

Aspose Word 模板使用总结

命名空间：

```
1.  using Aspose.Words;
2.  using Aspose.Words.Saving;
3.  using System.IO;
4.  using System.Data;
```

添加 dll：

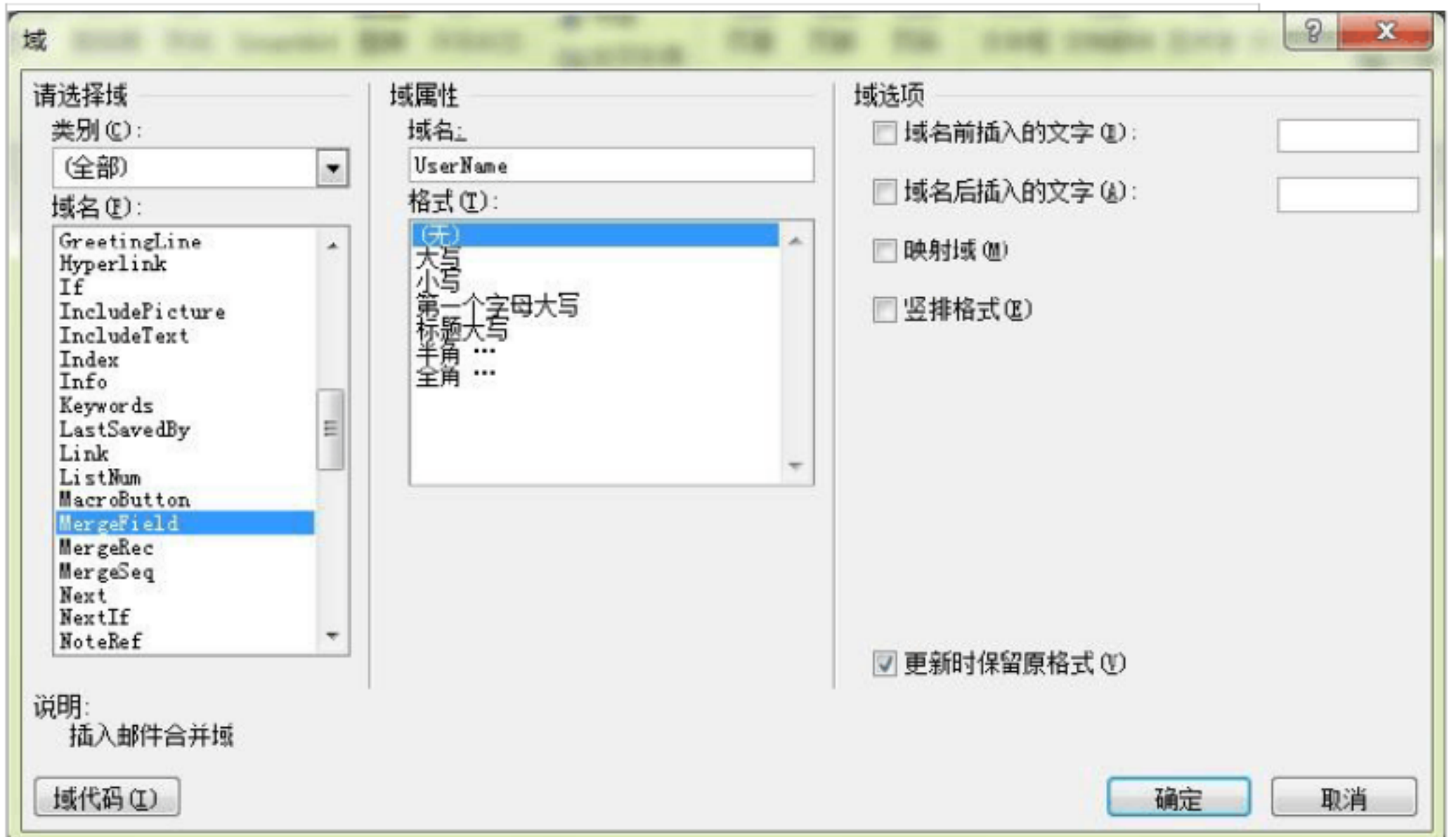
链接： <http://pan.baidu.com/s/1pJG899T> 密码： bv3k

1.创建 word 模版，使用 MergeFeild 绑定数据

新建一个 Word 文档，命名为 Template.doc

姓名：	«UserName»	性别：	«Gender»
出生年月：	«BirthDay»	家庭住址：	«Address»

