



**CSE 1530**

**Introduction to Computer Use II:  
Programming**

Winter 2006 (Section M)

Topic F: External Files and Databases -  
Using Classes and Objects

Monday, March 20 2006

Bill Kapralos

CSE 1530, Winter 2006, Bill Kapralos

---

---

---

---

---

---

---

---

**Overview (1):**

- ▣ **Before We Begin**
  - ▣ Some administrative details
  - ▣ Some questions to consider
- ▣ **As An Aside**
  - ▣ Pull Down Lists (ComboBoxes)
  - ▣ Message Boxes
- ▣ **Topic Overview**
  - ▣ Overview/Introduction
  - ▣ As An Aside

---

---

---

---

---

---

---

---

**Overview (2):**

- ▣ **Opening and Reading a File**
  - ▣ Standard Dialog Window
  - ▣ Using the Standard Dialog Window

---

---

---

---

---

---

---

---

## Before We Begin

---

---

---

---

---

---

---

---

### Administrative Details (1):

- **No Lab Exercise to Submit This Week**
  - Nothing to submit today (Monday, March 20)
- **Correction For This Week's Exercise**
  - [Submit Exercise 6-8 and not Exercise 6-9](#)
  - Due March 27 2006
- **Test 2 And Various Exercises That Haven't Been Picked Up**
  - Available at the end of the lecture

---

---

---

---

---

---

---

---

### Administrative Details (2):

- **Test Annulment Forms**
  - Will be available from [March 27 - April 21](#) 2006 from the Computer Science Engineering Undergrad Office located in CSEB 1003
    - Office hours → 10:00am - 12:00pm & 2:00-4:30pm
  - Must be completed if you wish to drop either of your test grades (Test 1 and/or Test 2)

---

---

---

---

---

---

---

---

### Some Questions to Consider (1):

- What is the purpose of data validation ?
- How does Visual Basic allow us to validate data ?
- What is the `CausesValidation` property ?
- What is the `Validate` event handler ?
- What is the purpose of "white-space" use ?

---

---

---

---

---

---

---

---

### As An Aside...

---

---

---

---

---

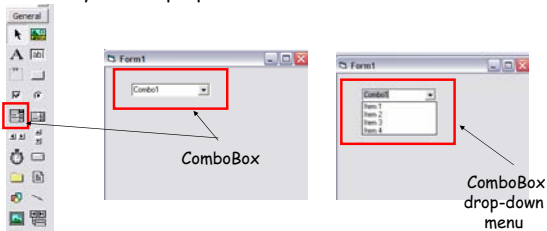
---

---

---

### ComboBox (1):

- What is a "Drop-Down" ComboBox ?
  - Allows you to have a list of items from which the user can make a selection
  - Very similar properties to ListBoxes



---

---

---

---

---

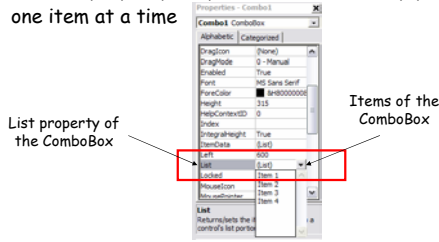
---

---

---

### ComboBox (2):

- Using a **ComboBox (Adding Items)**
  - Two ways to adding items
    - Design mode → in the properties window, choose the **List** property drop-down menu and simply add one item at a time



---

---

---

---

---

---

---

---

### ComboBox (3):

- Using a **ComboBox (Adding Items) (cont.)**
  - Two ways to adding items (cont.)
    - Run mode → using the `AddItem` procedure of the `ComboBox` (similar to the `ListBox` object)
    - Assume we have a `ComboBox` called `myComboBox`, the following will add three items to the `ComboBox`

```
myComboBox.AddItem("Item 1")
myComboBox.AddItem("Item 2")
myComboBox.AddItem("Item 3")
```
    - Can use the `Clear` method to "clear" the `ComboBox`

---

---

---

---

---

---

---

---

### Message Box (1):

- What is a Message Box ?**
  - A special type of Visual Basic window ("Dialog Box") that is used to display a message to the user
  - Can be used to convey a message but can also be called as a function that will return a value back to the caller indicating the user's response
  - In addition to the message, you can also include the following in the message box
    - Icon
    - Title bar caption
    - Command button

