


WCF RIA Services Inside Out

Brian Noyes
www.idesign.net

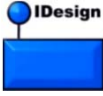
©2010 Brian Noyes, IDesign Inc. All rights reserved



About Brian

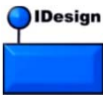
<p>Chief Architect IDesign Inc. (www.idesign.net)</p>	<p>Publishing</p> <p><i>Developing Applications with Windows Workflow Foundation, LiveLessons training DVD, June 2007</i></p> <p><i>Smart Client Deployment with ClickOnce, Addison Wesley, January 2007</i></p> <p><i>Data Binding in Windows Forms 2.0, Addison Wesley, January 2006</i></p> <p><i>MSDN Magazine, MSDN Online, CoDe Magazine, The Server Side .NET, asp.netPRO, Visual Studio Magazine</i></p>
<p>Microsoft Regional Director (www.theregion.com)</p>	<p>Speaking</p> <p><i>Microsoft TechEd US, Europe, Malaysia, Visual Studio Connections, DevTeach, INETA Speakers Bureau, MSDN Webcasts</i></p>
<p>Microsoft MVP Connected Systems</p>	

E-mail: brian.noyes@idesign.net
Blog: <http://briannoyes.net>
Twitter: @briannoyes



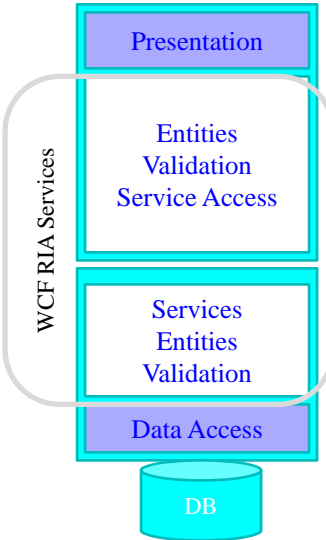
Agenda

- Quick Intro to WCF RIA Services
- LINQ Magic
- The services behind the curtain
- Entity glue – metadata



WCF RIA Services Overview

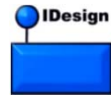
- Simplifies building N-tier applications
- Focused on Line of Business (LOB) applications
 - Highly dependent on pull-push of data from back end
- Architecture and tools for building the glue code between the client and the back end
 - Silverlight, ASP.NET; future: WPF, other clients
 - Entity Framework, LINQ to SQL; future: WCF Data Services (Astoria), SQL Azure



The diagram illustrates the WCF RIA Services architecture. It consists of several layers:

- Presentation**: The top layer, representing the user interface.
- Entities Validation Service Access**: A layer containing logic for entities, validation, and service access.
- Services Entities Validation**: A layer containing logic for services, entities, and validation.
- Data Access**: A layer responsible for data access.
- DB**: The database layer at the bottom.

 A vertical label 'WCF RIA Services' is positioned to the left of the middle three layers, and a rounded rectangle highlights the 'Entities Validation Service Access' and 'Services Entities Validation' layers.



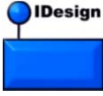
What do you need?

- Silverlight Tools for Visual Studio 2010
 - Core bits
- RIA Services Toolkit
 - Other endpoint types (SOAP, JSON)
 - ASP.NET support
 - Client tools for proxies



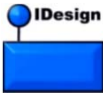
WCF RIA Services Overview

- Link client project to server project
- Define server side domain services
 - Base classes: DomainService, LinqToEntitiesDomainService, LinqToSqlDomainService
- Compile
- WCF RIA Services code generates client side “proxy” code
 - Domain context
 - Entities
 - Validation
 - Shared code



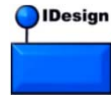
Application Architecture

- Web project
 - Silverlight host
 - Domain service host
 - Domain Services
 - ▲ EnableClientAccess attribute
- Client project (i.e. Silverlight)
 - Generated code folder
 - DomainContext classes
 - ▲ One per domain service
 - DomainDataSources
 - ▲ Facilitates simple data binding
- Supporting class libraries as needed



Query Operations

- Read operations
 - Return object, IEnumerable<T>, IQueryable<T>
 - Query attribute on methods
 - ▲ IsComposable=false when returning single entity
 - Naming convention without Query attribute
 - Use LoadOperation<T> on client side to load data
 - Load related types with a metadata class with [Include] on related members
- CUD Operations (CRUD – R)
 - Naming convention in domain service to identify methods
 - ▲ InsertXXX, UpdateXXX, DeleteXXX where XXX is entity type
 - DomainContext contains an EntityContainer
 - EntityContainer tracks changes on entities with EntityChangeSet



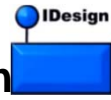
DomainDataSource

- Silverlight only, eventually WPF as well
- Similar to ObjectDataSource and XmlDataSource in WPF
- Allow you to execute a DomainContext method to query data
 - Including updates
- Drag/drop data binding is VS 2010
- Supports parameterized queries
- Supports sorting, grouping, filtering, paging
- Not a good fit with MVVM
 - Puts query “logic” into the view



Agenda

- Quick Intro to WCF RIA Services
- [LINQ Magic](#)
- The services behind the curtain
- Entity glue – metadata



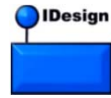
IQueryable<T> and Deferred Execution

- RIA Services serializes the expression tree from the client the server
- LINQ operations specified by the expression tree are executed on the server
- When the underlying provider supports it (i.e. Entity Framework and LINQ to SQL), deferred execution goes all the way to the DB



Agenda

- Quick Intro to WCF RIA Services
- LINQ Magic
- [The services behind the curtain](#)
- Entity glue – metadata



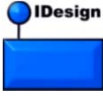
Service Magic

- WCF service created at runtime based on domain service
- REST style
- Binary encoding
- Address scheme:
 - [BaseURI] + [DomainServiceNameWithNamespace].svc with all "." in the DomainServiceNameWithNamespace replaced by "_"
- Client proxy code generated at compile time
- Generated service contracts:
 - Query methods return QueryResult<T>
 - CUD methods signature:
 - ▲ ChangeSet SubmitChanges(ChangeSet changeSet)



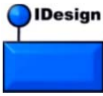
Agenda

- Quick Intro to WCF RIA Services
- LINQ Magic
- The services behind the curtain
- Entity glue – metadata



Entity Glue

- Define a partial class for the entity type
- Define a nested metadata class inside the entity type
 - Not required, but convention
 - “Buddy class”
- Add a MetadataType attribute on the entity partial class to tie in the metadata type
- Add attributes to properties on the metadata type that match the entity properties
- Those attributes show up on the client entity type or influence entity transmission
 - Validation
 - Include
 - Exclude
 - Composition



Resources

- WCF RIA Services
 - WCF RIA Services (10 parts – in work):
<http://www.silverlightshow.net/items/WCF-RIA-Services-Part-1-Getting-Started.aspx>
- Weblogs:
 - Team blog: <http://www.riaservicesblog.net/Blog/>
 - Nikhil Kothari: <http://www.nikhilk.net/>
 - Deepesh Monani: <http://blogs.msdn.com/b/deepm/>
 - Jeff Handley: <http://jeffhandley.com/>
 - Saurabh Pant: <http://blogs.msdn.com/b/saurabh/>