注意：这里并不是输入 "«"和"»"就可以了，而是必须在菜单的 "插入 文档部件 域" 找到 MergeField
并输入相应的域名



2. 使用数组提供数据源

```
1.  string tempPath = Server.MapPath("~/Docs/Temp/Template.doc ");
2.  string outputPath = Server.MapPath("~/Docs/Output/Template.doc ");
3.  // 载入模板
4.  var doc = new Document(tempPath);
5.  // 提供数据源
6.  String[] fieldNames = new String[] {" UserName", " Gender", " BirthDay", " Address"};
7.  Object[] fieldValues = new Object[] {" 张三 ", " 男 ", " 1988-09-02 ", " 陕西咸阳 "};
8.  // 合并模版，相当于页面的渲染
9.  doc.MailMerge.Execute(fieldNames, fieldValues);
10. // 保存合并后的文档
11. doc.Save(outputPath);
12. // 在 WebForm 中,保存文档到流中，使用 Response.BinaryWrite 输出该文件
13. var docStream = new MemoryStream();
14. doc.Save(docStream, SaveOptions.CreateSaveOptions(SaveFormat.Doc));
15. Response.ContentType = "application/msword ";
16. Response.AddHeader("content-disposition ", " attachment; filename=Template.doc ");
```

2. 使用数组提供数据源

```
1.  string tempPath = Server.MapPath("~/Docs/Temp/Template.doc ");
2.  string outputPath = Server.MapPath(" ~/Docs/Output/Template.doc ");
3.  // 载入模板
4.  var doc = new Document(tempPath);
5.  // 提供数据源
6.  String[] fieldNames = new String[] { " UserName", " Gender", " BirthDay", " Address" };
7.  Object[] fieldValues = new Object[] { " 张三 ", " 男 ", " 1988-09-02 ", " 陕西咸阳 " };
8.  // 合并模版，相当于页面的渲染
9.  doc.MailMerge.Execute(fieldNames, fieldValues);
10. // 保存合并后的文档
11. doc.Save(outputPath);
12. // 在 WebForm 中,保存文档到流中，使用 Response.BinaryWrite 输出该文件
13. var docStream = new MemoryStream();
14. doc.Save(docStream, SaveOptions.CreateSaveOptions(SaveFormat.Doc));
15. Response.ContentType = "application/msword ";
16. Response.AddHeader("content-disposition ", "attachment; filename=Template.doc ");
```

2. 使用数组提供数据源

```
1.  string tempPath = Server.MapPath("~/Docs/Temp/Template.doc ");
2.  string outputPath = Server.MapPath(" ~/Docs/Output/Template.doc ");
3.  // 载入模板
4.  var doc = new Document(tempPath);
5.  // 提供数据源
6.  String[] fieldNames = new String[] { " UserName", " Gender", " BirthDay", " Address" };
7.  Object[] fieldValues = new Object[] { " 张三 ", " 男 ", " 1988-09-02 ", " 陕西咸阳 " };
8.  // 合并模版，相当于页面的渲染
9.  doc.MailMerge.Execute(fieldNames, fieldValues);
10. // 保存合并后的文档
11. doc.Save(outputPath);
12. // 在 WebForm 中,保存文档到流中，使用 Response.BinaryWrite 输出该文件
13. var docStream = new MemoryStream();
14. doc.Save(docStream, SaveOptions.CreateSaveOptions(SaveFormat.Doc));
15. Response.ContentType = "application/msword ";
16. Response.AddHeader("content-disposition ", "attachment; filename=Template.doc ");
```

```
19. var userScoreTable= GetUserScoreDataTable();

20. dataSet.Tables.Add(userTable);

21. dataSet.Tables.Add(userScoreTable);

22. dataSet.Relations.Add(new

        DataRelation(" ScoreListForUser",userTable.Columns[" Id"],?userScoreTable.Columns["UserId"]));

23. // 合并模版，相当于页面的渲染

24. doc.MailMerge.ExecuteWithRegions(dataSet);

25. var docStream = new MemoryStream();

26. doc.Save(docStream, SaveOptions.CreateSaveOptions(SaveFormat.Doc));

27. return base.File(docStream.ToArray(), "application/msword ", "Template.doc");
```

7.模版上使用书签，插入标记位置

选中文档中的文字，在菜单的 “插入 书签” 指定书签的名称，排序依据选定为位置，添加一个新书签。选中的文字为书签的 Text 属性，这里是为了方便查看。也可以直接插入一个书签并指定位置，只是不明显。

