Visual C Sharp Fast-track

Sample manual - first two chapters

TABLE OF CONTENTS (1 of 6)

TABLE OF CONTENTS (2 of 6)

TABLE OF CONTENTS (3 of 6)

TABLE OF CONTENTS (4 of 6)

TABLE OF CONTENTS (5 of 6)

TABLE OF CONTENTS (6 of 6)

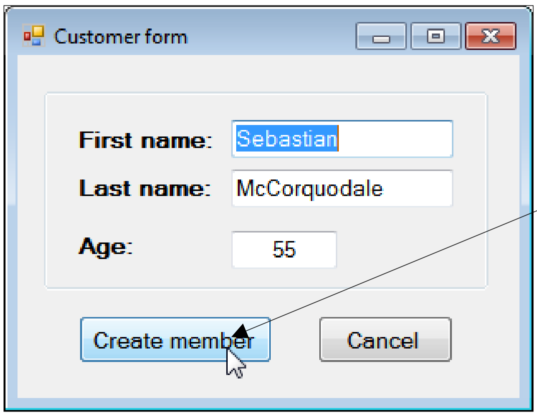
CHAPTER 1 - VISUAL STUDIO PRIMER

1.1 Windows Forms

There are three main types of application you can develop using *Visual Studio* ((Microsoft's development tool for .NET programmers):

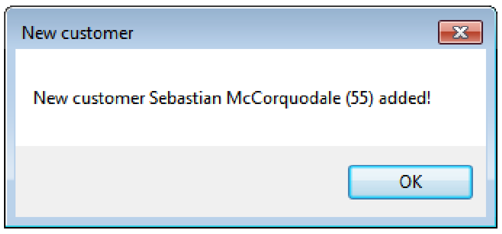
Type of application	Use for
WinForms (Windows Forms)	Creating basic business standalone applications to run within Windows.
WPF (Windows Presentation Foundation)	Creating Windows applications with fancy graphics (WPF takes longer to learn but is more powerful than Windows forms).
ASP.NET webforms	Creating websites using forms-based ASP.NET.
ASP.NET MVC / MVC Core	Creating websites using the model-view-controller method.

This courseware uses WinForms exclusively, because it's the simplest of the 3 types of application (the aim of the course is to teach C#, not drawing!).


Customer form

First name:	Sebastian
Last name:	McCorquodale
Age:	55

This is an application form for the *Wise Owl Dating Agency* (*WODA*, not to be confused with YODA). Clicking on the button could show a message like this:


New customer

New customer Sebastian McCorquodale (55) added!

To create a Windows Forms application like this, you'll first need to learn to use Visual Studio, as shown in the rest of this chapter.

Wise Owl's Hint

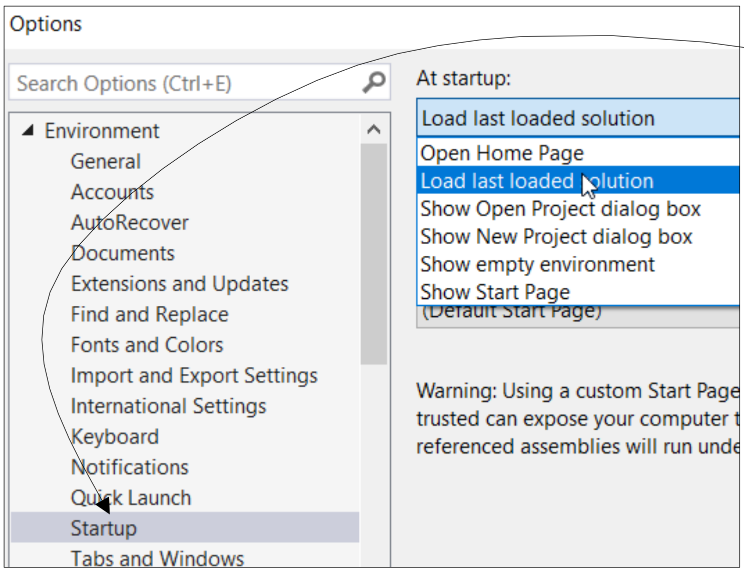
This courseware uses Visual Studio 2017, but you should find that all functionality is pretty much identical, regardless of which version of Visual Studio you use.

1.2 Customising Visual Studio

Before getting started using Visual Studio, it's a good idea to make a couple of changes.

Setting the Default Start-up Page

To change what you see each time that you go into Visual Studio, from the menu select: **Tools Options...** and then:



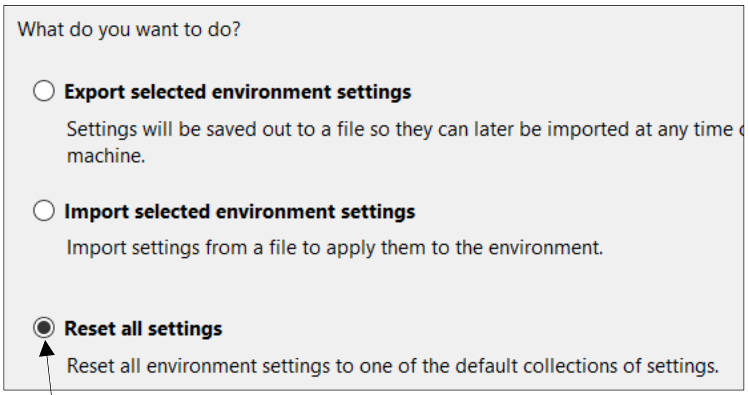
Warning: Using a custom Start Page
trusted can expose your computer t
referenced assemblies will run unde

a) Select the **Startup** option under the **Environment** section.

b) Choose the option you want (the Wise Owl preference is to show the last thing you were working with, as here).

