



On-the-Job Training in CRIL

June 1, 2009

Geoffrey D. Reyes (g.reyes@irri.org)

Crop Research Informatics Laboratory (IRRI)

Introduction

Geoffrey D. Reyes

- TRACE College
- 4th Year B.S. Computer Science
- Two 300 hour periods of On-the-Job Training
 - (1) April-May
 - (2) June-October
- Projects in CRIL:
 - Excel Import/Export/Delete VB Components Add-In
 - Access Import/Export/Delete VB Components Tool
 - R Import ODS table to Dataframe Script

Excel Import/Export/Delete VB Components Add-In

Description

- Import/Export/Delete VB Components in XLS or XLA files and installed add-ins
- MS Excel 2003/2007
- Able to process multiple import/export/delete

VB Components

- Excel Document Objects (worksheets/ThisWorksheet - codes only)
- User Forms (.frm files)
- Modules (.bas files)
- Class Modules (.cls files)

Notes

- Enable running macros
- Check the "Trusted access to Visual Basic Project" and "Trust all installed add-ins and templates" in the macro security section

Excel Import/Export/Delete VB Components Add-In

– References required:

- Microsoft VBA Extensibility 5.3 extensions or better

C:\Program Files\Common Files\Microsoft
Shared\VBA\VBA6\VBE6EXT.OLB

- Microsoft Shells Controls and Automation

C:\WINDOWS\system32\SHELL32.dll

Problems encountered

- Excel Document Objects (Worksheets and ThisWorkbook) are imported/exported as class modules instead of Excel Document type objects
 - Import as class module
 - Copy the codes to the corresponding worksheet
 - Delete the imported class module

Excel Import/Export/Delete VB Components Add-In

- Some virus detection softwares detected codes as a poly macro virus
 - Auto_Open and Workbook_Activate event procedures
 - Automatically run macro upon opening the workbook
 - Workbook_AddinInstall event procedure to automatically create the custom toolbars upon add-in installation

References

- Craig S. Taylor
 - This file is the main reference that was used.
 - <http://sourceforge.net/projects/vba-cvs-impexp/>
- Chip Pearson (cpearson@kc.rr.com)
 - Importing excel document objects
 - <http://www.cpearson.com/Excel/VBE.aspx>
 - Getting the folder directory path
 - <http://www.cpearson.com/excel/BrowseFolder.aspx>

Excel Import/Export/Delete VB Components Add-In

- Helpful sites
 - Changing module names of worksheet objects
 - http://exceltip.com/st/Change_the_worksheet_codemodule_names_using_VBA_in_Microsoft_Excel/406.html
 - Adding custom toolbars
 - http://www.vbaexpress.com/kb/getarticle.php?kb_id=808
 - Creating add-ins
 - <http://www.ozgrid.com/VBA/excel-add-in-create.htm>

How to use / Demonstration

Question / Comments / Suggestions

Access Import/Export/Delete VB Components Tool

Description

- Import/Export/Delete VB Components in MDB files
- MS Access 2003/2007
- Able to process multiple import/export/delete
- Converted version of the Excel Import/Export/Delete VB Components Add-In

VB Components

- Access Document Objects (Forms - codes only)
- User Forms (.frm files)
- Modules (.bas files)
- Class Modules (.cls files)

Notes

- Enable running macros
- Check the "Trusted access to Visual Basic Project" and "Trust all installed add-ins and templates" in the macro security section
- User forms and Access forms are different - User forms are created/used in the VB editor while Access forms are created/used in the main application.

Access Import/Export/Delete VB Components Tool

- References required:
 - Microsoft VBA Extensibility 5.3 extensions or better
C:\Program Files\Common Files\Microsoft Shared\VBA\VBA6\VBE6EXT.OLB
 - Microsoft Shells Controls and Automation
C:\WINDOWS\system32\SHELL32.dll\
 - Microsoft Forms 2.0 Object Library
C:\WINDOWS\system32\FM20.DLL
 - Microsoft Office 12.0 Object Library
C:\Program Files\Common Files\Microsoft Shared\OFFICE12\MSO.DLL

Limitations

- Does not support MDA files and installed Add-ins
- Importing Access Document Objects (Forms) would require manual work if the object to be imported does not exist
- Requires to open the file before any function can be done
- Not an add-in (single file that needs to be open to use)

Access Import/Export/Delete VB Components Tool

Problems encountered

- Access VB editor behaves differently than Excel VB editor Import as class module
 - In Excel, every active spreadsheet shares one VB Editor and getting all active worksheets with VB Components are listed in the VB Editor.
 - In Access, every active database has its own VB Editor. This makes it hard for programmers to call and refer different objects.
 - This also made the tool not able to list all the active databases. Listing all active databases is needed for the user to be able to select to which they will need to import/export/delete VB Components.
 - To overcome this, the tool opens the file and sets it as the active access object. From there on, importing/exporting/deleting VB Components can be achieved.

