

Silverlight[™] with AxCMS.net[®]

Abstract / Summary

Silverlight is a new Microsoft technology aiming for increasing interactivity of the web applications, increasing the power of the client, at the same time avoiding the typical problems of a thick client. Silverlight enables developers to create rich GUI with animations and video support regardless of the browser using all the familiar features, languages and tools of the .NET framework.



AxCMS.net supports Silverlight in many ways:

- You can develop Silverlight modules as a part of your application (and consume the data from AxCMS.net or any other sources).
- You can publish the same content you already have in AxCMS.net in a Silverlight channel.
- To help you mastering this technique Axinom offers a Silverlight Template Project – a Visual Studio solution in source code demonstrating how to create a Silverlight channel and how to skin your dynamic controls to use Silverlight.
- AxCMS.net itself starting from the version 9 (released on 9.9.2009) uses Silverlight to provide editors with a new user experience in management system.

History

Before we touch Silverlight, let us explain the context in which it appeared.

We postulate that the whole development of information technologies can be seen as a pendulum between two extremes: everything on the client and everything on the server.

In the **mainframe** era there was no question: everything runs on the server (“computer”) and the users are working with a dumb terminal. Such terminals evolved from a simple typewriter to keyboards+alphanumeric display to a graphical station with flat screen and a mouse – which all does not influence the separation of the roles.

A revolution occurred with the appearance of the **personal computers** (let’s count from the beginning of 80s – first IBM PC). For the first time “big” computational power was available on a desk. In this euphoria numerous applications for desktops appeared followed by a rapid operation systems evolution: MS DOS, MS Windows and by Microsoft growing and soon dominating the market.

Very soon it was evident, that personal computers should be connected in networks, otherwise their usefulness is quite limited. As LAN-technologies evolved, so did the software-architectures and eventually it came to the **client-server** architecture. Here all computers are connected through a network, and the responsibilities of an application are distributed between the clients and the servers. Whereas data storage was a clear server responsibility and presenting a GUI to the user – typical client responsibility, the application’ business logic was moving around between the client and the server depending on an application architecture, development tool and current philosophy. It is there as the terms like three-tier-architecture appeared. Client-server architecture had a lot of advantages – it leveraged the power of a computer network and was flexible. The disadvantages were – lack of standards, high administrative costs (deployment and maintenance for the clients), incompatible

development tools – steep learning curve, traffic congestion on a network (depending on the protocols used).

The next milestone is the birth of **WWW** (World Wide Web), let's date it back to 1993, as Mosaic – first graphical web browser – appeared. Technically the Web is using client-server architecture – with clear responsibilities of the server and the client and clear standards: **HTTP** for data exchange between the server and the client and **HTML** – a markup language used to describe the user interface. It gave a birth to a standard client – a **browser**, understanding HTML and allowing navigating to any web server (and the hypertext plays also a role). The metaphor was very successful – we witnessed an explosion of the number of web-sites in mid-90s. The browser-client turned out to be relative dumb – again. HTTP was hardly changed over the last 20 years: from 0.9 to 1.0 to 1.1. At the same time HTML undergo dramatic changes, which started immediately after the first browsers where around and which were used in the browser-wars of mid-90s. All about expression power of the user interface. HTML evolved from a simple text-markup to an instrument to describe challenging graphical user interface.

As the web servers were clearly separated from the browsers with HTML/HTTP standards, they started their own evolution and offered better and better development environments – from CGI-application through scripting languages to full-featured development platforms, like Microsoft .NET. At some point alongside with the web servers delivering HTML for the browsers appeared **web services**, delivering XML-data to other applications, which enabled a lot of new integration scenarios.

