

The Application of VBA in Data Conversion for Network Platform Database

Guoliang Dong^{1, a}, Fujia Liu^{1, b}

¹Research Institute of Highway Ministry of Transport, Beijing, China

^a GL.DONG@rioh.cn, ^b FJ.Liu@rioh.cn

Keywords: VBA, Heterogeneous data, Database, Data conversion, Excel, Word

Abstract. Heterogeneous data sources are very common in web site building. All sorts of materials should be processed and converted to a standard format to be imported in the database for a web site about motor vehicle techniques. An Excel file embedded VBA codes can easily extract information from Word files, transform application data to a standard format and automatically create reply paper letters.

Introduction

It's a routine thing to manage user accounts and application data for a networking platform. Data should be pre-processed and prepared when the platform is deployed [1]. Data engineers process different kinds of information and convert them to a standard format which is a very odd and time-exhausted work. Heterogeneous data [2] should be processed and proper methods could free data engineers from the odd and hard work. A web site of motor vehicle techniques is built to manage information about auto manufacturers, ordinary users, vehicle repair techniques, etc. Hundreds and thousands of raw materials are collected and engineers have to process the Word and Excel format files. Build some VBA functions and subroutines in an Excel file to process the information and convert them to a standard format Excel file which will be import to the web site database later.

Data proceeding procedure

Web site users provide paper documents and their electronic version to the site operators who check the validity of these materials. The engineers process all the materials, convert them to standard Word or Excel files by some template files and import the processed Excel files to the web site [3,4]. A reply paper letter will be sent to the qualified site user to notice them the account and password to logon in the site.

Implement of data process

Data should be processed and prepared before the platform deployment. The process and preparation of background data include user information extraction, application data conversion, Excel file merge, etc. Also the offline business such as making a paper reply letter, printing and sending back the letters to the qualified user should be done.

3.1 User information extraction

The site users provide paper and electric version materials which include the user name, user type, phone, address etc. The site managers check the information and authorize them as a qualified user once the materials check passed. Data engineers process the Word files, extract information form the tables, save them as an Excel file by a template format file and import them to the platform database. The main VBA codes embedded in an Excel [5,6] to implement the above function are list as below.

```
myfile = Dir(aPath & "*.doc")  
docNos = docNos + 1  
DocNmArr(docNos) = myfile  
Do While myfile <> ""
```

```

myfile = Dir
If myfile = "" Then
    Exit Do
End If
docNos = docNos + 1
DocNmArr(docNos) = myfile
Loop
Set wdApp = CreateObject("Word.Application")
With wdApp
For i = 1 To docNos
    taskName = bPath & DocNmArr(i)
    .Documents.Open Filename:=taskName, revert:=False
    tbs = .ActiveDocument.Tables.count
    AAstr(1) = .ActiveDocument.Tables(1).Cell(1, 1).Text
    .....
    .ActiveDocument.Close savechanges:=wdDoNotSaveChanges
    With Workbooks(ThisXslFileName).Worksheets("Sheet1")
        .Cells(1, 1).Value = AAstr(1)
    .....
    End With
Next i
End With

```

3.2 Business data transformation/Word Table to Excel

Some of the electric business documents are Word files. Information should be extracted from the tables in the Word files [7] and be converted to Excel [8] which will be imported in the web site database. The main VBA codes implementing the above function are list as below.

```

With wdApp
For i = 1 To docNos
    taskName = bPath & DocNmArr(i)
    .Documents.Open Filename:=taskName, revert:=False
    With .ActiveDocument
        Set xlapp = CreateObject("Excel.application")
        xlapp.Visible = True
        Set appbook = xlapp.Workbooks.Add
        wdApp.ActiveDocument.Select
        wdApp.Selection.Tables(1).Select
        wdApp.Selection.Copy
        appbook.Worksheets(1).Range("A1").Paste
        appbook.SaveAs bPath & DocNmArr(i) & ".xls", FileFormat:=xlExcel8
        xlapp.DisplayAlerts = True
        xlapp.Quit
        Set xlapp = Nothing
    End With
    .ActiveDocument.Close savechanges:=wdDoNotSaveChanges
Next i
End With

