**1.基本介绍**

Aspose.Words是一个商业.NET类库，可以使得应用程序处理大量的文件任务。Aspose.Words支持Doc，Docx，RTF，HTML，OpenDocument，PDF，XPS，EPUB和其他格式。使用Aspose.Words可以在不使用Microsoft.Word的情况下生成、修改、转换和打印文档。在项目中使用Aspose.Words可以有以下好处。

**1.1丰富的功能集**

其丰富的功能特性主要有以下4个方面：

1)格式转换。Aspose.Words具有高质量的文件格式转换功能，可以和Doc，OOXL，RTF，TXT等格式互相转换。

2)文档对象模型。通过丰富的API以编程方式访问所有的文档元素和格式，允许创建，修改，提取，复制，分割，加入，和替换文件内容。

3)文件渲染。可以在服务器端转换整个文档或者页面为PDF，XPS，SWF格式，同样可以转换文档页面为图像格式，或者.NET Graphics对象，这些功能和Microsoft.Word是一样的。

4)报表。可以从对象或者数据源填充模版生成文件。

**1.2不需要Microsoft.Word**

Aspose.Words可以在没有安装Microsoft Office的机器上工作。所有的Aspose组件都是独立，不需要微软公司的授权。总之， Aspose.Words在安全性、稳定性、可扩展性、速度、价格和自动化功能方面，是一个很不错的选择。

**1.3独立的平台**

Aspose.Words可以运行在Windows，Linux和Mac OS操作系统上面。可以使用Aspose.Words去创建32位或者64位的.NET应用程序，包括Asp.NET、WCF、WinForm等等，还可以使用Com组件在Asp、Perl、PHP和Python语言中使用，同样可以在Mono平台上使用Aspose.Words建立.NET应用程序。

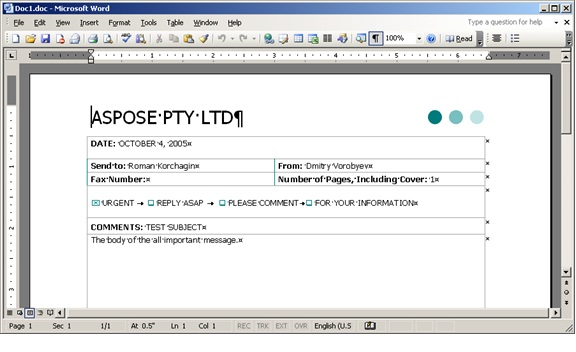
**1.4性能和可伸缩性**

Aspose.Words可以运行在服务器和客户端，它是一个独立的.NET程序集，可以被任何.NET应用程序复制和部署。使用Aspose.Words可以在短时间内产生成千上万的文档，可以打开文档，并修改格式和内容，填充数据并保存。Aspose.Words是多线程安全的，不同的线程在同一时间处理不同的文档。

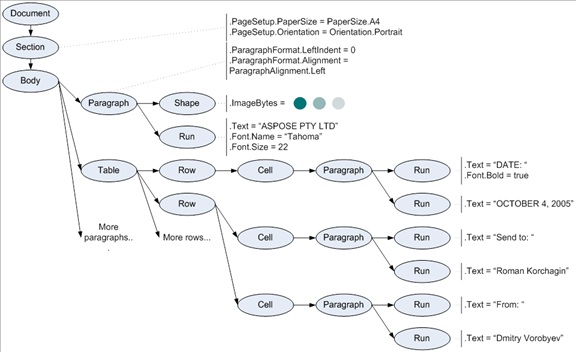
**2.文档对象模型概述**

**2.1 DOM介绍**

Aspose.Words的文档对象模型(以下简称DOM)是一个Word文档在内存中的映射，Aspose.Words的DOM可以编程读取、操作和修改Word文档的内容和格式。理解DOM的结构和相应的类型，是使用Aspose.Words灵活编程的基础，这一点非常重要。下面的一个Word文档例子和其结构如下图所示:



当上述文档被Aspose.Words的DOM读取时，会创建如下结构的树形对象：



从上图的结构和对应的Word文档，我们可以看到大概的DOM中相关对象的结构，有了这些基本概念，就可以很流程的操作Word文档了。Document, Section, Paragraph, Table, Shape, Run 以及图中的其他椭圆形的都是Aspose.Words对象，这些对象具有树形的层级结构，图中的注释同样说明这些文档对象树中的对象具有多个属性。