The advantages of a light-weight client (e.g. no client administration and high standardization) came at a price of relative poor user experience. All over the times attempts were made to improve it. One way was – to extend HTML. Another way – to enable client-side scripting (**JavaScript**). It does open possibilities for richer user interactions, but as every browser has its own JavaScript implementation, it makes very difficult to develop JavaScript applications. And yet another way – to create browser-plug-ins for special kinds of content. Every major browser provides an API for such plug-ins. Plug-ins are widely used to play videos and animations as well as show 3D-graphics (like VRML or 3DMLW). Plug-ins – of course – enable new wonderful possibilities, but they must be installed by every user for the first time, get updated as new versions are released, and published applications have to consider the versions of the plug-ins installed – the same issues which apply for the desktop applications.

An exceptional position among all plug-ins has **Adobe Flash**, used for adding animations and interactivity to web pages. Flash can manipulate vector and raster graphics, supports streaming of audio and video and contains a scripting language called ActionScript. Due to its big install base Flash is considered “almost” a standard in Internet.

Recently one more technique to increase the interactivity of the web pages appeared – **AJAX** (stays for asynchronous JavaScript + XML). It relies on JavaScript, web services and DHTML. Pieces of JavaScript run on the client (in browser) and make web service calls in a background on their own. They render the returned data manipulating the HTML code of the page in browser directly. AJAX brings some advantages – it avoids the need to reload the whole page with a new server request on every click and achieves better response times, better user experience, higher level of interactivity. AJAX drawbacks are – much more complex and error-prone development and a gap to the accepted practices in web (bookmarking, browser history, deep links, visibility for search engines – all these should be reinvented with AJAX). Besides that, AJAX-application depends heavily on JavaScript implementation in a particular browser (and of course JavaScript is not allowed to be turned off by the end user).

What is Silverlight?

As an official definition from silverlight.net says, “Microsoft Silverlight is a free framework empowering you to build new types of applications for the Web regardless of target platform or browser, using all the familiar features, languages and tools of the .NET framework.”

Silverlight aims for increasing interactivity of the web applications and for this purpose increasing the power of the client, at the same time trying to avoid the typical problems of a thick client.

First of all, instead of the HTML- and JavaScript-based programming model, a full-featured programming environment is used. The rich user interface includes much interaction, animations, incl. 3D-animations, video, etc. It is described with XAML – a declarative XML-language (same way as in WPF – Windows Presentation Foundation). For Designers full featured tools (Expression-family) are available. Designers can work independently on the same project with developers; their results are stored as XAML-files. Developers keep using Visual Studio – it supports now Silverlight development. Programming is accomplished with a subset of .NET Framework with usual .NET languages, like C#. It means: .NET developers, who are get used to ASP.NET and/or WPF will find it easy to switch to Silverlight.

Silverlight applications can easily consume web services, so they can everything what AJAX can and better. Silverlight application can interact on the browser side with HTML-DOM / JavaScript.

A Silverlight-application is loaded to the client (as a part of a host HTML-page) and is executed in a local .NET runtime environment. The runtime environment for Silverlight is available as a browser plug-in (small size) for the major browsers on Windows, Mac OS 10 and Linux (Moonlight). The plug-in provides a sandbox for client applications, mitigating the problem of different browser implementations of e.g. JavaScript. There exists even an out-of-browser version of Silverlight allowing creating standalone applications with Silverlight for offline scenarios.

Silverlight can out of the box play videos in WMV/VC-1/H.264 format, supports HD video, PlayReady-DRM and different streaming methods, like Progressive Download and Adaptive Streaming.

Silverlight provides functionality which Adobe Flash also provides, but fits much better into the Microsoft Software Development world.

Silverlight Showcases

You can find many Silverlight showcases at <http://silverlight.net/showcase/>. It shows the new kind of user interactions possible in the modern web.

AxCMS.net goes Silverlight

AxCMS.net supports Silverlight in many ways.

