

Builder Design Pattern in C#

Ancelia Patrao
Siddhant Magar
Shreya Rao

Builder Design Pattern

Definition

The Builder Design Pattern is a creational design pattern that separates the construction of a complex object from its representation, allowing the same construction process to create different representations of that object.

Use Case

Report Generator (PDF & Excel)

Laptop Assembly

Components of Builder

Builder

Defines the steps to construct the product (as an interface or abstract class).

Concrete Builder

Implements each step to build a specific version of the product.

Director

Controls the building sequence

Client

Triggers the building process

Product

Complex Object that is built step by step

Problems in Normal Object Creation

```
public class Report
{
    public string Title;
    public string Content;
    public string Footer;
    public string Header;
    public string Logo;
    public string Author;
}
```

```
Report report = new Report("Sales",
    "Data",
    "Page 1",
    "Header1",
    "LogoPath",
    "Swabhav");
```

Too many constructor parameters make object creation hard to read and maintain.

Constructors force all fields to be passed, even when some are optional.

Parameter order must be remembered, increasing chances of mistakes.

Advantages

Builds complex objects in a controlled, ordered way.

Keeps construction logic separate from the object itself.

Clear and organized code without constructor overloads.

Each builder can be tested independently.

Easily add new product types without changing existing logic.

Final product can be read-only after construction.

Example FlowChart

