index_Safari_SupportedBrowser]);

// ++++++ Silverlight-Control-HTML-Code zusammenbauen
X_Kette1="";

// ----- Beginn-Tag Teil 1 OBJECT-Einleitung
X_Kette2=X_SilverlightControl_OBJECT_Tag1;
if(X_Flag){X_Kette2=X_SilverlightControl_EMBED_Tag1;} // Safari
X_Kette1+=X_Kette2;

// ----- Beginn-Tag Teil 2 Attribut TYPE
X_Kette1+=X_SilverlightMimeTypeAttribut;           // Mime-Typ TYPE

// ----- Beginn-Tag Teil 3 restliche Attribute laut Elemente 0 bis 2 in X1_Daten_OBJECT_ElementBezeichner
for(X_Index=0;X_Index<3;X_Index++)
{X_Kette2=Y_ZeichenZuUnicode(X1_Daten_OBJECT_ElementBezeichner[X_Index]);
// Bezeichner des Datenelementes holen
X_Kette3=Y_ZeichenZuUnicode(X_Daten_OBJECT[X_Kette2]); // Wert des Datenelementes holen
X_Kette1+=(" " + X_Kette2 + "=" + X_Kette3 + "");
// Hinweis: X_Daten_OBJECT.name ist identisch mit X_Daten_OBJECT[name]
//         wobei name ein String ist, der den Bezeichner des
//         Datenelementes enthält (und keine Punktnotation
//         enthält, da jeder Punkt einen Unter-Index anzeigt).
}

// ----- Beginn-Tag Teil 4 OBJECT-Ende
X_Kette2='>';
if(X_Flag){X_Kette2="";} // Safari
X_Kette1+=X_Kette2;

// ----- PARAM's im OBJECT
X_Kette2=X_SilverlightControl_PARAM_Tag1;
X_Kette3=X_SilverlightControl_PARAM_Tag2;
X_Kette4=X_SilverlightControl_PARAM_Tag3;
if(X_Flag) // Safari

```





```

{X_Kette2= '';
 X_Kette3= '""';
 X_Kette4= ""';
}
for(X_Index in X_Daten_PARAM)           // Daten PARAM abklappern
{if(X_Daten_PARAM[X_Index])
{X_Kette1+=( X_Kette2 + Y_ZeichenZuUnicode(X_Index)
  + X_Kette3 + Y_ZeichenZuUnicode(X_Daten_PARAM[X_Index])
  + X_Kette4
);                                     // Hinweis: X_Daten_PARAM[X_Index] ist identisch mit X_Daten_PARAM.X_Index
                                     // wobei X_Index ein String ist, der den Bezeichner des
                                     // Datenlementes enthält (und keine Punktnotation
                                     // enthält, da jeder Punkt einen Unter-Index anzeigt).
}
}

// ----- Ende-Tag des OBJECT
X_Kette2=X_SilverlightControl_OBJECT_Tag2;
if(X_Flag)                             // Safari
{X_Kette2= X_SilverlightControl_EMBED_Tag2
  + X_SilverlightControl_IFrameTag;
}
X_Kette1+=X_Kette2;

alert('HTML-Code des Silverlight-HTML-Objektes:\n\n' + X_Kette1);

return X_Kette1;
}
};

// +++++ HTML-Code des Silverlight-Online-Install-Controls erzeugen +++++
function Y_SilverlightOnlineInstallControl_HTMLCode_ermitteln(X_Daten)
// benötigt gefüllt X_SilverlightEinsetzbar true so Silverlight einsetzbar
// Silverlight.ua
// Routine MUSS manuell aktualisiert werden bezüglich
// Abfrage auf X_TitelFeld_SupportedBetriebssystem[] je nach Browser
// Aktualisierung der LinkID-Ziffernfolgen 9xxxxx je nach Browser
{var X_TextFeld=new Array
('Microsoft Silverlight erhalten',
'Mit Klicken',
'akzeptieren Sie da',
'Silverlight License Agreement',
'Silverlight startet automatisch, ',
'Silverlight Privacy Statement',
'Mehr erfahren'
);
var b="92822";
var e="";
var m="0x409";
var n="";
var i="";
var h="";
var g="";
var k="";
var X_DownloadUrl_Root=Silverlight.fwlinkRoot;
var X_OS=Silverlight.ua.OS;
var X_DownloadLink_HTMLCode="";

// +++++ prüfen ob Silverlight einsetzbar ist
if(X_SilverlightEinsetzbar)
{if(X_Daten.inplaceInstallPrompt)
{n="98109";
if(Silverlight.available)
{e="96189";
i="96422"
}
}
}

```



```

else
{e="96188";
i="96422";
}
h="93481";
g="93483";
if(X_OS==X_TitelFeld_SupportedBetriebssystem[X_Index_Windows_SupportedBetriebssystem])
{b="92799";
h="92803";
g="92805";
}
else
{if(X_OS==X_TitelFeld_SupportedBetriebssystem[X_Index_MacIntel_SupportedBetriebssystem])
{b="92808";
h="92804";
g="92806";
}
else
{if(X_OS==X_TitelFeld_SupportedBetriebssystem[X_Index_MacPPC_SupportedBetriebssystem])
{b="92807";
h="92815";
g="92816";
}
}
}
}

```

```

X_DownloadLink_HTMLCode =
'<TABLE BORDER="0" CELLPADDING="0" CELLSPACING="0" WIDTH="206px">
+ '<TR>
+ '<TD>
+ '<IMG STYLE="display: block; cursor: pointer; '
+ 'BORDER= 0;" '
+ 'TITLE="' + X_TextFeld[1] + '" '
+ 'ALT="' + X_TextFeld[1] + '" '
+ 'onclick="javascript:Y_DownloadLink_OnClick({0});"'
+ 'SRC="{1}" '
+ '>'
+ '</TD>'
+ '</TR>'
+ '<TR>
+ '<TD STYLE="width: 206px; '
+ 'margin: 0px; '
+ 'background: #FFFFFF; '
+ 'color: #C7C7C7; '
+ 'border-left-style: solid; '
+ 'border-right-style: solid; '
+ 'border-color: #c7c7bd; '
+ 'border-width: 2px; '
+ 'padding-left: 6px; '
+ 'padding-right: 6px; '
+ 'padding-top: 3px; '
+ 'padding-bottom: 0px; '
+ 'font-size: 55%; '
+ 'font-family: Verdana; '
+ 'text-align: left'
+ '""
+ '>'
+ X_TextFeld[2]
+ '<B>' + X_TextFeld[1] + '</B>'
+ X_TextFeld[3]
+ '<BR>'
+ '<A TITLE="' + X_TextFeld[4] + '" '
+ 'HREF="{2}" '
+ 'TARGET="_top" '
+ 'STYLE="text-decoration: underline; color: #96C5E1"'
+ '>'

```



```

+ '<B>' + X_TextFeld[4] + '</B>'
+ '</A>';
+ '</TD>'
+ '</TR>'
+ '<TR>'
+ '<TD>'
+ '<IMG SRC="{5}" '
+ 'STYLE="border: 0; display: block" '
+ '>'
+ '</TD>'
+ '</TR>'
+ '<TR>'
+ '<TD STYLE="width: 206px; '
+ 'margin: 0px; '
+ 'background: #D8EFF9; '
+ 'color: #C7C7C7; '
+ 'border-left-style: solid; '
+ 'border-right-style: solid; '
+ 'border-width: 2px; '
+ 'border-color: #c7c7bd; '
+ 'padding-left: 6px; '
+ 'padding-right: 6px; '
+ 'padding-top: 0px; '
+ 'padding-bottom: 2px; '
+ 'font-family: Verdana; '
+ 'font-size: 55%; '
+ 'text-align: left'
+ ''
+ '>'
+ X_TextFeld[5]
+ '<A TITLE="' + X_TextFeld[6] + '" '
+ 'HREF="{3}" '
+ 'TARGET="_top" '
+ 'STYLE="text-decoration: underline; color: #96C5E1"'
+ '>'
+ '<B>' + X_TextFeld[7] + '</B>'
+ '</A>';
+ '</TD>'
+ '</TR>'
+ '<TR>'
+ '<TD>'
+ '<IMG ALT="" '
+ 'SRC="{4}" '
+ '>'
+ '</TD>'
+ '</TR>'
+ '</TABLE>';

X_DownloadLink_HTMLCode = X_DownloadLink_HTMLCode.replace("{2}", X_DownloadUrl_Root + h);
X_DownloadLink_HTMLCode = X_DownloadLink_HTMLCode.replace("{3}", X_DownloadUrl_Root + g);
X_DownloadLink_HTMLCode = X_DownloadLink_HTMLCode.replace("{4}", X_DownloadUrl_Root + i);
X_DownloadLink_HTMLCode = X_DownloadLink_HTMLCode.replace("{5}", X_DownloadUrl_Root + n);
}
else
{if(Silverlight.available){e="94377";}
else{e="92801";}

if(X_OS==X_TitelFeld_SupportedBetriebssystem[X_Index_Windows_SupportedBetriebssystem]){b="92800";}
else
{if(X_OS==X_TitelFeld_SupportedBetriebssystem[X_Index_MacIntel_SupportedBetriebssystem]){b="92812";}
else
{if(X_OS==X_TitelFeld_SupportedBetriebssystem[X_Index_MacPPC_SupportedBetriebssystem]){b="92811";}
}

X_DownloadLink_HTMLCode =
'<DIV STYLE="display:block; width: 205px; height: 67px;">'

```



```

+ '<IMG onclick="javascript:Y_DownloadLink_OnClick({0});"'
+ 'STYLE="border:0; cursor:pointer" SRC="{1}" '
+ 'TITLE="' + X_TextFeld[0] + '" '
+ 'ALT="' + X_TextFeld[0] + '" '
+ '>'
+ '</DIV>';
}
X_DownloadLink_HTMLCode = X_DownloadLink_HTMLCode.replace("{0}", b);
X_DownloadLink_HTMLCode = X_DownloadLink_HTMLCode.replace("{1}", X_DownloadUrl_Root + e + "&clid=" + m);

return X_DownloadLink_HTMLCode;
}
};

// ++++++
//                               Event-Handler
// ++++++

// +++++ Download-Link onClick-Handler ++++++
function Y_DownloadLink_OnClick(a)
{top.location = Silverlight.fwlinkRoot + String(a);}

// +++++ Standard-Error-Handler ++++++
function Y_Silverlight_EventHandler_Standard(e, X_FehlerObjekt)
// e ist Platzhalter
{// +++++ nachfolgende Text-Variablen sind anpassbar
var X_Text='\nSilverlight-Fehlermeldung';
var X_FehlerKomponenten_TextFeld=new Array // Indexe sind nicht veränderbar !
(X_Text,                                // 0
'\nFehler-Code: ',                      // 1
'\nFehler-Typ: ',                      // 2
'\nMeldung: ',                        // 3
'\nim XAML-File: ',                   // 4
'\nin Zeile: ',                       // 5
'\nan Position: ',                   // 6
'\nin Routine: '                      // 7
);
var X_FehlerMeldungGesamt=X_Text + ': Fehler-Daten nicht verfügbar!';

// +++++ ab hier nichts ändern !
if(X_FehlerObjekt!=null)
{
var X_FehlerTyp;
var X_FehlerCode;
var X_FehlerTyp = X_FehlerObjekt.ErrorType;
var X_FehlerZeilenNummer=X_FehlerObjekt.lineNumber;
var X_FehlerKomponentenFeld=new Array(); // indexsynchron zu X_FehlerKomponenten_TextFeld

// +++++ Fehlerkomponenten-Feld füllen (indexsynchron zu X_FehlerKomponenten_TextFeld)
X_FehlerKomponentenFeld[0]=""; // immer Leerkette
X_FehlerKomponentenFeld[1]=X_FehlerObjekt.ErrorCode;
X_FehlerKomponentenFeld[2]=X_FehlerTyp;
X_FehlerKomponentenFeld[3]=X_FehlerObjekt.ErrorMessage;

if(X_FehlerTyp==X_FehlerKennung_ParserError)
{X_FehlerKomponentenFeld[4]=X_FehlerObjekt.xamlFile;}
else
{X_FehlerKomponentenFeld[4]="";
X_FehlerKomponenten_TextFeld[4]="";
}

if(X_FehlerZeilenNummer!=0)
{X_FehlerKomponentenFeld[5]=X_FehlerZeilenNummer;
X_FehlerKomponentenFeld[6]=X_FehlerObjekt.charPosition;

```



```

}
else
{X_FehlerKomponentenFeld[5]="";
X_FehlerKomponentenFeld[6]="";
X_FehlerKomponentenFeld[5]="";
X_FehlerKomponentenFeld[6]="";
}

if(X_FehlerTyp==X_FehlerKennung_RuntimeError)
{X_FehlerKomponentenFeld[7]=X_FehlerObjekt.methodName;}
else
{X_FehlerKomponentenFeld[7]="";
X_FehlerKomponenten_TextFeld[7]="";
}

// +++++ Fehlermeldungs-Gesamt Text erzeugen
for(X_Zahler=0;X_Zahler<8;X_Zahler++)
{X_FehlerMeldungGesamt+=X_FehlerKomponenten_TextFeld[X_Zahler];
X_FehlerMeldungGesamt+=X_FehlerKomponentenFeld[X_Zahler];
}
}

// +++++ Fehlermeldung-Gesamt anzeigen
alert(X_FehlerMeldungGesamt);
};

// +++++ Eventhandler entfernen ++++++
function Y_Window_Eventhandler_OnUnload()
{var X_Index=0;
var X_Kette="";
for(X_Index=Silverlight._silverlightCount-1;X_Index>=0;X_Index--)
{X_Kette=X_Index.toString();
window[X_WindowEventhandler_Load_Kennung + X_Kette] = null; // Zeiger des Eventhandler laut Kennung löschen
window[X_WindowEventhandler_Error_Kennung + X_Kette] = null; // Zeiger des Eventhandler laut Kennung löschen
}
// unload deattachieren beim IE (andere Browser Listener abschalten)
if(window.removeEventListener)
{window.removeEventListener("unload", Y_Window_Eventhandler_OnUnload, false);}
else{window.detachEvent("onunload", Y_Window_Eventhandler_OnUnload);}
};

