

Silverlight 3: CoreCLR+Small BCL

Concepts, Assemblies, Types, Reflection, Attributes, Threading, Garbage Collection, System.IO, Isolated Storage, Security

When Flash or Javascript developers first move to Silverlight, one of the things that amazes them is the richness of the underlying runtime, the CoreCLR. The .NET Common Language Runtime (CLR) is the foundation for all aspects of the .NET initiative. The CoreCLR is a stripped down version for Silverlight.

The CoreCLR provides a “managed” environment, in the sense that a VM-like entity, separate from your application, is assisting with its execution. Its modern feature-set includes a rich object model, garbage collection, granular security, generic collections, innovative metadata, sophisticated type loading, thread management, and virtualized contracts

between types (so that their physical layouts may be defined and optimised at runtime). All these features are exposed to your application code via the Small BCL (Base Class Library – System.*). To build an optimized Silverlight app, you must understand how it can leverage what CoreCLR + SmallBCL offers.

Silverlight is becoming the platform of choice for new RIA projects. It is highly regarded and is gaining a reputation as a solid foundation for innovative solutions. Every Silverlight developer needs to appreciate what the CoreCLR and Small BCL are. Taking this course is a vital step in achieving this goal.

<i>Learning Topics</i>	
<p>Target Audience: software engineers who need to rapidly get up to speed with CoreCLR programming.</p> <p>Prerequisites: Participants need general understanding of .NET concepts and a working knowledge of C#.</p> <p>Participants should have completed Clipcode's “Silverlight 3: Fundamentals” course or have equivalent experience.</p> <p>Little or no previous CLR or CoreCLR experience needed.</p>	<div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p style="text-align: center;">CoreCLR Architecture</p> <p>Feature-rich runtime for many languages What services does it provide? Intermediate Language (IL) Relationship between CLR and CoreCLR Small BCL and full BCL Functionality of Small BCL (System.*)</p> <p style="text-align: center;">Assemblies</p> <p>An assembly is a (DLL-like) delivery & management unit for CLR types Strong names Versioning Modules and assemblies Locating assemblies</p> <p style="text-align: center;">Types</p> <p>Importance of type CLS & CTS Types and different languages System.Object & Identity System.Type Type features</p> <p style="text-align: center;">Reflection</p> <p>Deeper look at what an assembly is What data is stored along with code Manifest & metadata tables Peeking inside an assembly Discovering types Dynamically activating an instance</p> <p style="text-align: center;">Custom Attributes</p> <p>Need for custom metadata Advantages of using attributes in IL Pseudo-custom attributes Detecting attributes Creating attributes</p> </div> <div style="width: 48%;"> <p style="text-align: center;">Garbage Collection</p> <p>Memory management in the browser Differences between CLR/CoreCLR IDisposable Weak references Resurrection No induced collections</p> <p style="text-align: center;">Threading</p> <p>Working with threads How to create and manage threads Synchronization Importance of UI thread for UI work Timers Application Services (long running functionality with no UI)</p> <p style="text-align: center;">System.IO</p> <p>IO architecture File systems in a browser context Programming with streams System.Text and encodings MemoryStream</p> <p style="text-align: center;">Isolated Storage</p> <p>A mini-file system Application Settings Isolated Storage[File Stream] System.IO.IsolatedStorage types Persisting isolated storage</p> <p style="text-align: center;">Security</p> <p>The Silverlight security story Identity in a Silverlight app Plug-in security Integration with ASP.NET Membership Transparent libraries / SafeCritical Hashing, AES, CryptoStream, ..</p> </div> </div>