



# INTRODUCING VISUAL STUDIO

Presented by  
**STEVE HALLAM**

WebTech Training and Development  
[www.webtechcorp.co.uk](http://www.webtechcorp.co.uk)  
[hallam.steve@gmail.com](mailto:hallam.steve@gmail.com)



# Agenda

- Solutions and Projects
- Debugging
- Working with databases
- Deployment

# What version of Visual Studio?

- This presentation concentrates on Visual Studio 2010 Professional
- Most of this stuff will work in the free Express Version of VS 2010
- There is more functionality in VS Premium and Ultimate versions, at a high price – versions aimed at:
  - Architects and designers
  - Development
  - Database
  - Testing
  - Team managers

# Solutions and Projects Overview

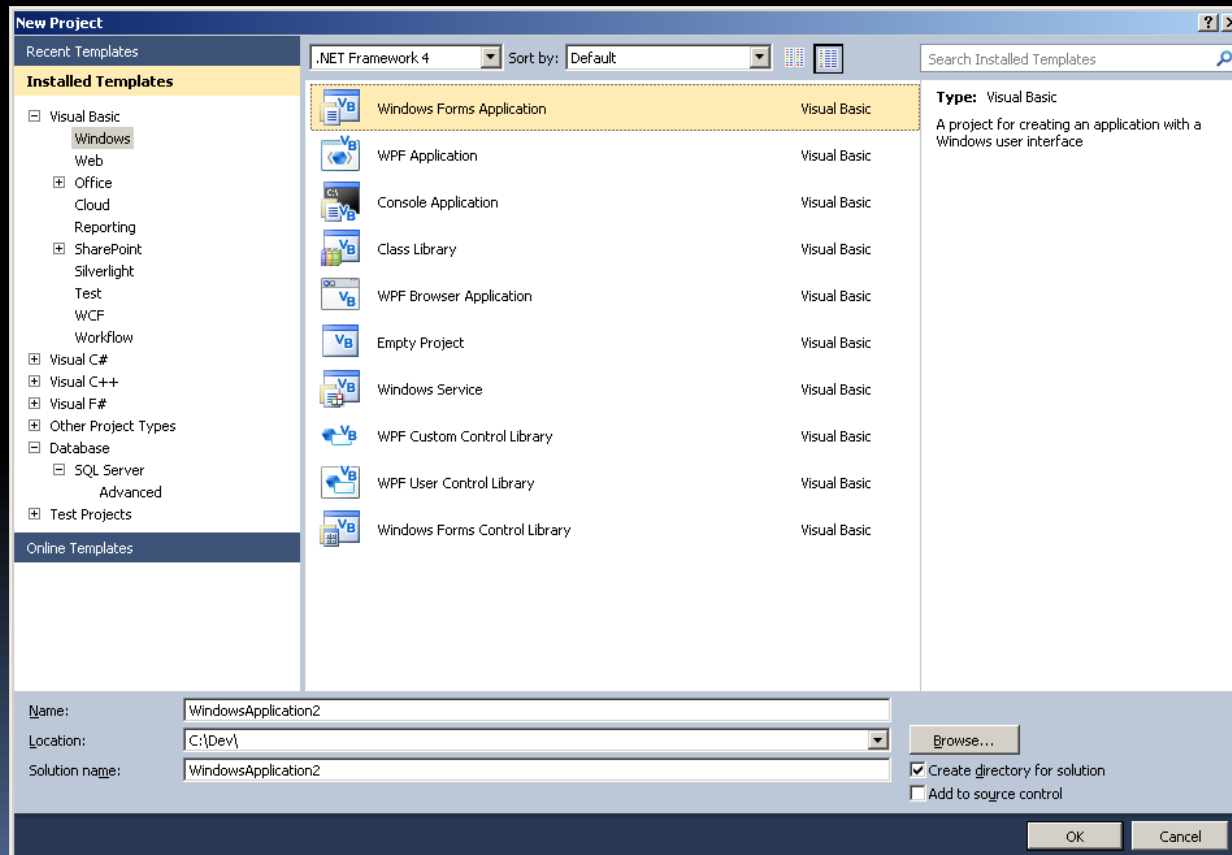
- The foundation of all VS based software development
- In a nutshell... to do anything useful:
  - You must have a Solution
  - The Solution must contain at least one Project
- A Project is needed for each build output
  - Usually a DLL or an EXE
- Projects contain source files

# Solutions

- A solution contains
  - A collection of one or more Projects
  - Info on the dependencies of the Projects
- You can only have one solution open at a time in VS
  - You can open multiple instances of VS
- Solutions cannot be nested
- A Project can belong to several solutions
  - Useful for large builds – but third-party build tools are more appropriate

# Creating a Solution

- Note location, Framework version dropdown, and “Create directory for solution” checkbox



# Project Types

- There are LOTS of project types!
- They will depend on:
  - The Framework version you are targeting
  - Any third-party tools you have installed
- All common project types can be built in either VB.NET or C#
- C++ has a different set of project types (e.g. ATL)

# Windows Projects

- An important and useful set of project types:
  - Windows Forms apps (EXE)
  - Class Library (DLL)
  - Console apps (command-line) (EXE)
  - Windows Services
  - WPF (Windows Presentation Foundation) apps (EXE)
  - Windows Forms and WPF Control Libraries (DLL)



# Creating a Solution for Windows Forms

(DEMO)



# Web Projects

(DEMO)

# Adding Projects to a Solution

- If you can see the Solution in the Solution Explorer, right-click the Solution and “Add > New Project” or “Add > Existing Project”
- If you cannot see the Solution in the Solution Explorer, you can add a Project from the File menu