// +++++ Zeichenkette nach Unicode durchsuchen und bei Unicode das Zeichen durch '&# ...;' ersetzen
function Y_ZeichenZuUnicode(X_Kette1)
{var X_Unicode=0;
var X_Kette2="";
var X_Zeichen="";
var X_Laenge=0;
var X_Index=0;
if(X_Kette1!=null)
{X_Laenge=X_Kette1.length;
if(X_Laenge>0)
{for(X_Index=0;X_Index<X_Laenge;X_Index++)
{X_Zeichen = X_Kette1.charAt(X_Index); // ASCII-Zeichen holen
X_Unicode = X_Zeichen.charCodeAt(0); // Unicode holen
if( ((X_Unicode > 43) && (X_Unicode < 58) && (X_Unicode != 47)) // ,. und 0 bis 9 jedoch nicht /
|| ((X_Unicode > 64) && (X_Unicode < 91)) // A bis Z
|| (X_Unicode == 95) // _
|| ((X_Unicode > 96) && (X_Unicode < 123)) // a bis z
)
{X_Kette2+=X_Zeichen; // unverändert
else{X_Kette2+=("&#" + X_Unicode + ";");} // konvertieren
}
}
}
return X_Kette2;
}
}

```



```
// ++++++
//                               Silverlight-Einsetzbarkeit ermitteln (Init-Status wird nicht ermittelt)
// ++++++

// +++++ Verfügbarkeit von Silverlight-ActiveX bzw. -PlugIn ermitteln ++++++
function Y_SilverlightEinsetzbarkeitErmitteln_AufActiveXbzwPluginPruefen(X_Daten_SilverlightControl_Version)
{var X_ReturnWert=false;
var X_ActiveXControlErzeugen=false;
var X_TestControl=null;
var X_DIVContainer=null;
var X_Zahler=0;
var X_LaengeFeld=0;
var X_IndexAufFeld=0;
var X_ControlArt=0;
var X_PluginKennung="";
var X_TitelBrowser="";
var X_ControlCode="";

// +++++ Feld-Index zum erkannten Browser (laut Silverlight.ua.Browser) ermitteln
X_LaengeFeld=X_TitelSupportedBrowser.length;
X_IndexAufFeld=-1;
Annahme: Index nicht gefunden
for(X_Zahler=0;X_Zahler<X_LaengeFeld;X_Zahler++)
{X_TitelBrowser=X_TitelSupportedBrowser[X_Zahler];
if(Silverlight.ua.Browser == X_TitelBrowser){X_IndexAufFeld=X_Zahler;}
// Feld der Titel der Browser abklappern
// Titel des Browsers holen
// und mit gemerkten vergleichen
// wenn identisch so Index erkannt
}

// +++++ Index gefunden ?
X_ReturnWert=(X_IndexAufFeld>=0);

if(X_ReturnWert)
{
// +++++ Silverlight-Control-Daten holen (für Test-Control)
X_ControlArt=X_SilverlightTestControl_Art_SupportedBrowser[X_IndexAufFeld];           // Control-Art
X_ActiveXControlErzeugen=(X_ControlArt==1);                                       // true so ActiveX
X_PluginKennung=X_SilverlightTestControl_PluginAktivexKennung_SupportedBrowser[X_IndexAufFeld]; // Plugin-Kennung
X_ControlCode=X_SilverlightTestControl_HTMLCode_SupportedBrowser[X_IndexAufFeld];   // Control-Code

// +++++ init Zeiger auf DIV-Container und Test-Control
X_DIVContainer=null;
X_TestControl=null;

// +++++ Silverlight-Control als Test-Control erzeugen und prüfen ob Version laut Funktionsparameter
//                               unterstützt wird
// per try-catch jede Fehlerart abfangen und Returnwert belegen
try
{
// +++++ Test-Control erzeugen

// ----- ActiveXControl erzeugen
if(X_ActiveXControlErzeugen)
{X_TestControl=new ActiveXObject(X_PluginKennung);}
else
{
// ----- nicht-Windows-Browser kennen nur PlugIn
// prüfen ob Plugin von Silverlight installiert ist

X_ReturnWert=(window.navigator.plugins.length>0);
if(X_ReturnWert){X_ReturnWert=(window.navigator.plugins[X_PluginKennung]!=null);}
if(X_ReturnWert)
{
// ----- Plugin ist installiert

// ----- DIV-Container erzeugen (anhängen an document)
X_DIVContainer=document.createElement("DIV");           // erzeugen

```



```

X_ReturnWert=(X_DIVContainer!=null);
if(X_ReturnWert)
{X_DIVContainer=document.body.appendChild(X_DIVContainer);// anhängen
X_ReturnWert=(X_DIVContainer!=null);
}
if(X_ReturnWert)
{X_DIVContainer.innerHTML = X_ControlCode;           // wird sofort gerendert
// Kind des DIV erzeugen (Test-Control im DIV)
X_TestControl = X_DIVContainer.childNodes[0];       // einzigste Kind im DIV-Container ist
// das Test-Control
// wer will kann das Test-Control auch per createElement() und div_container.appendChild() erzeugen
// und desse Attribute per Punktnotation (z.B. zeiger_auf_control_aus_appendchild.type = wert;) füllen
// Achtung: Microsoft Explorer benötigt kein getElementID etc., sondern kann mit Zeigern, die
// anhand des ID-Attributes implementiert wurden, umgehen: ID-Attributwert ist der Zeiger.
// Warnung: ID-Attribut darf in HTML-Code, der in createElement() verwendet wird,
// NICHT kodiert sein (Zeiger laut createElement() verwenden).
// Zeiger, die aus HTML-DOM-Funktionen wie appendChild() resultieren, müssen in jedem
// Browser direkt verwendbar sein (ohne Umweg über getElementID etc.) (im Internet Explorer sowieso).
}
}
}

// +++++ prüfen ob Test-Control erzeugt wurde
X_ReturnWert=(X_TestControl!=null);

if(X_ReturnWert)
{
// +++++ prüfen ob Silverlight-Version supported wird
X_ReturnWert=X_TestControl.IsVersionSupported(X_Daten_SilverlightControl_Version);

// und im Objekt auch merken
Silverlight.available=X_ReturnWert;

// +++++ Testobjekt aus Dokument entfernen (nicht bei ActiveX-Control)
if(!X_ActiveXControlErzeugen){document.body.removeChild(X_DIVContainer);}

// +++++ Zeigerinit
X_TestControl=null;
X_DIVContainer=null;
}
}
catch(e)
{// +++++ im Fehlerfall (JEDER Art)
X_ReturnWert=false;
}
} // Index gültig

return X_ReturnWert;
};

// +++++ auf unerlaubte Kombinationen von Betriebssystem und Browser prüfen +++++
function Y_SilverlightEinsetzbarkeitErmitteln_AufUnerlaubteKombinationBrowserBetriebssystemPruefen()
// benötigt gefüllt Silverlight.ua
// Returnwert: wenn true geliefert so mindestens 1 unerlaubte Kombination erkannt
// false geliefert so keine unerlaubte Kombination erkannt
// füllt nicht X_SilverlightEinsetzbar
{var X_Flag=false;
var X_Wert=0;
var X_Zahler=0;
var X_OS=Silverlight.ua.OS;
var X_Browser=Silverlight.ua.Browser;

// +++++ prüfen ob Betriebssystem nicht supported (ist ja auch unerlaubt)
X_Flag=(X_OS==X_Kennzeichen_NichtSupported);
// +++++ prüfen ob Browser nicht supported (ist ja auch unerlaubt)
X_Flag= ( X_Flag

```



```

        ||(X_Browser==X_Kennzeichen_NichtSupported)
    );
// +++++ prüfen ob unerlaubte Kombination aus Betriebssystem mit Browser anliegt
X_Wert=X_UnerlaubteKombination_TitelFeld_SupportedBetriebssystem.length;
if(X_Wert>0)
{for(X_Zahler=0;X_Zahler<X_Wert;X_Zahler++)
{X_Flag= ( X_Flag
    || ( (X_OS==X_UnerlaubteKombination_TitelFeld_SupportedBetriebssystem[X_Zahler]) // Kombination aus Betriebssystem
        && (X_Browser==X_UnerlaubteKombination_TitelFeld_SupportedBrowser[X_Zahler]) // mit Browser
    )
    );
}
}
return X_Flag;
};

// +++++ prüfen ob Silverlight von Betriebssystem UND Browser unterstützt wird +++++
// UND Silverlight-ActiveX- bzw. -Plugin verfügbar ist
// also ob Silverlight einsetzbar ist
function Y_SilverlightEinsetzbarkeitErmitteln()
// füllt Silverlight.ua
// füllt nicht X_SilverlightEinsetzbar
// liefert true so Silverlight einsetzbar (ob Silverlight-Objekt initialisiert ist, wird nicht beachtet)
{
    var X_Kette1="";
    var X_Kette2="";
    var X_KettenFeld;
    var X_ZaehlerEndWert=0;
    var X_Zaehler=0;
    var X_VersionErkannt=0;
    var X_SupportedHauptVersion=0;
    var X_SupportedUnterVersion=0;
    var X_UserAgentZeiger = window.navigator.userAgent;
    var X_Flag=false;

// +++++ Silverlight-Objekt-Elemente init auf Annahme, dass
// weder erkanntes Betriebssystem
// noch erkannter Browser
// von Silverlight unterstützt wird.
Silverlight.ua=
{OS: X_Kennzeichen_NichtSupported,
 Browser: X_Kennzeichen_NichtSupported
};

// +++++ von Silverlight unterstützte Betriebssystem erkennen und im Silverlight-Objekt merken

// ----- supported Betriebssysteme vorhanden ?
X_ZaehlerEndWert=X_KennzeichenFeld_SupportedBetriebssystem.length; // ja wenn > 0

// ----- supported Betriebssysteme abklappern
if(X_ZaehlerEndWert>0)
{for(X_Zaehler=0;X_Zaehler<X_ZaehlerEndWert;X_Zaehler++) // Feld-Index
{
    // ----- ERST auf supported prüfen
    X_Kette1=X_KennzeichenFeld_SupportedBetriebssystem[X_Zaehler]; // supported Kennung holen

    if(X_UserAgentZeiger.indexOf(X_Kette1)>=0)
    {
        // ----- DANN auf nicht eingeschränkt supported prüfen
        X_Kette1=X_KennzeichenFeld_EingeschraenktSupportedBetriebssystem[X_Zaehler];
        // eingeschränkt-supported Kennung holen

        if(X_UserAgentZeiger.indexOf(X_Kette1)===-1)
        {
            // ----- supported ohne Einschränkung
            Silverlight.ua.OS=X_TitelFeld_SupportedBetriebssystem[X_Zaehler]; // Titel merken

```





```

    }
    }
  } // for
}

// +++++ von Silverlight unterstützte Browser erkennen und im Silverlight-Objekt merken
//   Achtung: window.navigator.userAgent muss Zeichenkette enthalten in der Form "wxyz"
//           wobei w laut X_Kennung_SupportedBrowser           z.B. "Firefox/"
//           x       für Hauptversion als Ziffer               z.B. "2" oder "20"
//           y       laut X_VersionsNummerTrenner              z.B. "."
//           z       für Unterversion als Ziffer               z.B. "5" oder "25"
//           Die Syntax wird NUR dann geprüft, wenn erkannter Browser
//           mit Kennung laut navigator.userAgent in X_Kennung_SupportedBrowser hinterlegt ist
//           und Hauptversionsnummer > 0 in X_Hauptversion_SupportedBrowser.
//           Ein Browser, dessen Kennung laut navigator.userAgent nicht enthalten ist in
//                                           in X_Kennung_SupportedBrowser
//                                           gilt automatisch als nicht supported !

// ----- prüfen ob erkanntes Betriebssystem ohne Einschränkung supported wird
if(Silverlight.ua.OS!=X_Kennzeichen_NichtSupported)
{
  // ----- supported Browser vorhanden ?
  X_ZaehlerEndWert=X_Kennung_SupportedBrowser.length;           // ja wenn > 0

  if(X_ZaehlerEndWert>0)
  {
    // ----- supported Browser abklappern
    for(X_Zaehler=0;X_Zaehler<X_ZaehlerEndWert;X_Zaehler++)      // Feld-Index
    {
      // ----- supported Kennung holen
      //       an Beispiel Firefox "Firefox/"
      X_Kette1=X_Kennung_SupportedBrowser[X_Zaehler];           // supported Kennung holen

      // ----- und supported Kennung suchen im userAgent-String
      if(X_UserAgentZeiger.indexOf(X_Kette1)>=0)
      {
        // ----- Kennung als supported erkannt (Browser supportbar)
        //       Browser muss mit seiner Kennung an o.g. Syntax halten
        //       (ansonsten gilt der Browser als nicht supported)

        // ----- jetzt Titel holen
        X_Kette2=X_Titel_SupportedBrowser[X_Zaehler];

        // ----- supported Hauptversion holen
        X_SupportedHauptVersion=X_Hauptversion_SupportedBrowser[X_Zaehler];

        // ----- supported Hauptversion prüfen nur wenn > 0 (== 0 bedeutet immer supported)
        if(X_SupportedHauptVersion>0)
        {
          // ----- Hauptversionsnummer erkennen am Beispiel für Firefox
          // - - - splitten direkt hinter "Firefox/" und Teilketten nach Feld
          //           "Firefox/" wird nicht übernommen
          //           mindestens 1 Feldelement entsteht
          //           alles bis vor "Firefox/"
          //           eventuell alles ab hinter "Firefox/"
          X_KettenFeld=X_UserAgentZeiger.split(X_Kette1);

          // - - - hinter Firefox/ muss die Versionsabgabe laut o.g. Syntax liegen

          // - - - prüfen ob Teilkette der Versionsangabe als Feldelement verfügbar
          if(X_KettenFeld.length>1)                                // Länge ab 1
          {
            // - - - verfügbar, also alles splitten am Trenner laut X_VersionsNummerTrenner und wieder als Feldelemente
            //           Trenner wird nicht übernommen
            X_KettenFeld=X_KettenFeld[1].split(X_VersionsNummerTrenner); // Index ab 0

```



```

// - - - Hauptversionsnummer liegt vor dem ERSTEN X_VersionsNummerTrenner also im Feldelement mit Index 0
//                                     prüfen ob Feldelement der Hauptversionsnummer verfügbar
if(X_KettenFeld.length>1)                                     // Länge ab 1
{
    // - - - Hauptversion nach numerisch
    X_VersionErkannt = parseInt(X_KettenFeld[0]);              // Index ab 0

    // ---- prüfen ob erkannte Hauptversion mindestens die supported Hauptversion
    if(X_VersionErkannt>=X_SupportedHauptVersion)
    {
        // ---- prüfen ob identisch mit Hauptversion, denn nur dann wird Unterversion relevant
        if(X_VersionErkannt==X_SupportedHauptVersion)
        {
            // ----- supported Unterversion holen
            X_SupportedUnterVersion=X_Unterversion_SupportedBrowser[X_Zaehler];