Creating Appropriate Settings

In Visual Studio you can save *settings*, letting you switch between different ways of working. From the menu choose **Tools → Import and Export Settings…** and then:

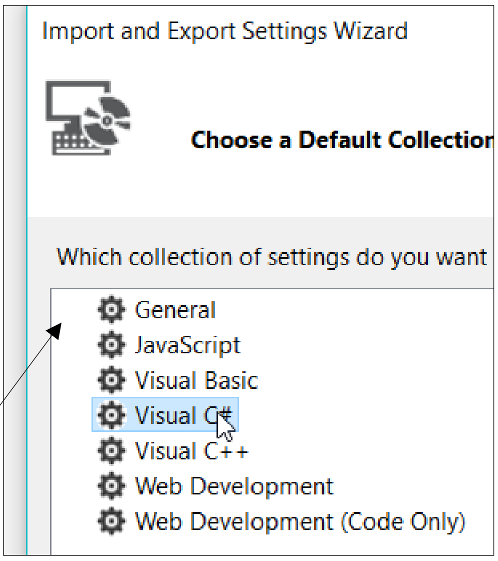


What do you want to do?

○ **Export selected environment settings**
Settings will be saved out to a file so they can later be imported at any time o
machine.

○ **Import selected environment settings**
Import settings from a file to apply them to the environment.

● **Reset all settings**
Reset all environment settings to one of the default collections of settings.


Import and Export Settings Wizard
Choose a Default Collectior
Which collection of settings do you want
General
JavaScript
Visual Basic
Visual C#
Visual C++
Web Development
Web Development (Code Only)

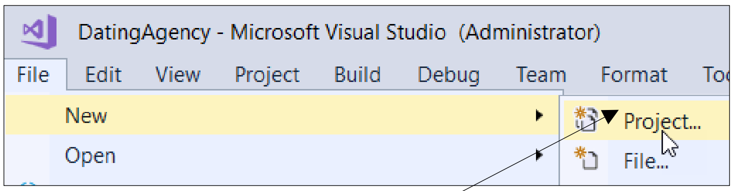
a) Choose to reset your settings (the next dialog box asks if you want to save your existing ones, but there never seems much point).

b) Choose to become a C# developer by default!

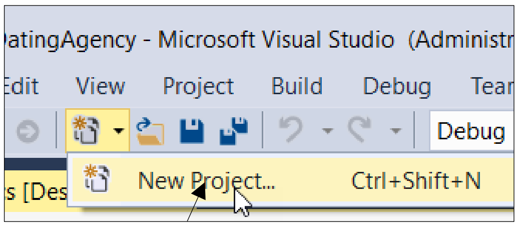
1.3 Creating Projects

A *project* is the term for the container for all of the files which make up your application. Projects can contain any number of folders and files. To create a project:

1) Choose one of the ways shown below to create a new project:


DatingAgency - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Team Format
New
Open
Project...
File...

Either choose this menu option (you can also press Shift + Ctrl + N to do this) …


atingAgency - Microsoft Visual Studio (Administr
Edit View Project Build Debug Tear
Debug
New Project... Ctrl+Shift+N

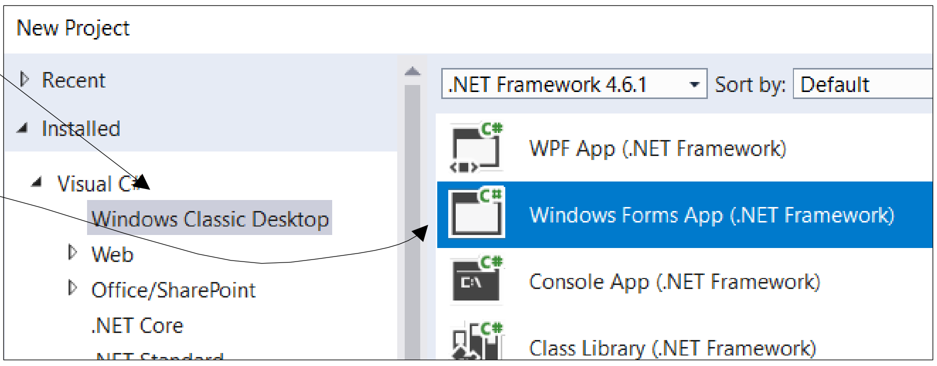
… or click on this tool to create a new project.