```

3.3 The imported data check /Merge Excel files

It is necessary to check the data integrity and consistency after data imported in the site database. Merging hundreds of Excel files into a new one and checking data line by line or random selection is

a good method. It is easy to process files through a folder and combine them as a new one by embedded VBA codes [9]. The main codes are list as below.

```
FName = Dir(MyPath & "*.xls*")
Do While FName <> ""
Set W = Workbooks.Open(MyPath & FName)
With Workbooks(1).Sheets(1)
    For G = 1 To Sheets.count
        W.Sheets(G).UsedRange.Copy .Cells(.Range("A65536").End(xlUp).Row + 1, 1)
    Next
W.Close False
End With
FName = Dir
Loop
```

3.4 Reply letters automatic creating

The site managers should print reply letters and send them to the qualified users. In the paper letter the below items should be included: user name, account, password, date etc. Editing hundreds of such letters is an odd thing. By the aid of Bookmark inserted in the Word file, automatic creating letters can be implemented [10,11]. Edit a Word file, define the replaced text as a Bookmark and save the file as a template document. An Excel file embedded VBA codes will load user's information and replace the Bookmarks in the template file and save it as a new one (reply letter file) named by the user name. The main codes are list as below.

```
With wdApp
For i = 1 To docNos
    taskName = bPath & DocNmArr(i)
    .Documents.Open Filename:=taskName, revert:=False
    With .ActiveDocument
        Set xlapp = CreateObject("Excel.application")
        xlapp.Visible = True
        Set appbook = xlapp.Workbooks.Add
        wdApp.ActiveDocument.Select
        wdApp.Selection.Tables(1).Select
        wdApp.Selection.Copy
        appbook.Worksheets(1).Range("A1").Paste
        appbook.SaveAs bPath & DocNmArr(i) & ".xls", FileFormat:=xlExcel8
        xlapp.DisplayAlerts = True
        xlapp.Quit
        Set xlapp = Nothing
    End With
Next i
End With
```

Conclusions

An Excel file embedded VBA codes easily extract information from Word files and convert application data to imported format. Hundreds of user tables have been extracted and 16 thousands of application records have been imported into web site database. The program codes work well in Windows 7, Office 2010 and Visual Basic 6.0 environment.

References

- [1] PANG Zhi-nian and NIU Di: Design of Information System Network Platform, Computer & Network[J],2013,vol.12.
- [2] Li Yu-tong: Heterogeneous Database Conversion[D],2012.
- [3] PENG Lei and LI Xian-guo: Design and implementation of Excel mass data importing system, Modern Electronics Technique,vol.14,2014.
- [4] Gao Zhi-chao and Zhang Zhi-kui: Importing Excel into MySQL, Computer Programming Skills & Maintenance,vol.19,2012.
- [5] Yunzhou Stiduo, Excel 2000 VBA[M], Posts & Telecom Press,2000.
- [6] Yunzhou Stiduo, Word 2000 VBA[M], Posts & Telecom Press,2000.
- [7] Microsoft Corporation, Microsoft Word Help[DB/OL].
- [8] Microsoft Corporation,Microsoft Excel 2000 Language Reference [M], Beijing Hope Electronic Press,1999.
- [9] Yao Wen-tao, Excel VBA Application Programming Instances[M] , Tsinghua University Press, 2009.
- [10] Wang Shun-xing,and Yan Mei-dai,Word 2000 VBA Tutorial and Instances Analysis[M] ,Beijing University Press,2001.
- [11] Mingri Tech., Gao Chun-yan,Liu bin-bin,and Wang Bin, Visual Basic Programming Bible[M] , Posts & Telecom Press,2007.