Adding Silverlight to your applications

If you develop your own application build on AxCMS.net, you can use Silverlight in it, like if it were any other ASP.NET application. You implement functionality you wish as a Silverlight module and add to your AxCMS.net application either as a template or as a custom control. If your module needs data (and it probably does), it can be exposed as a web service. In the example below you see a dynamic page list rendered with Silverlight. It takes advantages of Silverlight animations for better user experience. The data for the list is delivered via XML from a Web Service specially created for this control. It exposes the same data, which is usually used in a dynamic page list.



Fig. 1 A video-player in AxCMS.net programmed in Silverlight and using a documented uploaded into AxCMS.net as a source

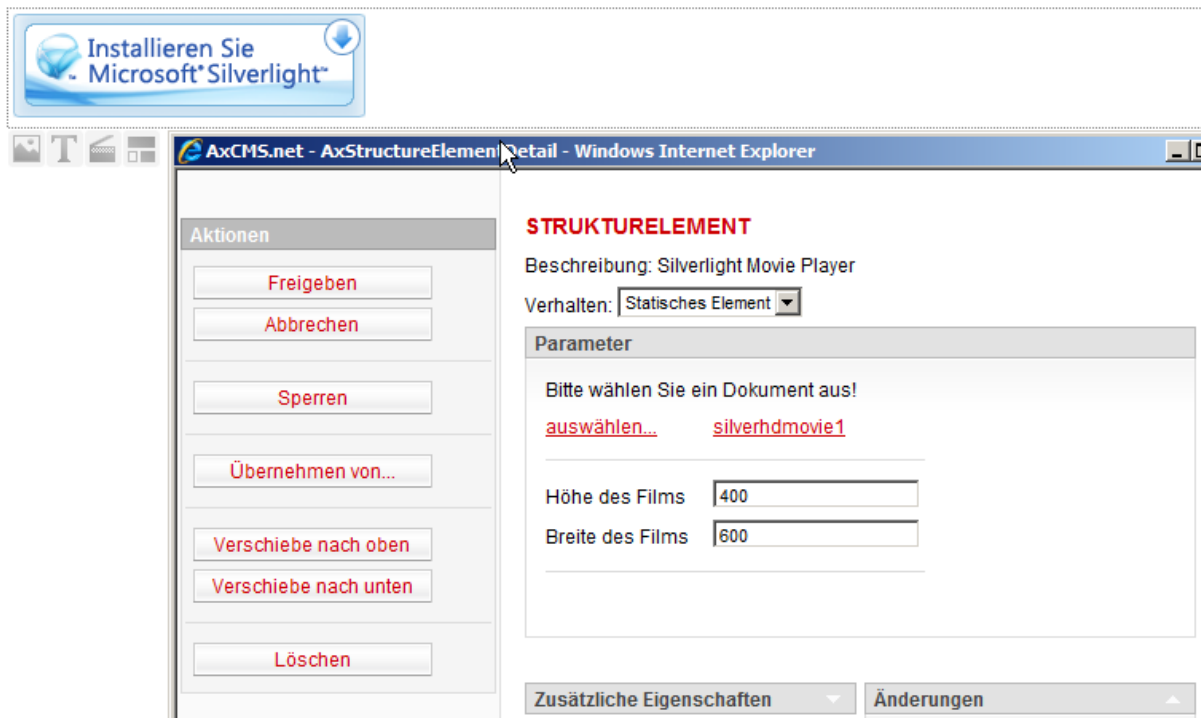


Fig. 2 The same control being edited in Management System

It is also possible to develop the whole GUI of your application in Silverlight with AxCMS.net as backend providing an administration interface and web services for the Silverlight frontend.

Publishing your content in Silverlight

Silverlight applications have a nice look & feel, but how you can manage content, shown in them?

You can combine the advantages of Silverlight with the power of AxCMS.net and present the content you already have in AxCMS.net through a Silverlight **output channel**.

You will need two versions of your template: a usual one in ASP.NET and another one in Silverlight. AxCMS.net publishes those two versions of your web site for you: one in HTML and one in XAML – for Silverlight. To do this you take advantage of AxCMS.net Channels and Rendering Engines.



