Microsoft Visual Studio .NET: An Integrated Windows Program Development Environment

Using Microsoft Visual Studio .NET

- Self-contained environment for Windows program development:
  - creating
  - compiling
  - linking
  - testing/debugging
- IDE that accompanies Visual C++, Visual Basic, Visual C#, and other Microsoft Windows programming languages

Visual Studio Capabilities

- Generate starter applications without writing code
- View a programming project in many different ways
- Edit source and include files
- Build the application’s user interface visually
- Compile and link
- Debug an application while it runs
- Obtain online help
- Lots of others

Some Visual Studio Components

- **The Editors**
  - **C/C++, C#, VB source program text editors**
    - cut/paste color cues, indentation,
    - generates text files
  - **Resource Editors**
    - icons, bitmaps, cursors, menus, dialog boxes, etc.
    - graphical, WYSIWYG, Integrated
    - generates resource script (.rc) files
    - integrated with text editor
    - Done visually

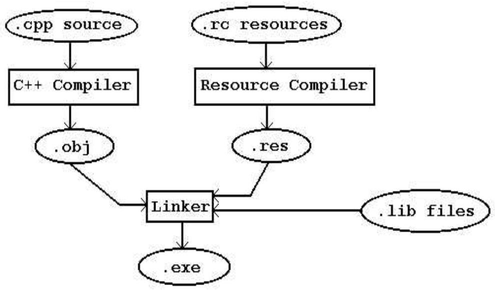
VC++ Unmanaged Code Compilers

- **C/C++ Compiler**
  - translates source programs to machine language
  - detects and reports errors
  - generates object (.obj) files for linker
- **Resource Compiler**
  - Reads .rc file
  - Generates binary resource (.res) file for linker

The Linker

- Reads compiler .obj/.res files
- Accesses C/C++/Windows libraries
- Generates executable (.exe or .dll)

VC++ Program Build Process



.cpp source
C++ Compiler
.obj
.rc resources
Resource Compiler
.res
Linker
.lib files
.exe

The Debugger

- powerful source code debugger
- integrated with all parts of Dev Studio
- Features
  - breakpoints
  - tracing through/over functions
  - variable watch windows
  - much more

The Wizards

- **AppWizard**
  - Windows code generator for MFC apps
  - automatically creates working program templates & skeleton code
- **ClassWizard**
  - facilitates easy extension of AppWizard-generated classes
  - creation of new classes and response functions
  - used to tailor AppWizard-generated MFC & .NET skeletons
  - Replaced by **Properties Window** in .NET

Help

- Can be accessed by:
  - 'Start Page' – 'Search Online'
    - To access the MSDN Online Library
  - Help Menu Item
    - Dynamic Help – context sensitive
      - pin it up as a fly-out
    - Search
    - Index

Win32 API Online Help

- 'Help – Contents'
  - Filtered by: (no filter)

MSDN Library
    Windows Development
        Win32 API
            SDK Documentation
                Reference
                    Functions by category
                    Functions in alphabetical order

MFC Online Help

- 'Help – Contents'
  - Filtered by: (Visual C++)

Visual Studio .NET
  Visual C++
    Visual C++ Reference
      Visual C++ Libraries
        MFC Reference
          Hierarchy Chart
          MFC Classes
          Class Library Overview
          MFC Classes (!!)
          etc.

MSDN Library (on Web)

- **Go to:** http://msdn.microsoft.com
  - **Search MSDN for desired topic**
  - Good URL for MFC:

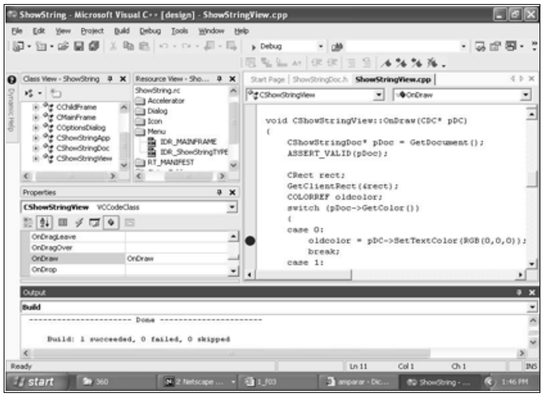
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/_mfc_class_library_reference_introduction.asp

Using Visual Studio .NET

- To prepare many kinds of applications
  - Win32 Console Applications (DOS programs)
  - Win32 API Apps
  - MFC apps
  - DLLs
  - Managed C++ Apps
  - ATL Projects
  - Web Services
  - Others

Visual Studio Layout

- Menu bar
- Several tool bars
- View Windows (left)
  - Solution Explorer
  - Class View
  - Resource View
  - Properties Window
- Working Area (right)
  - Text Editor to enter/modify source code
  - Resource Editors
- Output Window & Status Bar (bottom).
  - System messages (errors)