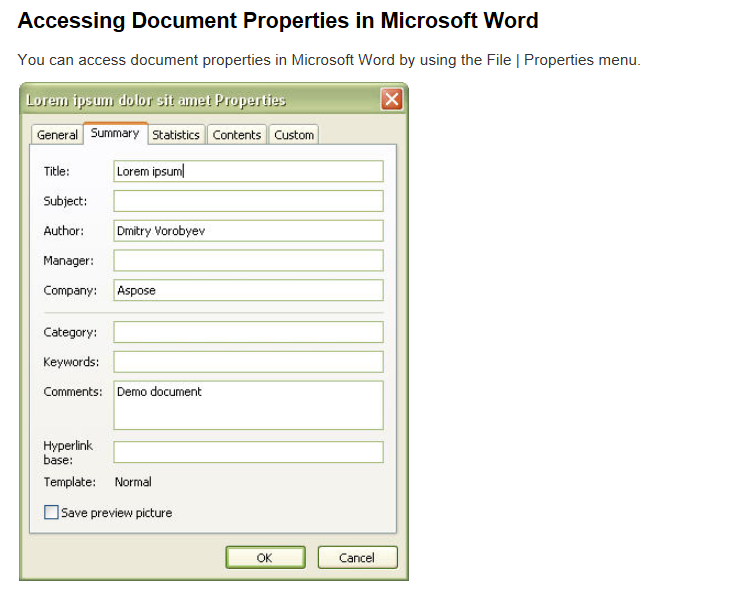
Aspose.Words中的DOM有以下特点：

1.所有的节点(node)类最终都继承于Node类，它是Aspose.Words DOM的基本类型。

2.节点可以包含(嵌套)其他节点，例如Section和Paragraph都继承自CompositeNode类，而CompositeNode类来源与Node类。

**2.1.1 document对象**

* 在word中

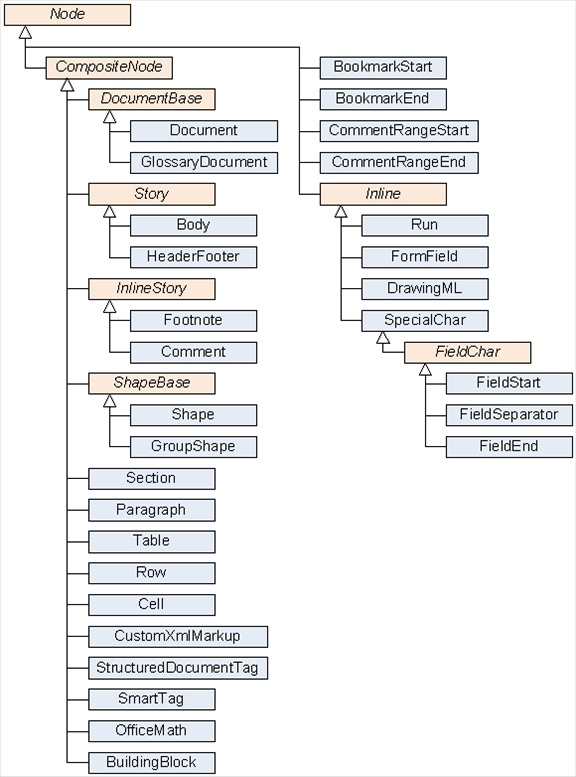


在aspose中

**2.2 Node类型**

当Aspose.Words读取Word文档到内存中时，不同类型的文档元素被不同的类型对象来替代，每一个文本框的text, paragraph, table, section都是Node对象,甚至文档本身都是一个Node。Aspose.Words为每一种文档节点类型都定义了一个类。

下面是一个UML类图，表示DOM中不同node类型之间的关系。抽象类的名字用斜体表示。注意，Aspose.Words DOM中同样包括了一些非节点类型的类，例如Style, PageSetup, Font等等，它们没有在这幅图里面显示。