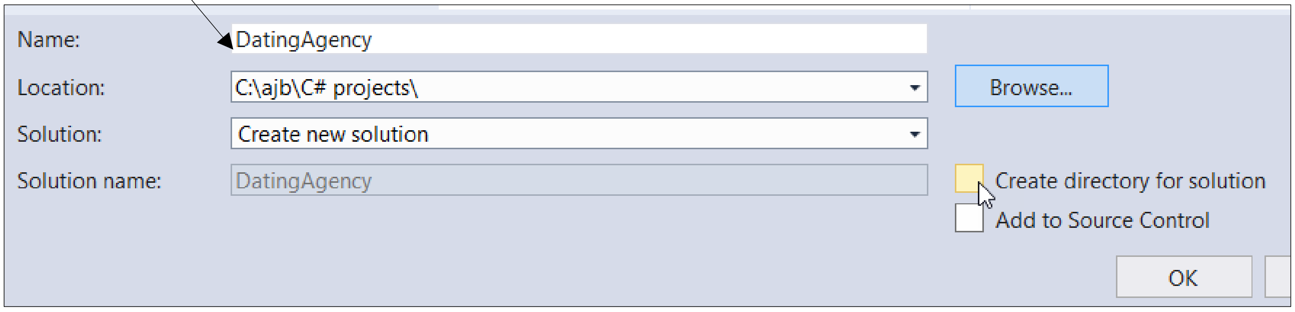
2) Complete the dialog box which appears as shown below, then select OK.

a) Choose to create a Windows application using C#.

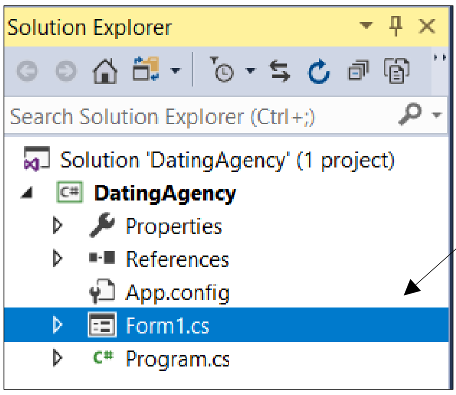
b) Choose to create a WinForms application, not a WPF one.

c) Give your project a name, and choose where to store it.

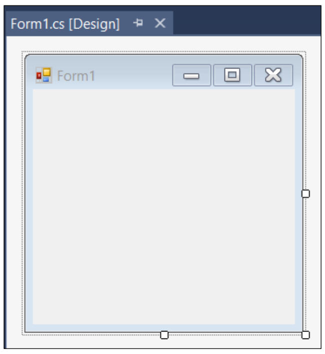




Name: DatingAgency
Location: C:\ajb\C# projects\
Solution: Create new solution
Solution name: DatingAgency
Create directory for solution
Add to Source Control



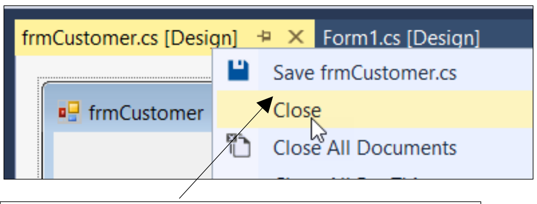
d) Visual Studio automatically creates a new form called **Form1** in your new project, and shows this on screen:



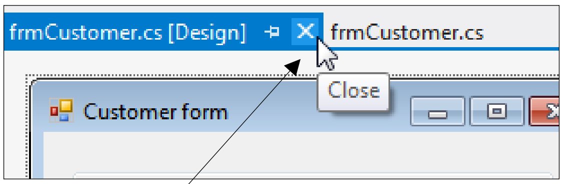
1.4 Saving and Closing Files

Closing One Window

To close a single window, either right-click on its tab or use the cross:



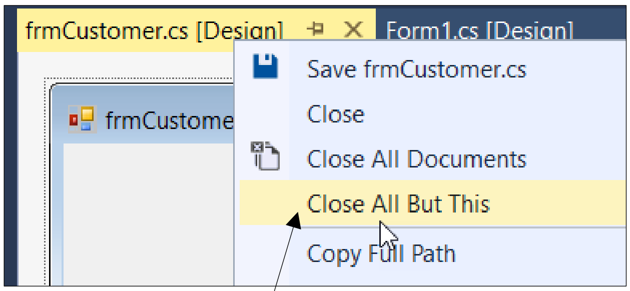
Either right-click on the window tab and choose the option shown to close it …



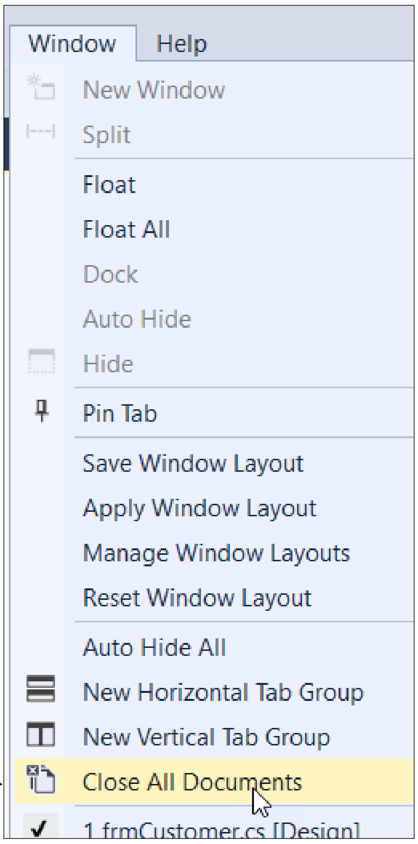
… or click on this cross to close the window in question.

Closing All or Nearly All Windows

Windows accumulate quickly in Visual Studio – before you know where you are you have 8 or 10 open! Two quick ways to close all or nearly all of your windows are shown below:



Right-click on a window and choose this menu option to close all the windows except this one.

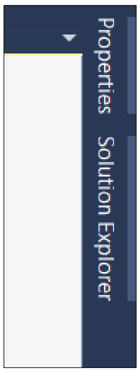


Select this menu option to close every window that you have open.