Fig. 3 An AxCMS.net page rendered for Silverlight output channel in the Management System

Silverlight Template Project

To simplify this task for you, we offer a **Silverlight Template Project** – a template project for AxCMS.net delivered as source code (Visual Studio solution) demonstrating how to render AxCMS.net content in Silverlight.

Silverlight Template Project defines a Silverlight channel, which converts standard AxCMS.net controls (texts, images, custom controls) to XAML. These XAML files are then loaded by a special Silverlight control that can display them.

The project's look & feel resembles the standard AxCMS.net Template Project with a Silverlight version added – same navigation, same pages, same texts and pictures, couple of structural controls, static control and some dynamic controls. We added the Silverlight version to the following controls:

- Dynamic: List control, Image Rotator control
- Static: Site Map
- Structural: image left, image right.

Silverlight version of controls has some moderate Silverlight effects - to show the benefits of Silverlight.

The real application you develop on top of the Silverlight Template Project should make much more use of Silverlight – otherwise the power of Silverlight remains unleveraged.

```

1  <?xml version="1.0" encoding="utf-8" standalone="yes"?>
2  <UserControl
3      xmlns="http://schemas.microsoft.com/client/2007"
4      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
5      xmlns:custom="clr-namespace:SilverlightDemo;assembly=SilverlightDemo">
6      <Grid Width="Auto"
7          Margin="0,0,0,0"
8          VerticalAlignment="Stretch"
9          HorizontalAlignment="Stretch">
10         <Grid.ColumnDefinitions>
11             <ColumnDefinition Width="570" />
12             <ColumnDefinition Width="200" />
13         </Grid.ColumnDefinitions>
14         <Grid.RowDefinitions>
15             <RowDefinition MinHeight="600" Height="*" />
16         </Grid.RowDefinitions>
17         <ScrollViewer x:Name="scrollView"
18             MinHeight="559"
19             HorizontalScrollBarVisibility="Disabled"
20             VerticalScrollBarVisibility="Auto"
21             BorderThickness="0" Margin="0,32,0,0">
22             <Grid Grid.ColumnSpan="2" Grid.Row="0" Grid.Column="0"
23                 Margin="0,0,0,0"
24                 VerticalAlignment="Stretch"
25                 HorizontalAlignment="Stretch">
26                 <StackPanel Grid.Row="0" Grid.Column="0">
27                     <Axinon:AxContentPlaceholder id="basePlaceholder"/>
28                 </StackPanel>
29             </Grid>
30         </ScrollViewer>
31         <!-- rightPlaceholder -->
32         <Grid x:Name="rightPlaceholder"
33             HorizontalAlignment="Right" Margin="0,5,0,0" Width="183"
34             Grid.Row="0" Grid.Column="1">
35             <Grid VerticalAlignment="Top">
36                 <Axinon:AxContentPlaceholder id="rightPlaceholder"/>
37             </Grid>
38         </Grid>
39     </Grid>
40 </UserControl>

```

Fig. 4 AxCMS.net Page Template in XAML format (from Silverlight Template Project)

There is one “side effect” of Silverlight Template Project – you can see in practice how to create new channels and rendering engines and this enables you to create even more channels besides Silverlight. Maybe you wish a version for mobile devices or an XML version for integration with your other applications? Multi channel AxCMS.net architecture has many usages.

Silverlight in AxCMS.net Management System

In AxCMS.net version 9.0 we have enhanced AxCMS.net Management System with a new user interface based on Silverlight.

