Using Visual Studio

An Integrated Development Environment (IDE)
Terms

• Solution
  – A collection of projects
  – A complete application
  – May have multiple projects

• Project
  – A collection of related files
  – Linked to build a program

• Configuration –
  – A selection of projects & program files
  – Multiple ways to build a solution
Using the Wizard - 2

Database Support

Overview
Application Type
Compound Document Support
Document Template Properties
Database Support
User Interface Features
Advanced Features
Generated Classes

Database support:
- None
- Header files only
- Database view without file support
- Database view with file support

Client type:
- OLE DB
- ODBC

Data source:
Data Source...

Generate attributed database class
Bind all columns
Type:
- Dynaset
- Snapshot
### User Interface Features

#### Main frame styles:
- [x] Thick frame
- [x] Minimize box
- [x] Maximize box
- [ ] Minimized
- [ ] Maximized
- [x] System menu
  - [ ] About box
  - [ ] Initial status bar
  - [ ] Split window
- Child frame styles:
  - [x] Child minimize box
  - [x] Child maximize box
  - [ ] Child maximized

#### Command bars (menu/toolbar/ribbon):
- [ ] Use a classic menu
- [ ] Use a classic docking toolbar
- [ ] Use a browser style toolbar
- [ ] Use a menu bar and toolbar
  - [ ] User-defined toolbars and images
  - [ ] Personalized menu behavior
- [ ] Use a ribbon

#### Dialog title:
- MFC-mw2
Using the Wizard - 4

Advanced Features

Overview
Application Type
  Compound Document Support
  Document Template Properties
Database Support
User Interface Features
  Advanced Features
  Generated Classes

Advanced features:
- Printing and print preview
- Automation
- ActiveX controls
- MAPI (Messaging API)
- Windows sockets
- Active Accessibility
  - Common Control Manifest
  - Support Restart Manager
    - Reopen previously open documents
    - Support application recovery

Advanced frame panes:
- Explorer docking pane
- Output docking pane
- Properties docking pane
- Navigation pane
- Caption bar

- Advanced frame menu items
- Show/activate panes

Number of files on recent file list: 4
Configurations (advanced topic)

- There are 2 different uses of the term
  - Project Configuration
    - Options for compiling, linking, debugging, etc.
    - Accessed from Project/Properties
  - Solution Configuration
    - Selection of projects to be built into the solution
    - Accessed from the configuration manager
Solution example 1

• Solution: an accounting application
• Projects:
  – Manage checkbook
  – Manage investments
  – Compute taxes
• Programs
  – Change viewing options (a utility)
  – Read/write the database (another utility)
  – Convert data formats (another utility)
  – Buy/sell/trade investments (specific programs for the app)
  – Utilities for other projects in the solution
Solution example 2

• Solution: testing student programs

• Projects:
  – Assignment 1
    • Student 1 A1
    • Student 2 A1
  – Assignment 2
    • Student 1 A2
    • Student 2 A2
  – Common code
    • File 1
    • File 2
Creating configurations

• Define a configuration
  – Name
  – Active vs debug

• Set it as the active configuration
  – Do this in project properties

• Change configuration options
  – Exclude/include files from build
  – Load/unload projects

• Changes saved immediately
Project configuration

1. Select "base-hello" in the Configuration dropdown.

2. Set the Item Type to "C/C++ compiler".

(C) D. Foreman 8/21/19
Activate configuration
Major Areas of the IDE

• **Menu**
  – Commands: compile, build, run etc.

• **Solution Explorer**
  – Lists all files (.H, .C or .CPP, .RC)
  – Allows add/remove/load/unload, etc.

• **Properties**
  – Lists all properties of the selected object
  – Allows changes

• **Source window** – human readable text

• **Output**
  – Results of compile, link, build stages
Create a "project"

• Choose a good name for your "solution"
  – "my-win32-apps"
  – Change project name to the SPECIFIC app
    • e.g.; Basic-window, Advanced-Window, etc.

• Add new items
  – Stored in "my documents/Visual Studio 2013/Projects"
  – Projects folder can be ANYWHERE you specify
  – Or leave it in default and create a separate "source-code" folder somewhere (trickier)
Modify VS settings

• Specify location of any #include files
• Specify if using pre-compiled headers
• Specify build-type
  – DLL (your "solution" will be part of a DLL)
  – Type of program:
    • Win32API
    • Console (not for CS360)
    • MFC
• Specify language C++, C#
• Add any post-build operations
  – Rename executable
  – Move executable to another folder
Primary Commands (buttons)

• Compile
• Build solution (compile & link)
• Run (with or w/o debugging)
• Configurations
  – Allows for inclusion of different files for Build
  – One "solution" can = multiple runnable programs
    • Config 1: files A, B, C
    • Config 2: files A, C
    • Config 3: files A, B
  – Post-build allows renaming output file
  – Produce separate executables

(C) D. Foreman 8/21/19