A useful short-cut is to press Shift + Ctrl + S *to save every window that you have open – this means that you can then close all of your open windows without having to confirm that you want to save the changes for each.*

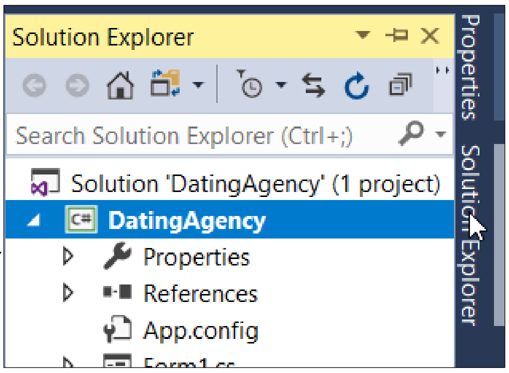
1.5 Auto-hiding windows

There are a lot of windows in Visual Studio, and they can quickly clutter up the screen. The best thing to do is to *auto-hide* them, so that they show up as icons on the edge of the screen:

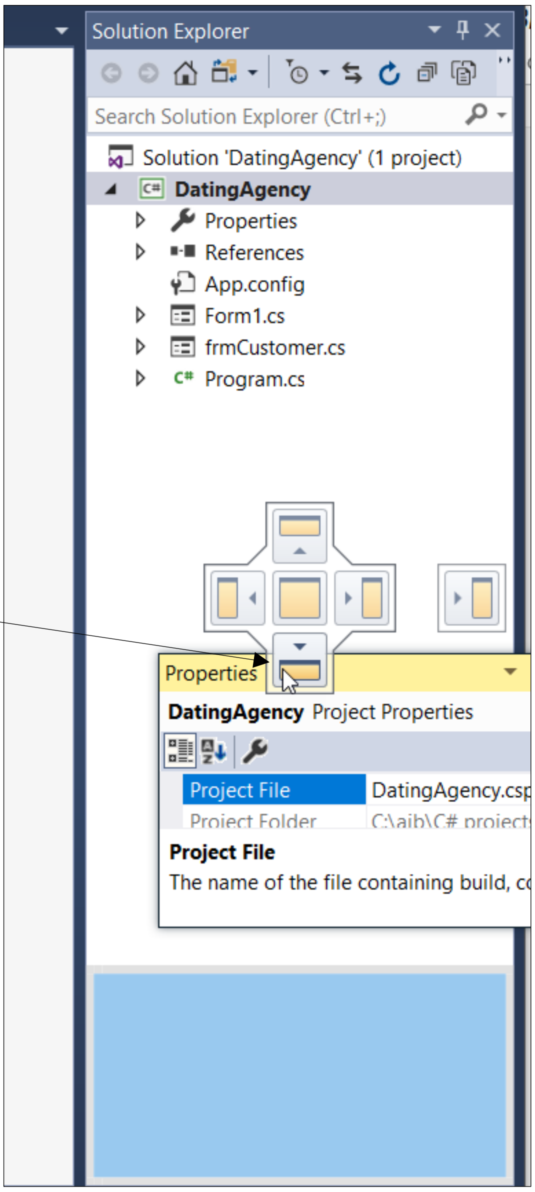


Auto-hidden windows will show up as icons on the edge of the screen …

… but when click onto them, the window expands (when you click off them, it contracts back down to an icon again).

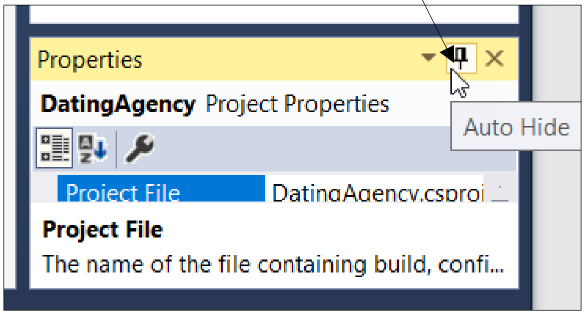


To set windows to *auto-hide* like this:



a) If necessary, drag the window to the top, bottom, left or right of the Visual Studio screen. When you position the window on one of these icons, it will lock into place – you can then release the mouse button.

b) Click on the vertical map pin icon to auto-hide the window – the map pin will then be horizontal.


Properties
Auto Hide
DatingAgency Project Properties
Project File
DatingAgency.csproj
Project File
The name of the file containing build, confi...

1.6 The Three Most Useful Windows

Displaying Windows

Visual Studio contains 3 toolbars that you'll need to have open most of the time – here's how to display them:

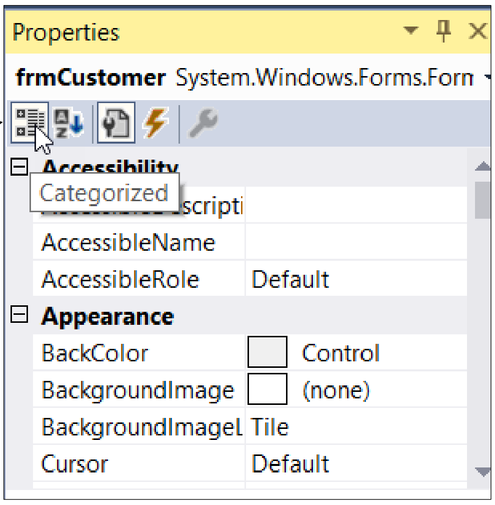
Windows	Menu option	Short-cut key
Solution Explorer	**View Solution Explorer**	Alt + Ctrl + L
Properties	**View Properties Window**	F4
Toolbox	**View Toolbox**	Alt + Ctrl + X

Properties Window

You can choose to display properties by category or alphabetically:

If you choose to display properties by category (the best option, this owl thinks) …

… you'll see the properties grouped into categories (**Accessibility**, **Appearance**, etc.).


