

Research and Implementation of Word Documents Generation Based on C#

Cai Xiaoyan¹, Shen Wei², Li Longteng³, Ge Yu⁴, Lu Lina⁵, Zeng Li⁶

1. Wuhan Ordnance Non-commissioned Officer Academy of PLA, Wuhan, China

2. Powerchina Hubei Electric Engineering Corporation, Wuhan, China

Email:xycai1984@163.com

Abstract—To solve the problem that report forms generated in information management system is not convenient for modification and conservation, a method of generating Word documents with texts and images automatically is put forward based on COM components of Office. The realization of the method is discussed in detail, which is chiefly used for in depth office function development. At present, the method has been applied in the information management system of faculty member successfully to implement the automatic generation of Word documents. The operation practices show that the method has high practical value and good generality.

Keywords—C#; Word Documents; COM components

I. INTRODUCTION

The generation of reports is fairly significant link in the process of development of database application system. Transferring data of database into fine reports for users is a basic need for database application system. Although mounts of development tools of application program have their own reports generation system, Word, as output tool of application system, has many irreplaceable advantages. Firstly, it can create documents with images, texts and tables. Secondly, it can generate more complex reports, which are including multi-level nested tables and diagonal tables, and so on. Thirdly, users can customize and modify the document templates [1]. Fourthly, users do further processing for generated documents. Fifthly, users can save and publish the generated documents.

It is the problem developers often have to face that how to combine Word's powerful document processing capacity with database application system, to replace traditional report output and automatically create Word documents with texts and images. For example, in the information management system of faculty member, we need to combine C# and SQL Server 2005 database to generate a Word document with information including the personal information, teaching information, edited material information, published paper information and scientific research information, etc. During the process of design and development, I have found that we could solve this problem effectively with Office COM components provided by Microsoft. Based on the design and development of faculty member's information management system, the paper discusses the key technology of development of information system deeply, and the specific

ideas and method to export data of database into Word documents with texts and images.

II. DESIGN IDEAS

In order to export data of database into Word document, steps are as follow. First, we need to import COM components, create and open Office document in the program. Then read data or images of database and write down it into Word. At last, save the corresponding Word document and release the memory of the Word object. The flow chart of specific design idea is shown in Fig.1 below.

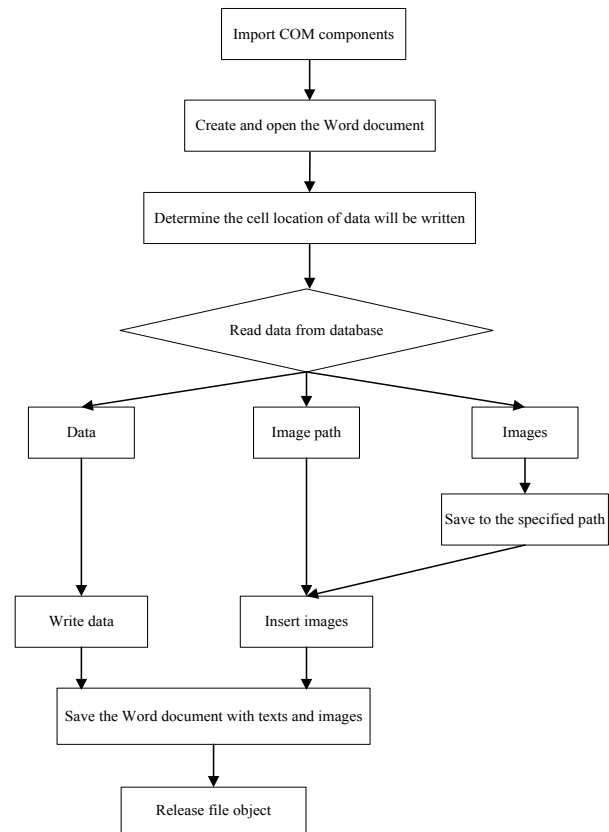


Fig.1. The flow chart of generating Word document

As can be seen from above, generated Word document is divided into several steps as follow.

a) COM components provide C# with basic functions and properties of Word document. After importing, the classes and interfaces in the namespaces of Microsoft.Office.Core and Microsoft.Office.Interop.Word can be used.

b) Create and open a new Word document by using Add and Open method of Microsoft.Office.Interop.Word.Application.Documents interface.

c) Using the method of Microsoft.Office.Interop.Word.Document.Tables.Add to insert tables.

d) After inserting a table into new Word document, we need to determine the cell location of data will be written.

e) Read data from database, and choose different written ways according to different types of data.

f) Save the document through the method of Microsoft.Office.Interop.Word.Document.SaveAs.

