Lecture 17 – Visual Basic for Applications: Object Models

Karl R. Wilcox
K.R.Wilcox@reading.ac.uk

Objectives

- Object Refresher
- Finding object information
- Code for manipulating objects
- The Word Object Model
- Today's practical
  - Automating Microsoft Word

Objects - Encapsulation

- **Encapsulation**
  - Implementation details are hidden
- **Properties**
  - Attributes / characteristics of the object that are externally visible
  - Can be used like ordinary variables
  - May be read-only
- **Methods**
  - Actions that can be carried out on an object
  - Can be used like function calls

Objects – Inheritance

- Objects are members of a class
  - All class members share the same properties and methods
- Class can be hierarchical
  - A sub-class “extends” a super-class
    - Can also be thought of as a child / parent relationship
  - The sub-class inherits all of the properties and methods of its super-class
    - Child class inherits from parent class
  - The child class may “override” some of the parent’s methods and properties
  - The child class may have additional properties and methods

Finding Object Information

- **Most "object based" development tools have very rich functionality**
  - **E.g. Java, Visual C++, VBA**
  - **100's of objects, 1000's of methods and properties**
  - **There is almost certainly one that does what you want, the problem is finding it!**
  - **There may also be several closely related objects / methods with slightly different characteristics**
    - **More of a problem with Java than Microsoft Office object models**

The Object Model

- **Recall, classes are hierarchical**
  - **The "tree" of child / parent relationships is the object model**
- **In Microsoft Office applications the topmost object is called *application***
  - **All other objects are children / grand children / … of the *application* object**
- **To use a specific method or property we must navigate through this hierarchy to the object that provides the method or property**

Using the Macro Recorder

- Use the macro recorder (covered in lecture 16)

1. Start the recorder
2. Carry out the action you want to perform (or something similar)
3. Stop the recorder
4. Study the resulting code to see what objects / methods and properties have been used
5. Cut / paste and edit as required

Using the Object Browser

- View → Object Browser or F2 from the VBA editor
- Use the browser to navigate the hierarchy
- Use the search function to find keywords
- Includes brief details of types / arguments
  - But has no narrative / description of usage etc.
- (Brief demonstration follows)

Using Help

- F1 from the VBA editor
  - But be careful! If you don't have a keyword selected you will get help on VBA and the VBA editor, not the application object model
- Can navigate the object model in a similar way
  - See "Microsoft *application* objects" under "Microsoft *Application* Visual Basic Reference"
  - Includes much more narrative + copy / pastable code examples
- Caution! The index search also goes through the application help, not just the object model
  - Use the Answer Wizard instead

Using List Properties / Methods

- This is a feature of the VBA editor
- Includes both:
  - Word completion
    - Type partial word then ctrl-spacebar
  - Automatic listing of methods and properties for the object you have just typed
    - As soon as the "." is typed
- Controlled by Tools → Options in VBA editor (default is on)

Typical Use

- You will usually find that you need to use all three previous methods in combination
  - List properties / methods for those you are familiar with and to save you typing
  - The object browser to navigate the object model and search methods and properties names exclusively
  - The help to find descriptions and examples

VBA Code For Objects

- There are some specific code features in VBA to manipulate objects
  - Properties can be treated exactly like variables
  - Methods can be treated exactly like function calls
- The *Object* data type can be used to refer to any object
  - Can use it as “shorthand” for an object deep in the hierarchy
  - Also has performance improvements in loops
- *Collections* are sets of similar objects
  - E.g. The *Documents* collection is the set of currently open objects of the type *Document* in Microsoft Word

The For Each … Next Loop

- **VBA provides specific syntax for navigating collections**

```
For Each Object In Collection
   [statements]
   [Exit For]
   [statements]
Next [Object]
```

- ***Object* must be of the *Object* type**

The "Is" Operator

- In conditional statements we can also use the "Is" operator
  - Similar to "=" , "<>", etc.
- It is used to compare two variables, both of which must be of the *Object* type
- It is TRUE if both variables refer to the SAME object
  - Not if they have the same contents e.g.
    - `Obj1 = Application.ActiveDocument`
    - `Obj2 = Application.ActiveDocument`
    - `Obj3 = Obj1`
  - Obj1 Is Obj2 = FALSE; Obj3 IS Obj1 = TRUE

The Word Object Model

- Is pretty complicated
- In reality, most important / useful stuff is in:
  - The *Documents* collection
  - The *Selection* object
  - The *Windows* collection
- Each *Document* object "exposes" almost every feature of the document
  - Down to individual words & letters

Points to Note

- Some objects can be used without going through the *Application* object
  - Document, Selection etc.

- Not everything is “exposed”
  - It is not possible to find the width of a particular word or character
  - This makes absolute positioning of text difficult
  - In most cases this should not be a problem

Next Week

- Custom Dialog Boxes
- Input validation
- Event handling
- This will be the last lecture this term
  - Week 10 will be the second COTS Test (5%)

Today's Practical

- Writing VBA Code
- Understanding the Word Object Model
- Automating some Word Operations

- **REMEMBER TO SIGN OFF ON THE REGISTRATION SHEET!**