Properties

frmCustomer System.Windows.Forms.Form

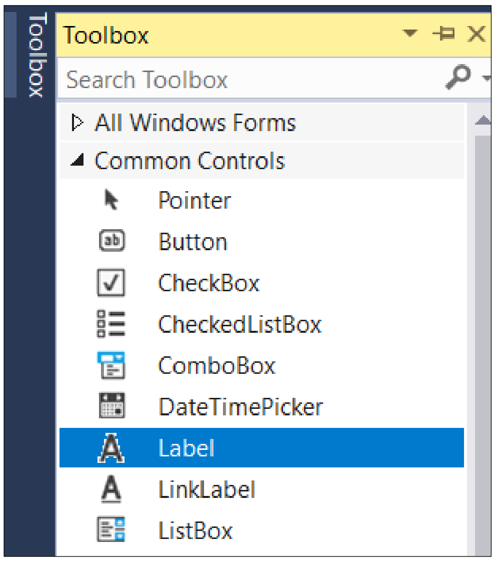
Accessibility	
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	Control
BackgroundImage	(none)
BackgroundImageLayout	Tile
Cursor	Default

The Toolbox

The *toolbox* allows you to add things to a form quickly:

Click on any tool and drag it onto the form to add it. The main tools we'll use are:

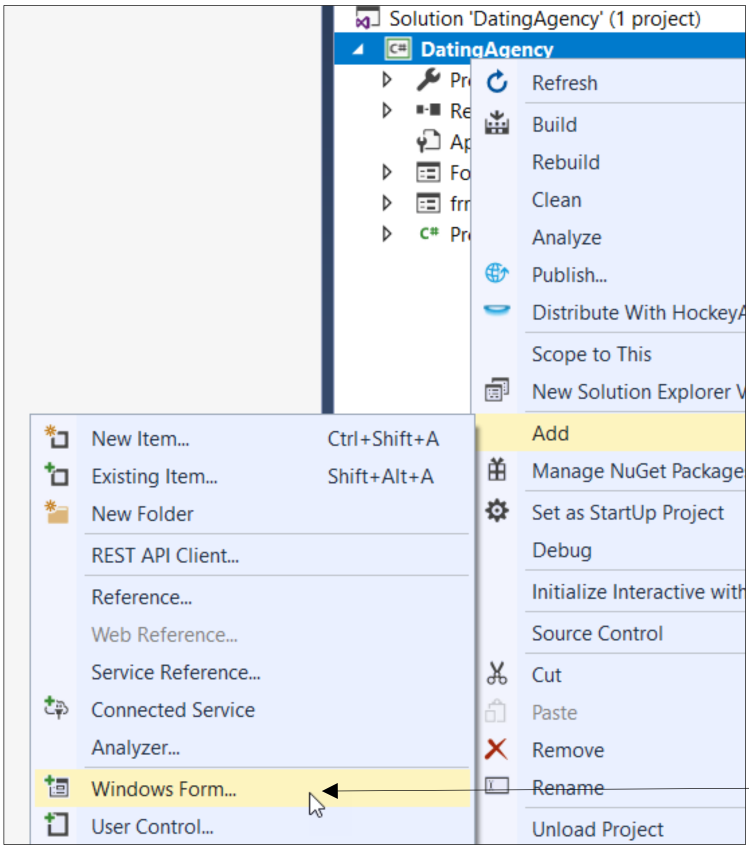
Tool	What it does
Label	Displays text on a form.
TextBox	Allows a user to type text into a box.
Button	A clickable button, with events attached.
ComboBox	A dropdown list.
GroupBox	Used to draw rectangles on forms.