III. IMPLEMENTATION OF THE KEY TECHNOLOGY

A. Import COM Components

Suppose we would like to generate a Word document automatically in Visual Studio 2008, we should accomplish it by using the “Microsoft Office 11.0 object library” COM components of Office. Adding this component should operate in the environment of VS2008, and under the circumstance that the building project application under open, you should select the project menu and select the add reference, and then choose COM from pop-up menu of add reference, select “Microsoft Word 11.0 object library”, then click the OK button. By this way, user can access or create a Word document in the project [2].

Following two kinds of classes are mainly involved in encoding.

a) Microsoft.Office.Interop.Word.Application.Class is the application class of Word and provides application handing function to Word.

b) Microsoft.Office.Interop.Word.Document is the document class of Word and provides document handing function in proportion to specific disk files.

B. Create Word Document

All the Word documents are made up of mounts of objects, and these objects together with properties value determine the structure and format of the documents, as shown in Fig.2 below. The object of Word represents an element of Word, Such as: document, paragraph, bookmarks or character. Collection is an object, and property is a feature of the object or one aspect of this object’s operation. For example, Document’s properties are including name, content, condition, and so on [3].

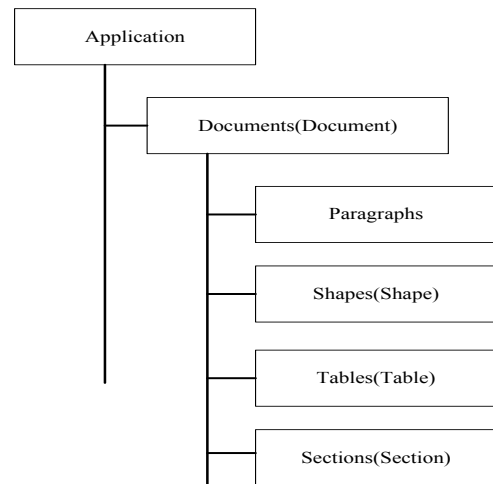


Fig.2. The structure of Word documents

Supposing someone would like to operate the Word document, he/she should create a Word object first. As Word is a controller and can't be operated direct, we need save the Word object into Document object. Then the Word document could be operated.

```
Word.Application wordApp = new
Word.ApplicationClass();
```

```
Word.Document wordDoc =
wordApp.Documents.Add(ref Nothing, ref Nothing, ref
Nothing, ref Nothing);
```

```
wordApp.Visible = true;
```

C. Insert and Set Table in Word

1) Insert Table

The collection of tables is member of Microsoft.Office.Interop.Word.Document, Microsoft.Office.Tools.Word.Document, Selection and Range. It means we can create a table in any context, and can insert table in designated range by using Add method of tables' collection.

In order to insert table in Word, we should set the rows and columns of table and the foot style of cell. These operations are all set in Tables of document object wordDoc.

```
Object missingValue = Type.Missing;
```

```
wordDoc.Tables.Add(wordApp.Selection.Range, row number
of table, column number of table, ref missingValue, ref
missingValue); //insert row and column information of table
```

The table will be added to tables' collection of Microsoft.Office.Interop.Word.Document when it is created. And then, you can use the Item property to refer to the table according to the Item number.

2) Merge Cells

Merge each cell as its position in table by using Merge method.

```
wordDoc.Tables.Item(1).Cell(1,5).Merge(wordDoc.Tables
.Item(1).Cell(5,6));
```

The code above is merging the row from first to sixth of the fifth column in table.