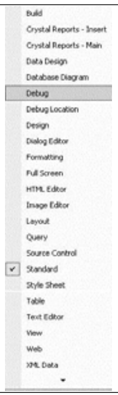


```
void CShowStringView::OnDraw(CDC* pDC)
{
    CShowStringDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);

    CRect rect;
    GetClientRect(&rect);
    COLORREF oldcolor;
    switch (pDoc->GetColor())
    {
    case 0:
        oldcolor = pDC->SetTextColor(RGB(0,0,0));
        break;
    case 1:
```

```
---------------------- Done ----------------------

    Build: 1 succeeded, 0 failed, 0 skipped
```

Toolbars

- Contain icons--instant routes to main menu functions
- Many of them
- May not be visible
- If not, right click on any visible toolbar
- Brings up following popup window
- Can activate a toolbar by clicking on its check box



Solutions and Projects

- Solution
  - A single application
  - Can contain one or more projects
    - In Managed applications, can be in different languages
  - Overall solution information stored in a .SLN file
  - Open this when you want to work on a solution
- Project
  - Collection of files:
    - Source, headers, resources, settings, configuration information

Important Visual Studio Generated Files

`.sln`	**Solution**
`.vcproj`	**Project**
`.c or .cpp`	**C/C++ source**
`.h`	**C/C++ header**
`.rc`	**Resource script**
`.res`	**Compiled resource**
`.ico`	**Icon**
`.bmp`	**Bitmap image**
`.exe`	**Executable program**
`.dll`	**Dynamic Link Library (if used)**

Temporary Visual Studio generated files

- **Many are very big and can (should) be removed!**

`.obj`	**Compiler machine code translation**
`.ilk`	**Incremental link file**
`.pch`	**Precompiled header (huge!!!)**
`.pdb`	**Precompiled debugging info**
`.idb`	**Incremental debug info**
`.ncb`	**Supports viewing classes**
`.aps`	**Supports viewing resources**
others	

Program Configurations

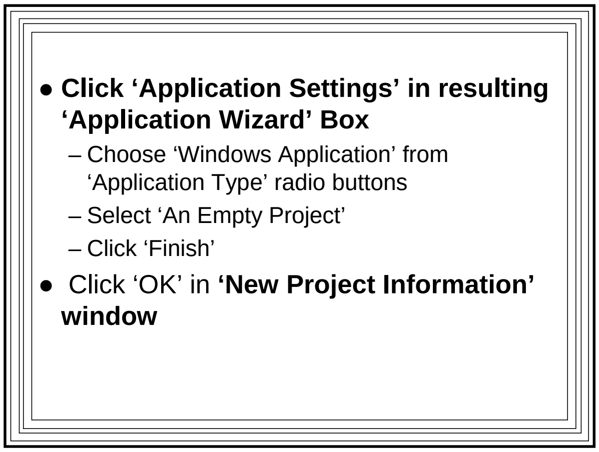
- Debug
  - appends debugging info
  - produces more and larger files
- Release
  - no debugging information
  - optimized for size & efficiency

Setting the Configuration

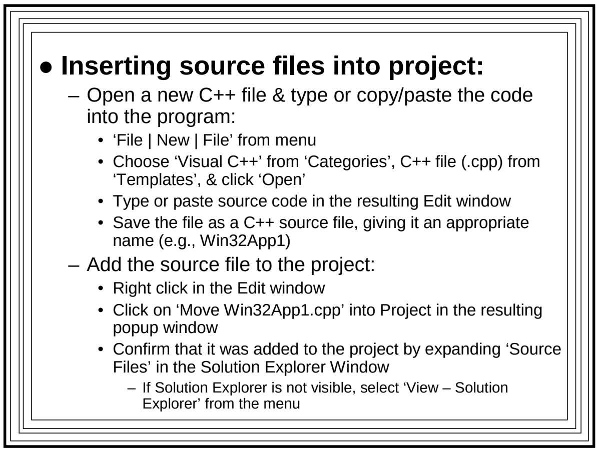
- Click "Build" on Main Menu
- Choose "Configuration Manager”
- Choose desired configuration (“Debug” or “Release”) in “Active Configuration Box”
- Default is “Debug”