            // ----- prüfen ob Unterversion > 0 (== 0 bedeutet immer supported)
            if(X_SupportedUnterVersion>0)
            {
                // ----- Unterversionsnummer erkennen am Beispiel für Firefox
                X_VersionErkannt = parseInt(X_KettenFeld[1]);    // Feldelemente nach dem ERSTEN
                                                                // X_VersionsNummerTrenner nach
                                                                // numerisch (Unterversion)

                // ---- prüfen ob erkannte mindestens die supported Unterversion ist
                if(X_VersionErkannt>=X_SupportedUnterVersion){Silverlight.ua.Browser=X_Kette2;}
                                                                // supported also Titel merken

            } // supported Unterversion > 0
            else{Silverlight.ua.Browser=X_Kette2;}              // supported Unterversion ist 0
                                                                //
                                                                // also wird Browser immer supported
                                                                // also Titel merken

            // erkannte Hauptversion ist identisch mit supported Hauptversion
            else{Silverlight.ua.Browser=X_Kette2;}              // erkannte Hauptversion grösser als supported
                                                                //
                                                                // also Unterversion egal
                                                                // also Titel merken

            // erkannte Hauptversion ist mindestens supported Hauptversion
            // // Feldelement der Hauptversionsnummer verfügbar
            // // Teilkette der Versionsangabe als Feldelement verfügbar
            // // supported Hauptversion > 0
            else{Silverlight.ua.Browser=X_Kette2;}              // supported Hauptversion ist 0
                                                                //
                                                                // also wird Browser immer supported
                                                                // also Titel merken

            // // supported Kennung im userAgent-String gefunden: Browser supportbar
            // // for supported Browser abklappern
            // // supported Browser verfügbar
            // // erkanntes Betriebssystem wird ohne Einschränkung supported

            // +++++ auf verbotene Kombinationen von Betriebssystem und Browser prüfen
            //                                     (inklusive Prüfung auf X_Kennzeichen_NichtSupported)
            // wenn ja, dann Silverlight nicht einsetzbar
            // wenn nein, dann Silverlight einsetzbar
            X_Flag=Y_SilverlightEinsetzbarkeitErmitteln_AufUnerlaubteKombinationBrowserBetriebssystemPruefen();
            // benötigt gefüllt Silverlight.ua
            // Returnwert: wenn true geliefert so mindestens 1 unerlaubte Kombination erkannt, also Silverlight nicht einsetzbar
            // false geliefert so keine unerlaubte Kombination erkannt, also Silverlight einsetzbar
            return !X_Flag;
        }
    }

    // ++++++
    //                                     Silverlight-Init der Daten
    // ++++++

```



```
// +++++ Silverlight-Objekt-Struktur füllen mit Daten +++++
function Y_SilverlightVerwaltung_Aktivieren_InitUndStart(X_Daten)
// X_Daten wird festgelegt in GrandPiano.js
//           bezüglich Elemente-Bezeichner und Werte
// benötigt gefüllt X_SilverlightEinsetzbar true so Silverlight einsetzbar
// aufgerufen von Y_SilverlightVerwaltung_Aktivieren()
{var X_Daten_OBJECT={};
var X_Daten_PARAM={};
var X_Index;
var X_KettenFeld;
var X_Wert=0;
var X_Zahler=0;
var X_Kette="";
var X_Zeiger;
var X_Flag=false;

// +++++ prüfen ob Silverlight einsetzbar ist
if(X_SilverlightEinsetzbar)
// +++++ prüfen ob Datenstruktur verfügbar
if(X_Daten!=null)
{
// +++++ Daten OBJECT füllen
//           Es werden auch in OBJECT benötigte Datenelemente aus X_Daten übernommen
//           wobei Bezeichner der Datenelemente sich mit Übernahme ändern könnte
//           zu OBJECT siehe Y_SilverlightControl_HTMLCode_ermitteln()
X_Wert=X1_Daten_OBJECT_ElementeBezeichner.length;
for(X_Zahler=0;X_Zahler<X_Wert;X_Zahler++)
{// ----- eval-Kette der Übernahme bilden
// zu übernehmendes Datenelement adressieren
X_Kette=X_Zeiger=X_Daten.' + X1_Daten_OBJECT_ElementeBezeichner[X_Zahler] + ';;
// und prüfen ob existent
X_Kette+=X_Flag=(X_Zeiger!=null);
// und übernehmen
X_Kette+=('if(X_Flag){X_Daten_OBJECT.' + X1_Daten_OBJECT_ElementeBezeichner[X_Zahler] + '=X_Zeiger;}');
// ----- übernehmen
eval(X_Kette);
}

// +++++ Daten PARAM füllen
//           Es werden nur in PARAM benötigte Datenelemente aus X_Daten übernommen
//           wobei Bezeichner der Datenelemente sich mit Übernahme ändern,
//           zu PARAM siehe Y_SilverlightControl_HTMLCode_ermitteln()
X_Wert=X1_Daten_PARAM_ElementeBezeichner.length;
for(X_Zahler=0;X_Zahler<X_Wert;X_Zahler++)
{// ----- eval-Kette der Übernahme bilden
// zu übernehmendes Datenelement adressieren
X_Kette=X_Zeiger=X_Daten.' + X1_Daten_PARAM_ElementeBezeichner[X_Zahler] + ';;
// und prüfen ob existent
X_Kette+=X_Flag=(X_Zeiger!=null);
// und übernehmen
X_Kette+=('if(X_Flag){X_Daten_PARAM.' + X1_Daten_PARAM_ElementeBezeichner[X_Zahler] + '=X_Zeiger;}');
// ----- übernehmen
eval(X_Kette);

// Hinweis: Folgendes geht nur, wenn der Index selbst keine Punktnotaion enthält (da
//           jeder Punkt einen Unter-Index anzeigt)
//           X_Daten_PARAM[X1_Daten_PARAM_ElementeBezeichner[X_Zahler]]
//           = X_Daten[X1_Daten_PARAM_ElementeBezeichner[X_Zahler]];
}

// +++++ Eventhandler, die nicht onLoad oder onError sind, werden mit den Zeigern
//           aus der DatenStruktur.events nach Daten PARAM übernommen
// ----- Struktur abklappern
for (X_Index in X_Daten.events)
{// ----- wenn Element verfügbar ist
```



```

//      und Bezeichner nicht 'onload' ist
//      und Bezeichner nicht 'onError' ist
//      Bezeichner laut GrandPiano.js
if( X_Daten.events[X_Index]
    && (X_Index != "onLoad")
    && (X_Index != "onError")
)
{
// ---- dann Zeiger des Handlers nach Daten PARAM
X_Daten_PARAM[X_Index] = X_Daten.events[X_Index];
}
}

// +++++ Verfügbarkeit von Silverlight-ActiveX bzw. -PlugIn ermitteln
if (Y_SilverlightEinsetzbarkeitErmitteln_AufActiveXbzwPluginPruefen(X1_SilverlightControl_Version))
{
// +++++ wenn Zähler auf 0 so onUnload-Event-Überwachung aktivieren
//      > 0 so onUnload-Event      überwachen
if (Silverlight._silverlightCount == 0)
{if (window.addEventListener){window.addEventListener("onunload", Y_Window_Eventhandler_OnUnload, false);}}
else{window.attachEvent("onunload",Y_Window_Eventhandler_OnUnload);}

// +++++ Kennung der Eventhandler erzeugen und in Kopie-Datenstruktur Properties merken für
//      Y_Window_Eventhandler_OnUnload
// ---- variablen Teil der Kennung bilden
X_Wert = Silverlight._silverlightCount++;
X_Kette=X_Wert.toString();
// ---- Kennung merken
X_Daten_PARAM.onLoad = X_WindowEventhandler_Load_Kennung + X_Kette;
X_Daten_PARAM.onError = X_WindowEventhandler_Error_Kennung + X_Kette;

// +++++ und Eventhandler an window übergeben unter gerade erstellter Kennung
//      Hinweis: Zeiger Löschen per Y_Window_Eventhandler_OnUnload()
// ---- onLoad
window[X_Daten_PARAM.onLoad] = function(X_FehlerObjekt)
{
if (X_Daten.events.onLoad)
{X_Daten.events.onLoad(document.getElementById(X_Daten_OBJECT.id),
X_Daten.context,X_FehlerObjekt);
}
};
// ---- onError
window[X_Daten_PARAM.onError] = function(e, X_FehlerObjekt)
{
if (X_Daten.events.onError)
{X_Daten.events.onError(e, X_FehlerObjekt);}
else
{Y_Silverlight_EventHandler_Standard(e, X_FehlerObjekt);}
};

// +++++ Silverlight-HTML-Code ermitteln (OBJECT mit PARAM)
X_Kette = Y_SilverlightControl_HTMLCode_ermitteln(X_Daten_OBJECT, X_Daten_PARAM);
}
else{X_Kette = Y_SilverlightOnlineInstallControl_HTMLCode_ermitteln(X_Daten);}
//      nicht verfügbar also Download der Silverlight-Variante
//      OBJECT ohne PARAM

// +++++ wenn Eltern existieren, dann HTML-Code in den Eltern ablegen (Eltern neu rendern)
if (X_Daten.parentElement){X_Daten.parentElement.innerHTML = X_Kette;}
else{return X_Kette;}

} // Datenstruktur verfügbar
} // Silverlight einsetzbar
};

// +++++ Silverlight-Objekt erzeugen und Struktur füllen mit Daten +++++
function Y_SilverlightVerwaltung_Aktivieren(X_Daten)

```



```

// X_Daten wird festgelegt in GrandPiano.js
//           bezüglich Elemente-Bezeichner und Werte
// benötigt gefüllt X_SilverlightEinsetzbar true so Silverlight einsetzbar
{var X_DatenRette;
var X_ReturnWert;
// +++++ prüfen ob Silverlight einsetzbar ist
if(X_SilverlightEinsetzbar)
{if(X_Daten!=null)
{X_DatenRette = X_Daten;
X_ReturnWert = Y_SilverlightVerwaltung_Aktivieren_InitUndStart(X_DatenRette);
if (X_DatenRette.parentElement==null){return X_ReturnWert;}
}
}
};

// ++++++ Silverlight-Init und Start der Silverlight-Verwaltung
// ++++++

// +++++ Silverlight-Objekt Silverlight als Unterobjekt von window prüfen
var X_SilverlightEinsetzbar=false; // true, so Browser und Betriebssystem passen zu Silverlight
//                               // und Silverlight ist als ActiveX bzw. Plugin verfügbar
//                               // ob Silverlight-Objekt initialisiert ist, wird nicht erfasst

// +++++ Silverlight komplett erzeugen
function createSilverlight() // aufgerufen in index.html
// liefert true wenn Silverlight erfolgreich erzeugt und aktiviert wurde
{var EventVerwaltung; // Eventverwaltung-Objekt
var X_EventDelegate; // Funktionszeiger auf Routine, die Eventhandler einbindet
var X_Kette='errorHandle';

// +++++ Annahme: Silverlight nicht einsetzbar
X_SilverlightEinsetzbar=false;

// +++++ Sys und Silverlight je als Objekt von window leer erzeugen
if(!window.Sys){window.Sys = {};}
if(!window.Silverlight){window.Silverlight = {};}

// +++++ Silverlight-Eventhandler in aktive Silverlight-Eventverwaltung einbinden
// wird auch in anderen js-Dateien benötigt
Silverlight.createDelegate = function(EventVerwaltungInstanz, Handler_SilverlightEvent)
{return function()
{return Handler_SilverlightEvent.apply(EventVerwaltungInstanz, arguments);}
// arguments Collection der Funktionsargumente (Parameter)
}

// +++++ Silverlight-Schnittstellen init
Silverlight._silverlightCount=0;
Silverlight.ua =null;
Silverlight.available =false;
Silverlight.fwlinkRoot =X_SilverlightDownloadUrl;
Silverlight.default_error_handler =Y_Silverlight_EventHandler_Standard;
Silverlight.followFWLink =Y_DownloadLink_OnClick;
Silverlight.__cleanup =Y_Window_Eventhandler_OnUnLoad;
Silverlight.buildHTML =Y_SilverlightControl_HTMLCode_ermitteln;
Silverlight.buildPromptHTML =Y_SilverlightOnlineInstallControl_HTMLCode_ermitteln;
Silverlight.isInstalled =Y_SilverlightEinsetzbarkeitErmitteln_AufActiveXbzwPluginPruefen;
Silverlight.detectUserAgent =Y_SilverlightEinsetzbarkeitErmitteln;
// füllt Silverlight.ua
// füllt nicht X_SilverlightEinsetzbar
// liefert true so Silverlight einsetzbar (ob Silverlight-Objekt initialisiert ist,
// wird nicht beachtet)
Silverlight.supportedUserAgent =
Y_SilverlightEinsetzbarkeitErmitteln_AufUnerlaubteKombinationBrowserBetriebssystemPruefen;
// benötigt gefüllt Silverlight.ua

```



```

        // Returnwert: wenn true geliefert so mindestens 1 unerlaubte Kombination erkannt
        //                false geliefert so keine unerlaubte Kombination erkannt
        // füllt nicht X_SilverlightEinsetzbar
Silverlight.createObject      =Y_SilverlightVerwaltung_Aktivieren_InitUndStart;
                                // aufgerufen von Y_SilverlightVerwaltung_Aktivieren;
Silverlight.createObjectEx    =Y_SilverlightVerwaltung_Aktivieren;

// +++++ prüfen ob Silverlight von Betriebssystem UND Browser unterstützt wird
//                UND Silverlight-ActiveX- bzw. -Plugin verfügbar ist
//                also ob Silverlight einsetzbar ist
// (ob Silverlight-Objekt initialisiert ist, wird nicht beachtet)
X_SilverlightEinsetzbar=Y_SilverlightEinsetzbarkeitErmitteln();
        // füllt Silverlight.ua
        // füllt nicht X_SilverlightEinsetzbar
        // liefert true so Silverlight einsetzbar (ob Silverlight-Objekt initialisiert ist,
        //                wird nicht beachtet)

if(X_SilverlightEinsetzbar)
{
// +++++ Eventverwaltung erzeugen und starten
//                X1_GrandPiano erzeugt in GrandPiano.js
EventVerwaltung = new X1_GrandPiano.EventVerwaltungInstanzieren();
                                // aus GrandPiano.js