Access Import/Export/Delete VB Components Tool

- Access VBA limitations encountered
 - In Excel, you can create a worksheet and change its properties using VBA programming
 - In Access, you cannot do this without administrator rights or having to interact with the user for input.
 - This made the tool it not able to programmatically create an Access document object if the user imports one and the corresponding object does not exist in the Access file
 - Another problem is that Access doesn't automatically generate the corresponding VB Component of Access document objects. For Access document Objects to have a corresponding VB Component in the VB Editor, the user must manually add a command to the object. The object must have at least one command in it. Thus, requiring users to manually create the Access document object and add a command to generate the corresponding VB Component

Access Import/Export/Delete VB Components Tool

- Access Document Objects (Forms) are imported/exported as class modules
 - Import as class module
 - Copy the codes to the corresponding worksheet
 - Delete the imported class module
- Creating an Access Add-in requires some editing in the windows registry
 - Creating and running an add-in requires some editing in the windows registry. Because of this, users will need administrator rights to be able to run an add-in.

Access Import/Export/Delete VB Components Tool

References

- Microsoft Excel Import/Export/Delete VB Components Add-In
 - Main reference that was used.
- Chip Pearson (cpearson@kc.rr.com)
 - Importing excel document objects
 - <http://www.cpearson.com/Excel/VBE.aspx>
 - Most of the codes are also applicable to Access VBA

How to use / Demonstration

Question / Comments / Suggestions

R Import ODS table to Dataframe Script

Description

- Imports the table data of an ODS file to an r dataframe
- Source: ROpenOffice package by Duncan Lang
- This function reads the “content.xml” file in the ODS file (an ODS file is an archive)

Syntax: `<data frame name> = read.ods("<ods file>", "<sheet name>")`
ex. `df = read.ods("Spreadsheet.ods", "Sheet1")`

Notes

- Requires the XML package
- Copy and paste script code to R

R Import ODS table to Dataframe Script

Functions

- `read.ods`
 - Main function for importing data
 - Uses the `unz` function in the base package to connect to the ODS archive
 - Uses the `readLines` function to copy all text in the “content.xml”
 - Uses the `xmlParse` function in the XML package to parse the XML data to R
- `read.odsList`
 - Function to show all worksheets in the ODS file
 - `read.odsList(file)`

R Import ODS table to Dataframe Script

Limitations

- Proper table structure:
 - Rectangular table with at least 2 columns and 2 rows (one is for the headers and one is for a row data value)
 - Column headers located at the first row
 - Column headers must start at the first column of the sheet
 - May contain empty row cell values
 - Full blank rows are not included in the dataframe (skipped)
 - If all sheets are to be imported, each sheet should contain values and contains the proper table structure
- Table value types are stored in the dataframe as character and numeric only
 - Date, Time, Percentage, Scientific, fraction and Boolean

R Import ODS table to Dataframe Script

Problems encountered

- Read xml file as pointer
 - By extracting the XML file in the ODS archive, you will be able to easily parse the XML to R. However, it uses more resources and sometimes would require administrator rights.
 - Use the unz and readLines function
- Subscript out of range error
 - It occurs when importing an ODS file with a blank sheet in the last sheet
 - Import specific sheet using sheet name and implement proper table structure

R Import ODS table to Dataframe Script

Reference

- Duncan Lang
 - <http://www.omegahat.org/ROpenOffice/>
 - This package uses the Rcompression package to extract the XML file in the ODS archive then parses the data by using the XML package

How to use / Demonstration

Question / Comments / Suggestions

OJT Experience

- Help with the transition from being a student to a worker
- CV/Resume
- Time management
- Studying
- Working with other people
- Free food
- Talking to the boss (in english)
- Suggestions/Comments
 - Hire more OJTs and let them work together
- **Thanks!!!**
 - To Thomas, Carlos, Achu, Warren, Lou, CRIL boys, to the people who opens the door for me, to the people who greets me, to everyone

Cheers!