Create a Win32 Application with Visual Studio

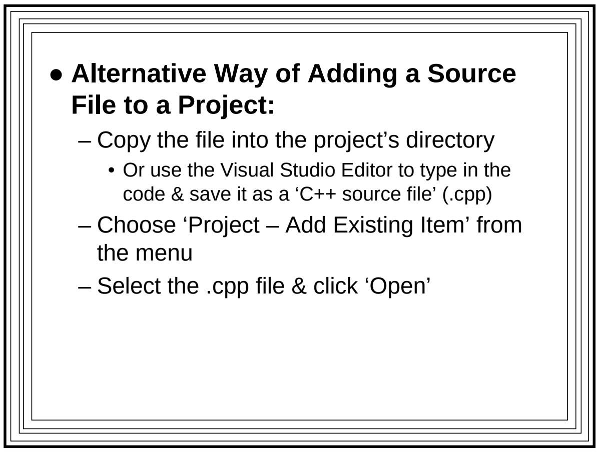
- **Startup**
  - click ‘Start’ on Task Bar – ‘All Programs’
  - ‘Microsoft Visual Studio .NET’ – ‘Microsoft Visual Studio .NET’
- **Creating a new solution**
  - ‘File’ – ‘New’ – ‘Project’ from menu bar
  - In ‘New Project’ box, select ‘Visual C++ Project’ from ‘Project Types:’ & click on ‘Win32 Project’ in ‘Templates’
  - Set the ‘Location’ to a convenient directory &name the project (e.g. win32app1)
  - ‘OK’



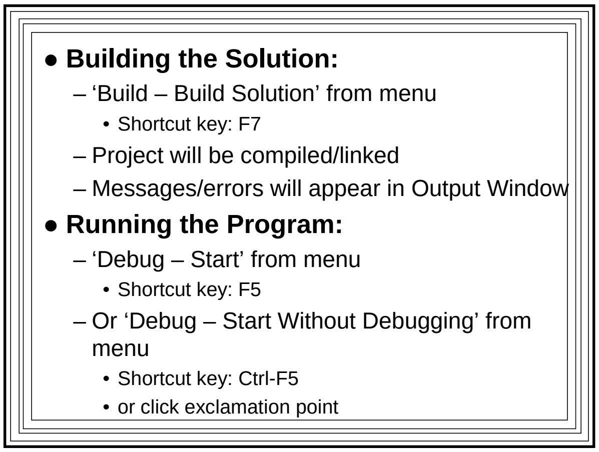
- **Click 'Application Settings' in resulting 'Application Wizard' Box**
  - Choose 'Windows Application' from 'Application Type' radio buttons
  - Select 'An Empty Project'
  - Click 'Finish'
- Click 'OK' in **'New Project Information' window**



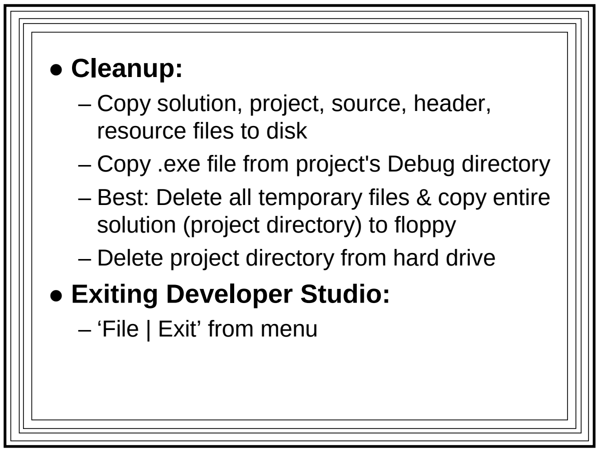
- **Inserting source files into project:**
  - Open a new C++ file & type or copy/paste the code into the program:
    - 'File | New | File' from menu
    - Choose 'Visual C++' from 'Categories', C++ file (.cpp) from 'Templates', & click 'Open'
    - Type or paste source code in the resulting Edit window
    - Save the file as a C++ source file, giving it an appropriate name (e.g., Win32App1)
  - Add the source file to the project:
    - Right click in the Edit window
    - Click on 'Move Win32App1.cpp' into Project in the resulting popup window
    - Confirm that it was added to the project by expanding 'Source Files' in the Solution Explorer Window
      - If Solution Explorer is not visible, select 'View – Solution Explorer' from the menu



- **Alternative Way of Adding a Source File to a Project:**
  - Copy the file into the project's directory
    - Or use the Visual Studio Editor to type in the code & save it as a 'C++ source file' (.cpp)
  - Choose 'Project – Add Existing Item' from the menu
  - Select the .cpp file & click 'Open'



- **Building the Solution:**
  - 'Build – Build Solution' from menu
    - Shortcut key: F7
  - Project will be compiled/linked
  - Messages/errors will appear in Output Window
- **Running the Program:**
  - 'Debug – Start' from menu
    - Shortcut key: F5
  - Or 'Debug – Start Without Debugging' from menu
    - Shortcut key: Ctrl-F5
    - or click exclamation point



- **Cleanup:**
  - Copy solution, project, source, header, resource files to disk
  - Copy .exe file from project's Debug directory
  - Best: Delete all temporary files & copy entire solution (project directory) to floppy
  - Delete project directory from hard drive
- **Exiting Developer Studio:**
  - 'File | Exit' from menu