

如何操作 Office Open XML 格式文档

发布日期： 2007-06-28 | 更新日期： 2007-06-28

摘要： Office Open XML 格式文件代替了早期的二进制 Office 系统文件。本文档向您介绍了包含在一个格式化文档中的组件以及展示这些文件功能的一些场景。

Frank Rice, 微软公司

适用于： Microsoft Office Excel 2007, Microsoft Office PowerPoint 2007, Microsoft Office Word 2007

下载 [2007OfficeManipulatingOpenXMLFilesSample.exe](#).



本页内容

↓ [概述](#)

↓ [创建一个 Office Open XML 格式文件](#)

↓ [浏览 Office Open XML 格式文件](#)

↓ [手动编辑由 Office Open XML 格式创建的文档](#)

↓ [通过编程来操作 Office Open XML 格式文档](#)

↓ [结论](#)

↓ [额外资源](#)

概述

在 Microsoft Office 的早期版本中，由 Microsoft Office Excel, Microsoft Office PowerPoint 和 Microsoft Office Word 创建的文件以独有的，单一文件格式进行保存，它们称之为二进制文件。Microsoft Office System 2007 的发布为 Microsoft Office Excel 2007, Microsoft Office PowerPoint 2007 和 Microsoft Office Word 2007 引入了一个新的文件格式： Office Open XML 格式。

Office Open XML 格式是基于 XML 和 ZIP 归档技术创建的。和早期 Microsoft Office 版本类似，2007 的文档保存在一个单一的文件或者容器中，所以管理这些文档的过程仍然是简单的。但是，与早期文件不同的是 Office Open XML 格式的文件能够被打开显示器中的组件，使得您能够访问此文件的结构。

在本文中，您可以手动的打开文件并且浏览组成此文档那个的每一个部件来查看 Office Open XML 格式文件。

另外您也可以对此文档进行编程。在本文中适用到的文件可以通过下载获得，

[2007OfficeManipulatingOpenXMLFilesSample.exe](#)。如果您不能够下载，您可以使用自己的程序文件和图像

文件进行替代。

🏠 [返回页首](#)

创建一个 Office Open XML 格式文件

此部分中，您将检测一个包含文本，图像和属性的示例 Word 2007 文档的 XML 文件格式。

在 Word 中创建一个 Office Open XML 格式文档

- 1 启动 Word 2007。
.
- 2 在新建文档中，粘贴以下文本：
.
Soaring with the American Bald Eagle

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla rutrum. Phasellus feugiat bibendum urna. Aliquam lacinia diam ac felis. In vulputate semper orci. Quisque blandit. Mauris et nibh. Aenean nulla. Mauris placerat tempor libero.

Pellentesque bibendum. In consequat, sem molestie iaculis venenatis, orci nunc imperdiet justo, id ultricies ligula elit sit amet ante. Sed quis sem. Ut accumsan nulla vel nisi. Ut nulla enim, ullamcorper vel, semper vitae, vulputate vel, mi. Duis id magna a magna commodo interdum.
- 3 高亮显示 Soaring with the American Bald Eagle，然后在 开始标签，在样式组中，设置样式为标题。
.
- 4 下一步，在文档中插入一张图片：
 1. 将鼠标指针放置在第一段落的末尾，并且点击回车插入新行。
 2. 然后，点击插入标签，点击图片，找到一个图片文件（例如包含在下载中的 Eagle1.gif 文件），然后点击插入。
- 5 现在添加一个文档属性：
 1. 点击 Microsoft Office 按钮，指向准备，然后点击属性。
 2. 在文档属性面板中，添加作者名称，标题，主题和备注，类似于以下：

表 1.文档属性设置

属性	描述
作者	Nancy Davolio
标题	Soaring with the American Bald Eagle
主题	Bald Eagles
备注	A study of the bald eagle

- 6 下一步，在文档中添加一些批注。
 - 1 在审阅标签下，点击新建批注。

- 2 在批注框中，输入 This is my comment。此时您的文档类似于图 1。



图 1. 示例 Office Word 2007 文档

- 7 下一步，保存文档：

1. 单击 Microsoft Office 按钮，然后单击另存为。
2. 在另存为文件类型列表中，选择 Word Document (*.docx)，然后在文件名称框中输入 SampleWordDocument.docx。
3. 单击保存。