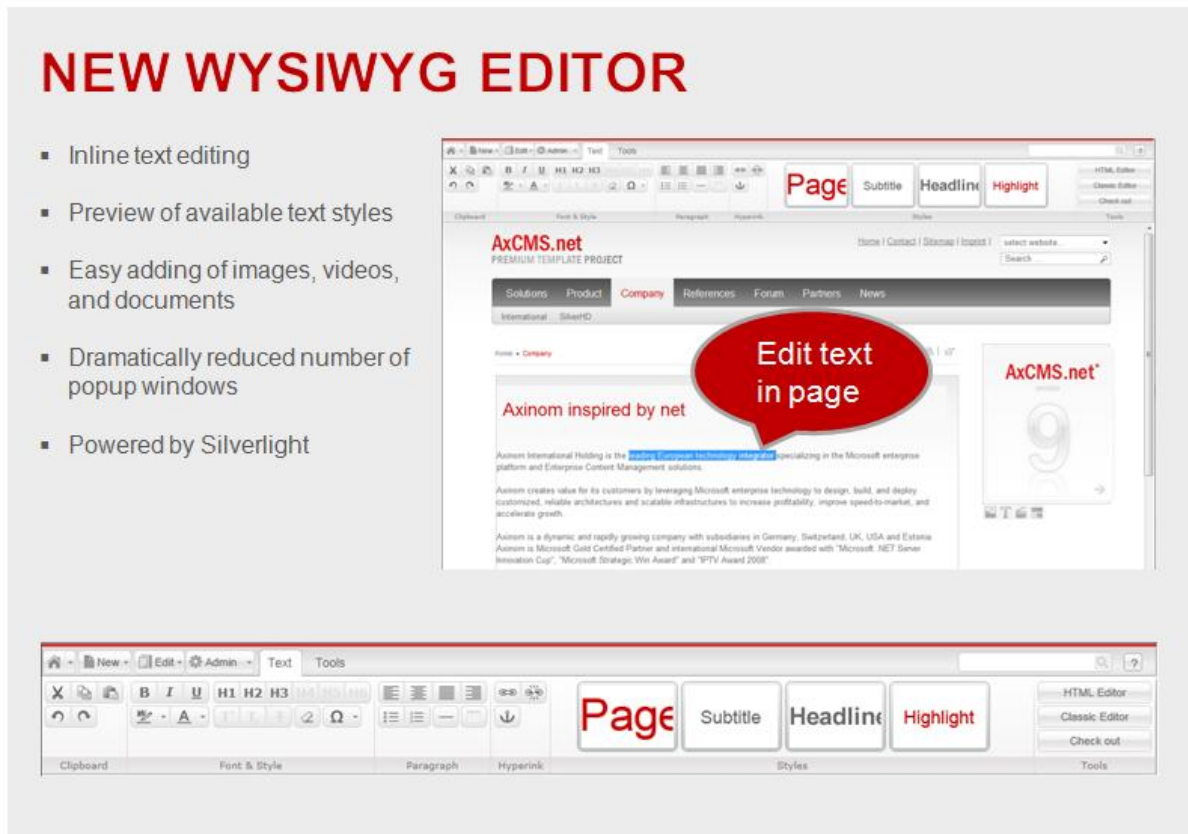


Fig. 5 AxCMS.net 9 – user interface based on Silverlight

AxCMS.net 9 is available since 9.9.2009, partner release AxCMS.net 9.1 – since January, 2010.

SILVERHD

Axinom's SILVERHD is a set of cutting edge products for content owners focused on publishing high definition (HD) images, audio and video content (optionally protected by DRM) over the internet. SILVERHD provides significant economic benefits through higher advertising revenues and increased audience engagement. SILVERHD combines the power of Microsoft Silverlight Technology with the latest Axinom Web Platform AxCMS.net to provide the easy, scalable deployment and brilliant user experience.

SILVERHD Portal

SILVERHD Portal enables companies to manage and share and end-users to explore high-definition (HD) images, audio and video content in a new and interactive way.