 看看这些主要的类及作用

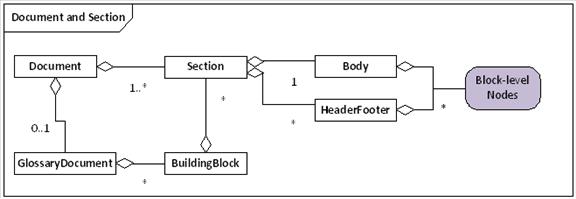
|  |  |  |
| --- | --- | --- |
| Aspose.Words类 | 类别 | 描述 |
| Document | Document | Document对象是文档树的根节点，提供访问整个文档的入口 |
| Section | Document | Section对象对应一个文档中的一节 |
| Body | Document | 是一节中的主要文本容器 |
| HeaderFooter | Document | 一节中的特殊页眉或者页脚容器 |
| GlossaryDocument | Document | 代表一个Word文档中词汇表的根条目 |
| BuildingBlock | Document | 代表一个词汇表文档，如构件，自动图文集或一个自动更正条目 |
| [Paragraph](http://www.cnblogs.com/asxinyu/admin/aspose.words.paragraph.html) | Text | 一个文本段落，保护内联的节点 |
| [Run](http://www.cnblogs.com/asxinyu/admin/aspose.words.run.html) | Text | 一个格式一致的文本块 |
| [BookmarkStart](http://www.cnblogs.com/asxinyu/admin/aspose.words.bookmarkstart.html) | Text | 一个书签的起点标记 |
| [BookmarkEnd](http://www.cnblogs.com/asxinyu/admin/aspose.words.bookmarkend.html) | Text | 一个书签的结束标记 |
| [FieldStart](http://www.cnblogs.com/asxinyu/admin/aspose.words.fields.fieldstart.html) | Text | 一个特殊的字符指定一个单词字段的开始 |
| [FieldSeparator](http://www.cnblogs.com/asxinyu/admin/aspose.words.fields.fieldseparator.html) | Text | 单词字段的分隔符 |
| [FieldEnd](http://www.cnblogs.com/asxinyu/admin/aspose.words.fields.fieldend.html) | Text | 一个特殊的字符指定一个单词字段的结束 |
| [FormField](http://www.cnblogs.com/asxinyu/admin/aspose.words.fields.formfield.html) | Text | 一个表单字段 |
| [SpecialChar](http://www.cnblogs.com/asxinyu/admin/aspose.words.specialchar.html) | Text | 特殊字符类型，没有具体的 |
| [Table](http://www.cnblogs.com/asxinyu/admin/aspose.words.tables.table.html) | Tables | Word文档中的表格 |
| [Row](http://www.cnblogs.com/asxinyu/admin/aspose.words.tables.row.html) | Tables | 一个表格对象的行 |
| [Cell](http://www.cnblogs.com/asxinyu/admin/aspose.words.tables.cell.html) | Tables | 表格行的单元格 |
| [Shape](http://www.cnblogs.com/asxinyu/admin/aspose.words.drawing.shape.html) | Shapes | Word文档中的图像，形状，文本框或者OLE对象 |
| [GroupShape](http://www.cnblogs.com/asxinyu/admin/aspose.words.drawing.groupshape.html) | Shapes | 一组Shapes对象 |
| [DrawingML](http://www.cnblogs.com/asxinyu/admin/aspose.words.drawing.drawingml.html) | Shapes | 一个文档中的Sharp或者图像，图表 |
| [Footnote](http://www.cnblogs.com/asxinyu/admin/aspose.words.footnote.html) | Annotations | 文档中包括文本的脚注或者尾注 |
| [Comment](http://www.cnblogs.com/asxinyu/admin/aspose.words.comment.html) | Annotations | 文档中包含文本的注释 |
| [CommentRangeStart](http://www.cnblogs.com/asxinyu/admin/aspose.words.commentrangestart.html) | Annotations | 一个相关的注释区域的开始 |
| [CommentRangeEnd](http://www.cnblogs.com/asxinyu/admin/aspose.words.commentrangeend.html) | Annotations | 一个相关的注释区域的结束 |
| [SmartTag](http://www.cnblogs.com/asxinyu/admin/aspose.words.markup.smarttag.html) | Markup | 在一个段落内围绕一个或多个内嵌结构的智能标记 |
| [CustomXmlMarkup](http://www.cnblogs.com/asxinyu/admin/aspose.words.markup.customxmlmarkup.html) | Markup | 文档中的某些结构的自定义XML标记 |
| [StructuredDocumentTag](http://www.cnblogs.com/asxinyu/admin/aspose.words.markup.structureddocumenttag.html) | Markup | 文档中的一种结构化的文档标签（内容控制） |
| [OfficeMath](http://www.cnblogs.com/asxinyu/admin/aspose.words.math.officemath.html) | Math | 代表Office的数学对象，如函数，方程或者矩阵 |

**2.3组成模式**

Aspose.Words文档的结构树非常重要，下面的设计吐可以更清晰的理解各个节点之间的包含关系。

**2.3.1 Document and Section**

文档和节：



从上图可以看出：

1.一个Document有1个或者多个Section(节)节点；

2.Section有1个Body(正文)，没有或者有多个HeaderFooter节点；

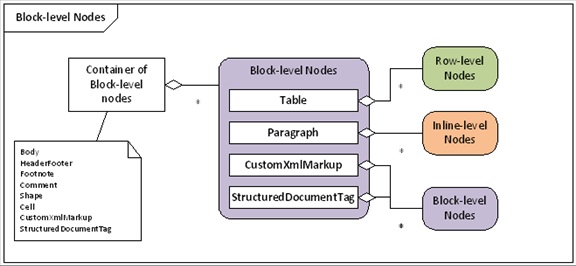
3.Body和HeaderFooter可以包含多个块级节点；

4.1个Document能够有一个[GlossaryDocument](http://www.cnblogs.com/asxinyu/admin/aspose.words.buildingblocks.glossarydocument.html).

