

Leveraging Component Scalable Logical Architecture Framework

Course Length: 3 days

Price: \$2,250

Summary

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.

Who Should Attend?

In this course you will learn to make the most of using the CSLA .NET Framework. Onsite classes can be delivered using either C# or VB programming languages. Public classes give students the option to use either language for lab work. Students must be familiar with .Net programming techniques and frameworks.

Course Outline

Application Architecture

- Importance of proper architecture
- Advantages of a good architecture
- Architecture Layering
 - 3-Tier Architectures
 - N-Tier Architectures

Fundamentals

- Encapsulation

- Communication between layers
- Abstraction

Tiers vs. Layers

- Local vs. Distributed
- Determining layer location
- Distributed Architecture
 - 5-layer architecture
 - Remote Communications
 - Remoting
 - Web Services
 - WCF

Object Oriented Design

- Fundamentals
- Different objects, different roles
 - DAL Objects
 - Data-transfer objects
 - Business Objects
 - UI Helper Objects

Data-Centric object design vs. Domain Specific

- Combinations of both

Database Design

- Fundamentals
- Setting up standards and conventions
- Best Practices

Application Frameworks

- Why use them
- Identifying code commonalities

- Providing abstractions
- Providing implementations
- Examples in the .NET Framework

The Component Scalable Logical Architecture Framework

- Framework Goals and Design
- Framework Organization
- Features
 - Business Object base classes
 - N-Level undo
 - Data Binding
 - Data Portal
 - Authentication
 - Authorization
 - Helper Types
 - Other
- Framework Abstractions
 - Base Classes
 - Interfaces
- Creating a project using CSLA
 - Folder structure
 - Referencing CSLA
 - Implementing CSLA functionality
 - Creating Business Objects
 - Life Cycle

- Class Structure
- Business object lists
- Accessing data (the CSLA way)
- Validation
- Security
- Distributing objects
- Client Interfaces
 - Windows Forms
 - Accessing Business Objects
 - Data Binding
 - User Controls
 - Extender Controls
 - Using N-Level Undo
 - ASP.NET
 - Accessing Business Objects
 - State Management
 - Declarative Data Sources
 - Limitations (in comparison to Windows Forms)
 - Interface Design
 - Security
 - Web Services
 - Fundamentals
 - Services vs. Components
 - Authentication

- Web Methods
 - Component-based
 - Service-Oriented
- A Consumer Client
- Remote Data Portals
 - Remoting
 - Web Services
 - Enterprise Services
 - Windows Communication Foundation (what's next)

Code Generation

- Advantages of Code-generation
- A base/derived or partial class layer approach
- Code Generation and CSLA
- Implementing code-generation with CodeBreeze using the CSLA templates
- Database changes and re-generation

Conclusion and Review

- Summarization of good architecture principles
- Advantages of using a framework (CSLA)
- Advantages of code-generation