- 8 关闭 Word。

[返回页首](#)

浏览 Office Open XML 格式文件

在此部分中，您将浏览您刚刚创建的示例文档。

浏览 Office Open XML 格式文档

1. 打开 Windows 资源管理器。
2. 找到 SampleWordDocument.docx 文件，右键单击此文件，然后选择重命名。

重要：

使用下面的三个步骤来抽取 Office Open XML 格式文件。根据您使用的操作系统不同（Windows XP

或 Windows Vista)，第 5 步有所不同。

3. 在文件名称后面添加一个.zip 扩展名，文件名称现在变为 SampleWordDocument.docx.zip。
4. 当弹出警告信息时，点击确定。
5. 抽取容器文件：
 1. (如果您使用 Windows XP:) 右键点击 SampleWordDocument.docx.zip 文件，指向打开方式，然后点击 Compressed (zipped) Folders。此时组成文档的文件夹和部件将会显示在资源管理器中。
 2. (如果您使用 Windows Vista:) 右键点击 SampleWordDocument.docx.zip 文件，然后点击 Extract All。在 Extract Compressed (Zipped) Folders 对话框中，接受默认位置或者选择一个新的位置，然后点击 Extract。此时组成文档的文件夹和部件将会显示在资源管理器中。

在下面的步骤中，您将检测包含在文档中的关键部件。

6. 检测 [Content_Types].xml 部件：
 1. 使用 Windows 资源管理器，找到名为[Content_Types].xml 的文件。
 2. 右键点击此文件，指向打开方式，然后点击 Internet Explorer。每一个 Office Open XML 格式文档的根下都有一个 [Content_Types].xml 组件。[Content_Types].xml 组件的作用就是确定文档中每个唯一类型组件。每个组件需要在这个组件中列出它自己的类型。组件需要有可以识别的类型，以便应用程序知道当渲染此文档时如何使用这个组件。类型也能够使您了解到组件的作用以及如何使用。
 3. 关闭文件。
7. 检测.rels 文件夹：

关联代表着两个组件之间的连接。关联是存储在_rels 子文件夹中的组件。任何组件都有相关联的组件，包含在同结构的_rels 文件夹中，此文件夹包含了一个定义它们关系的.rels 组件。子文件夹创建在与组件相同的文件夹下。一个关联的名称通过在原始组件的文件名后加上.rels 扩展名获得（用于文档文件的关联组件是一个例外：它的名称为“.rels”）。

 1. 在 Windows 资源管理器中，双击_rels 文件夹，然后右键点击.rels 文件。
 2. 指向打开方式，点击选择程序，点击 Internet Explorer，然后点击确定。
 3. 完成检测后，关闭 IE。

8. 检测 docProps 文件夹：

2007 版本中的文档属性在三个 Microsoft Office System 程序中是始终保持结构化的。分成三个逻辑 XML 组件，它们被存储自 docProps 子文件夹下。这使得用户可以很容易的进行访问，因为他们它们

在相同的位置并且不会和其它的文档内容混淆。

在 Windows 资源管理器中，双击 docProps 文件夹，右键单击 core.xml 文件，指向打开方式，然后单击 Internet Explorer。

9. 检测 core.xml 组件：

打开 core.xml 组件，然后观察显示的您之前输入的属性。 core.xml 组件持有用户输入的用来标识文档的属性，例如标题，主题和作者。

10. 检测 custom.xml 组件：

从 Windows 资源管理器中，在 Internet Explorer 中打开 custom.xml。

custom.xml 组件包含了被用户，开发人员或者通过自定义逻辑添加到文档任何自定义文档属性。

11. 检测 app.xml 组件：

从 Windows 资源管理器中，在 Internet Explorer 中打开 app.xml。

app.xml 组件包括在应用程序级别指定到文档的唯一属性，例如文章的页数，文本行的数量，应用程序的版本等等。

12. 检测 word 文件夹：

大部分的内容组件都驻留在 word 子文件夹下面。另外，在里面也有一个_rels 关联子文件夹。

双击 _rels 文件夹。

在_rels 子文件夹中，用于连接所有文档组件的关联名称为 document.xml.rels。

13. 检测 document.xml.rels 组件：

在 Internet Explorer 中打开 document.xml.rels 。

关联使用 IDs 和 Uniform Resource Identifiers (URIs)来定位组件。它允许在没有任何硬编码引用的前提下所有非关联的组件是无效的。在后面的章节中将会详细讨论。