// +++++ DANN Delegate der Eventverwaltung erzeugen
X_EventDelegate=Silverlight.createDelegate(EventVerwaltung, EventVerwaltung.Handler_SilverlightEvent_onLoad);
                                // EventVerwaltung.Handler_SilverlightEvent_onLoad laut GrandPiano.js

// +++++ DANN init der Pflichtelemente der Silverlight-Daten
X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_parentElement_Index]]
        = document.getElementById(X1_SilverlightControlHost_ID);
                                // Zeiger auf DIV-Container aus index.html;
                                // parenElement

// event wurde bereits als Array() erzeugt
X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_events_Index]]
        [X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_onError_Index]]=X_Kette;
                                // entspricht events.onError
X1_SilverlightDaten[X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_events_Index]]
        [X1_Silverlight_DatenElementeBezeichner[X1_Silverlight_DatenElementeBezeichner_onLoad_Index]]=
                                X_EventDelegate;
                                // entspricht events.onLoad

// +++++ DANN Silverlight-Objekt erzeugen und Animation starten
Silverlight.createObjectEx(X1_SilverlightDaten);
}
return X_SilverlightEinsetzbar;
}

```



**GrandPiano.xml**

```

<!--
//
// grandpiano.xml
//
// Â© 2007 Microsoft Corporation. All Rights Reserved.
//
// This file is licensed as part of the Silverlight 1.0 SDK, for details look
// here: http://go.microsoft.com/fwlink/?LinkID=89144&clcid=0x409
//
//
-->
<Canvas
  xmlns="http://schemas.microsoft.com/client/2007"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
>
  <Image           Height="500" Width="1024"           Source="bg.jpg"           />
  <Image x:Name="imgC" Height="500" Width="1024" Canvas.Top="0" Source="C.jpg" Opacity="0"/>
  <Image x:Name="imgC2" Height="500" Width="1024" Canvas.Top="0" Source="C2.jpg" Opacity="0"/>
  <Image x:Name="imgD" Height="500" Width="1024" Canvas.Top="0" Source="D.jpg" Opacity="0"/>
  <Image x:Name="imgD2" Height="500" Width="1024" Canvas.Top="0" Source="D2.jpg" Opacity="0"/>
  <Image x:Name="imgE" Height="500" Width="1024" Canvas.Top="0" Source="E.jpg" Opacity="0"/>
  <Image x:Name="imgF" Height="500" Width="1024" Canvas.Top="0" Source="F.jpg" Opacity="0"/>
  <Image x:Name="imgF2" Height="500" Width="1024" Canvas.Top="0" Source="F2.jpg" Opacity="0"/>
  <Image x:Name="imgG" Height="500" Width="1024" Canvas.Top="0" Source="G.jpg" Opacity="0"/>
  <Image x:Name="imgG2" Height="500" Width="1024" Canvas.Top="0" Source="G2.jpg" Opacity="0"/>
  <Image x:Name="imgA" Height="500" Width="1024" Canvas.Top="0" Source="A.jpg" Opacity="0"/>
  <Image x:Name="imgA2" Height="500" Width="1024" Canvas.Top="0" Source="A2.jpg" Opacity="0"/>
  <Image x:Name="imgB" Height="500" Width="1024" Canvas.Top="0" Source="B.jpg" Opacity="0"/>

  <MediaElement x:Name="sndC" AutoPlay="false"           Source="C.wma"           />
  <MediaElement x:Name="sndC2" AutoPlay="false"          Source="C2.wma"          />
  <MediaElement x:Name="sndD" AutoPlay="false"           Source="D.wma"           />
  <MediaElement x:Name="sndD2" AutoPlay="false"          Source="D2.wma"          />
  <MediaElement x:Name="sndE" AutoPlay="false"           Source="E.wma"           />
  <MediaElement x:Name="sndF" AutoPlay="false"           Source="F.wma"           />
  <MediaElement x:Name="sndF2" AutoPlay="false"          Source="F2.wma"          />
  <MediaElement x:Name="sndG" AutoPlay="false"           Source="G.wma"           />
  <MediaElement x:Name="sndG2" AutoPlay="false"          Source="G2.wma"          />
  <MediaElement x:Name="sndA" AutoPlay="false"           Source="A.wma"           />
  <MediaElement x:Name="sndA2" AutoPlay="false"          Source="A2.wma"          />
  <MediaElement x:Name="sndB" AutoPlay="false"           Source="B.wma"           />

  <Canvas x:Name="pathCanvas">
    <Path x:Name="CPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="161.666656494141"
      Height="272.333312988281" Canvas.Left="135.5" Canvas.Top="138.83332824707" Data="M272.66667,140 L136,330 147.33333,336
      162.66667,410 66667 247.33333,410 234.66667,330 66667 278,248 254,242 66667 258.66667,200 66667 296.66667,139 33333 272.66667,140" />
    <Path x:Name="C2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="91" Height="136.333319306374"
      Canvas.Left="263.5" Canvas.Top="105.500005066396" Data="M325.33333,106.66667 L264,202 66667 266,241.33333 307.33333,238.66667
      354,138 351.33333,106.66666 326.66667,106" />
    <Path x:Name="DPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="131" Height="274.333297729492"
      Canvas.Left="247.499999999999" Canvas.Top="137.499984741212" Data="M358,142.66667 L311.33333,243.33333 288.66667,246
      248,330 66667 258,338 266.66667,411 33333 354,410 66665 347.33333,330 66668 374,246 66668 350,243 33336 352.66667,202 66669
      378,138 00005 360.66667,140 00005" />
    <Path x:Name="D2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="70.3333129882813"
      Height="134.333319425583" Canvas.Left="356.83334350586" Canvas.Top="107.500003457069" Data="M394.66667,108.66667
      L357.33333,204 359.33333,241.33333 398,240 66667 426.66667,134 66667 425.33333,108" />
    <Path x:Name="EPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="117.666656494141"
      Height="274.333343505859" Canvas.Left="358.833343505859" Canvas.Top="137.5" Data="M431.33333,140 L403.33333,245.33333
      381.33333,248 359.33333,330 368.66667,339 33333 372,411 33333 458,411 33333 455.33333,331 33333 476,138 433.33333,138 66667" />
    <Path x:Name="FPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="99.6666564941406"
      Height="273.666680037975" Canvas.Left="470.83334350586" Canvas.Top="138.833345949649" Data="M486.66667,140 66667

```