1个Word文档包含1个或多个节，一个节可以定义自己的页码，边距，方向，以及页眉页脚的文字；一个节保护主要的问题，如页眉，页脚(首页，奇数页，偶数页)。

**2.3.2 Block-level节点**

Block-level节点的关系图如下所示：



从上图中可以看到：

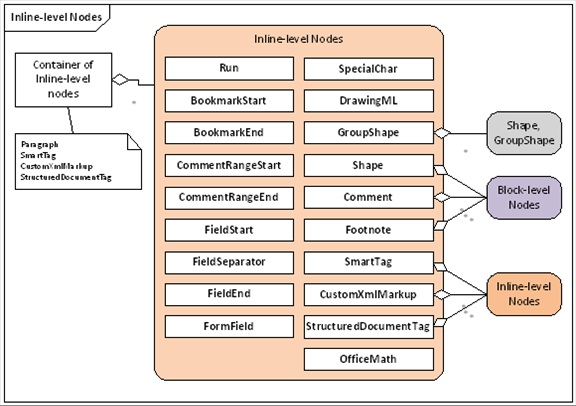
1.Block-level元素可以出现在文档中的很多地方，如Body的子节点，脚注，评论，单元格其他元素。

2.最重要的Block-level节点是表格和段落；

3.1个表格有0行或者多行；

CustomXmlMarkup 和StructuredDocumentTag可以包含其他Block-level节点；

**2.3.3 Inline-level节点**



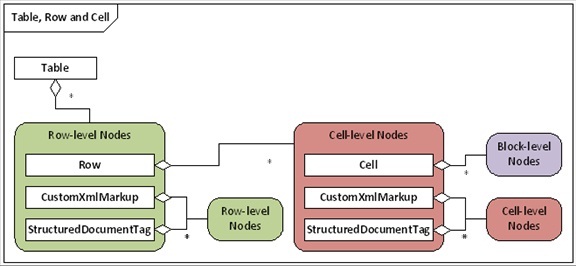
从上面的图表可以看到下列关系：

1.Paragraph是最经常出现的Inline-level节点；

2.Paragraph可以包含不同的Run格式节点；也可以包含书签(bookmarks)和注释(annotations)

3.Paragraph还可以包含形状，图像，绘图对象等，以及智能标签；

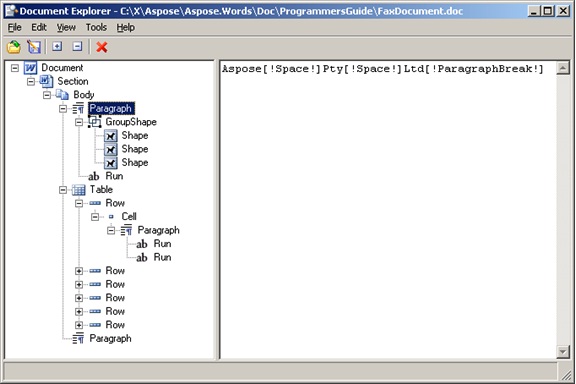
**2.3.4 表格行单元格**



Table可以包含很多行，行可以包含单元格，单元格可以包括block-level节点。

**2.4 设计模式与导航**

Aspose.Words将文档表示为一个有节点组成的树，因此就可以在节点之间互相切换。Aspose.Words提供了一个“文档浏览器”(DocumentExplorer)，这是一个项目例子Demo。如下图所示：



可以通过Node类的ParentNode属性来访问上级节点，因此很方便获取父节点。文档对象模型是有大量的对象组合而成，他们的关系如下：

1.Node类是所有节点类的基类；

2.CompositeNode类是组合节点的基类；

3.Node类的中，没有子节点管理的接口，子节点管理的方法只出现在CompositeNode；

4.从Node类中移除子节点管理的方法，更干净，可以减少很多额外的转换；

第一篇基本介绍就介绍到这里，我们可以看到这个组件的功能是非常强大的，特别是文档的转换保存，操作也很灵活，相比DocX当然要强大。只不过唯一不足它是商业的，自己业余玩玩倒是没问题，源代码等下一篇写例子了，再发上来吧，至于官方的资料，是商业的，大家都懂的。至于途径吗，官方网站在这里：<http://www.aspose.com/>，国内的途径吧，CSDN比较多，呵呵。

参考文档：

中文博客：<http://www.cnblogs.com/asxinyu/p/3242754.html>

官方文档：

<https://docs.aspose.com/display/wordsjava/Aspose.Words+Document+Object+Model>

API文档