8.在书签位置插入另一个文档的内容

```
1. // 载入模板

2. var doc = new Document(tempPath);

3. var doc1 = new Document(tempPath1); // 新文档
```

```
4. // 找到名称为 PositionFlag 的书签

5. var bookmark= doc.Range.Bookmarks[" PositionFlag"];

6. // 清空书签的文本

7. bookmark.Text = "";

8. // 使用 DocumentBuilder 对象插入一些文档对象，如插入书签，插入文本框，插入复选框，插入一个
   段落，插入空白页，追加或另一个 word 文件的内容等。

9. var builder = new DocumentBuilder(doc);

10. // 定位到指定位置进行插入操作

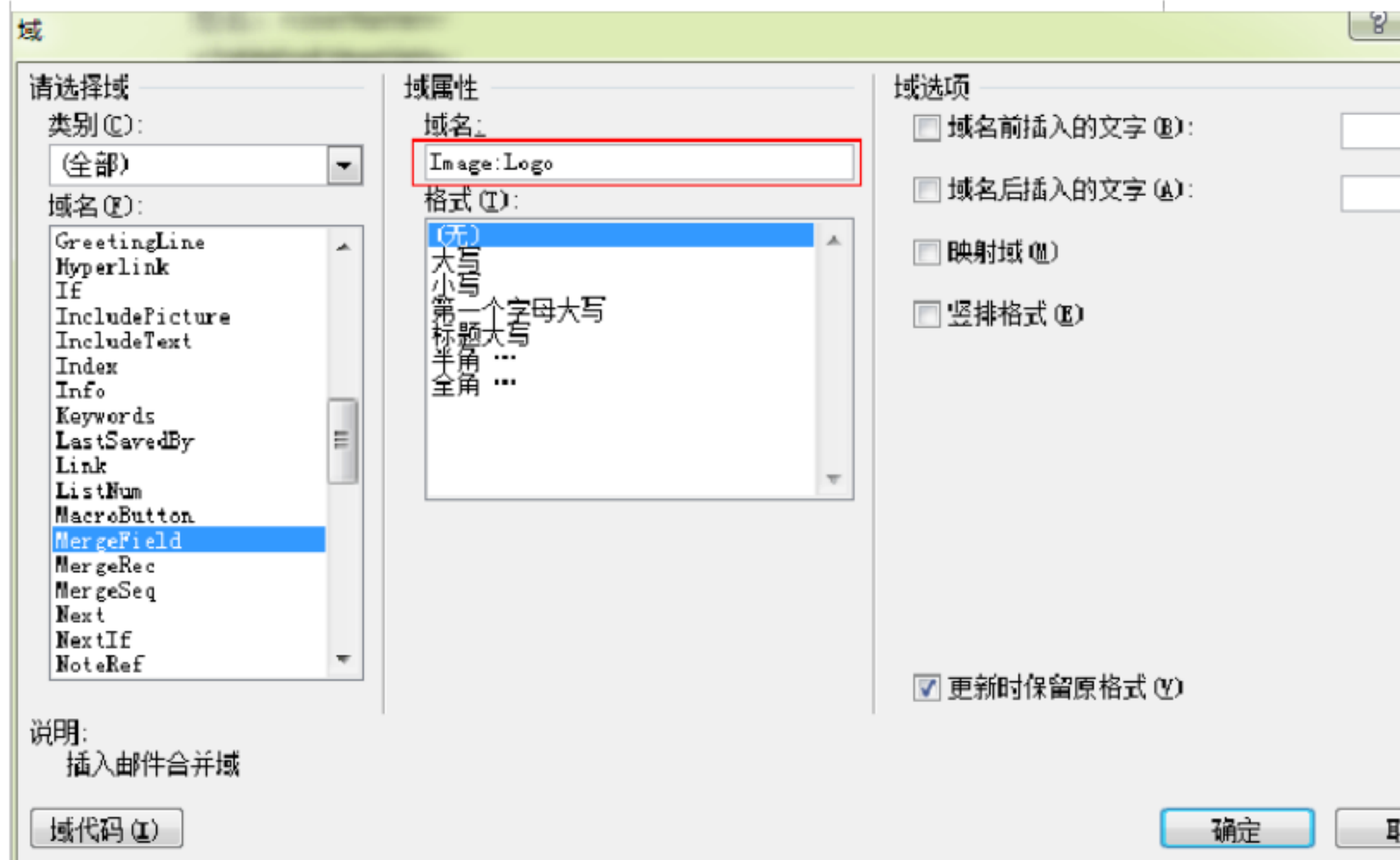
11. builder.MoveToBookmark(" PositionFlag");

12. // 在 PositionFlag 书签对应的位置，插入另一个文档的内容。

13. //InsertDocument 方法可以在
    http://www.aspose.com/docs/display/wordsnet/How+to++Insert+a+Document+into+another+Document
    找到

14. InsertDocument(bookmark.BookmarkStart.ParentNode, doc1);
```