```
L471.33333,332 477.33333,334 476.66667,412 563.33333,411.33335 564,339.33334 570,328 560,246.66665 535.33333,243.33331 530,139.3333
486.66667,139.33329" />
<Path x:Name="F2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="51" Height="133.666656494141"
Canvas.Left="534.166687011719" Canvas.Top="108.16665649414" Data="M538.66667,109.33333 L534.66667,137.33333 542,241.33333
584,240.66666 584.66667,205.33334 568,108.66667 540.66667,108.66668" />
<Path x:Name="GPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="111" Height="274.333343505859"
Canvas.Left="570.166687011719" Canvas.Top="137.5" Data="M570.66667,248 L582.66667,336 582,411.33333 669.33333,410.66667
672.66667,332.66667 680.66667,330 655.33333,245.33333 631.33333,244 601.33333,138 578.66667,138 594.66667,244.66667 571.33333,248" />
<Path x:Name="G2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="76.3333129882813"
Height="131.666656494141" Canvas.Left="603.5" Canvas.Top="110.166664123534" Data="M612,111.33333 L604,136.66667 636,238.66667
636.66667,241.33333 679.33333,241.33333 679.33333,206.66667 641.33333,111.33334 612.66667,110.66668" />
<Path x:Name="APath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="138.333312988281"
Height="274.333343505859" Canvas.Left="655.5" Canvas.Top="137.5" Data="M665.33333,246 L694.66667,336.66667 686,411.33333
774.66667,410 785.33333,335.33333 793.33333,328.66667 750.66667,246.66667 722,243.33333 673.33333,138 656,138.66667
688.66667,216.66667 685.33333,245.33333 666,246" />
<Path x:Name="A2Path" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="95" Height="131"
Canvas.Left="676.166687011719" Canvas.Top="110.833320617675" Data="M682,111.33333 L676.66667,138.66667 730.66667,240.66667
768.66667,241.33333 770.66667,204.00001 713.33333,111.33334 683.33333,111.33335" />
<Path x:Name="BPath" Opacity="0.01" Fill="#FFFFFF" Stretch="Fill" Stroke="#FF000000" Width="172.333312988281"
Height="275.666687011719" Canvas.Left="733.5" Canvas.Top="136.833320051431" Data="M758,247.33333 L806,333.33333 790,412 878,412
894.66667,334 905.33333,327.33333 764.66667,138.66667 734,137.33333 776.66667,207.33334 778.66667,246.66668 758.66667,248" />
</Canvas>
</Canvas>
```





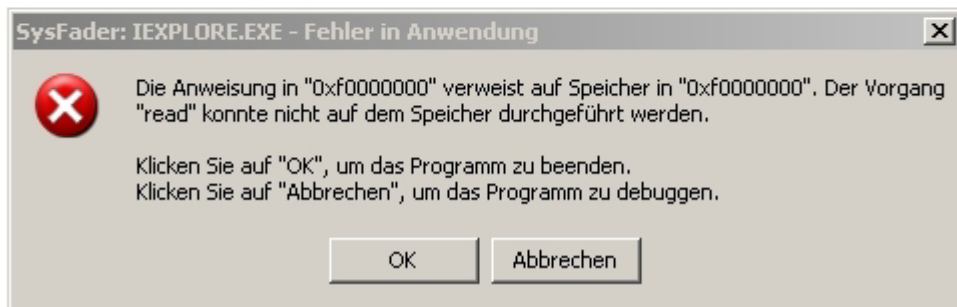
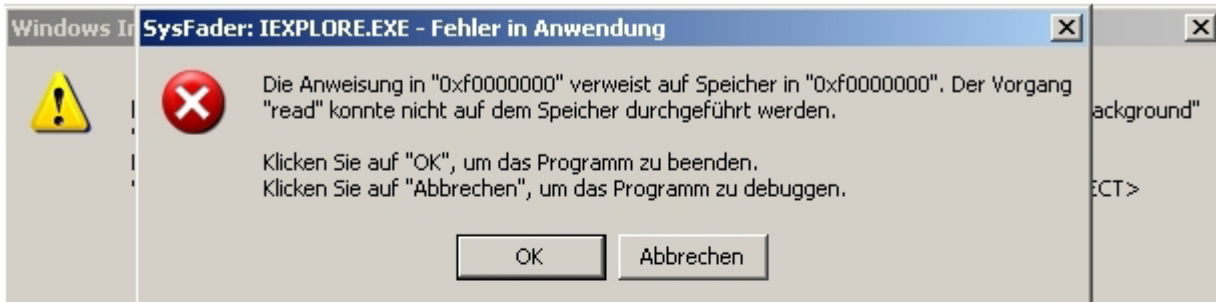
### Adressfehler des Internet Explorers bei Abarbeitung bei Silverlight

Sollte Der Internet Explorer mit Start der index.html regelmäßig einen Adressfehler bringen, der auch die Adresse 0x00000000 umfassen kann, ist das Silverlight-ActiveX-Control defekt und es muss neu installiert werden: Dazu lade man sich die aktuellen Updates

zum IE per <http://msdn2.microsoft.com/en-us/library/bb188743.aspx>

zu Silverlight per <http://silverlight.net/Default.aspx>

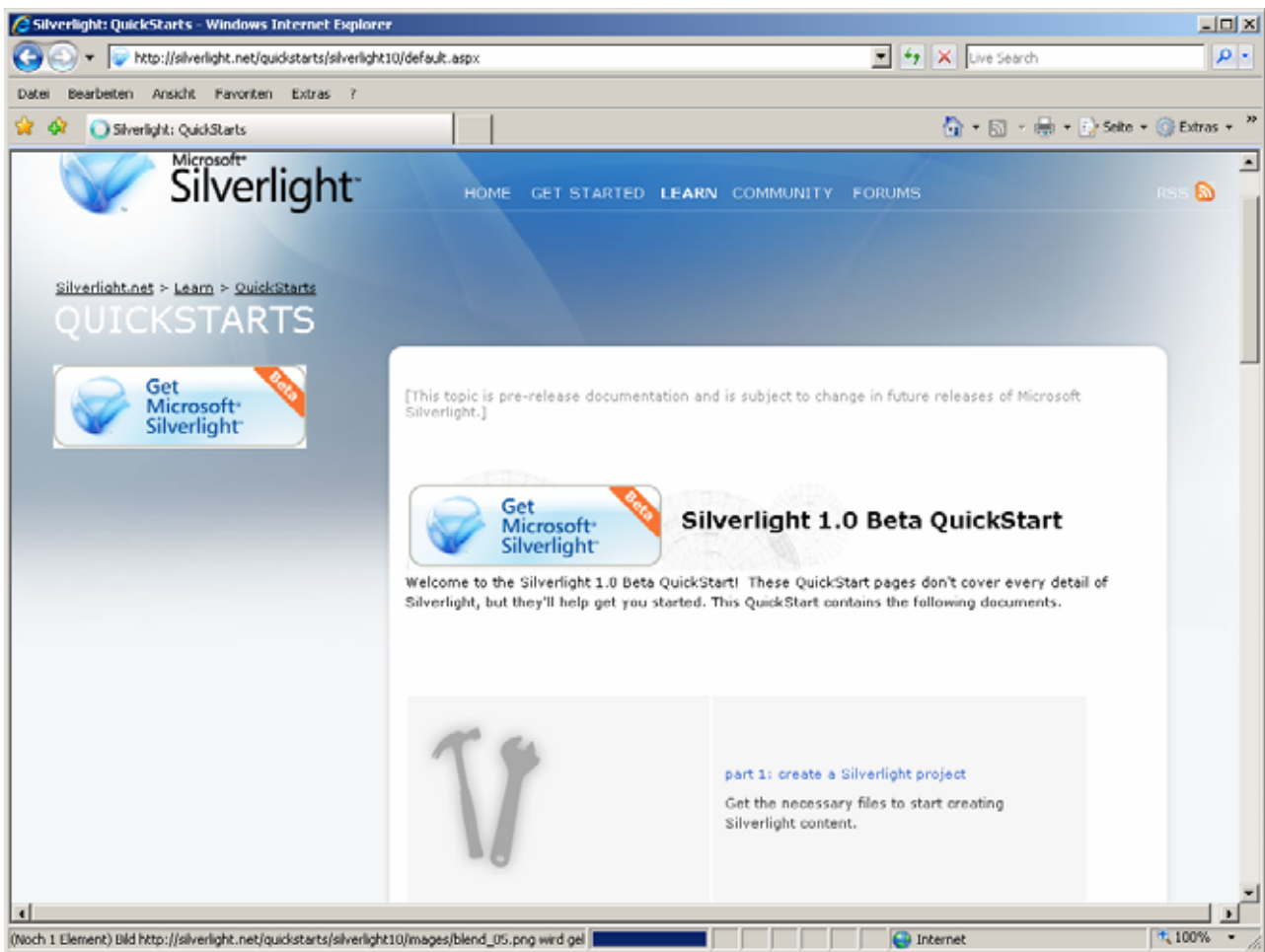
herunter und installiere diese. Man beachte Nicht-Kompatibilität von Silverlight-Varianten (inkl. deren Beta-Versionen).

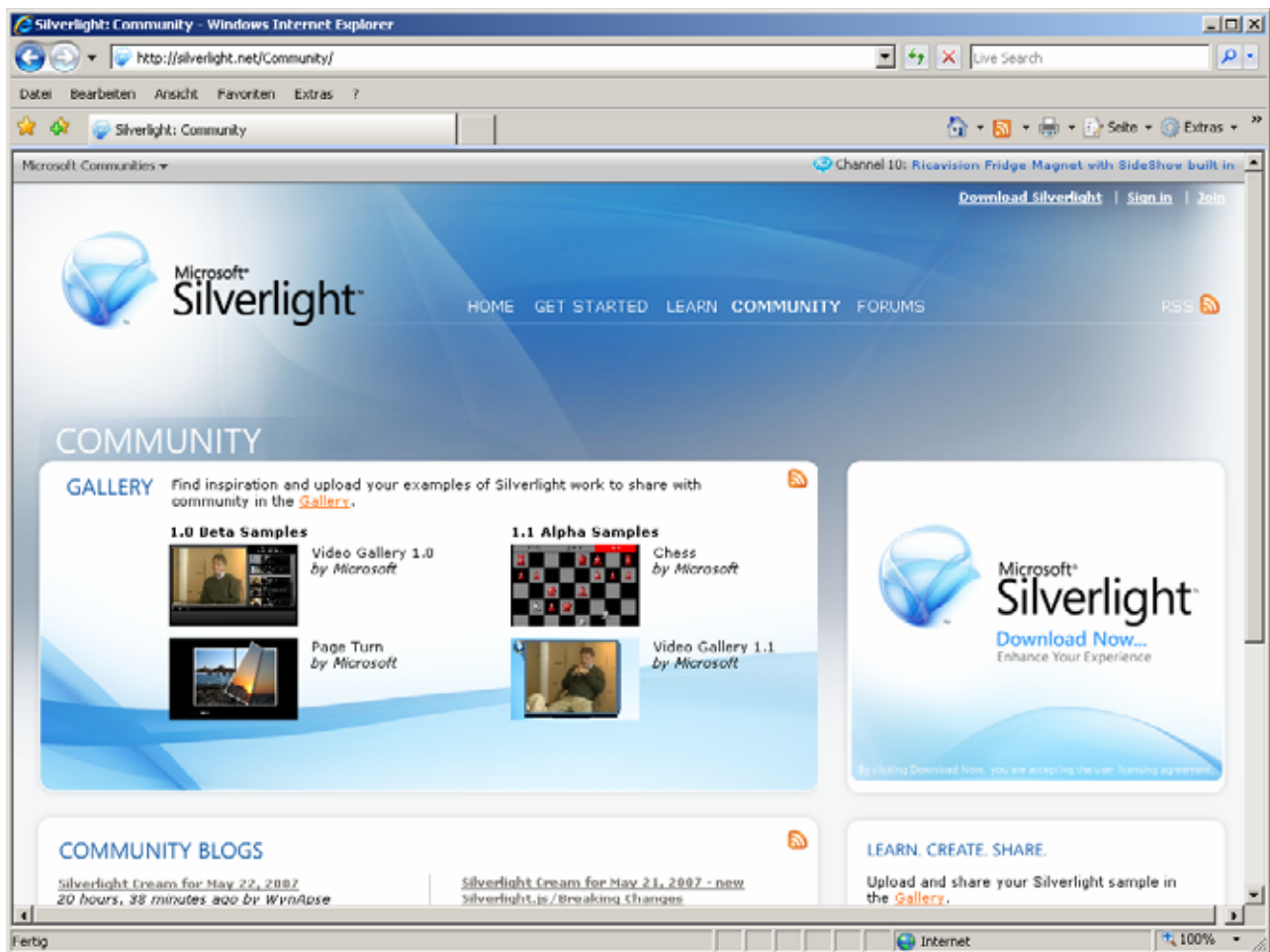


## **Silverlight-Webseite silverlight.net (u.a. Download von Silverlight-Anwendungen)**

<http://silverlight.net/Default.aspx>







<http://silverlight.net/community/communitygallery.aspx>

Hier können auch (kostenlose) Silverlight-Anwendungen getestet und gedownloadet werden, für die immer die Lizenzbedingungen zu beachten sind. Für den Test der Anwendungen muss Silverlight installiert sein, wobei die Version von Silverlight zu beachten ist. Wer eine Silverlight-Anwendung erzeugt hat, kann diese auf der Webseite auch online stellen.



## CATEGORY: 1.0 BETA SAMPLES

Show by: Most Recent | [Top Rated](#)



★★★★★  
[Silverlight Map](#)

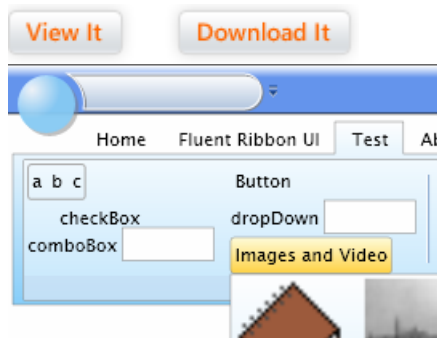
Uploaded on May 18

Created by: Michael Scherotter

Transforming a Mindjet MindManager Map into a Silverlight application, Michael Scherotter a Microsoft evangelist, has hosted this application on silverlight.live.com. Zoom, pan, expand, collapse, click on hyperlinks and attachments. The download is a small client application (with source code) that transforms and uploads the map.

By clicking





Download It you  
accept the [license](#)



[Office 2007 Ribbon UI for Websites](#)

Uploaded on May 18

Created by: Michael Scherotter

This uses the Microsoft Office 2007 RibbonX XML to define Website ribbons that are transformed into Silverlight XAML using XSLT. This is a work-in-progress by Michael Scherotter a Microsoft evangelist that, once done, will comply with the 2007 Microsoft Office System User Interface Guidelines.



By clicking  
Download It you  
accept the [license](#)



[Tile Text](#)

Uploaded on April 29

Created by: Microsoft

A cool text animator that displays the phrase you give it in a tiled artistic fashion.



By clicking  
Download It you  
accept the [license](#)







[Ink Tattoo Studio](#)

Uploaded on April 29

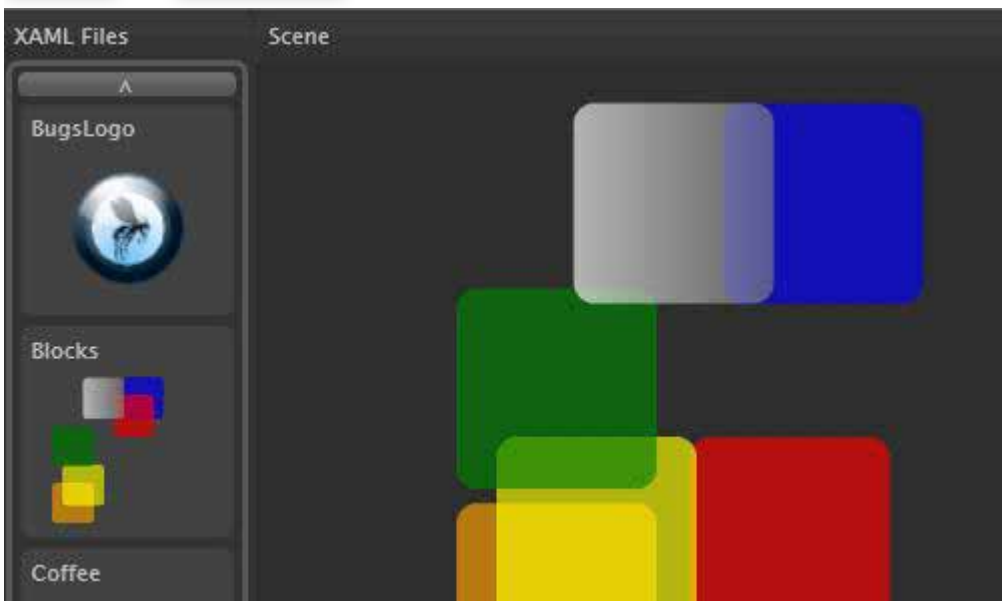
Created by: Microsoft

Add some ink on your arm with this virtual tatoo studio. Pick a color, and go to work (without having to worry about permantent mistakes!)

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)



[Silverlight Pad](#)

Uploaded on April 29

Created by: Microsoft

View the XAML source behind the scene as you click and drag silverlight xaml objects.

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)





[Page Turn](#)

Uploaded on April 29

Created by: Microsoft

Simulate the turning of pages in a book or magazine with this great demo of animation and media in Silverlight.

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)



[Grand Piano](#)

Uploaded on April 29

Created by: Microsoft

Try your music skills with this keyboard driven Silverlight Grand Piano. Keyboard strokes trigger animation and sound.

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)







[Clock](#)

Uploaded on April 29

Created by: Microsoft

Real time display of hours minutes and seconds using an analog clock.

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)



[Video Library](#)

Uploaded on April 29

Created by: Microsoft

Browse through and play a collection of videos with this Silverlight Video Library and video player example.

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)



[Sprawl](#)

Uploaded on April 29

Created by: Microsoft

Sprawl is a basic point and click game where you race the computer opponent in a challenge to take over the most real estate on the board.

[View It](#)

[Download It](#)

By clicking  
Download It you  
accept the [license](#)



## Microsoft Silverlight-Webseite auf microsoft.com

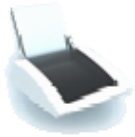
<http://www.microsoft.com/silverlight/default01.aspx>











## FREQUENTLY ASKED QUESTIONS



### General questions

#### What is Silverlight?

Silverlight is a cross-browser, cross-platform plug-in for delivering the next generation of Microsoft .NET-based media experiences and rich interactive applications for the Web.

#### Is Silverlight free?

Yes, Microsoft will make the Silverlight browser plug-in freely available for all supported platforms.

#### Is Silverlight the official name for "WPF/E"?

Yes. Silverlight was formerly code-named "WPF/E."

#### What is the long-term goal or vision for Silverlight?

Silverlight is a cross-browser, cross-platform plug-in for delivering the next generation of .NET-based media experiences and rich interactive applications (RIAs) for the Web. Silverlight offers a flexible programming model that supports AJAX, Visual Basic .NET, C#, Python, and Ruby and integrates with existing Web applications. Silverlight media capabilities include fast, cost-effective delivery of high-quality audio and video to all major browsers including Mozilla Firefox, Apple Safari, and Windows Internet Explorer running on Mac OS or Microsoft Windows. By using Microsoft Expression Studio and Microsoft Visual Studio, designers and developers can collaborate more effectively using the skills they have today to light up the Web of tomorrow.

#### What are the main features and benefits of Silverlight?

Key benefits of Silverlight include:

##### 1. Compelling cross-platform user experiences

- Deliver media experiences and rich interactive applications (RIA) for the Web that incorporate video, animation, interactivity, and stunning user interfaces (UIs).
- Seamless, fast installation for users, thanks to a small, on-demand, easy-to-install plug-in that is under 2 megabyte (MB) in size and works with all leading browsers.
- Consistent experiences on Windows and on Mac OS without any additional installation requirements.
- Create richer, more compelling Web experiences that take greater advantage of the client for increased performance.
- Stunning vector-based graphics, media, text, animation, and overlays enable seamless integration of graphics and effects into any existing Web application.
- Enhance existing standards/AJAX-based applications with richer graphics and media and improve their performance and capabilities by using Silverlight.

##### 2. Flexible Programming Model with Collaboration Tools

- Based on the Microsoft .NET Framework, Silverlight enables developers and designers to easily use existing skills and tools to deliver media experiences and RIAs for the Web.
- Choice of programming languages such as AJAX, Visual Basic .NET, C#, Python, and Ruby offers developers and designers the flexibility to use their existing skills without the need to learn a new language.
- Simple integration with existing Web technologies and assets means Silverlight works with any back-end Web platform or technology. No "rip and replace" required. Silverlight integrates





with your existing infrastructure and applications, including Apache and PHP, as well as with JavaScript and XHTML on the client.

- Role-specific tools for both designers and developers that take advantage of Web standards and the breadth of the Microsoft .NET-connected software features.
  - Designers will like that Expression Studio creates interactive UIs and media rich experiences, prepares media for encoding and distribution, and creates World Wide Web Consortium (W3C) standards-compliant sites by using modern XHTML, XML, XSLT, CSS, and Microsoft ASP.NET.
  - Developers will use Visual Studio for development of client and server code with full Microsoft IntelliSense, powerful debugging, rich language support, and more.
  - Consistent presentation model using XAML, the declarative presentation language used in Windows Vista applications. Controls, visual designs, media, and other elements can be presented with full design fidelity in both Silverlight and Windows-based applications.
  - Dramatically improved performance for AJAX-enabled Web sites with the power, performance, and flexibility of Silverlight and .NET-connected software.
- 3. High-quality media, low-cost delivery**
- Unified media format scales from high definition (HD) to mobile with Windows Media Video (WMV), the Microsoft implementation of the Society of Motion Picture and Television Engineers (SMPTE) VC-1 video standard, as well as support for Windows Media Audio (WMA) and MP3 audio.
  - Add vector-based graphics and overlays to media with support for integration of graphics that scale to any size and broadcast-style overlays for tickers and closed captioning.
  - Flexible ad-insertion solutions with video and animation including the ability to deliver fluid, broadcast-style video or animated advertisements without any loss of visual fidelity or motion quality.
  - Lower-cost media streaming with Emmy Award-winning Windows Media technologies that can significantly lower the cost of streaming delivery with the flexibility to work with existing Windows Media streaming deployments. Even further cost reductions are possible with the upcoming Microsoft Internet Information Services (IIS) Media Pack for Windows Server Code Name "Longhorn."
  - Broad ecosystem of media tools, servers, and solutions compatible with the Windows Media operating system.
  - Powerful encoding tools for live and on-demand publishing of media experiences with Expression Media Encoder, including hardware-accelerated encoding of WMV and VC-1 at up to 15 times the performance of software alone when paired with a Tarari Encoder Accelerator board (based on Tarari internal tests).
- 4. Connected to data, servers, and services**