在 Word 文件夹下面，注意 styles.xml 组件。

14. 检测 styles.xml 组件：

在 Internet Explorer 中 styles.xml。

这个包含了一些可用的重音和底纹，您可以在文档中使用它们。

15. 必须的和可选的组件

16. 在 Office Open XML 格式文件中这些组件的使用使得文档能够以高度模块化的方式存储。一些组件对文档来说是必须有效的，例如 document.xml 组件和 fontTable.xml 组件。

17. 检测 document.xml 组件：

1. 在 Internet Explorer 中打开 document.xml 。 document.xml 组件包含了文档主体的文本。

2. 查看完此文件后，关闭 Internet Explorer。

18. 检测 fontTable.xml 组件：

1. 在 Internet Explorer 中打开 fontTable.xml 组件。

fontTable.xml 组件包含了文档的字体设置。

2. 查看完此文件后，关闭 Internet Explorer。

如果组件描述的功能在文档中使用不到的话，这些组件就不是必需的。示例中包括了备注，页眉组件和页脚组件，这些组件对 Word 文档来说是可选的。这就使得用户能够轻松的通过文档结构进行操纵，而不需要研究没有使用的内容。

XML 设计用于结构化的内容，并不会本地支持二进制的內容，例如图像或者 OLE 对象。二进制数据能够被编码到字符并且以 XML 形式进行存储，但是它需要一个编码和解码的过程，这就使得对于应用程序或者开发人员来说效率低下。在 2007 发布之后，此时就不在需要编码二进制对象因为他们能够作为二进制组件以他们自身的格式进行存储。因此在 Office 文档中访问二进制对象变得非常容易。媒体文件存储在 media 文件夹中。

19. 检测 word\media 和 word\embeddings 文件夹：

在 Windows 资源管理器中，双击 media 子文件夹。

注意.gif 媒体文件表示您之前插入的图像。

备注：

您可能注意到图像的文件名称已经从 Eagle1.gif 更改为 image1.gif 了。更改的目的是为了解决隐私考虑，因为一个恶意用户可以从文档中的组件名称中获得关于此组件的一些内容，例如一个图像文件。例如，一个作者可能通过加密文档中的文本部件来保护文档内容。但是，如果两个图像 old_widget.gif and new_reinforced_widget.gif 被插入。即使文本被保护起来了，恶意用户也能够知道 widget 已经被更新了。使用像 image1 和 image2 这样的通用图像文件可以在 Office Open XML 格式文件中添加额外的保护。

20. 关闭 SampleWordDocument.docx.zip ，不保存。

[返回页首](#)

手动编辑由 Office Open XML 格式创建的文档

Office Open XML 格式拥有很多的优势。其中一个优势就是具有可以在没有 Office 程序的情况下使用由 Office 2007 系统创建的文档的能力。这就使得您能够创建基于服务器的解决方案，用于在可伸缩的环境下集合，访问和编辑文档。

在下面的步骤中，您将手动的编辑一个 Word 2007 文档。请注意这些场景只是这个新的文件格式的一个很小的示例。在大部分的场景中，用户不会以这种方式手动编辑文档。但是对于开发人员，在不需要编写代码的情况下浏览由 Office 2007 创建的文档将带来很大的好处，尤其是当设计解决方案或者编写程序模型的时候。如您之前

看到的，在访问此文档的容器文件后，您能够轻松的浏览独立的组件。这也就意味这您能够编辑，替换甚至添加组件。通常情况下，您可以修改文档的备注，更新文档属性。

利用 XML 修改 Office Open XML 格式文档

1. 在 Word 2007 中，打开 SampleWordDocument.docx 文档。

备注：

在打开文件之前，请先删除文件的.zip 扩展名

2. 点击 Microsoft Office 按钮，指向准备，然后点击属性。注意作者，标题，主题和备注框体，然后关闭文档。
3. 打开 Windows 资源管理器，浏览到 SampleWordDocument.docx 文档。
4. 使用 [steps described for your operating system](#) 抽取文档文件。

