



Training – Consulting - Mentoring

Official Course Syllabus

CSLA Master Class

5 days

DUNN Training and Consulting

4805 Rambling Rose Drive

Cumming, GA 30040

770 653-6364

<mailto:info@dunntraining.com>

<http://www.dunntraining.com/>

OBJECTIVES

The CSLA .NET framework is an application development framework that reduces the cost of building and maintaining applications.

The framework enables the use of object-oriented design as the basis for creating powerful applications. Business objects based on the framework support many advanced features to simplify the creation of Windows Forms, Web Forms and Web Services interfaces.

CSLA .NET is designed to allow great flexibility in object persistence, so business objects can use virtually any data sources available. The framework is designed to enable single tier and n-tier models through the concept of mobile objects. This provides the flexibility to optimize performance, scalability, security and fault tolerance with no changes to code in the UI or business objects.

PREREQUISITES

Students are expected to possess the following prerequisite skill sets:

- Solid understanding and experience programming Microsoft .NET-based applications.
- Extensive knowledge and experience in each of the clients used in this class (Windows, Web, WPF, etc.) is not necessary as primers will be given. Students however should be seasoned enough to be able to get around all environments without getting lost.
- A good base for object oriented programming is essential for this class.
- If any of these are missing, being a quick-study and having eagerness to learn in a fast paced advanced environment will most likely suffice.

Additionally, students are expected to provide for themselves with the following equipment & technology:

- Their own laptop computer
- Visual Studio 2008 (*Visual Studio 2005 w/.NET 3.5 can work but 2008 is preferred*)
- SQL Server 2005 or SQL Server Express 2005 with Management Studio
- Sufficient security privileges on their computer to attach and access a provided SQL server database. If students possess a company laptop, they should test for this ahead of time.

NOTES

All classroom labs will have solution starters so students can hit the ground running with the essential material the lab covers. This 5-day class will use and cover all parts of the latest version of the CSLA Framework. Visual Studio 2008 with .NET 3.5 will be used during this class.

Students will receive a class CD containing the following:

- All completed applications in both VB and C# including database
- All class lab starters as well as completed lab code
- The CSLA Framework 3.5x including source
- The CslaEx component including source
- The WebControlsEx ASP.NET control library
- A trial version of CodeBreeze

In addition, students will receive a class manual with all presentation slide printouts, lab walkthroughs, and notes section. The class manual will also include instructions for perform certain classroom tasks such as attachment of databases, etc.

CLASS OUTLINE

- General Frameworks
 - Life before the .NET Framework & the CLR
 - Plumbing areas handled by the .NET Framework
 - Object brokering
- Business Frameworks
 - What are they
 - Why we need them
 - Goals of a business framework
- What is CSLA?
 - Rockford Lhotka
 - THE book
 - Evolution of CSLA
 - Its synchronicity with the .NET Framework
 - Business Object practices (<=VB6)
 - CSLA 1.x (pre-generics)
 - CSLA 2.x
 - CSLA 3.0 (enter W*F)
 - CSLA 3.5
 - Goals of the CSLA Framework
 - Hide plumbing
 - Object feature normalization

- Enforce standards
 - Object mobility
 - Features of the CSLA Framework
 - A base for business objects
 - Bindable
 - Business rule encapsulation
 - Security encapsulation
 - more...
- Software Architecture
 - Layers of application architecture
 - Various application architectures
 - Logical vs. Physical layers
 - Object distribution
 - Where CSLA falls into the mix
 - Service Oriented Architecture & CSLA
- Software Configuration Management
 - Installing CSLA
 - Assembly separation
 - Combined vs. separate data access layer
 - The CslaEx component
 - The CSLA Store Application
 - Project breakdown
 - Demo
 - Standards & Conventions

- Database
 - Project separation & naming
 - Coding conventions
- Business Objects
 - What makes a class a business object?
 - Full encapsulation
 - Features of a business object
 - Full state information
 - Data access
 - Data storage
 - Feature container
 - Collection container
 - Validation capable
 - Secure
- Overview of the CSLA Base Classes
 - BusinessBase
 - BusinessListBase
 - EditableRootListBase
 - NameValueListBase
 - ReadOnlyBase
 - ReadOnlyListBase
 - CommandBase
 - CriteriaBase
- Pre-requisites Primer

- .NET Generics
- LINQ
- Developing a CSLA Business Object
 - Inheriting from BusinessBase
 - Lockdown
 - Properties in a CSLA object
 - Property details
 - Attributes
 - Can-methods
 - Dirtiness
 - 3.5 property enhancements
 - The Console test-harness
 - Can-methods
 - Exposing a public interface & introducing the Data Portal
 - Factory methods
 - Creating a new object
 - Retrieving an existing object
 - The Data Portal lifecycle in detail
 - Criteria objects
 - Creating a unique signature
 - Two alternatives for criteria classes
 - The SafeDataReader object
 - Using the CslaEx component
 - State of the object

- Validity
- Dirtiness
- Newness
- Savability
- N-Level Undo
- The SmartDate object
- Adding authorization rules
 - Reading
 - Writing
 - Execution
- Validation
 - Adding validation rules
 - Dissecting a business rule
 - Developing custom business rules
 - Simple rules
 - Complex rules
 - Strongly-typed rules
 - Database-driven rules
 - Different levels of broken rules
- Saving an object
 - Testing for newness
 - Need for cloning (3.5 enhancement)
 - Data Portal methods
 - Returning the saved object

