

# **Silverlight for Windows Embedded Compact 7 in i.MX53 development platform**

## **Abstract**

Silverlight for Windows Embedded is a native (C++) UI development framework for Windows Embedded Compact devices that is based on Microsoft Silverlight 3. Silverlight for Windows Embedded is interoperable with Microsoft Expression Blend 3. It provides tools to generate template application code that is based on Expression Blend 3 XAML projects.

## **Overview**

Silverlight for Windows Embedded provides helper template methods that automate safe implicit-type conversion of generic object types. The programming model in Silverlight for Windows Embedded provides a visual host that hosts a Silverlight visual tree in a Microsoft Win32 window. To handle UI events, you can attach C++ delegates to UI objects through methods that are available on various classes in the C++ API.

With Silverlight for Windows Embedded, you can create a Silverlight application for an embedded device, create and add custom user controls to your application, add Win32 controls to your application, and support hardware-accelerated graphics in the OS.

Silverlight provides a retained mode graphics system similar to Windows Presentation Foundation (WPF) and integrates multimedia, graphics, animations and interactivity into a single runtime environment.

In Silverlight applications, user interfaces are declared in Extensible Application Markup Language (XAML) and programmed using a subset of the .NET Framework. XAML can be used for marking up the vector graphics and animations. Silverlight can also be used to create Windows Sidebar gadgets for Windows Vista.

Silverlight supports H.264 video, Windows Media Video (WMV) without using media player provided by the Windows.

A Silverlight project contains the Silverlight.js and CreateSilverlight.js files which initializes the Silverlight plug-in for use in HTML pages, a XAML file for the UI and code-behind files for the application code.

Silverlight applications are debugged in a manner similar to ASP.NET applications. Visual Studio's CLR Remote Cross Platform Debugging feature can be used to debug Silverlight applications running on a different platform as well.

**Microsoft Silverlight history includes the following versions:**

**Silverlight 1:** It was released in 2007 and included the features of user interface interactivity, basic UI controls, graphics and animation, media playback etc.

**Silverlight 2:** It was released in 2008 and included .NET framework. It was similar to Silverlight 1 except the .NET inclusion.

**Silverlight 3:** The Beta version of this was released in 2009 and included more controls.

**Silverlight 4:** The final version of this was released in 2010. One new feature included was Google Chrome browser, webcam and microphone, more mouse support, new notification support to send messages to users etc.

**Silverlight 5:** This was released in 2011. New features include GPU accelerated video decoding, 3D graphics, playback speed controls and so on.

Silverlight for Windows Embedded is a native (C++) UI development framework for Windows Embedded Compact devices that is based on Microsoft Silverlight 3. You can use Silverlight to do the following:

- Define visual UIs for embedded applications in XAML.
- Collaborate with UI designers by using XAML projects.
- Separate C++ programming logic and UI design.
- Add, modify, and customize the UI at run time.
- Create interactive multimedia UIs for embedded devices.

Silverlight for Windows Embedded Compact 7 is compatible with Silverlight 3 XAML and provides a set of equivalent classes for supported XAML elements.

## Glossary

Acronyms	Description
2-D	2-Dimensional
3-D	3-Dimensional
API	Application Programming Interface
LCD	Liquid Crystal Display
UI	User Interface
WinCE 7	Windows Embedded Compact 7
XAML	Extensible Application Markup Language
XML	Extensible Markup Language

## Reference

Silverlight for Windows Embedded:

<http://msdn.microsoft.com/en-us/library/ee502198.aspx>

## Features and Benefits of Silverlight in Windows Embedded Compact 7

### It provides the following features:

- 1) Silverlight provides the API's that integrate with the Windows Embedded operating system. So it helps shorten the study span for the developers who are familiar in developing windows embedded compact.
- 2) C++ classes that developers use for the development can use the same to create the best UI elements.
- 3) Supported for advanced features like gradients, animations etc so that developers can provide interactive control UI's in their embedded applications.
- 4) Compatibility with Silverlight 3 XAML and a set of equivalent classes for supported XAML elements for developers and designers who are familiar with using Silverlight 3 XAML.

- 5) Run-time support for displaying XAML UIs so that if they prefer, OEMs can design UIs for applications that are entirely in XAML.
- 6) Ability to dynamically change the UI at run time by using C++.
- 7) Separation of programming logic and UI design to encourage XAML designers to focus on designing user experiences; and to help developers to focus on integration, programming logic, and run-time behavior.