Word 2007 文档中的备注存储在名称为 comments.xml 的组件中。这种文档主体的分离使您能够轻松的定位和修改组件。

5. 将 comments.xml 组件从压缩文档中拖拽到 Windows 桌面。
6. 右键点击 comments.xml 组件，指向打开方式，然后选择文本编辑器或者 XML 编辑器打开，例如记事本。
7. 定位到下面的文本：

```
<w:t>A study of the bald eagle</w:t>
```

8. 替换或者编辑 <w:t> 元素的文本，例如，将其更改为：

```
<w:t>A detailed study of the bald eagle</w:t>
```

9. 保存并且关闭文件。
10. 从 Windows 桌面中，将 comments.xml 组件拖拽到压缩文件的 word 文件夹中。
11. 当弹出确认文件替换警告消息时，点击确定。

下一步，您可以更改文档属性，然后确认更改的结果。文档属性存储在压缩文件根目录下面的子文件夹中，使得他们能够轻松的访问和编辑。

12. 双击 docProps 文件夹。
13. 将 core.xml 组件从压缩文档中拖拽到 Windows 桌面。
14. 在文本编辑器中打开 core.xml。

15. 定位到下列文本：

```
<dc:creator>Nancy Davolio</dc:creator>
```

16. 替换或者编辑 <dc:Creator> 元素的文本。例如，使用您自己的名字替换里面的文本。

17. 保存并且关闭文件，然后将其拖拽回 docProps 文件夹。
18. 当弹出确认文件替换警告消息时，点击确定。
19. 通过点击回退箭头或者工具栏中的 up 图标导航到文档容器直到您定位到.zip 文件。
20. 从文件名称中移出.zip 扩展名，然后在 Word 2007 中打开文件。
21. 点击 Microsoft Office 按钮，指向准备，然后点击属性。

注意备注文本已经更改了。同时作者属性也已经更改。

在以前的步骤中，您通过在文档中编辑 XML 组件的方式修改了文档。利用新的文件格式，您也可以替换整个文档组件来更改内容，格式或者属性。这就使得您使用存在的文档组件来更新单个的文档或者更新整个文档库。

使用存在的组件来修改文档的一个例子就是更改文档使用的样式。如果您需要管理多个文档的样式，但是希望保留一个物理的版本，这种方式将很有用。更改文档使用的所有类型就是替换 styles.xml 组件。

此场景使您能够为所有的文档编译样式组件集合，然后创建一个应用程序，使得用户能够自动的选择不同的样式。

在后台，您的应用程序能够用一个预制的组件代替另外一个。在下面的步骤中，您将手动的执行此操作。

通过替换已存在组件的方式修改一个 Office Open XML 格式文档

1. 复制一份 SampleWordDocument.docx 文档，然后将其命名为 AnotherSampleWordDocument.docx。
2. 在 Word 2007 中打开 AnotherSampleWordDocument.docx。
3. 在开始页面，点击更改样式，指向样式集，然后点击独特。此文档看上去类似于图 2。



图 2. 在独特样式中的文档

4. 保存并且关闭文档。
 5. 使用 [steps described for your operating system](#) 来提取出 AnotherSampleWordDocument.docx 文档文件。
 6. 双击 word 文件夹，然后将 styles.xml 组件拖拽到 Windows 桌面。此组件将被用于更新您创建的第一个文档。
 7. 点击工具栏中的回退箭头或者 Up 图标导航到文档容器，直到定位到.zip 文件。
 8. 现在，在 Word 2007 中打开 SampleWordDocument.docx 文档，注意文档主体的样式。
 9. 关闭文档。
 10. 使用 [steps described for your operating system](#) 提取 SampleWordDocument.docx 文档文件。
 11. 双击打开 word 文件夹，然后将 styles.xml 从 Windows 桌面拖拽到 word 文件夹，替换原来的文件。
 12. 当弹出确认文件复制警告消息时，点击确定。
 13. 通过点击工具栏中的回退箭头或者 Up 图标导航到 SampleWordDocument.docx 文件的文档容器，直到定位到.zip 文件。
 14. 从文件名中移出.zip 扩展名，然后在 Word 2007 中打开文件。
- 注意文档样式已经更改为在 AnotherSampleWordDocument.docx 中看到的那样。