3) Add Rows and Columns

Cells are organized into rows and columns in Microsoft Office Word table. We can use the Add method of Rows object to add rows in table, so is the Column.

```
wordDoc.Tables.Item(1).Rows.Add(ref beforeRow);
```

```
wordDoc.Tables.Item(1).Columns.Add(ref beforeColumn);
```

D. Add Texts in Cells

Each table is including one group of cells and each cell object represents one cell in table. You refer to each cell by its location in the table and add texts into it, as well as apply the format settings.

```
wordDoc.Tables.Item(1).Cell(1,1).Range.Text = "name";
```

Way introduced above is directly generating word document of faculty information in program, and also inserting and setting the tables in Word as well as writing the faculty information. This way is really safety but not flexible. Once the faculty information is changed, you need to rewrite the program to design tables and regenerate the Word document of faculty information. If the format of Word document generated is fixed, you can create a table to insert data in the word template and the style of table is set manually outside the program.

Therefore, before the Word document of faculty information can be generated in the application, you can save the fixed format of word as a Word document template, set the format of it (such as font style, size, paging, picture size, etc.) and define Word application object and Word document object in the program. After that you can extract the faculty's information data from the database, and automatically calls the Word template document in the program, then input the faculty information among them.

E. Insert Picture to the Word Document

Supposing the image is stored in a database. If you want to insert the image in the cell, first you need to read the image from the database and save the image to the PictureBox control which is created dynamically, and then save it to the specified path through the Image.Save() method of PictureBox, and then use the Inline Shapes. AddPicture() method to insert image under the specified path to the specified cell through the clipboard.

Mentioned above is just adding one image to the specified cell in the word, if you want to add multiple images of the database into the Word document, how to do?

If all images are stored in local disk, first you need to query images' path from the database and fill them to the DataSet, then call the ExportPic_To_Word() method, which will save the specified images to the Word document.

```
public void ExportPic_To_Word(DataSet DS, int flag, Word.Document wordDoc)
```

```
{
```

```
// DS, the image name of querying; wordDoc, Word document object
```

```
// flag, the number of images inserted to the document. For example, flag=1, insert one image; flag=3, insert three images
```

```
Object start = Type.Missing;
```

```
Object end = Type.Missing;
```

```
Word.Range P_Range = wordDoc.Range(ref start, ref end); //the location of images will be inserted
```

```
Object P_Ranges = P_Range;
```

```
object missingValue = Type.Missing;
```

```
string[] str = newstring[flag];
```

```
for (int i = 0; i < DS.Tables[0].Rows.Count; i++)
```

```
{
```

```
for (int j = 0; j < flag; j++)
```

```
{
```

```
str[j] = DS.Tables[0].Rows[i][j].ToString();
```

```
if (File.Exists(PicFilePath + "\\" + str[j]))
```

```
{
```

```
object LinkToFile = false;
```

```
object SaveWithDocument = true;
```

```
wordDoc.InlineShapes.AddPicture(PicFilePath + "\\" + str[j], ref LinkToFile, ref SaveWithDocument, ref P_Ranges);
```

```
}
```

```
}
```

```
}
```

```
wordDoc.Save();
```

```
}
```

IV. CONCLUSION

In this paper, the author's innovations are as follows: by using the Visual Studio development platform and the COM components technology based on Office, combining with Microsoft SQL Server 2005 implement the automatic generation of Word documents of faculty information. It is convenient for faculty annual assessment or title appraisal for fixed format document. Users can not only modify and save the generated Word documents, and also can carry on the page setup, print control, etc. Seeing from the application, the Word documents generated automatically are illustrated and widely useful, and the effect is ideal and worth promoting.

REFERENCES

- [1] Ye Ming, Zhang Zheng. Development of Word report generation function based on C#.net. *Computer Engineering and Application*, vol. 44, pp. 104-106, September 2008.
- [2] Chen Xiumin, Yan Zhongwen, Zhu Meining, Yang Yanping, Liu Aiyong, Lin Zuo. The research of Some Key Technologies in .NET based Management System of Test Questions Library and Test Papers. *Journal of Hebei Normal University of Science & Technology*, vol. 23, pp. 59-63, April 2009.
- [3] Li Shengguang, Jia Lei, Wang Dezhou. Achieving Office Software Automation Using C++Builder. *Microcomputer Information*, vol. 18, pp.64-66, October 2002.