- Create mash-ups by incorporating data and services from anywhere on the Web by using Silverlight support for LINQ and LINQ-to-XML. Access data with common protocols like JSON, RSS, POX, and REST.
  - Increase discoverability of RIA content that can be indexed and searched, thanks to the Silverlight text-based XAML format.
  - Rapidly scale Silverlight applications by using Silverlight™ Streaming by Windows Live

**Is the re-released Silverlight Beta different from "WPF/E" February 2007 Customer Technology Preview (CTP)?**

Yes, the Beta is a new release. Read more about what is new in the Beta in the [What's New in the Silverlight February 2007 CTP whitepaper](#).

**How do Silverlight 1.0 Beta and Silverlight 1.1 Alpha differ?**

Silverlight 1.1 Alpha builds on the Silverlight 1.0 Beta to add managed code programmability and a managed object model.

**Can I deploy Silverlight-based applications with the CTP?**

Yes. Although Silverlight Beta releases are for evaluation and testing purposes, it has a go-live license enabling commercial deployment. Note that although commercial deployments are enabled, the software is still in Beta and the support is thus limited.

**Is there an expiration date for the Beta?**

Yes, the Beta expires on August 1, 2007. An update will be available prior to the expiration. Upon expiration, the client will prompt you with an expiration notice and ask you to upgrade. Web developers will be able to offer the update programmatically, making the experience easy for their users.



#### Is Microsoft on schedule to ship Silverlight in the first half of this year?

Yes.

#### What features are available in the Silverlight 1.0 Beta?

Silverlight 1.0 Beta includes:

- Enhanced Web audio and video streaming and playback with industry-leading Windows Media technologies tapping into the existing ecosystem of tools, servers, and solutions, as well as new features for interactive video, overlays, ad presentation, and more.
- A cross-platform, cross-browser plug-in with fast and easy install (Beta plug-in is approximately 2 MB) for Mac OS and Windows operating systems.
- Easy integration with client-based and server-based AJAX solutions including ASP.NET AJAX. This empowers developers to use standards-based Web techniques to enhance AJAX applications with Silverlight.
- Lower-cost deployment and distribution of streamed audio and video when compared to other comparable solutions (up to 50% less cost reported by content delivery networks (CDNs)).
- Broad support of tools for designers, developers, and video professionals, that tap into Web standards plus a breadth of Microsoft application platform features:
- For designers, Expression Design and Expression Blend 2 May Preview for creating XAML-based rich interfaces. Expression Web for Web-standards-based design using XHTML, XML, XSLT, and ASP.NET for building compelling modern Web sites.
- For video professionals, Expression Media, a full-featured digital asset management and video encoding solution for the enhancement, compression, and publishing of video for Silverlight.
- For developers, Visual Studio–based support for Web standards development including ASP.NET AJAX with full IntelliSense editing for client script and server code.

#### What features are available in the Silverlight 1.1 Alpha?

The Silverlight 1.1 Alpha includes all the features available in Silverlight 1.0 Beta plus a number of new features focused primarily on improving the developer productivity and power, including:

- Managed code support
- Support for dynamic languages including managed Microsoft JScript and Python
- Rich UI control model based on WPF
- Improved networking stack with support for REST, RSS, JSON, and POX
- Enhanced, two-way HTML/AJAX bridge
- Comprehensive and consistent base class library
- Support for LINQ (LINQ to Objects, LINQ to XML)

#### Which platforms and browsers will Silverlight support?

Silverlight will support all major browsers on both Mac OS X and on Windows. Particular care is being taken to account for differences in platform and browser capabilities to ensure a consistent experience including experiences on FireFox, Safari, and Internet Explorer.

#### What Macintosh OS versions and hardware will be supported?

Macintosh OS X 10.4.8 or later will be supported for both Power PC–based and Intel-based hardware via a Universal Binary.

#### How will Silverlight be distributed by Microsoft?

Microsoft designed Silverlight to fit a broad range of uses that require reach and a compelling user experience. Many Microsoft projects will be using Silverlight for delivering richer user experiences and applications for the Web.

#### Is the installation experience of the Beta representative of what customers will see when released?

The installation experience for the Beta is close to the final installation experience. The final release will be even more refined.

#### Will I need more memory, a faster processor, or a better Graphics Processing Unit (GPU)?

Microsoft designed Silverlight with the ability to deliver high-fidelity experiences on the broadest set of system configurations. Some features, such as HD video, may benefit from newer personal computers. Final system requirements will be available at release.

#### Is Microsoft considering support for additional operating systems?





Microsoft is gathering feedback from customers like you on Silverlight and to help determine which platforms should be supported in the future.

#### Which devices will be supported?

Device platforms are being considered based on customer feedback.

## Silverlight installation and Web site issue

#### Where can I go to install Silverlight?

Go to the [Silverlight installation](#) page to start.

#### How do I ensure that the installation of Silverlight will be successful?

For most customers, these steps help:

- Before you download Silverlight, read the [system requirements for Macintosh computers](#) or the [system requirements for Windows-based computers](#) to ensure your system is compatible.
- Make sure you [uninstall any previous version of Silverlight](#), formerly code-named "WPF/E."
- If you are running Firefox or Safari, restart your browser.

#### I have still have problems with installation, where do I go for help?

Silverlight is still in beta. However, many people get answers to their questions about installation through the [Silverlight Installation and Setup forum](#).

#### I installed Silverlight successfully and I still have problems viewing web sites with Silverlight, where do I go for help?

The [installation and setup forum](#) can address those questions quite efficiently. For specific issues with this site you can also send an e-mail message to: [sbug@microsoft.com](mailto:sbug@microsoft.com).

#### How do I uninstall Silverlight?

Follow the instructions on the [Uninstalling previous versions of Silverlight](#).

## Development related

#### Can I redistribute the Silverlight 1.0 Beta release?

Yes, the Silverlight 1.0 Beta release offers a Go-Live license. Hence, you can begin building applications that target Silverlight immediately. Silverlight 1.1 Alpha does not offer a Go-Live license at this time.

#### Can I redistribute the Silverlight 1.1 Alpha release?

No, Silverlight 1.1 Alpha does not offer a Go-Live license at this time.

#### Where can I find Silverlight and associated Software Development Kits (SDKs) for download?

Silverlight betas and released versions can be found on the [Microsoft Silverlight](#) site. Developer and designer resources such as the SDK and CTP releases can be found at the [MSDN Silverlight](#) site. Another great resource for getting started with Silverlight is the newly launched Silverlight community site.

#### How can I build experiences and applications with Silverlight?

Silverlight development tools include role-specific productivity tools for both designers and developers:

- **Expression Studio** empowers designers to create interactive UI and media-rich experiences, prepare media for encoding and distribution, and create W3C standards-compliant sites using modern XHTML, XML, XSLT, CSS, and ASP.NET. Expression Design includes support for exporting XAML for Silverlight. At MIX 07, Microsoft released Expression Blend 2 May Preview and Expression Media Encoder Preview to enable designers to build media experiences and RIAs.
- **Visual Studio** empowers developers to develop client and server code using full IntelliSense, powerful debugging, rich language support, and more.



By using Expression Studio and Visual Studio, designers and developers can collaborate more effectively using the skills they have today. Additionally, Silverlight supports a consistent subset of XAML (eXtensible Application Markup Language) for declarative programming, the same format found in .NET 3.0. Because XAML is toolable, there is always the potential for third-parties to provide additional XAML-based Silverlight tools in the future.

**Are the features of the Macintosh and Windows releases of Silverlight fully compatible?**

Yes.

**When would a customer use Silverlight instead of ASP.NET AJAX?**

Silverlight integrates with existing Web applications, including ASP.NET AJAX applications. Thus, ASP.NET AJAX and Silverlight are designed to be complementary technologies. In the broader sense, Silverlight can talk to any AJAX application, both client-side and server-side. In addition, ASP.NET AJAX can be used to control Silverlight-based visualization of data or delivery of rich experiences. Examples might include mapping applications or video playback with rich presentation.

ASP.NET AJAX and Silverlight at final release will also benefit from being a fully supported technology from Microsoft with the benefits of technical support around the clock and the breadth support of the Microsoft development community. AJAX is a fundamental technology supported in Silverlight and now, in ASP.NET.

Customers can enhance existing ASP.NET or ASP.NET AJAX applications by using Silverlight's media experiences and RIAs.

**What is the ASP.NET Futures (May 2007) release?**

The Microsoft ASP.NET Futures May 2007 ("Futures") release contains an early developer preview of features providing a wide range of new functionality for both ASP.NET and Silverlight. The Futures release includes early experimental versions of features currently under consideration for future versions of ASP.NET and the .NET Framework. The Futures releases assume prior knowledge of the core features of ASP.NET, including the previous ASP.NET AJAX Futures January CTP.

**How does Silverlight make the Microsoft development system better?**

Silverlight is a cross-browser, cross-platform plug-in for delivering the next generation of media experiences and rich interactive applications (RIAs) for the Web. Examples include:

- For ASP.NET-based Web applications, Silverlight provides a rich UI front-end that, with a consistent programming model, adds support for richer interactivity, media, and audio.
- For Microsoft SharePoint-based content, Silverlight offers the ability to create rich Web parts.
- For Windows Live services, Silverlight offers the ability to consume services and APIs more effectively.

**When would a customer use Silverlight instead of Windows Presentation Foundation (WPF)? Is**

**Silverlight more appropriate for certain types of applications?**

WPF and Silverlight are complementary platforms, based on XAML for presentation:

- WPF provides a unified programming model for building the best Windows Vista era of smart-client user experiences that incorporate UI, 3D, media, and documents.
- Silverlight is a cross-browser, cross-platform plug-in for delivering the next generation of media experiences and RIAs for the Web by using a consistent subset of WPF features and the flexibility of AJAX for programmability.

Thus, both WPF and Silverlight enable developers and designers to develop visually stunning user experiences, but they are designed for different scenarios.

**What are the key features of the ASP.NET Futures May 2007 release?**

ASP.NET Futures includes a number of new, innovative solutions that help developers become more productive and enables them to create better user experiences for their customers. Silverlight-related features include:

- **Silverlight Controls for ASP.NET.** Enables the ability to include media (video, audio) vector graphics and animations by using familiar and powerful ASP.NET server controls



- **ASP.NET Application Services.** Offers new application services to make AJAX and Silverlight Web applications more discoverable from search engines and easily searchable from within the site. Additional services help you gather and analyze data about client errors.

**Where can I go to connect with other Silverlight developers to ask questions?**

For Silverlight-related forums, blogs, and community resources for the CTP, visit the [Silverlight Support and Community](#) page.

## Media (audio and video)

**What is the relationship between Silverlight and Windows Media technologies?**

Silverlight builds on top of Windows Media in a fashion similar to the Xbox XDK (Xbox Development Kit), enabling developers and designers to collaborate in building media experiences and RIAs. Silverlight is released by the Server and Tools Division at Microsoft as a part of the .NET Framework.

**Is Silverlight a new media player?**

No. Silverlight is a cross-browser, cross-platform plug-in for delivering media experiences and RIAs. It is not a desktop application or stand-alone media player.

**What is the relationship between Silverlight and Windows Media Player?**

The Silverlight browser plug-in is a separate component, independent of Windows Media Player. Silverlight is designed for delivery of cross-platform, cross-browser media experiences and rich interactive applications (RIAs) inside a Web browser combining audio, video, animation, overlays, and more. Windows Media Player delivers a breadth of local playback and user focused experiences, while also offering support for application and Web page embedding.

**Do I need to have the latest version of Windows Media Player installed?**

No. Silverlight is completely independent and when installed is less than 2 MB in size.

**What audio or video formats are supported in Silverlight?**

Silverlight supports Windows Media Audio and Video (WMA, WMV7–9) and VC-1, as well as MP3 audio. Additional formats may be available by the final release based on customer feedback.

**Will Silverlight support all the codecs Windows Media Player supports?**

Since Silverlight is a lightweight cross-platform technology, it only carries the most common codecs that are needed for Web playback. However, we are gathering information from customers about the needed codecs and can update Silverlight when necessary.

**What is SMPTE VC-1?**

VC-1 is an industry-standard video format, recognized by the Society of Motion Picture and Television Engineers (SMPTE), and most notably ships in all HD-DVD and Blu-ray Disc–certified electronics, hardware, and tools. Windows Media Video 9 (WMV-9) is the Microsoft implementation of the SMPTE VC-1 standard video codec. Microsoft initiated development of the standard with the release of WMV-9 to SMPTE.

**Will Silverlight support HD quality?**

Yes, Silverlight supports 720p, HD quality with considerable performance benefits over other solutions. Performance is dependent upon the central processing unit (CPU) capabilities of your computer and configurations. Generally, in testing, a 3-gigahertz (GHz) CPU and/or dual-core support greatly benefit the HD playback experience.

**Will Silverlight work with my new or existing Windows Media services platform for streaming?**

Yes, in the final release. The CTP supports progressive download and playback from any server. In its final release, Silverlight will take advantage of Windows Server features for streaming.

**Will Silverlight support live streaming events as well as downloading media?**

Yes, in the final release. The February CTP is optimized for progressive "download and play" scenarios to test the platform.



**Does Silverlight support MPEG4 and H.264 video, or Advanced Audio Coding (AAC) audio, or Flash video?**

No. However, content from many of these formats can be transcoded into formats that are supported by Silverlight, such as by an automated server function (many available third-party solutions support this workflow), and then incorporated into a Silverlight-based application.

**Will Silverlight support the full range of APIs offered by the Windows Media Player ActiveX control today?**

We are actively speaking to customers and partners about their needs.

**Will Silverlight support digital rights management?**

For content providers, Silverlight will support digital rights management (DRM) built on the recently announced Microsoft PlayReady content access technology on Windows-based computers and Macintosh computers.

**Will Silverlight-based applications run in Media Center? What about Media Center Extender?**

Silverlight-based applications can run in Media Center but may have rendering issues on Media Center Extenders. Applications that are implemented using the Media Center SDK will provide a better experience. The engineering team is actively evaluating customer requirements for a variety of scenarios moving forward.

**Are there any new tools for creating and publishing media content with Silverlight?**

Yes. The recently announced Expression Media Encoder, a feature of Expression Media, will support live and on-demand encoding and template-based publishing of Silverlight-based experiences and applications. In addition, Silverlight works with the broad range of Windows Media encoding tools and utilities available today. Additional details will be made available shortly.

## Server and services

**Will Silverlight-based applications and content run on any Web Server? What are the benefits to running it on servers running Windows?**

Silverlight works with any web server just like HTML. Video and audio content can also be progressively downloaded and played back from any Web server platform. Benefits of Windows server-based distribution of Silverlight applications include Windows Media Services with Fast Stream (instant playback) and Fast reconnect technologies, lower distribution costs (streaming users only download what they watch), and tap into the full Windows server ecosystem of platform components and partner solutions. Those benefits will be enhanced in the future version of Windows Server (code name “Longhorn”) and with Internet Information Server 7 (IIS).

**What features are missing from Silverlight presentation markup that will be supported in WPF (Windows Presentation Foundation)?**

Some high-end Windows specific features of WPF, such as real 3D, hardware-based video acceleration, and full document support, will not be supported in Silverlight. This is by design in order to serve Silverlight’s cross-browser, cross-platform reach scenario that demands a light weight plug-in. That being said, Silverlight will offer a uniform runtime that can render identical experiences across browsers on both Mac OS and Windows.

**When would a customer use Silverlight versus ASP.NET AJAX?**

Silverlight integrates with existing Web applications, including ASP.NET AJAX applications. Consequently, ASP.NET AJAX and Silverlight are designed to be complementary technologies. In the broader sense, Silverlight can communicate with any AJAX application, both client and server-side. ASP.NET AJAX can additionally be used to control Silverlight-based visualization of data or



delivery of rich experiences. Examples might include mapping applications or video playback with rich presentation.

ASP.NET AJAX and Silverlight at final release will also benefit from being a fully supported technology from Microsoft, with the benefits of 24-hour technical support and the breadth of support of the Microsoft development community. AJAX is a fundamental technology supported in Silverlight and now by ASP.NET.

You can enhance existing ASP.NET or ASP.NET AJAX applications by using the Silverlight media experiences and rich interactive applications.

#### Is Microsoft making new server investments for creation and delivery of digital media?

Windows Server “Longhorn” raises the bar significantly on security, reliability, and robustness, and it is already considered to have excellent live and streaming delivery. New investments are being made in two key areas: IIS 7.0 and Windows Media Services.

- **Windows Media Services (WMS):** Under development for Windows Server “Longhorn” to provide advanced streaming media support, enabling significantly more scalable live and on-demand broadcasts that build on the industry-leading availability, cost-efficiency, and uptime offered by Microsoft Windows Server 2003 today. Silverlight joins the list of client platforms that can connect to and present WMS streamed audio and video as an integrated part of media-enabled applications for the Web, whether you are building a simple cross-platform, browser-based media experience or a rich interactive application (RIA) for the Web.
- **IIS 7.0:** For customers who prefer to use IIS for progressive download of media-enabled applications and experiences, Microsoft is announcing the IIS 7.0 Media Pack. The IIS 7.0 Media Pack reduces the cost of delivering downloaded content, such as media, thanks to bit-rate throttling. The IIS 7.0 Media Pack will be a free download for customers of Windows Server “Longhorn” later this year.

## Globalization and localization

#### Is Silverlight supported on various locales?

Silverlight installs on localized versions of Macintosh computers and Windows. At this time, the installation is available in an international English format. Final releases will render international text (using double-byte characters) and support the full 64K Unicode character set. Silverlight uses simple input mechanism that treats all the languages in the same way.

#### What are the different ways to display text with Silverlight?

Silverlight supports displaying static preformatted text that is comprised out of glyph elements and also dynamic text that uses TextBlock. With glyphs, one needs to position the characters individually while TextBlock supports simple layout.

#### What kinds of fonts are supported with Silverlight?

Beyond standard and western fonts, Silverlight also supports East Asian characters, double-byte characters, and can work with any East Asian font or Middle Eastern font by using the glyphs element and a supporting TrueType font file that supports the requested glyph.

## Silverlight Streaming

#### What is Microsoft® Silverlight™ Streaming by Windows Live™?

Microsoft® Silverlight™ Streaming by Windows Live™ offers a free cloud-based hosting and streaming solution for quickly delivering high-quality, high-scale, cross-platform, cross-browser, media-enabled RIAs.

#### How much does Silverlight™ Streaming cost?

While the product is in Beta, hosting is free of charge. Up to 4 GB of data and streaming is free of charge up to 700 kilobit/s. At the conclusion of the Beta program, the developer can chose to enable





Microsoft-sponsored advertising in the application for continued free use of the service to or subscribe to a pay-for-use service that is free of advertisements.

**How much is the pay-for-use service if I chose not to use Microsoft-sponsored advertising?**

We're not prepared to discuss the final pricing of the nonadvertising-based product at this time except to say that it will be extremely cost competitive. The advertising-based product will continue to be free in perpetuity.

**How much storage do I get and what is the bit-rate?**

A user receives 4 GB of free storage for Silverlight™ Streaming applications. Each media element in a Silverlight™ Streaming application cannot exceed a 10-minute uninterrupted video stream at 300 kilobit/s. The peak outbound bit rate for Silverlight™ Streaming is 700 kilobit/s. As the limit is set based on byte size instead of duration, by using the full bandwidth of 700 kilobit/s, the video will be truncated to approximately 4 minutes. Customers requiring additional capabilities are encouraged to contact a [Windows Media Streaming Hosting Provider](#).

