Login (/login.aspx)   Join Now (/join.aspx)

(/)

.NET Design Patterns

Design patterns are solutions to software design problems you find again and again in real-world application development. Patterns are about reusable designs and interactions of objects.

The 23 Gang of Four (GoF) patterns are generally considered the foundation for all other patterns. They are categorized in three groups: Creational, Structural, and Behavioral (for a complete list see below).

To give you a head start, the C# source code for each pattern is provided in 2 forms: *structural* and *real-world*. Structural code uses type names as defined in the pattern definition and UML diagrams. Real-world code provides real-world programming situations where you may use these patterns.

A third form, *.NET optimized*, demonstrates design patterns that fully exploit built-in .NET 4.5 features, such as, generics, attributes, delegates, reflection, and more. These and much more are available in our .NET Design Pattern Framework 4.5 (/products/net-design-pattern-framework). You can see the Singleton (/net/singleton-design-pattern) page for a .NET 4.5 Optimized example.

Creational Patterns

Abstract Factory (/net/abstract-factory-design-pattern)	Creates an instance of several families of classes
Builder (/net/builder-design-pattern)	Separates object construction from its representation
Factory Method (/net/factory-method-design-pattern)	Creates an instance of several derived classes

| Prototype (/net/prototype-design-pattern) | A fully initialized instance to be copied or cloned |
| Singleton (/net/singleton-design-pattern) | A class of which only a single instance can exist |

Structural Patterns

Adapter (/net/adapter-design-pattern)	Match interfaces of different classes
Bridge (/net/bridge-design-pattern)	Separates an object's interface from its implementation
Composite (/net/composite-design-pattern)	A tree structure of simple and composite objects
Decorator (/net/decorator-design-pattern)	Add responsibilities to objects dynamically
Facade (/net/facade-design-pattern)	A single class that represents an entire subsystem
Flyweight (/net/flyweight-design-pattern)	A fine-grained instance used for efficient sharing
Proxy (/net/proxy-design-pattern)	An object representing another object

Behavioral Patterns

Chain of Resp. (/net/chain-of-responsibility-design-pattern)	A way of passing a request between a chain of objects
Command (/net/command-design-pattern)	Encapsulate a command request as an object
Interpreter (/net/interpreter-design-pattern)	A way to include language elements in a program
Iterator (/net/iterator-design-pattern)	Sequentially access the elements of a collection
Mediator (/net/mediator-design-pattern)	Defines simplified communication between classes
Memento (/net/memento-design-pattern)	Capture and restore an object's internal state
Observer (/net/observer-design-pattern)	A way of notifying change to a number of classes
State (/net/state-design-pattern)	Alter an object's behavior when its state changes
Strategy (/net/strategy-design-pattern)	Encapsulates an algorithm inside a class

Template Method (/net/template-method-design-pattern)	Defer the exact steps of an algorithm to a subclass
Visitor (/net/visitor-design-pattern)	Defines a new operation to a class without change

- Better Code
- Better Career
- Better Lifestyle

.NET Design Pattern Framework 4.5 (/products/net-design-pattern-framework)

(/products/net-design-pattern-framework)

C# and VB (/products/net-design-pattern-framework)

Includes:

- Gang of Four Patterns
- Head First Patterns
- Enterprise Patterns
- Multi-Tier Patterns

- Repository Pattern
- Unit-of-Work Pattern
- Active Record Pattern
- CQRS Pattern

- Model View Controller
- Model View Presenter
- Model View ViewModel

- Spark^{TM} RAD Platform
- Art Shop MVC application
- More...

(/products/net-design-pattern-framework)
Get started today!

Click here for details
(/products/net-design-pattern-framework)

-- Instant Access --
Instant Download

Company

- About Us (/about)
- Our Story (/story)
- Services (/services)
- Training (/training)
- Contact Us (/contact)

- Privacy (/privacy)
- Terms (/terms)
- Licensing (/licensing)

Customers

- Our Customers (/customers)
- Customer Stories (/customers/stories)

Community

- Questions (/topic/search.aspx)
- Ask Question (/topic/add.aspx)
- Explore (/topic/topics.aspx)
- Tags (/tag/tags.aspx)
- Users (/user/users.aspx)

Reference Guides

- .NET Design Patterns (/net/design-patterns)
- JavaScript Design Patterns (/javascript/design-patterns)

- JavaScript Tutorial (/tutorial/javascript)
- SQL Tutorial (/sql/tutorial)

- Connection Strings (/reference/connection-strings)
- Visual Studio Shortcuts (/reference/visual-studio-shortcuts)
- C# Coding Standards (/reference/csharp-coding-standards)
- HTML Colors (/reference/html-color-codes)

Our Products

- .NET Design Pattern Framework (/products/net-design-pattern-framework) [TM]
- PRO .NET Design Pattern Framework (/products/pro-net-design-pattern-framework) [TM]

- JavaScript + jQuery Pattern Framework (/products/javascript-jquery-design-pattern-framework) [TM]
- SQL + Database Pattern Framework (/products/sql-database-design-pattern-framework) [TM]

- Products and Pricing (/products)