---

---

---

---

---

---

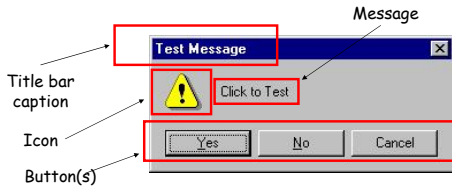
---

---

## Message Box (2):

### What is a Message Box (cont.)

- Can be used in many situations
  - When user has entered invalid data
  - When user has neglected to enter required data
  - To convey some form of information to the user



---

---

---

---

---

---

---

---

## Message Box (3):

### Creating a Message Box

- The MsgBox Statement General Form  
`MsgBox "Message String" [, Button/Icon] [, "Caption of title bar"]`
- Message String
  - Message you want to appear in the message box
- Button/Icon
  - Optional → determines the command buttons that will be displayed in the message box and any icons that will appear

---

---

---

---

---

---

---

---

## Message Box (4):

### Creating a Message Box (cont.)

- Button/Icon Options

Button/Icon	Value	Constant
OK Button	0	vbOkOnly
Critical Message Icon	16	vbCritical
Warning Query Icon	32	vbQuestion
Warning Message Icon	48	vbExclamation
Information Message Icon	64	vbInformation

---

---

---

---

---

---

---

---

### Message Box (5):

- **Creating a Message Box (cont.)**

- The MsgBox Statement General Form

`MsgBox "Message String" [, Button/Icon] [, "Caption of title bar"]`

- Caption of title bar
  - Optional → caption displayed in the message box title bar
  - If this is omitted, then the default caption will be the project name → this is considered sloppy programming practice!!

---

---

---

---

---

---

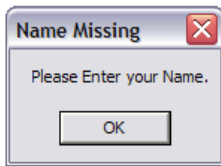
---

---

### Message Box (6):

- **Example of Using a Message Box**

```
If (txtName.text = vbNullString) Then  
    MsgBox "Please Enter Your Name.", vbOKOnly, "Name Missing"  
End If
```



---

---

---

---

---

---

---

---

## Topic F Overview

---

---

---

---

---

---

---

---

**Overview (1):**

- **So Far...**
  - Up until this point, all required user input has been given directly by the user, typically via TextBoxes
  - Next step is to write programs that access a file not already connected to the program
    - Provides much greater flexibility
    - Can make user input/output much more quicker thus increasing computation speed → displaying anything to the screen is VERY computationally expensive!

---

---

---

---

---

---

---

---

**Overview (2):**

- **What is a File ?**
  - Collection of stored data that is referred to by a specific name
  - Data can be read and modified
    - We can add new data to the file or change the existing data on the file
  - "Permanent" storage of data
    - Permanent when considering RAM that is only active while the computer is ON

---

---

---

---

---

---

---

---

**Overview (3):**

- **"Road Map"**
  - The focus of this chapter is file processing
    - We will examine how to input data to a program by reading the data from a file
    - We will examine how to access files through the local computer system → this will involve not only reading data from a file but also writing data to a file (e.g., output)
  - We will use standard Microsoft Windows dialog boxes for browsing the file system
    - The same for any Windows application → should be familiar to you!

---

---

---

---

---

---

---

---

**Overview (4):**

▫ **Working With Files Summary**

- Using the standard Microsoft file dialog we will obtain the name of the file
- The file will be opened using "new" Visual Basic classes called `FileSystemObject` and `TextStream` that provide the necessary tools for opening, reading and writing files
- Working with additional classes, we will be able to work with the file
  - We will focus files that have been constructed to contain fields and records (a simple `database`)

---

---

---

---

---

---

---

---

**Overview (5):**

▫ **Working With Files Summary (cont.)**

- We will see how to perform common operations
  - Deleting and adding records
  - Searching for a record
  - Scroll through the records

---

---

---

---

---

---

---

---

**Overview (2):**

▫ **Chapter Challenges**

- Working with files will allow us to understand new Object Oriented Programming concepts
  - Using new classes → before we can use the new class, we should understand the properties and methods of the class