**What video encoding formats are supported?**

The designer or developer is free to use any encoding format for their video supported by the Windows Media Video codec. This includes Variable Bit Rate (VBR) encoding for DVD-quality video and the use of the VC-1 codec for high-definition content. However, for HD content, be aware that the maximum output rate from the service is 700 kilobit/s, which means the client will not receive real-time delivery of HD video.

**What is the difference between Silverlight™ Streaming and other video sharing services?**

Silverlight™ Streaming is focused on developers who want to build their own media-rich applications or Web sites. Unlike other video sharing services, there are no third-party branding requirements for the use of Silverlight™ Streaming, and the developer is in full control over their rich media experience within the context of their Web site. This includes items that are not presently supported on other sites such as 16:9 aspect ratio video, DVD-quality video, stereo audio, and customized UI (XAML).

**Does Silverlight™ Streaming replace other video and file sharing services?**

This service does not replace other video or file sharing services. Silverlight™ Streaming is intended to allow developers the power and flexibility to build such services themselves, and more. Customers requiring additional capabilities are encouraged to contact a [Windows Media Streaming Hosting Provider](#).

**What steps are being taken to ensure copyrighted content or illegal material is not uploaded?**

The media owners as well as the Silverlight developers rendering content in their third-party Web sites are responsible for respecting the copyright of the content they expose. Microsoft reserves the right to remove from our servers any copyrighted content brought to our attention. The front page of our service has a Report Abuse button and the free-of-charge XAML templates that we provide include a similar button.

**Can I string together longer clips made of 10-minute files in a playlist?**

Yes, so long as the total size of your Silverlight application content does not exceed the 4-GB storage limits of this free service.

**What scalability does this system deliver? What if I have a large number of visitors to my Web site at the same time?**

This service uses the Microsoft content delivery network specifically optimized for long-form video content. It has a provisioned global aggregate data rate in excess of 1 terabit/second.

**Does this solution compete with content delivery networks (CDNs)?**

No, this service uses the Microsoft CDN for the low latency delivery of media, and it does not compete with CDNs because it is specifically optimized for use with Silverlight RIAs. Customers requiring additional capabilities including guaranteed service levels are encouraged to contact a [Windows Media Streaming Hosting Provider](#).



#### Why is this service branded with Windows Live™?

This service is part of the Windows Live™ Platform.

#### It's free—what's the catch?

There is no catch. This is a new offering designed to accelerate the development of the next generation of media rich applications.

#### Can I tap into other Windows Live™ services?

Yes, customers are able to use Windows Live ID™ and other Windows Live APIs today and in the future. Silverlight provides a great platform to consume these services.

#### Do you support digital rights management to protect my videos?

In the future, Silverlight™ Streaming will provide support for DRM-encoded video as an optional paid turnkey offering.

#### What applications will Microsoft provide to make hosting easy?

Microsoft is building a simple uploading tool and working to add publishing support directly to Silverlight™ Streaming via Expression™ Media Encoder, a feature of Expression™ Media. In addition, third-party companies are adding support to their own applications for Silverlight™ Streaming.

#### How is my content secured from unauthorized access?

You will have to be signed into the Silverlight™ Streaming service to manage your account and your Silverlight applications. Your Silverlight™ Streaming ID and secret key, associated to your Windows Live ID, will authenticate you as the unique and legitimate owner of the applications and content you upload to the service. You will also need this information to manage your Silverlight applications using the API. The Silverlight™ Streaming ID is public. However, the secret key should be kept confidential.

#### How do I get started?

To sign up for your free account, visit [streaming.live.com](http://streaming.live.com). Anyone with a Windows Live ID can participate.

#### Can Microsoft® Silverlight™ Streaming service be used for non-video files such as music files?

Yes. All valid content encoded for Silverlight applications and presented with Silverlight can be used. This includes music files such as WMA and MP3. Business logic written in JavaScript can also be stored and streamed to the client, and in the future, any Silverlight™ Dynamic Language Runtime assemblies can also be used.

#### How does the service stream content?

Content is streamed progressively using a progressive download mechanism today. Active streaming support using Windows Media Services is being considered based on customer feedback in the future.

#### Can I stream live content/events?

No, the service only supports on-demand content today. Customers requiring additional capabilities are encouraged to contact a [Windows Media Streaming Hosting Provider](#).

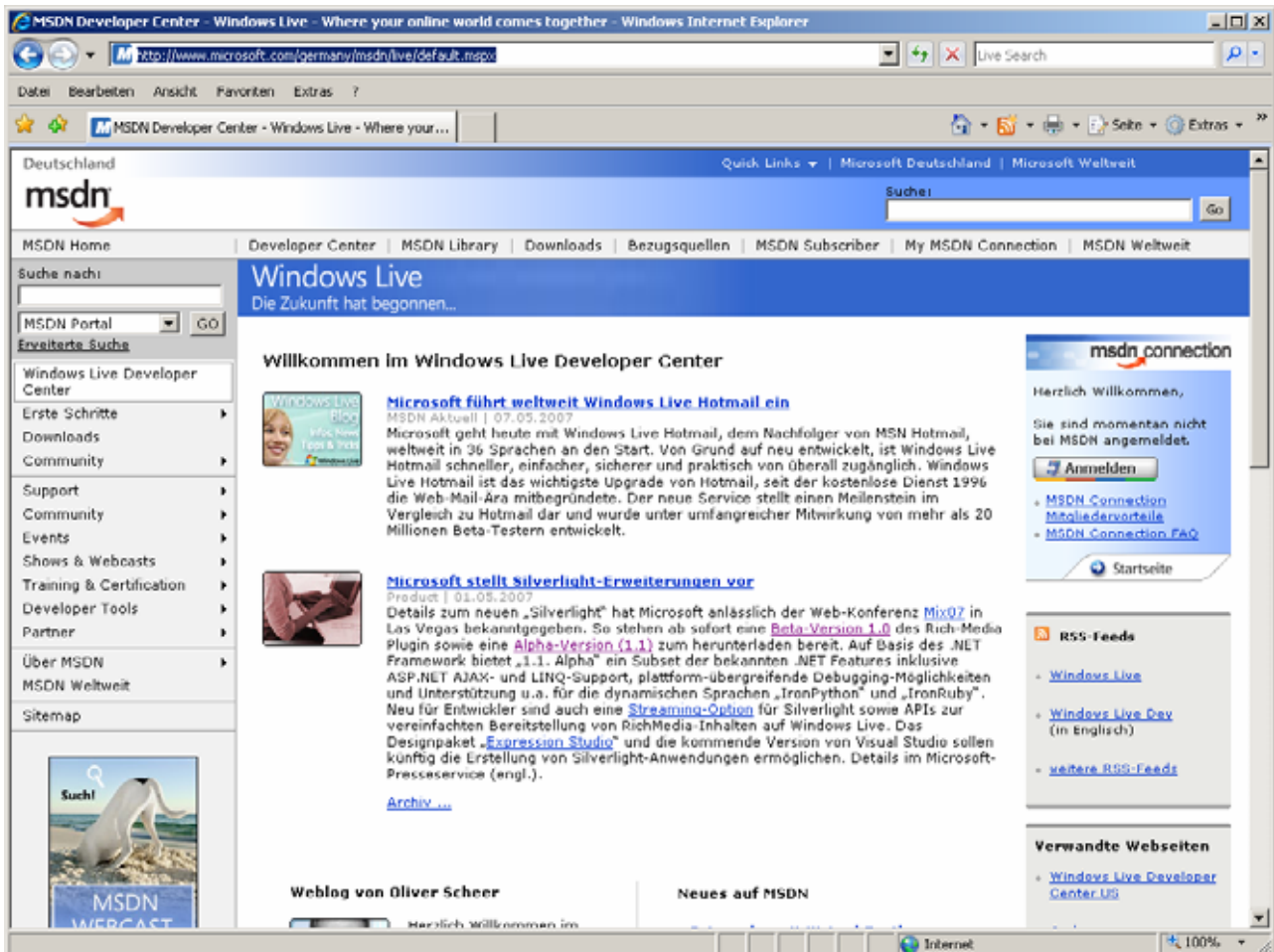
[Manage Your Profile](#)

**Microsoft** © 2007 Microsoft Corporation. All rights reserved. [Terms of Use](#) | [Trademarks](#) | [Privacy Statement](#)



## Microsoft Silverlight-Webseite auf microsoft.com/germany

<http://www.microsoft.com/germany/msdn/live/default.mspix>





<http://www.microsoft.com/germany/msdn/asp.net/default.msp>

The screenshot shows the MSDN Developer Center - ASP.NET page in a Windows Internet Explorer browser window. The address bar shows the URL <http://www.microsoft.com/germany/msdn/asp.net/default.msp>. The page is in German and features a search bar, navigation links, and several news items.

**MSDN Developer Center - ASP.NET**

Suche nach:

MSDN Portal GO

Erweiterte Suche

ASP.NET Home

ASP.NET Grundlagen

ASP.NET Steuerelemente

Web Forms

Support

Community

Events

Shows & Webcasts

Training & Certification

Developer Tools

Partner

Über MSDN

MSDN Weltweit

Sitemap

**Microsoft ASP.NET Developer Center**

**Final-Version: Internet Explorer Developer Toolbar**

Nach drei Beta-Vorläufern hat Microsoft die finale Version der „Internet Explorer Developer Toolbar“ bereit gestellt. Mit dieser kostenlosen Toolbar für den Internet Explorer 6 und 7 können u.a. Inhalte auf einer Webseite getestet oder angepasst werden; auf Knopfdruck sind auch Simulationen diverser Bildschirmauflösungen möglich. Weiterhin können Informationen zu HTML-Objekten (z.B. ID oder Class-Name), die Dateigröße und Maße von Bildern oder der direkte Link zum Speicherplatz abgefragt werden. Die Developer Toolbar kann darüber hinaus direkt HTML, CSS, WAI und RSS-Feed auf Standardkonformität testen. Weitere Details auf der englischsprachigen MSDN-Site.

**REMIX 2007 – Konferenz für Web-Entwickler, Designer und Business-Entscheider**

Entwickler, die Ende April keine Gelegenheit hatten, persönlich an der Web-Konferenz „MIX 2007“ in Las Vegas teilzunehmen, können jetzt im alten Europa Versäumtes nachholen: auf der REMIX 2007. Die kostenlose Veranstaltung informiert in zwei parallelen Tracks über neueste Trends beim Thema Web 2.0. Mit dabei sind internationale Top-Referenten, darunter Scott Guthrie, der mit seinem Team für die Entwicklung von IIS, ASP.NET AJAX oder den Visual Studio Tools für WPF verantwortlich ist. In Deutschland gibt's leider keinen REMIX-Termin, dafür aber Mitte Juni jeweils einen in Zürich und Amsterdam.

**Weblog von Oliver Scheer**

Hertzlich Willkommen im ASP.NET Developer Center. Mein Name ist Oliver Scheer und als Developer Evangelist bei Microsoft Deutschland beschäfige ich mich mit allen Themen rund

**Neues auf MSDN**

**\* Erste Schritte mit Silverlight**

White Paper | Laurence Moroney | 07.05.2007

Dieses Whitepaper bietet einen Überblick über Silverlight (ehemals „WPF/E“) und darüber, wie es sich in die Entwicklung der nächsten Generation von Webanwendungen einfügt.

**MSDN Library**

ASP.NET

Web Services

**MSDN Connection**

Hertzlich Willkommen, Sie sind momentan nicht bei MSDN angemeldet.

Anmelden

MSDN Connection Mitgliedervorteile

MSDN Connection FAQ

Startseite

**RSS-Feeds**

Web Development

Weitere RSS-Feeds

**MSDN Webcast**

Such!

MSDN WEBCAST



http://www.microsoft.com/germany/msdn/windowsvista/default.mspx

MSDN Developer Center - Windows Vista - Windows Internet Explorer

http://www.microsoft.com/germany/msdn/windowsvista/default.mspx

Suche nach:

MSDN Portal  [Erweiterte Suche](#)

Windows Vista Developer Center

Support  
Community  
Events  
Shows & Webcasts  
Training & Certification  
Developer Tools  
Partner  
Über MSDN  
MSDN Weltweit  
Sitemap

## Windows Vista

Sehen. Erleben. Einsetzen

**Buch-Tipp**  
Neues Fachkompendium vom MSDN-Experten Bernd Marquardt: in einem jetzt bei Microsoft Press erschienenen Crashkurs beschäftigt er sich mit Details des Benutzerschnittstellen-Frameworks "Microsoft Windows Presentation Foundation" (392 Seiten, 29,90 EUR). Mehr zum Titel in der Rubrik "Partner" der MSDN-Website.


**Microsoft stellt Silverlight-Erweiterungen vor**  
Details zum neuen „Silverlight“ hat Microsoft anlässlich der Web-Konferenz Mix07 in Las Vegas bekanntgegeben. So stehen ab sofort eine [Beta-Version 1.0](#) des Rich-Media Plugin sowie eine [Alpha-Version \(1.1\)](#) zum Herunterladen bereit. Auf Basis des .NET Framework bietet „1.1. Alpha“ ein Subset der bekannten .NET Features inklusive ASP.NET AJAX- und LINQ-Support, plattform-übergreifende Debugging-Möglichkeiten und Unterstützung u.a. für die dynamischen Sprachen „IronPython“ und „IronRuby“. Neu für Entwickler sind auch eine [Streaming-Option](#) für Silverlight sowie APIs zur vereinfachten Bereitstellung von RichMedia-Inhalten auf Windows Live. Das Designpaket „[Expression Studio](#)“ und die kommende Version von Visual Studio sollen künftig die Erstellung von Silverlight-Anwendungen ermöglichen. Details im Microsoft-Presseservice (engl.).

**msdn connection**  
Herzlich Willkommen,  
Sie sind momentan nicht bei MSDN angemeldet.  
  
• [MSDN Connection Mitgliedervorteile](#)  
• [MSDN Connection FAQ](#)

**RSS-Feeds**  
• [RSS-Feeds](#)

**MSDN Library**  
• [Windows Vista](#)

**Verwandte Themen**  
• [MSDN US Developer Center](#)

**Dirk Primbs Windows Vista Blog**  
  
Herzlich Willkommen im Vista Developer Center! Mein Name ist Dirk Primbs, ich bin Developer Evangelist bei Microsoft Deutschland mit Fokus auf Managed Code & den Technologien rund um die neue Windows-Version. Hier die aktuellsten Einträge zum Thema Vista und .NET Framework 3.0 aus [meinem Blog](#):

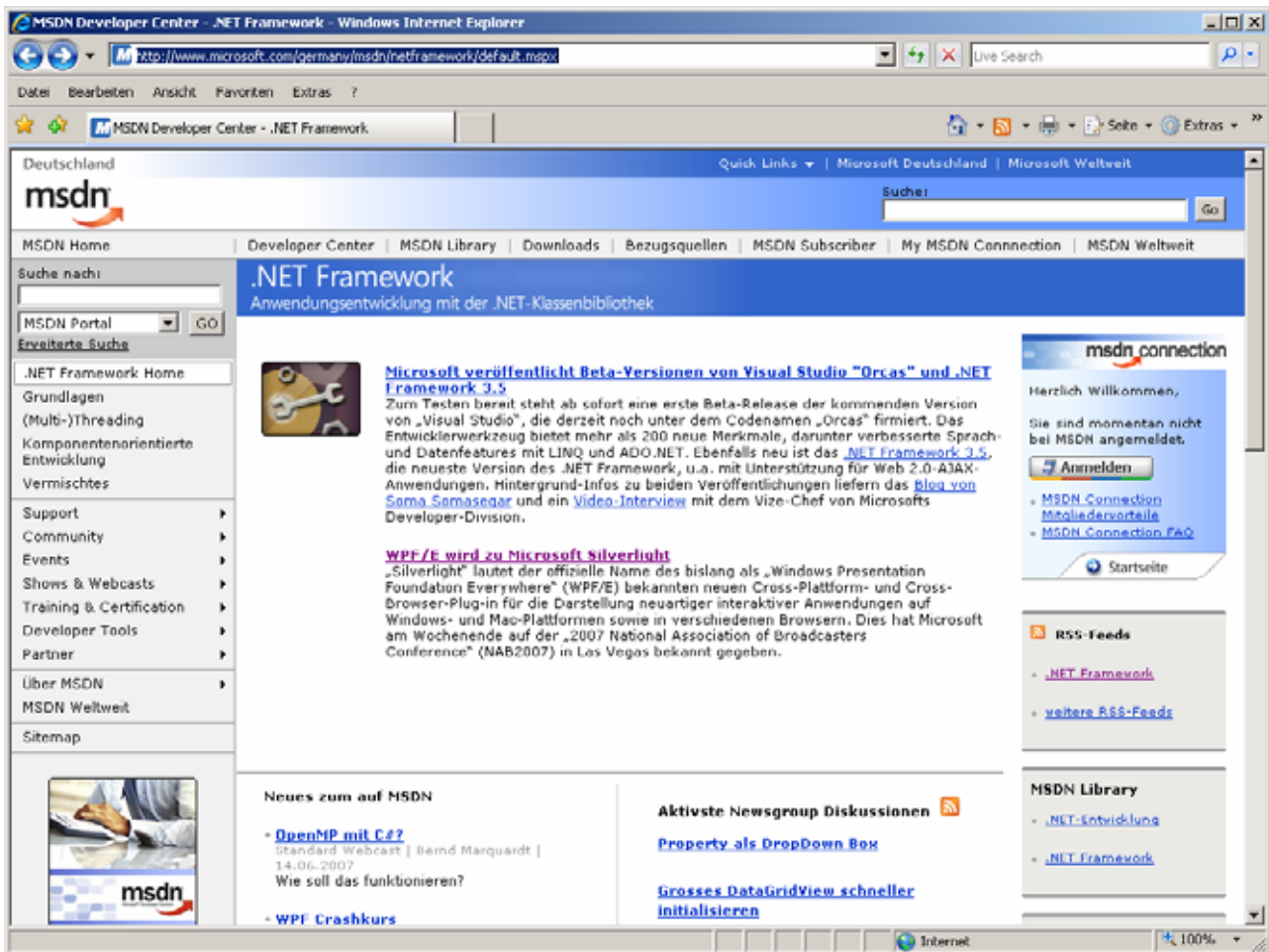
**Neues auf MSDN**  
• [Entwickeln für die Windows Vista-Shell](#)  
Technischer Artikel | Microsoft Corporation | 06.04.2007  
Dieser Artikel stellt Information zu Änderungen in verschiedenen Bereichen bereit.  
• [Willkommen bei der Pilotversion des MSDN Assistenten für vollautomatische Übersetzungen technischer Artikel über](#)

**MSDN WEBCAST FINDER**  
Ihr bequemer Zugang zu

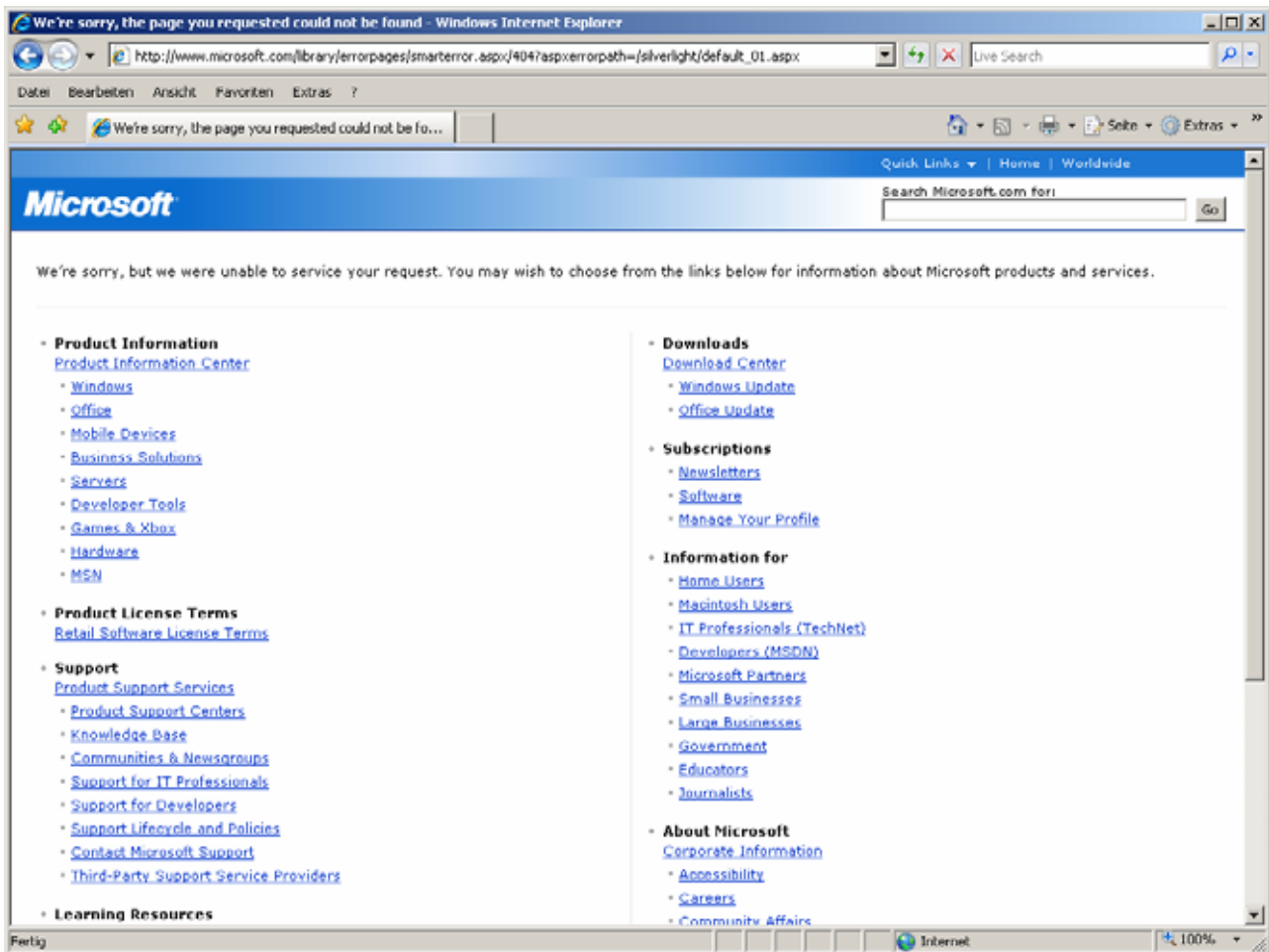
Internet 100%



<http://www.microsoft.com/germany/msdn/netframework/default.mspx>



[http://www.microsoft.com/silverlight/default\\_01.aspx](http://www.microsoft.com/silverlight/default_01.aspx)



### **Microsoft Expression-Produkte-Webseite auf microsoft.com**

Nachfolgende Bilder wurden in der Farbe unverändert übernommen. Wer also Bildinhalte wegen Farbproblemen nicht erkennen kann, bedenke, dass Microsoft Profi ist und daher einwandfreies Sehvermögen auch von seinen Kunden erwarten kann \*fg.

Man beachte, dass Microsoft Produkte anbietet, zu denen keine Informationen abrufbar sind: Siehe Bild How To Buy Expressions Products. Dafür wird aber Werbung für Wiederverkäufer gemacht - ohne Link auf Wiederverkäufer.

Man beachte, dass Silverlight mit Expressions-Produkten programmiert werden kann und daher als Link auftaucht.

<http://www.microsoft.com/expression/default.aspx>