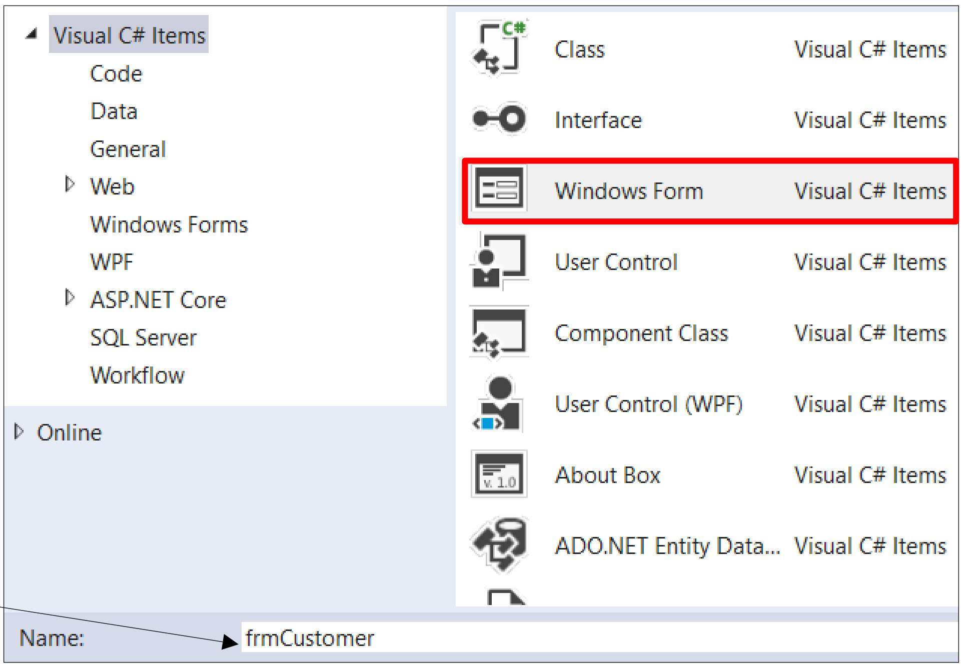
CHAPTER 2 - DRAWING FORMS

2.1 Creating a New Form

To create a new form, press Alt + Ctrl + L to bring up the Solution Explorer window, then:



a) Right-click on the project folder to which you want to add a form, and choose this option.


Visual C# Items
Code
Data
General
Web
Windows Forms
WPF
ASP.NET Core
SQL Server
Workflow
Online
Class Visual C# Items
Interface Visual C# Items
Windows Form Visual C# Items
User Control Visual C# Items
Component Class Visual C# Items
User Control (WPF) Visual C# Items
About Box Visual C# Items
ADO.NET Entity Data... Visual C# Items

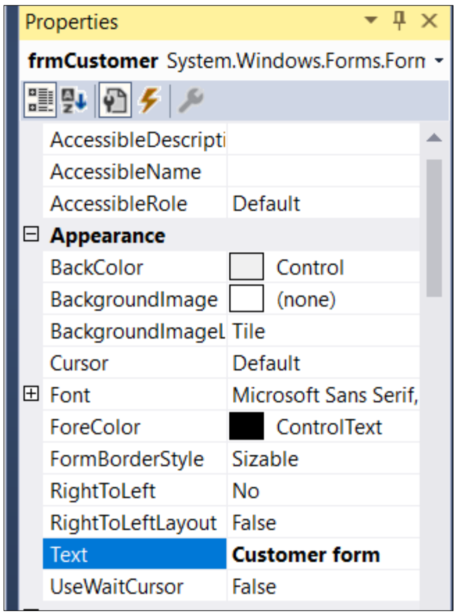
Name: frmCustomer

b) Choose to add a Windows form, then give it a sensible name.

2.2 Changing form properties

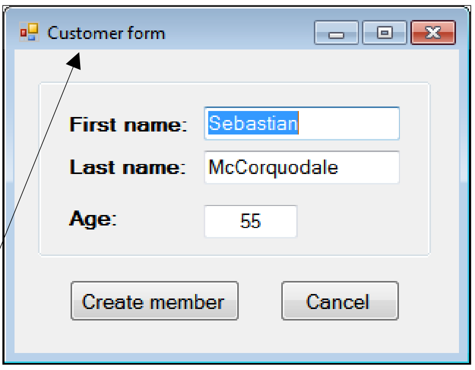
Click on the background of the form to make sure it’s selected, then change any properties:

For example, the caption of the form comes from its **Text** property, which is in the **Appearance** category.



frmCustomer System.Windows.Forms.Forn

AccessibleDescripti	
AccessibleName	
AccessibleRole	Default
Appearance	
BackColor	⬜ Control
BackgroundImage	⬜ (none)
BackgroundImageL	Tile
Cursor	Default
Font	Microsoft Sans Serif,
ForeColor	⬛ ControlText
FormBorderStyle	Sizable
RightToLeft	No
RightToLeftLayout	False
Text	**Customer form**
UseWaitCursor	False


Customer form

First name: Sebastian

Last name: McCorquodale

Age: 55

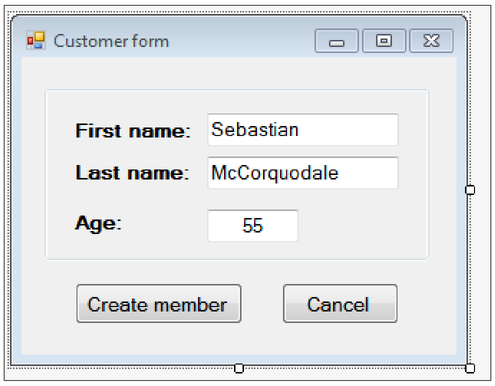
Here are some useful properties to set for a form:

Category	Property	Notes
Appearance	**Text**	The caption for the form (as above).
Layout	**StartPosition**	Change to **CenterScreen** to make a form appear in the middle of the screen.
Msc	**AcceptButton**	Set to the button which you want to be selected by default if a user presses ↵ .
	CancelButton	Set to the button which you want to be selected by default if a user presses Esc .