请首先将 AnotherSampleWordDocument.docx.zip 文件名的.zip 扩展名移出。

此样式场景的另外一个样子就是使用页眉页脚文档组件从一个 Word 2007 文档中快速的复制相同的设置到一个或者多个 Word 2007 文档中。页眉页脚组件能够被手动的更改，您将在后面的内容中了解到。当然，这个过程也能够使用代码进行自动化。这对于那些希望使用标准文档页眉和页脚，并且不需要在每个基本文档中付出管理任务的组织来说非常有用。另外，如果您的页眉页脚格式更改后，替换页眉页脚是很容易的。

在下面的步骤中，您将添加一个简单的页眉到 SampleWordDocument.docx 中。然后使用来自

AnotherSampleWordDocument.docx 文档的不同的页眉来更新它。

更新 Office Open XML 格式文档的页眉

1. 在 Word 2007 中打开 SampleWordDocument.docx。
2. 在插入标签中，点击页眉按钮的下拉键头，然后选择字母表型页眉。页眉将被添加到文档的标题中。此文档看上去类似于图 3。

Soaring with the American Bald Eagle

Let's begin with the eagle. The eagle is a symbol of strength and power. It is a bird of prey, and it is a bird of freedom. The eagle is a symbol of the American spirit, and it is a symbol of the American dream. The eagle is a symbol of the American people, and it is a symbol of the American future.



The eagle is a symbol of strength and power. It is a bird of prey, and it is a bird of freedom. The eagle is a symbol of the American spirit, and it is a symbol of the American dream. The eagle is a symbol of the American people, and it is a symbol of the American future.

图 3. 使用字母表型页眉的示例文档

3. 保存并且关闭文件。
4. 在 Word 2007 中打开 AnotherSampleWordDocument.docx。
5. 在插入标签中，点击页眉按钮的下拉键头，选择年刊型页眉。页眉将被添加到文档的标题中，并且带有年份。此文档那个看上去类似于图 4。

Soaring with the American Bald Eagle

Let's begin with the eagle. The eagle is a symbol of strength and power. It is a bird of prey, and it is a bird of freedom. The eagle is a symbol of the American spirit, and it is a symbol of the American dream. The eagle is a symbol of the American people, and it is a symbol of the American future.



The eagle is a symbol of strength and power. It is a bird of prey, and it is a bird of freedom. The eagle is a symbol of the American spirit, and it is a symbol of the American dream. The eagle is a symbol of the American people, and it is a symbol of the American future.

图 4. 带有年刊型页眉的示例文档

6. 保存并且关闭文件。
7. 使用 [steps described for your operating system](#) 为 AnotherSampleWordDocument.docx 提取文档文件。
8. 双击 word 文件夹，然后将 header1.xml 文件拖拽到 Windows 桌面。
9. 通过点击工具栏中的回退箭头或者 Up 图标导航到文档容器，直到定位到.zip 文件。
10. 从 AnotherSampleWordDocument.docx.zip 文件名中移出.zip 文件扩展名。
11. 使用 [steps described for your operating system](#) 为 SampleWordDocument.docx 提取文档文件。
12. 双击 word 文件夹打开它，将 header1.xml 文件从 Windows 桌面拖拽到 word 文件夹。
13. 当弹出确认文件替换警告消息时，点击确定。
14. 通过点击工具栏中的回退箭头或者 Up 图标导航到文档容器，直到定位到.zip 文件
15. Remove the .zip extension from the 从 SampleWordDocument.docx.zip 文件名中移出.zip 扩展名，然后在 Word 2007 中打开这个文件。

注意新的页眉。

文档中也能够包含二进制组件-例如图像文件或者 Microsoft VBA 项目，您可以将其作为 XML 组件那样轻松的访问。更换二进制组件会带来一些有趣的事情。例如您可以更换二进制组件来替换整个的 OLE 对象，如 Microsoft Office Visio 图像。手动的操作没有价值，您需要考虑一个使得图像能够自动从服务器更新的场景。编写一个工具来完成这样的操作相对来说是简单的人物。在下面的步骤中，您将使用一个图像来交换