---

---

---

---

---

---

---

---



**Overview (2):**

▪ **Main Concepts of This Chapter**

- Understanding and using new classes effectively
- Understanding the difference between **Private** and **Public** properties and subprograms
- Using the **Common Dialog** control
- Multiple Forms in a program
- Using the **FileSystemObject** and **TextStream** classes
- Creating and using a data source class
- Using the **RecordSet** class and **BindingCollection** class

---

---

---

---

---

---

---

---

**As An Aside (1):**

▪ **Other Approaches to File I/O**

- Although we will focus on an Objected Oriented approach to reading/writing to and from a file, this is not the only approach
  - We are of course using this approach to emphasize Object Oriented Programming
  - We want to obtain experience with creating/using classes/objects
- With Visual Basic, we can work with files in a non-OOP method
  - Using the VB **Open** statement

---

---

---

---

---

---

---

---

**As An Aside (1):**

▪ **Other Approaches to File I/O (cont.)**

- The **Open** statement is used in conjunction with the **Input** (for reading) and **Write** (for output) statements
  - Together with the **EOF** (End of File) function if the file was opened in **Input**, **Output** or **Append** mode or
  - In conjunction with the **Get** and **Put** statements and the **LOF** function if the file was opened in **Random** mode

---

---

---

---

---

---

---

---

## Opening and Reading a File

---

---

---

---

---

---

---

---

### Opening a File (1):

- **Standard Dialog Window**
  - Most MS Windows applications use a standard dialog window for locating and specifying a file to be opened
    - Allows the user to easily navigate through the file (directory) system to locate and open a file
    - Being a standard interface across all (most) MS applications ensures user familiarity → opening a file in Word is the same as opening a file in PowerPoint
    - Part of what is known as the [Microsoft Common Dialog Controls](#)

---

---

---

---

---

---

---

---

### Opening a File (2):

- **Standard Dialog Window (cont.)**
  - Since we're developing Windows applications with VB, we will of course employ this standard file dialog
    - Easy to incorporate in our VB applications

Standard Windows file dialog window



---

---

---

---

---

---

---

---

### Opening a File (3):

- **Common Dialog Control**
  - Although easy to use, the Common Dialog Control (and of course all its associated "controls") are not included in the standard VB development environment
    - Whenever we develop any programs that require its use, we must explicitly add it

---

---

---

---

---

---

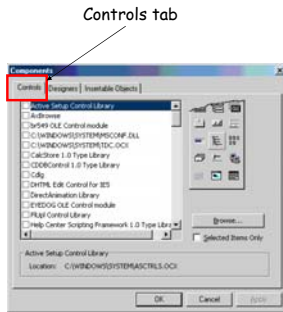
---

---

### Opening a File (4):

#### ▪ Adding the Common Dialog Control

- Begin a new standard VB project
- Under the **Project** menu select the **Components** option
  - This will cause the following window to appear



---

---

---

---

---

---

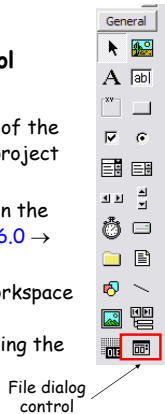
---

---

### Opening a File (5):

#### ▪ Adding the Common Dialog Control

- Ensure the **Controls** tab is selected
  - This allows you to add any number of the displayed components to your VB project that are needed
  - We are of course only interested in the **Microsoft Common Dialog Control 6.0** → select it & click "Ok"
  - Observe the toolbox in your VB workspace → you should observe a new icon representing the control for creating the standard MS dialog windows



---

---

---

---

---

---

---

---

### Opening a File (6):

- **Adding the Common Dialog Control (cont.)**
  - You can now add the file dialog control to your Form as you would add any other control from the tool box
  - This control is however slightly different from the other controls
    - Cannot be resized
    - When you run the project the control does not appear! → therefore, doesn't matter where on the form it is placed - its purpose is simply to make the `CommonDialog` object available to your program via the code you write

---

---

---

---

---

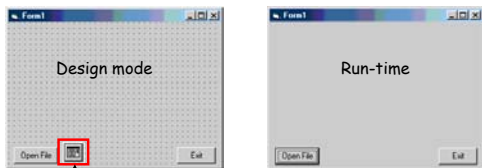
---

---

---

### Opening a File (7):

- **Adding the Common Dialog Control (cont.)**
  - Placing the Common Dialog Object on a Form and executing the program



Common Dialog control

---

---

---

---

---

---

---

---