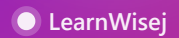


Using the Wisej Designer to Place Controls

Build a screen visually in the Designer — place, name and arrange controls before any behavior.



Lesson goal

Use the Wisej Designer to build a small page layout — placing, naming, resizing and organizing controls — before any behavior is added.

1. The Designer builds the screen visually

The Wisej Designer is where you build a screen **visually** instead of hand-writing every UI control. You drag controls from the Toolbox onto a **page** or **window**, arrange them, and set their properties. Behavior comes later: you write what happens when a button is clicked in the code-behind file.



Key idea

Design the layout first. Drag controls, set **Name** and **Text**, align them — then wire up behavior once the screen looks right.

2. The Designer workflow

Step	What to do	Why it matters
1	Open the page or window in Design view .	You can see and edit the interface visually.
2	Drag controls from the Toolbox onto the page.	This creates the UI objects the user interacts with.
3	Set the Name and Text properties.	Name is used in code; Text is what the user sees.
4	Resize and align the controls.	A clean layout makes the app easier to use.
5	Save and run locally.	Confirms the page appears correctly in the browser.

As you place controls, the Designer writes the layout into the **.Designer.cs** file **automatically** — you don't edit that file by hand.

3. Common controls to practice

Control	Use it for	Example name
Label	Headings, instructions, or status text	<code>lblStatus</code>
TextBox	Letting the user type input	<code>txtUserName</code>
Button	Running an action when clicked	<code>btnStart</code>
Panel	Grouping related controls	<code>pnlStatusArea</code>
ListBox	Showing a simple list of messages	<code>lstEventLog</code>
DataGridView	Showing rows of data	<code>gridTickets</code>

4. Name controls clearly

Clear names are a developer habit. Don't leave controls as `button1`, `label1` or `textBox1` once the page starts getting real behavior — clear names make event code far easier to read.

Weak name	Better name	Reason
<code>button1</code>	<code>btnStart</code>	Tells you it is the Start button.
<code>label2</code>	<code>lblStatus</code>	Tells you it shows the current status.
<code>textBox1</code>	<code>txtSearch</code>	Tells you the user types a search value.
<code>listBox1</code>	<code>lstEventLog</code>	Tells you it stores event messages.

5. Mini practice — a dashboard layout

You'll build this exact screen in the lab. For now, picture the pieces:

- A large title label: **System Dashboard**.
- A status label that starts at **Status: Idle**.
- Three service labels: **Server**, **Database** and **API Service**.
- Four buttons: **Start**, **Stop**, **Reset** and **Refresh**.
- An **Event Log** label and a ListBox/TextBox area for messages.

Suggested control names

```
lblTitle           // System Dashboard
lblStatus          // Status: Idle
lblServerStatus    // Server: Offline
lblDatabaseStatus  // Database: Offline
lblApiStatus       // API Service: Offline
btnStart, btnStop, btnReset, btnRefresh
lstEventLog        // Event Log messages
```



Completion check

You should be able to open a page in the Designer, place a Label, TextBox/ListBox, Button and Panel, rename them clearly, and change visible properties such as **Text** , **BackColor** , **Font** or **Size** .