AnotherSampleWordDocument.docx 文档中的图像二进制文件。

在 Office Open XML 格式文档中交换二进制组件

1. 使用 [steps described for your operating system](#) 为 SampleWordDocument.docx 提取文档文件。
2. 双击 Word 文件夹，然后双击 media 文件夹来定位到 Eagle1.gif 图像。
3. 右键点击 Eagle1.gif 然后点击预览。这是出现在文档中的图像。
4. 在下载文件中定位到 Eagle2.gif 图像，然后将其复制到 Windows 桌面。(或者您可以替换您自己的图像文件)
5. 在 Windows 桌面上，右键点击 Eagle2.gif 图像，然后点击预览。这是您将使用替换当前图像的图像文件。
6. 关闭预览窗口，右键点击 Eagle2.gif，然后点击重命名，将其名称更改为 Eagle1.gif。
7. 将重新命名过的 Eagle1.gif 从 Windows 桌面上拖拽到 media 文件夹中。

8. 当弹出确认文件替换警告消息时，点击确定。
9. 通过点击工具栏中的回退箭头或者 Up 按钮导航到文档容器，直到定位到.zip 文件。
10. 将名称中的.zip 扩展名移出掉，然后在 Word 2007 中打开文件。注意图像已经被更新了。

一些组件在 Microsoft Office 系统文档中是必须的，例如 Word 2007 中 document.xml 组件。但是，一些组件是可选的，只在需要某些功能的时候存在。这就意味着如果您不需要某些功能，您可以移出这些组件和相关的关系。包含项目代码的 2007 Office 文档被称为“macro-enabled”文档(用于 Word 2007 的.docm 扩展名，用于 Excel 2007 的.xlsm 扩展名和用于 PowerPoint 2007 的.pptm 扩展名)。不像“macro-free”文档，macro-enabled 文档在组件中存储代码。组件的类型依赖于文档中代码的类型。例如，包含 VBA 代码的 macro-enabled 文档在 vbaProject.bin 二进制组件中存储它的数据。

其他的项目类型包含使用 Excel 4.0-style macro(XLM macros)的 Excel 2007 工作簿中或含有命令按钮的 PowerPoint 2007 幻灯。这些特性位于他们独立的组件中，以便他们能够被轻松的确认和移出。

在下面的步骤中，您将创建一个简单的 macro-enabled 文档并且展示它的功能。然后，您将从文档中删除 vbaProject.bin 组件及其关系并且查看其效果。注意您也可以通过点击文件菜单中的另存为，或者通过保存文件为 macro-free 文件(.docx, .pptx, or .xlsx)来移出 macro-enabled 文件中的项目组件。但是，这需要您在 2007 Office 系统程序中打开此文档。使用下面过程中的步骤可以使您在不使用 Office 程序的前提下移出项目。

从 Office Open XML 格式文档中移出 VBA 项目

1. 使用 Word 2007 打开 SampleWordDocument.docx。
2. 在开发工具标签中的代码组中，点击 Visual Basic 来打开 Visual Basic 编辑器。您也可以按 ALT+F11 来打开 Visual Basic 编辑器。

备注：

如果您无法找到开发工具标签，您需要添加它。点击 Microsoft Office 按钮，然后点击 Word 选项，然后在常用标签中，选择“在功能区显示开发工具选项卡”。

3. 在 Visual Basic 编辑器代码窗口中，输入或者粘贴以下语句：

```
Sub SampleCode()  
  
Msgbox("Hello world")  
  
End Sub
```

备注：

如果您无法看到代码窗口，通过点击视图菜单，然后选择代码窗口。

4. 在运行菜单中，点击运行子过程/用户窗体来运行代码。您也可以点击 F5。

5. 点击确定关闭消息窗口，然后关闭 Visual Basic 编辑器。下一步，作为 macro-enabled 文档保存。
6. 点击 Microsoft Office 按钮，指向另存为，然后点击 Word 文档。
7. 在保存类型下拉列表中，选择 Word Macro-Enabled Document (.docm)，然后点击保存，关闭文档。
8. 使用 [steps described for your operating system](#) 为 SampleWordDocument.docm 提取文档文件。
9. 双击 word 文件夹，然后双击_rels 文件夹。
10. 键点击 document.xml.rels，点击打开方式，然后选择一个文本编辑器，例如 Notepad。