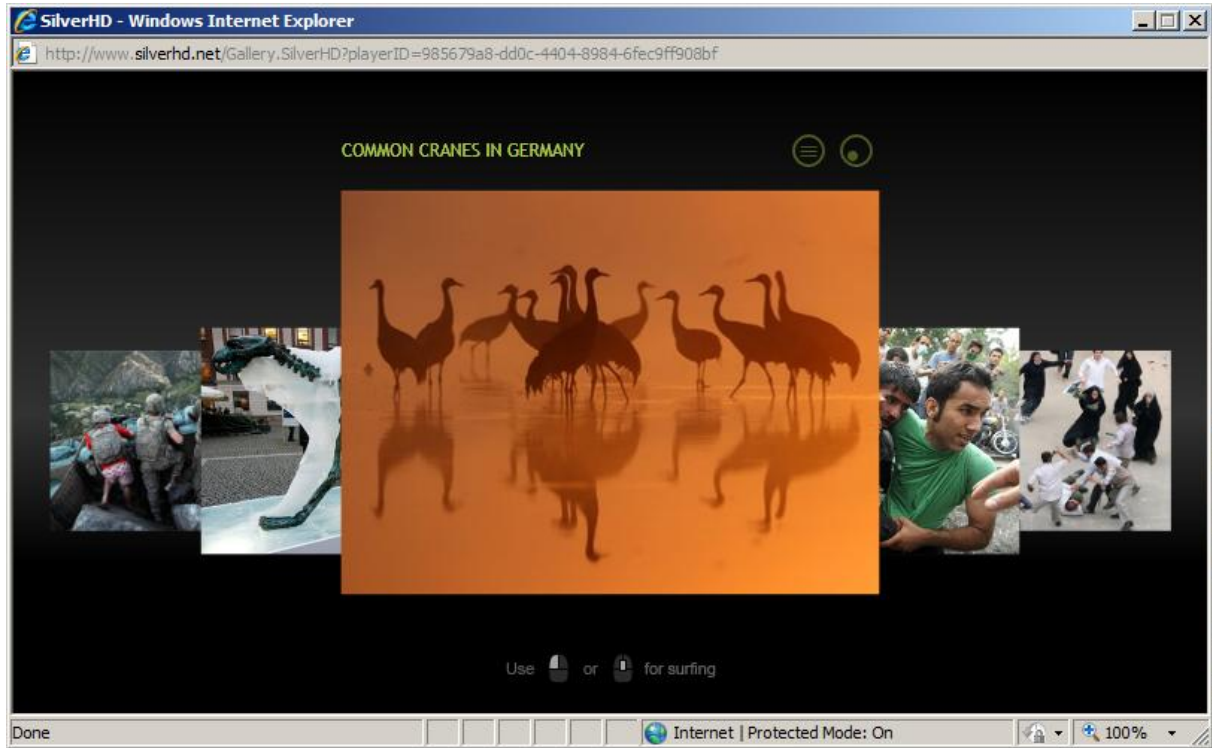


Fig. 6 A Silverlight Image Gallery on SilverHD-Portal

SILVERHD Viewer

SILVERHD Viewer presents pictures in high resolution fast and without waiting in an interactive gallery in your web browser. SILVERHD Viewer relies on Silverlight **DeepZoom** technology.