## Supported Silverlight UI Features

Silverlight for Windows Embedded introduces a subset of Silverlight 3 features to embedded devices, including the following UI features:

- **Advanced Graphics:** Graphics capabilities include brushes that produce gradient multicolor blend effects; image brushes that can paint the interior of UI elements; and *transforms*, which are used to rotate, scale, translate, or skew UI elements. Three-dimensional (3-D) graphics capabilities include 3-D transforms, which transform UI elements in three-dimensional space.
- **Layout System:** The layout system handles the tasks of drawing, resizing, and positioning the UI elements in the graphical window. This built-in functionality removes from the developer the responsibility of drawing and sizing elements on-screen by using C++ code. Silverlight supports layout container objects that work with the layout system to manage the arrangement of UI elements on-screen at run time.
- **Animation, Timers, and Storyboards:** With animation storyboards, UI elements can appear to dynamically change or move on the screen. This effect is achieved by quickly cycling through a series of images, or *key frames*, each slightly different from the previous one, over a specified duration of time. This visual effect can be achieved by defining each key-frame C++ object and implementing an animation storyboard.
- **Pixel Effects:** Pixel effects modify the appearance of a UI element, for example, by blurring the appearance of an element or adding a drop shadow. Pixel effects usually require hardware acceleration so that the graphics processing is done as fast as possible.

- **Text and Typography:** You can use both text controls and typography to display text in a Silverlight-based application. You can customize the display of text and provide unique customizations by changing visual properties or layout, or by applying transforms to the text. With built-in support for the XML Paper Specification (XPS), you can also use predefined glyphs in the Silverlight UI. A glyph is a rendered image that is a visual representation of a character in a font.
- **New Controls:** Silverlight for Windows Embedded introduces new Silverlight controls to Windows Embedded Compact developers, such as user controls, content controls, grids, canvases, paths, rectangles, and password boxes, which each inherit from the intermediate base class `IXRFrameworkElement`. Silverlight also provides intermediate base classes that are specifically intended for creating custom controls.
- **Styles and Templates:** Styles and templates collectively define the pieces that make up the appearance of a control and that provide the default visual behavior of the control. You can apply styles and templates to controls to define a consistent look for specific types of controls in your application.
- **Geometry:** Geometry objects can be used for rendering two-dimensional (2-D) graphic data on-screen.
- **Triggers and Events:** Triggers let developers define a custom visual behavior that is demonstrated in response to the **Loaded** event. Additionally, storyboard animations that show visual behavior can be started from within event-handler code. Custom visual behaviors can include changing the color, shape, or size of an object, or playing a short animation storyboard.

### **Expression Blend 3 a tool to create UI**

Microsoft Expression Blend 3 and Silverlight for Windows Embedded, combined with Microsoft Visual Studio, offer an end-to-end development environment.

Windows Embedded Compact 7 device designers and developers can use this environment to create first-class, cutting-edge application interfaces in a wide range of device categories. Expression Blend gives designers a wide variety of visual tools to use to create dynamic Silverlight for Windows Embedded applications.

Visual Studio gives developers a complete set of tools for modifying the native C++ code needed to drive embedded applications.

Designers and developers can efficiently work on a project in parallel using Expression Blend and Visual Studio respectively. Designers use the Expression Blend tool set to create a dynamic and compelling XAML UI design, while developers implement the corresponding business logic using C++ in Visual Studio.

A designer works on the appearance and interaction of an application by drawing and modifying objects on the design surface in Expression Blend.

A developer implements the underlying logic that interacts with the elements defined by the XAML files, using Visual Studio projects that contain C++ code files and resources derived for the XAML.

The XAML elements are attached to events in the C++ code files to complete the application. Each change the designer makes automatically modifies the underlying XAML and managed code files in the Expression Blend project.

The developer periodically imports the XAML changes and additions into Visual Studio project using Windows Embedded Silverlight Tools.

### **Sample Silverlight application:**

One sample Silverlight application that we developed for iWave i.MX53 WinCE 7 platform is as follows:

iWave i.MX53 is a development platform based on Freescale's i.MX53 processor with ARM cortex A8 core built with multimedia in mind. It is a development platform and it can be used for the development of applications. iWave has successfully ported WinCE 7 in i.MX53 board. Silverlight is an add-on tool which comes with the WinCE 7 that iWave has ported into i.MX53 board.

Here we go with an application that is developed for i.MX53 board. Only the snap shots of the result of the application have only been put here.

- 1) This application has an image of i.MX53 board and iWave's official logo and three Silverlight buttons.
- 2) Expression Blend 3 is used for the development.
- 3) When the application is deployed in the i.MX53 board, the LCD of the platform looks like the following snap shot.



There is a iWave official logo in the middle and two buttons in the either end of the main image and a button in the top right corner.

- 4) The logo will disappear when the right button is pressed .And the logo will appear back when the left button is pressed. Following screen shots make the concept clearer.







5) And the application exits when the top right corner button is pressed.

This application shows the working of the silverlight application in iWave i.MX53 development platform.

- Seetharam Nayak  
Member-Technical  
[seetharamn@iwavesystems.com](mailto:seetharamn@iwavesystems.com)