备注：

这个文件可能是只读的。如果是只读的话，关闭文件，右键点击文件名称，点击属性。然后清除只读属性复选框。然后重新打开文件。

11. 定位并且删除掉以下 XML 标记：

```
<Relationship Id="rId1"
Type="http://schemas.microsoft.com/office/2006/relationships/vbaProject"
Target="vbaProject.bin"/>
```

12. 保存并关闭文件。
13. 右键点击 document.xml.rels 然后选择复制。
14. 回到 SampleWordDocument.docm.zip 文件，然后打开它。
15. 双击 word 文件夹。
16. 右键点击 vbaProject.bin 项目，然后点击删除。当提示消息时，点击确定来删除。
17. 双击_rels 文件夹。
18. 右键点击，然后选择粘贴。
19. 当提示确认文件替换消息时（在 Windows Vista 中为复制和替换警告消息），点击确定。
20. 通过点击工具栏中的回退箭头或者 Up 按钮导航到文档容器中，直到定位到.zip 文件。
21. 右键点击 SampleWordDocument.docm 文件夹，然后点击删除来删除它。当提示时，点击确定。
22. 从 SampleWordDocument.docm.zip 文件名中移出.zip 扩展名。当提示时，在重命名警告框中点击确定，然后重新使用 Word 2007 打开此文件。
23. 按 ALT+F11 查看 VBA 项目。注意子程序已经丢失。

备注：

您可以将文件扩展名从.docm 更改成.docx 来获得相似的结果。带有 macro-free 扩展名的文件从来不会执行代码。

通过编程来操作 Office Open XML 格式文档

Office Open XML 格式的一个重要意义在于它对于自定义解决方案的无限潜能。您可以使用任何平台的能够处理 XML 或者 ZIP 文件的工具来访问并且修改文档内容。例如，您可以在 Microsoft Visual Studio 中利用托管代码编写一个服务器端的应用程序来省城 2007 Office 系统文档。您可以使用 Microsoft .NET Framework 的强大 XML 类库来编辑 Office Open XML 格式文件中包含的任何 XML 文档组件。

操作文档组件及其关系的一个强大方式就是使用 System.IO.Packaging 名称空间，它包含在 Microsoft Windows Software Development Kit 中。System.IO.Packaging 名称空间将在后面的内容中详细讨论。

备注:

您可以从微软的下载中心中下载 Microsoft Pre-Release Software WinFX Runtime Components—Beta2。在下面的步骤中，您将开发一个控制台应用程序在不使用 Word 2007 的前提下更改格式样式。这个项目通过利用从一个 Word 2007 文档中提取出来的样式组件来替换此文档中的样式组件来完成。结果就是目标文档采用了与提取样式那个文档相同的样式。

备注:

在此有很多这样的代码示例，您可以通过下载获得。在下载了这些预制的代码示例后，您将能够在 Microsoft Visual Studio 2005（在工具菜单中）中使用这些 Code Snippet Manager 特性来将它们插入到您的项目中。您可以下载 [2007 Office System Sample: Open XML File Format Code Snippets for Visual Studio 2005](#)。

执行下面的步骤来更改一个文档的格式。

在一个 Office Open XML 格式文档中更改样式

1. 首先，创建一个文件夹和子文件夹来宿主您将更新的文档和 styles.xml 组件。在这个练习中，命名文件夹为 WordOpenXMLFormatSample。在文件夹中，添加一个名称为 NewStylePart 的子文件夹。
2. 复制 SampleWordDocument.docx 到 WordOpenXMLFormatSample 文件夹中。
3. 导航到 AnotherSampleWordDocument.docx 文件，在文件名后添加.zip 文件扩展名，然后打开文件。
4. 在 word 文件夹中导航到 styles.xml 组件，右键点击它，然后点击复制。
5. 导航到 WordOpenXMLFormatSample 文件夹，然后导航到 NewStylePart 子文件夹。右键点击 NewStylePart 子文件夹，然后点击粘贴。文件夹现在包含了默认样式组件的 Word 2007 文档，子文件夹中包含了来自带有名称为“Fancy”样式的文档中的 styles.xml 组件。
6. 启动 Visual Studio 2005。
7. 在文件菜单中，点击新建项目。
8. 在新建项目对话框中，从左边的项目类型树视图中，点击其它语言，选择 Visual C#，然后从模板列表视图中选择控制台应用程序。