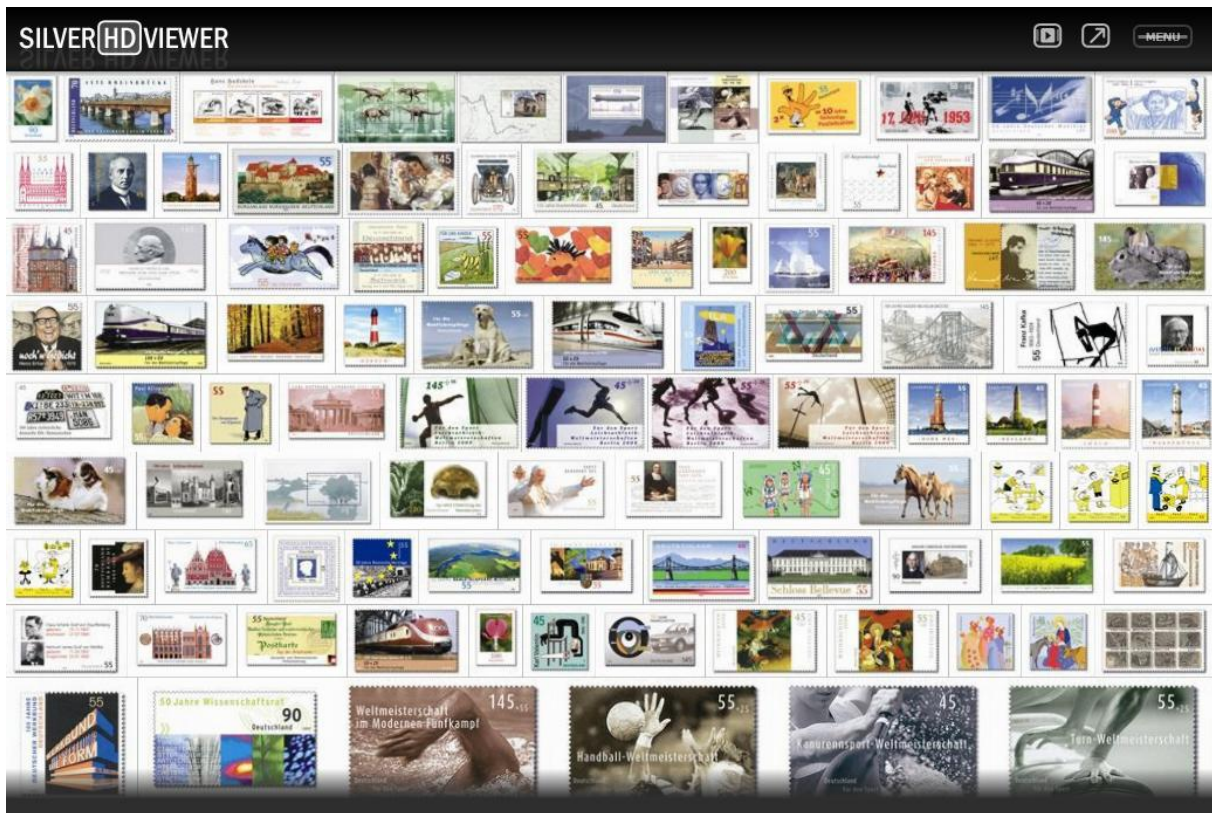


Fig. 7 SILVERHD Viewer – Post Stamps (created for: Deutsche Post AG)

SILVERHD Player and SILVERHD Player DRM

SILVERHD Player is an interactive Video Player based on Silverlight. With SILVERHD Player you can view videos, even High Definition videos, using streaming or progressive download – comfortable and platform independent.

For corporate video applications SILVERHD Player supports Digital Rights Management (DRM) using Microsoft PlayReady technology.



Fig. 8 SILVERHD Player

SILVERHD Geo

SILVERHD Geo is an interactive geographical presentation of your content. Usage of Microsoft technologies Silverlight and "Bing Maps for Enterprise" results in an innovative presentation and possibilities of user-interaction.



Fig. 9 SILVERHD GEO

References

- Official Silverlight Site
<http://www.silverlight.net>
- Silverlight Template Project – developer information
http://en.axcms.net/en_axcms_template_silverlight.AxCMS
- SILVERHD Portal
<http://www.silverhd.net>



Axiom International Holding is the leading European technology integrator specializing in the Microsoft enterprise platform and Enterprise Content Management solutions. Axiom creates value for its customers by leveraging Microsoft enterprise technology to design, build, and deploy customized, reliable architectures and scalable infrastructures. With AxCMS.net[®] Axiom provides a completely .NET based Enterprise Content Management System completely free of license costs. AxCMS.net[®] won the "Microsoft .NET Server Innovation Cup,, is "Certified for Microsoft Windows Server 2003" and is the base for the worldwide AxCMS.net[®] Solution Provider Program.