



# Lecture 15 – Introduction to Visual Basic for Applications

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



# Objectives

- To review software customisation
- To introduce VBA
- Today's practical
  - Recording macros in Word, assigning macros to menus and button bars



# Software Customisation Levels

- Used “out of the box” (default settings)
- Operating System (User Interface) settings
- Application preferences
- User Interface (menu / button) modifications
- Extensions / Plug-ins
- Macro based automation
- Program / script based automation



# What Is VBA?

- **VBA is a programming language**
- **Specifically, it is a scripting language for application automation**
- **It shares many concepts and syntax with Visual Basic**
- **It includes an Integrated Development Environment**
  - IDE
- **It has Object Orientated features**
  - In particular the COM object model for application automation



# Applications Supporting VBA

- MS Word
- MS Excel
- MS Powerpoint
- MS Outlook
  - (NOT Outlook express)
- MS Access
- MS Frontpage
- MS Project
- Wordperfect
- AutoCAD
- Visio
- + Increasing numbers of others



# What VBA Isn't

- It is NOT “write-once run-in-any-application” code
  - To save the currently open document
    - ActiveWorkbook.Save – Excel
    - ActiveDocument.Save – Word
    - ActivePresentation.Save – Powerpoint
  - The syntax, statements and generic functions are common, the application automation is specific
- It is NOT a general programming language
  - It cannot be used to develop stand-alone applications
- It is NOT a true Object based language
  - But it does have some OO features



# Where is VBA?

- It runs as part of the host application
- It crashes at the same time as the host application
- There is no VBA “executable”
  - However, VBA is not typically loaded into memory with the application
  - Loaded on demand to save memory and other resources



# Where Is The VBA Code?

- **It depends!**
- **Word**
  - Defaults to storing in Normal.dot
  - Can store in any document or template
- **Excel**
  - Default is Personal.xls macro workbook
  - Can store in any workbook or template
- **Powerpoint**
  - Has no default
  - Can store in any presentation or template
- **Project**
  - Default to Global.mpt, in ProjectGlobal
  - Can store in any project file



# Writing Code

- We can write code directly, using the Integrated Development Environment (IDE)
- We can use the editor of our choice and copy / paste into the IDE
- We can record user actions as a macro
  - This writes the code for us
  - This is the subject of today's practical



# Why Use the Macro Recorder?

- **It always produces working code**
  - Provided you run it under similar conditions to which it was recorded in
- **Quick and easy to use**
  - Although slightly different in every application
- **Sometimes the quickest way to find the object, method or property you need**
  - COM object models are often very big
  - There is often overlap or mismatch of names and concepts
  - Workbook Vs Document, Outline Vs Outline



# Why Not Use the Recorder?

- **It creates a lot of code – not all of it wanted**
  - If you change one item in a dialog box, the macro recorder includes the settings of every item in that dialog
- **Macro code will only work in the active document**
  - Cannot manipulate other documents
- **Some VBA features must be programmed**
  - Custom dialog boxes, user forms etc.
- **The “initial conditions” must be the same**
  - E.g. selection highlighted or not



# You Can't Use the Recorder...

- If the host application doesn't support it!
- Microsoft licences the VBA language to 3<sup>rd</sup> parties
  - NOT the macro recorder
- Most Microsoft applications include a macro recorder
  - 3<sup>rd</sup> party software (or ex-3<sup>rd</sup> party software) does NOT
  - e.g. Wordperfect, AutoCAD, Visio



# Why Customise?

- Automating repetitive tasks
- Assisting with repeated, labour-intensive tasks
- Automatic creation of files & reports
- Automating tasks for less skilled workers
- Creating “front end” menu programs (wizards)
- Ensuring consistency
- Avoiding errors
- Sharing with others (showing off!)
- *Which of these can be addressed by macros?*



# Further Reading

- Book – “Mastering VBA 6”,  
**Guy Hart-Davis, SYBEX, 0-7821-2636-7**
  - Do NOT confuse with books on VB !
- Website:
  - [www.msdn.microsoft.com/vba](http://www.msdn.microsoft.com/vba)



# Today's Practical

- Recording Macros
  - Assigning macros to toolbars
  - Starting the VB editor and finding the code
  - Using VB help
- 
- **REMEMBER TO SIGN OFF ON THE REGISTRATION SHEET!**