- Transactions
- Deleting an object
 - Immediate deletion
 - Marked for deletion
 - Saving
 - Post-deletion state
- Other Data Portal events
 - Invoke
 - InvokeComplete
 - Exception
- CSLA Collection Objects
 - Inheriting from BusinessListBase
 - Creating a new collection
 - Retrieving a collection of business objects
 - Saving the collection within a transaction
 - The business object as a Child object
 - The manual way
 - 3.5 enhancements for child objects
 - Inheriting from EditableRootListBase
 - Difference in Saving
 - ToArray()
- Name-Value Lists
 - Inheriting from NameValueListBase
 - Difference between this and a collection object

- Read-Only objects
 - When you should use them
 - Limitations
 - Inheriting from ReadOnlyBase
 - Inheriting from ReadOnlyListBase

- Commands
 - What are command objects
 - Creating a command
 - The factory method
 - The command class
 - Inheriting from CommandBase
 - The three jobs of a command class
 - Alternative placement for commands

- Parent-child Objects
 - The parent-child concept
 - Limitations of a child object
 - Controlling a child from its parent
 - Fetching, saving, and deleting
 - 3.5 enhancements for parent-child relationships
 - The new Data Portal methods
 - Overriding some state flags

- Switchable Objects
 - Description and examples
 - Object entry points

- Existing in two states
- The DataMapper
 - Mapping one object to another
 - Mapping a dictionary to an object
- Usage of CSLA Business Objects
 - Windows Applications
 - A Windows Forms data binding primer
 - Binding to CSLA objects
 - Binding to a grid
 - Binding to a data entry form
 - Adding field-level authorization
 - Error display
 - Saving
 - Step-by-step
 - Using the CslaEx component
 - ASP.NET Applications
 - An ASP.NET data binding primer
 - Overview of the DetailsView control
 - Introducing the CslaEx control enhancements
 - Binding to CSLA objects
 - The CslaDataSource control
 - Binding to a GridView control
 - Binding to a DetailsView control

- Deleting from a grid
 - Saving and Inserting using the DetailsView
- Windows Presentation Foundation Applications
 - A WPF primer
 - Binding to WPF forms
 - Authorization
- Windows Communication Foundation Applications
 - A WCF primer
 - Using CSLA business objects from WCF Services
 - Security considerations
- Workflow Foundation Applications
 - A WF primer
 - Workflow Activities & CSLA objects
- Sorted Lists
- Filtered Lists
 - Using filtered lists
 - Multiple filtration
 - Developing custom filters
- Object Adapter
- Context storage
 - Local
 - Client
 - Global
- LINQ in CSLA

- Querying CSLA collection objects using LINQ
- Security
 - Windows authentication (& ADS)
 - Custom authentication and authorization
 - A .NET Security primer
 - The IPrincipal & Identity interfaces
 - How roles fits in
 - Thread.CurrentPrincipal
 - HttpContext.Current.User
 - The Csla.ApplicationContext.User object
 - Building CSLA Principals
 - Inheriting from the BusinessPrincipalBase class
 - Perform authentication tasks
 - How CSLA business objects use the security principal
 - Object "Can-methods" vs. field-level security revisit
 - Security usage in a stateful environment
 - Security usage in ASP.NET
 - Custom membership providers
 - A primer
 - A CSLA membership provider
 - Persisting security information
 - Populating the context user
 - Repopulating using Global.asax

- Using a custom HTTP Module
 - WCF security
 - Best practices for naming
- Object Mobility
 - Situations that call for it
 - Remote Data Portals
 - A business object as a message (NOT SOA)
 - Options for remote data portals
 - Configuring remote data portals
 - Obsolete data portals
 - Remoting
 - Web Services
 - WCF Data Portals
 - Reconfiguring the client
 - Using Click-Once deployment
- Service Oriented Architecture
 - How CSLA fits in an SOA world
 - Various architecture possibilities
 - CSLA behind the service
 - CSLA in front of the service
 - CSLA on both sides
- Code Generation
 - Principles of code-generation
 - Generating CSLA objects

- Finding commonalities
 - Identifying exceptions (child objects, etc.)
- Using CodeBreeze
 - A CodeBreeze primer
 - Generating the data access layer
 - Generating business objects
- CSLA Community
 - CslaContrib project
 - The forums
 - Books available
 - CslaEx component update
- The future of CSLA

CLASS LABS

- Building and testing a CSLA business object
- Developing a custom business rule
- Building and testing a CSLA collection
- CSLA object usage from a Windows application
- CSLA object usage from an ASP.NET application
- CSLA object usage from a WPF application
- Using CSLA with Windows Workflow
- Querying CSLA using LINQ
- Implementing Security
- Creating and using a WCF Data Portal

CSLA MASTERS CLASS SUMMARY

- Business Frameworks
- CSLA - the what, why, and how
- Software Architecture
- Software Configuration Management
- CSLA Business Objects - ins and outs
- CSLA Collection Objects
- Name-Value Lists
- Read-Only objects
- Commands
- Parent-child Objects
- Usage of CSLA Business Objects
 - Windows Applications
 - ASP.NET Applications
 - Windows Presentation Foundation Applications
 - Windows Communication Foundation Applications
 - Workflow Foundation Applications
- Ancillary objects and functionality
- LINQ in CSLA
- Security
- Object Mobility (Remote Data Portals)
- Service Oriented Architecture
- Code Generation

- CSLA Community
- The future of CSLA

EXTRAS

- *The CslaEx component*

A component developed by us for the purpose of extending areas of the CSLA Framework. Among other things, this component includes a CSLA-compliant Data Access Layer and several enhanced controls that provide a nearly code-less approach to accessing certain functionality.

- *The WebControlsEx component*

An ASP.NET server control library containing several useful controls that are used throughout the sample application and offered as a freebie to CSLA Masters Class students.

- *CodeBreeze (trial version)*

CodeBreeze is a code-generation productivity tool that is generations ahead of all competition. It ships with CSLA templates whose generated code matches the style taught in the CSLA Masters Class.