2.3 Form Controls

Definition of Controls

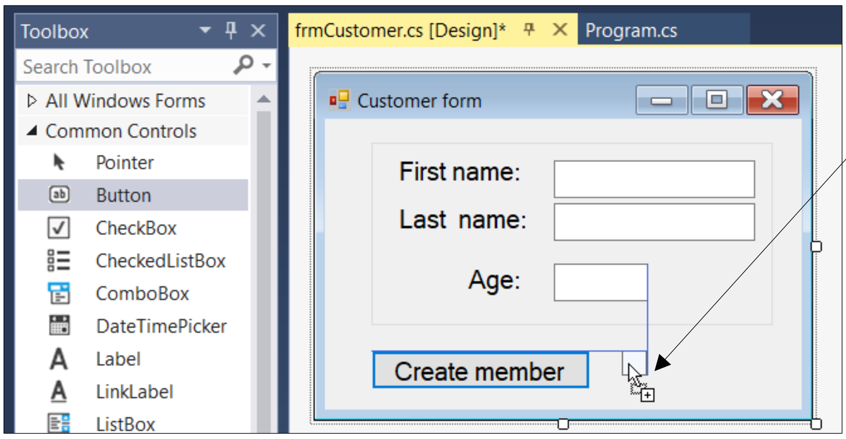
All the different widgets (labels, text boxes, command buttons, etc) that you add to a form are called *controls*:



This form contains 3 labels, 3 textboxes, 2 buttons and a groupbox (the thin rectangle).

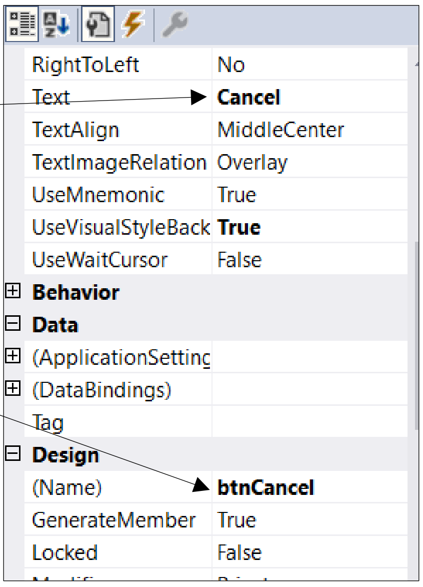
Adding controls

The easiest way to add a control to a form is to click and drag on it:



Here we're adding a clickable **Button** to a form. You can use the horizontal and vertical blue lines which appear to align controls.

You should then give your control a good name:



RightToLeft	No
Text	**Cancel**
TextAlign	MiddleCenter
TextImageRelation	Overlay
UseMnemonic	True
UseVisualStyleBack	**True**
UseWaitCursor	False
Behavior	
Data	
(ApplicationSettinc	
(DataBindings)	
Tag	
Design	
(Name)	**btnCancel**
GenerateMember	True
Locked	False

It's likely that you'll want to change the text displayed on a button.

The first **Design** category property of a control is always its name. Wise Owl use modified Hungarian notation (!) to name controls:

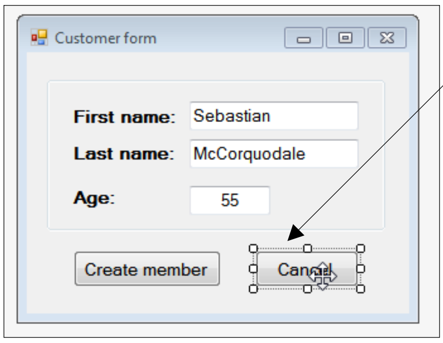
Type of control	Prefix to use
TextBox	**txt**
Label	**lbl**
Button	**btn**
ComboBox	**cmb**

2.4 Selecting Controls

Before you can move, copy, edit, delete, rename or format controls, you must first select them!

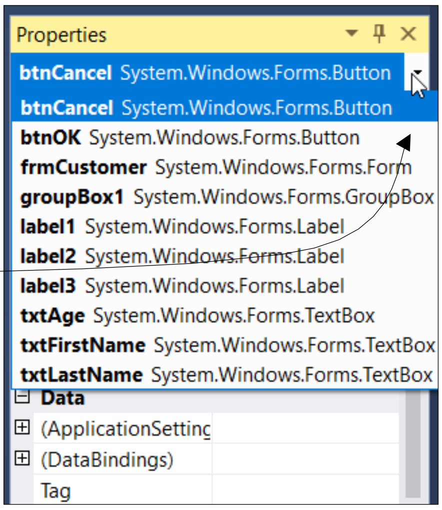
Selecting a Single Control

Here are two ways to select a single control:



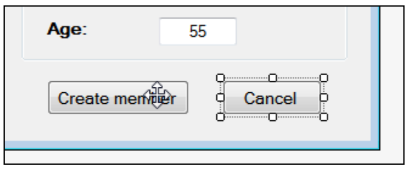
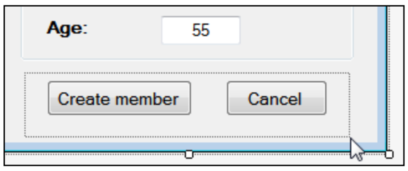
Click on a control to select it …

.. or click on the dropdown at the top of the **Properties** window to change what you have selected.