9.创建 word 模版，使用 MergeFeild 插入图片



«Image:Logo»+

10. 插入图片示例

```
1.  string tempPath = Server.MapPath("~/Docs/Temp/Template.doc ");
2.  string logoPath = Server.MapPath("~/Content/logo.jpg ");
3.  var doc = new Document(tempPath); // 载入模板
4.  // 提供数据源
5.  String[] fieldNames = new String[] { "logo", "Gender", " BirthDay", " Address"," Logo" };
6.  Object[] fieldValues = new Object[] { " 张三 ", " 男", " 1988-09-02 ", " 陕西咸阳 ",logoPath };
7.  // 增加处理图片大小程序
8.  //doc.MailMerge.FieldMergingCallback = new HandleMergeFieldInsertDocument();
9.  // 合并模版，相当于页面的渲染
10. doc.MailMerge.Execute(fieldNames, fieldValues);
11.
12. // 在 MVC 中采用 ,保存文档到流中，使用  base.File 输出该文件
13. var docStream = new MemoryStream();
14. doc.Save(docStream, SaveOptions.CreateSaveOptions(SaveFormat.Doc));
15. return base.File(docStream.ToArray(), "application/msword ", " Template.doc");
```

效果如下：



增加图片大小处理的程序

```
1. //Aspose.Word 提供了一个类似 Handler 的功能，IFieldMergingCallback 允许我们动态的处理 MergeField
2. void IFieldMergingCallback.FieldMerging(FieldMergingArgs e){} // 处理文本
3. void IFieldMergingCallback.ImageFieldMerging(ImageFieldMergingArgs args){} // 处理图片
4. // 这里我们处理图片写了一个自定义的类实现
5. class HandleMergeFieldInsertDocument : IFieldMergingCallback
6. {
7.     // 文本处理在这里，如果写在这一块，则不起作用
8.     void IFieldMergingCallback.FieldMerging(FieldMergingArgs e)
9.     {
10.
11.     }
12.     // 图片处理在这里
13.     void IFieldMergingCallback.ImageFieldMerging(ImageFieldMergingArgs args)
14.     {
15.         if (args.DocumentFieldName.Equals("Logo"))
16.         {
17.             // 使用 DocumentBuilder 处理图片的大小
18.             DocumentBuilder builder = new DocumentBuilder(args.Document);
19.             builder.MoveToMergeField(args.FieldName);
20.
21.             Shape shape = builder.InsertImage(args.FieldValue.ToString());
22.
23.             // 设置 x,y 坐标和高宽 .
24.             shape.Left = 0;
25.             shape.Top = 0;
26.             shape.Width = 60;
27.             shape.Height = 80;
28.         }
29.     }
30. }
```


效果如下：

姓名：	张三	头像：	
出生年月：	1988-09-02		
性别：	男		
家庭住址：	陕西咸阳		

11. 向模版插入 Html

姓名：	«UserName»	性别：	«Gender»
出生年月：	«BirthDay»	家庭住址：	«Address»
家乡简介：	«Desc»		

这里的家乡简介使用 html 格式

12. 插入 html 示例

```
1. string tempPath = Server.MapPath("~/Docs/Temp/Template.doc ");

2. string descHtml = ""; // 这里是 html 文本，由于太长略去

3. var doc = new Document(tempPath); // 载入模板

4. // 提供数据源

5. String[] fieldNames = new String[] { "UserName", "Gender", "BirthDay", "Address"," Desc"};

6. Object[] fieldValues = new Object[] { "张三 ", "男", "1988-09-02", "陕西咸阳 ", descHtml};

7. // 增加处理 html 程序

8. doc.MailMerge.FieldMergingCallback = new HandleMergeFieldInsertHtml();

9. // 合并模版，相当于页面的渲染

10. doc.MailMerge.Execute(fieldNames, fieldValues);

11. // 在 MVC 中采用 ,保存文档到流中，使用 base.File 输出该文件

12. var docStream = new MemoryStream();

13. doc.Save(docStream, SaveOptions.CreateSaveOptions(SaveFormat.Doc));

14. return base.File(docStream.ToArray(), "application/msword ", " Template.doc");

15. // 如果不增加 html 的处理程序，默认以文本的输出，这里我们写一个自定义的处理类
```

```
16. class HandleMergeFieldInsertHtml : IFieldMergingCallback

17. {

18.     // 文本处理在这里

19.     void IFieldMergingCallback.FieldMerging(FieldMergingArgs e)

20.     {

21.         if (e.DocumentFieldName.Equals(" Desc"))

22.         {

23.             // 使用 DocumentBuilder 处理图片的大小

24.             DocumentBuilder builder = new DocumentBuilder(e.Document);

25.             builder.MoveToMergeField(e.FieldName);

26.             builder.InsertHtml(e.FieldValue.ToString());

27.         }

28.     }

29.     // 图片处理在这里

30.     void IFieldMergingCallback.ImageFieldMerging(ImageFieldMergingArgs args)

31.     {

32.

33.     }

34. }

35. //IFieldMergingCallback 在循环结构中同样适用

36. // 小结：利用书签加上标志位，利用自定义的 IFieldMergingCallback 灵活处理各种需求，后边会继续

    尝试根据条件加载不同的模版
```