Lecture 16 – Visual Basic for Applications: The Language

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

Objectives

- To cover the concepts of the VBA language
- Today's practical
  - Writing simple programs in VBA

Variables

- **Naming conventions**
  - Up to 255 characters long, case sensitive
  - Alpha-numeric + underscore
  - NO reserved words (but not a good idea anyway)

- **Variable declaration**
  - Not required (implicit declaration)
  - Can use type declaration characters for type
    - e.g. Qnt% is integer, X1& is long etc.
  - Better to declare explicitly
    - `Option explicit` in modules
    - or Tools → Options → Require variable declaration

Variable Types

- **VBA supports the following data types:**

- Boolean
- Byte
- Currency
- Date
- Decimal
- Double
- Integer
- Long
- Object reference
- Single
- String
- Variant

The default type is Variant (contains any of the above)

Variable Scope

- **Procedure scope**
  - Default scope for variables (implicit or explicit by `Dim`)
- **Private scope**
  - Declare outside procedures (top of module)
  - Available to all procedures in the module (= file)
  - e.g. `Private Counter As Integer`
- **Public scope**
  - Declare outside procedures (top of module)
  - Available to all procedures in the project
  - e.g. `Public Enemy As Boolean`
- **Static scope**
  - Like procedure scope but retains value across calls

Language Syntax and Layout

- Statements occupy a single line
  - There is no statement termination character (";" in 'C')
  - To continue a line use space + underscore as the last two characters
- Whitespace is not significant
- Procedures are called sub-routines –

```
Sub ProcedureName()
    strExample = "Some text"
End Sub
```

- Functions are called functions

```
Function Twice(str as String) As String
    Twice = str & " " & str
End Function
```

For Loops

```
For Variable = start  To end [Step size]
   [statements]
   [Exit For]
   [statements]
Next [Variable]
```

- Questions
  - If *start* and *end* are expressions are they evaluated once or every time round the loop?
  - What happens if *start* = *end*?
  - Can we modify *Variable* in the middle of the loop?

Do Loops

```
Do While condition
   [statements]
   [Exit Do]
   [statements]
Loop
```

- Other types of Do loop:
  - Do … Loop While
  - Do Until … Loop
  - Do … Loop Until
- What is the difference?

Further Loops

- There is a While … Wend loop available for backwards compatibility
  - Superseded by Do … While
- Loops can be nested
  - But for readability use `Next Variable` in for loops
  - Exit only takes you out of the nearest enclosing loop
  - Loops can only nest 16 levels deep
- Infinite loops
  - Are usually a bad idea
  - Use ctrl-break to interrupt a looping procedure

Conditionals

- **Single line conditionals**
  - `IF Age < 18 Then MsgBox "No vote for you"`
- **More complex conditionals**

```
If condition Then
    statement
    [statements]
[ElseIf condition Then
    statements]
[Else
    statements]
End If
```

- **Note: ElseIf is one word, End If is two words…**

Conditions

- **In a condition you can use the following comparisons**
  - = equal
  - <> not equal
  - < less than
  - > greater than
  - <= less or equal
  - >= greater or equal

And the following logical operators

- And
- Not
- Or
- Xor
- Eqv(same value)
- Imp (implies)

Select Case

```
Select Case Variable
   Case Expression
        Statements
   [Case Expression
        Statements]
   [Case Else
        Statements]
End Select
```

```
Select Case marks
   Case Is < 0, 0
    Msg = "Eh?"
   Case 1 To 39
    Msg = "Fail"
   Case 40 to 70
    Msg = "Pass"
   Case Is > 70
    Msg = "Distinction"
End Select
```

Functions

- Functions have a type and return a value
- There are two ways to invoke a function:
  - `Call Function ( Arguments )`
  - `Function Arguments`
- Arguments can be passed by name or by position:
  - `DateSerial ( 2004, 1, 29 )`
  - `DateSerial ( Day:=29, Month:=1, Year:=2004 )`
- Arguments can be passed by value or reference
  - `Function TestFunction ( ByVal Arg1 )`
  - `Function TestFunction ( ByRef Arg1 )`
    - By reference is the default (so be careful!)

Function Groups

Function Groups

- **Built in functions cover the following areas:**

- Type conversion
- Formatting
- String manipulation
- String comparison
- Date & Time
- Date differences
- Directory access
- Mathematical

Most function groups are rich in functionality (except the mathematical which is pretty basic). Consult the on-line help for more information.

Input / Output

- I/O in VBA is very limited
- If the application has one, you can write to the status bar
  - `Application.StatusBar = "This is my status"`
  - The application may overwrite this
- You can use Message Boxes to display output
- You can use Input Boxes to get input
- You can define custom dialog boxes (covered later)
- In general, you use the application itself, not the VBA I/O facilities
  - Remember, this is not a general purpose language

Message Boxes

- `MsgBox (Prompt [, buttons] [, title] _ [, helpfile, context]`
- MsgBox – name of the VBA function
- Prompt – text that appears inside the box
- Buttons – see next slide
- Title – appears in the window title
- Helpfile – name of the corresponding help file
- Context – Index into the help file

- Use constant `vbCr` to insert new lines in prompt

Choosing Buttons

- You can choose the button types, icons, default buttons and window modality
- All by adding various constants
- 5 button groups:
  - `vbOKOnly, vbOKCancel, vbAbortRetryIgnore, vbYesNo, vbRetryCancel`
- 4 icon types
  - `vbCritical, vbQuestion, vbExclamation, vbInformation`
- 4 default buttons
  - `vbDefaultButton1, vbDefaultButton2, etc.`
- Modality option
  - `vbSystemModal`

Which Was Pressed?

- MsgBox is a function (not a procedure)
- Therefore it returns a value
  - We can assign to a variable or use it in an expression
- VBA defines constants for use in testing return value:
  - `vbOK`
  - `vbCancel`
  - `vbAbort`
  - `vbRetry`
  - `vbIgnore`
  - `vbYes`
  - `vbNo`

Input Boxes

- `InputBox (Prompt [, title] [, default] [, xpos] [, ypos] [, helpfile, context]`
- MsgBox – name of the VBA function
- Prompt – text that appears inside the box
- Title – appears in the window title
- Default – default value to return
- Xpos, Ypos – Numeric value for window position
  - Measured in “twips” (about 12 to the pixel)
- Helpfile – name of the corresponding help file
- Context – Index into the help file

Getting Input Values

- InputBox is a function, returning a string
- Assign to a variable, or use directly
- Use the conversion functions to test and convert to the required data type

Next Week

- Object Refresher
- Finding object information
- Code for manipulating objects
- The Word Object Model

Today's Practical

- Writing VBA Code
- Using message and input boxes
- String manipulation
- Simple loops and conditionals

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