Properties
btnCancel System.Windows.Forms.Button
btnCancel System.Windows.Forms.Button
btnOK System.Windows.Forms.Button
frmCustomer System.Windows.Forms.Form
groupBox1 System.Windows.Forms.GroupBox
label1 System.Windows.Forms.Label
label2 System.Windows.Forms.Label
label3 System.Windows.Forms.Label
txtAge System.Windows.Forms.TextBox
txtFirstName System.Windows.Forms.TextBox
txtLastName System.Windows.Forms.TextBox
Data
(ApplicationSetting
(DataBindings)
Tag

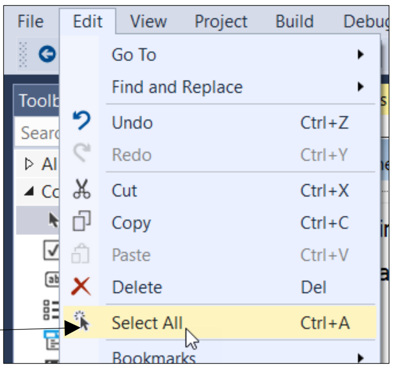
Selecting Several Controls

Suppose that you want to select the two buttons in the form above. Here are two ways to do this:

Method	Example diagram
Select one control, then hold down the Ctrl key and click on the others you want to add to/remove from your selection.	
Click and drag to draw a rectangle – anything it touches or encloses will be selected.	

Selecting All Controls

To do this, press Ctrl + A or select the menu option shown:



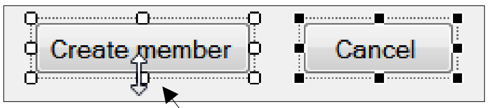
Select this option from the menu to select all of the controls on a form.

2.5 Basic Formatting

You can change how a form looks by *formatting* it. All of the examples below refer to the button, but the properties shown apply to most controls and to the form itself.

`Resizing Controls

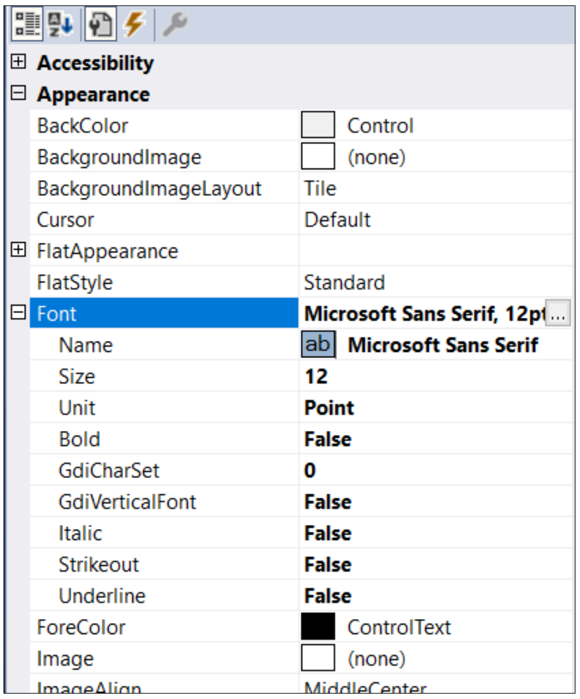
You can change the size of one or more controls using the handles surrounding them:



Clicking and dragging here, for example, would change the height of both buttons, since they are both selected.

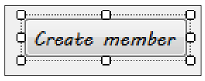
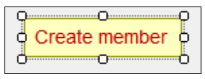
Changing how Controls Look

To do this, select the control(s) and change any of their properties – here are a couple of examples:



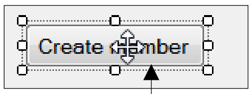
⊞ **Accessibility**	
⊟ **Appearance**	
BackColor	Control
BackgroundImage	(none)
BackgroundImageLayout	Tile
Cursor	Default
⊞ FlatAppearance	
FlatStyle	Standard
⊟ Font	**Microsoft Sans Serif, 12pt**
Name	**Microsoft Sans Serif**
Size	**12**
Unit	**Point**
Bold	**False**
GdiCharSet	**0**
GdiVerticalFont	**False**
Italic	**False**
Strikeout	**False**
Underline	**False**
ForeColor	ControlText
Image	(none)
ImageAlign	MiddleCenter

You can change the appearance of controls using these properties – here are some ideas!

Example	Notes
	This font is set to **MV Boli** – and looks the worse for it!
	Changing the **BackColor**, **ForeColor** and **FlatStyle** properties.

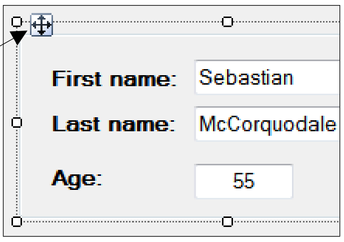
Moving Controls

To move controls, select them and then click and drag on them with the symbol:



Click and drag to move this textbox.

The only way to move this **GroupBox** is using this icon.



First name:	Sebastian
Last name:	McCorquodale
Age:	55

What we do!

		Basic training	Advanced training	Systems / consultancy
Office	Microsoft Excel	✓	✓	✓
	VBA macros	✓	✓	✓
	Office Scripts	✓		
	Microsoft Access	✓	✓	✓
Business Intelligence	Power BI	✓	✓	✓
	Power Apps	✓		
	Power Automate / PAD	✓		
SQL Server	SQL	✓	✓	✓
	Reporting Services	✓	✓	✓
	Report Builder	✓	✓	✓
	Integration Services	✓	✓	✓
	Analysis Services	✓		
Coding	Visual C# programming	✓	✓	✓
	VB programming	✓	✓	✓
	DAX	✓	✓	✓
	Python	✓	✓	

WiseOwl Training

Training | Internet | Intranet | Database systems
www.wiseowl.co.uk | (0161) 883 3606 | sales@wiseowl.co.uk