# Managing files in a Project

- To add a file to a Project, right-click and select “Add” then “New Item” or “Existing Item”
  - You can directly add Project-specific items like Windows Forms
- You can also copy and paste or drag and drop items into the Project
  - They will be copied into the Project folder
  - Double-click them to edit within VS (e.g. images, text files) or using an external editor (e.g. Word files)

# References

- When you add Projects to a Solution, you must tell VS about the dependencies between them
- This is done by setting up *references*
- Until you reference an external component, you cannot reference its types in your Project
- NB: *CopyLocal* property of the reference



# References

(DEMO)



# Build Order

- Inspect the build order by right-clicking on the Project in Solution Explorer
- Use the Dependencies tab to change the build order if needed

# Building the Solution

- *Build > Build Solution* - compile only the project files and components that have changed since the last build
- *Build > Rebuild Solution* - clean the solution, then build all project files and components
- *Build > Clean Solution* - delete any intermediate and output files, leaving only the project and component files
- *Running the Project will also trigger a build!*

# Configuration Manager

- Default Build types in VS:
  - Debug – full debugging info, no optimisation
  - Release – no debugging info, full optimisation
- Use toolbar dropdown, or use the Configuration Manager to switch
- You can create customised build types



# Debugging overview

- Starting the debugger
- Breakpoints
- Single-stepping
- Displaying variables and expressions
- The call stack
- Advanced features

# Launching to Debug

- *Debug > Start (F5)* – runs program and attaches the VS debugger
- *Debug > Start Without Debugging* – Ctrl+F5
- *Debug > Break All (Ctrl+Alt+Break)* – breaks the program: debug ends when the target program ends
- *Debug > Stop Debugging (Shift+F5)* – aborts the program
- *Debug > Detach All* – leaves program running

# Breakpoints

- To set a breakpoint on an executable line of code: F9 (*Debug > Toggle Breakpoint*)
- Breakpoints Window is very useful – Ctrl+Alt+B
- Breakpoint properties:
  - Hit Count
  - Condition

# Exception Assistant

- Occasionally, a program will break because of a runtime error in your code (!)
- By default, the Exception Assistant will be displayed
- Can often help you work out the problem or offer a solution



# Debugging

(DEMO)

# Stepping Through Code - 1

- When execution is suspended, yellow highlight shows the *next* line of code to be executed
- Some options:
  - *Continue* (F5) or *Stop Debugging* (Shift+F5)
  - *Step Into* (F11)
  - *Step Over* (call function, stop at next line) (F10)
  - *Step Out* of function (Shift+F11)

# Stepping Through Code - 2

- Right-click on a line of code:
  - Show Next Statement
  - Set Next Statement
  - Run to Cursor (temporary breakpoint)
- You can also drag the yellow arrow (within a procedure!)

# Edit and Continue

- When the program is in break mode, you can normally edit the code and run without restarting! 😊
- Not everything can be changed!
  - VB.NET is quite relaxed – Declaration statements are the main exclusion
  - C# is more fussy but still a very useful feature
- Does not work in a 64-bit environment 😞

# Displaying Variables and Expressions

- Data Tips (break mode only)
  - Can be pinned in VS2010
- Variable windows:
  - *Locals* – variables local to the context or scope
  - *Autos* – variables used in the current and the preceding line of code
  - *Watch windows* – you can add any valid expression recognized by the debugger
  - *QuickWatch* - can display only one variable or expression at a time
  - *Immediate*



# Visualizers

- Use a dialog box or grid to display a variable or object in a way appropriate to its data type
  - Some visualizers enable you to modify data
- VS2010 includes these visualizers:
  - Text, HTML, and XML visualizers, all of which work on string objects
  - The WPF Tree visualizer
  - Dataset visualizer
    - Works for DataSet, DataView, and DataTable objects

# The Call Stack

- Answers the question “*How did we get here?*”
- VS can tell you the complete chain of execution back to the start of the thread
- Call Stack Window (Ctrl+Alt+C)
  - Double-click to go to the source code of a calling line
  - Press F9 to set a breakpoint on a calling line



# Debugging

(DEMO)



# Data Tools Overview

- The Server Explorer
- Setting up Data Connections
- Browsing data
- Creating and modifying database objects
- Manage changes

A vertical bar on the left side of the slide, consisting of several colored segments: a small pink square at the top, a grey square, a yellow square, and a long pink rectangle at the bottom.

# Server Explorer

(DEMO)

# Deployment

- Not a problem for Web applications...
  - But what about EXE-based applications?
- Two choices:
  - Windows Installer (creates MSI)
  - ClickOnce deployment
- New option in VS2010 – InstallShield LE
  - A freebie replacement for the VS Windows Installer
  - Download and registration required

# Windows Installer





# ClickOnce

- Addresses problems with Windows Installer:
  - ClickOnce allows automatic installation of updates in a very controllable way
  - With ClickOnce deployment, each application is self-contained and cannot interfere with other applications
  - ClickOnce deployment does not require administrative permissions and allows non-administrative users to install

A vertical bar on the left side of the slide, consisting of several colored segments: a small pink square at the top, a grey square, a yellow square, and a long pink rectangle at the bottom.

# ClickOnce

(DEMO)

# Visual Studio 2010 features

- Multiple-monitor support
- Editors rebuilt in WPF
- Silverlight WYSIWYG designer
- IntelliSense improvements
- Better support for legacy Framework apps
- Debugging
  - Pinned DataTips, Export/Import Breakpoints, Group Breakpoints by name



# Any Questions?