9. 在名称对话框中，将项目命名为 **StyleSwapper**，然后点击确定。**Visual Studio** 在项目中创建了所有的文件，并且将他们存储在项目下面的目录中，例如 `drive\Visual Studio projects\StyleSwapper`。

为了看到确切地项目保存的位置，或者更改位置，执行下面操作：

1. 在工具菜单中，点击选项。
2. 在选项对话框中，点击项目和解决方案树型视图，然后点击常规标签。在此，您可以指定项目和模板存储的位置，或者设置其它选项。
3. 注意项目的位置，然后点击确定关闭窗口。

当您创建新的项目后，**Visual Studio** 自动的创建新的程序，并带有三个容器：属性，引用和 **Program.cs**。您可以在解决方案资源管理器中查看这三个容器。**Visual Studio** 同时也创建了用于添加大部分代码的空类。

备注：

Microsoft .NET Framework 3.0 (WinFX)包含了 **System.IO.Packaging** 名称空间。利用 **System.IO.Packaging** 名称空间，您能够添加文档组件，获取和更新内容，或者创建新的关系，使您能够构建新文档或者修改现有的文档名称空间中一些重要的成员包含 **Package** 对象，**PackagePart** 对象,和 **PackageRelationship** 对象。关于 **System.IO.Packaging** 名称空间的更多信息，请见 [System.IO.Packaging Namespace](#)。

下一步，您将从您的项目中添加一个引用到.NET Framework 3.0:

- 10 点击项目菜单，点击添加引用。
.
- 11 在浏览标签中，在查找框中，搜索到 **WindowsBase.dll**。根据此文档发布的时间，文件的位置位于
. `drive\Program Files\Reference Assemblies\Microsoft\WinFx\v3.0`.
- 12 选择 **WindowsBase.dll** 文件，然后点击确定。通过点击解决方案资源管理器中的引用来验证引用已经被创建。
.
- 13 在解决方案资源管理器中，右键点击 **Program.cs**，然后选择查看代码。
.
- 14 在代码窗口的名称空间上面输入下面代码：

```
using System.IO;  
  
using System.IO.Packaging;
```

要使用 2007 Office 系统文档的内容，您需要打开它。**System.IO.Packaging** 有一个称为 **Package** 的顶级成员，它相当于文档。在您代开 **Package** 项目后，您将检查它的结构，操作它的组件。**Packages** 能够作为只读，只写或者读/写形式打开。

- 15 在 `Class Program` 语句的括号(`{`)后面添加以下代码。以下行中设置了包含您 `Word` 文档和 `style.xml` 位置的变量。这些语句假设 `WordOpenXMLFormatSample` 文件夹在 `C` 盘中创建。

```
private static String stylePath =  
    @"C:\wordOpenXMLFormatSample\NewStylePart\styles.xml";  
  
private static String packagePath =  
    @"C:\wordOpenXMLFormatSample\SamplewordDocument.docx.zip";
```

备注:

以 `@` 符号开始的字符串被称为逐字字符串字面变量 (`verbatim string literals`)。这就告诉程序使用获取文件时出现的那些确定的字符串。

- 16 当运行项目时, `Main` 过程将自动的执行。在 `Main` 的开阔号 (`{`) 和闭括号 (`}`) 之间, 输入以下行:

```
SwapStylePart(packagePath, stylePath);
```

此行调用了 `SwapStylePart` 过程, 您将在下面添加。它将路径传输到了 `Word 2007` 文档和 `styles.xml` 组件中。

- 17 在 `Main` 后的闭括号 (`}`) 之后添加下面的代码:

```
static void SwapStylePart(String packagePath, String stylePath)  
{ }
```

下面步骤中, 您将添加代码到 `SwapStylePart` 程序中, 此程序用于打开作为 `Package` 对象并且拥有读/写访问权限出现的文档。注意 `using` 语句的使用。它的目的是自动的处理 `Package` 对象并且在语句完成之后清空使用的内存。

- 18 将下列代码输入到 `SwapStylePart` 程序中:

```
using (Package package = Package.Open(packagePath, FileMode.Open,  
    FileAccess.ReadWrite))  
{ }
```

为了能够使用 `2007 Office` 系统文档的任何组件, 您首先需要定位它。您可以使用 `URL` 来引用一个特定的文档组件, `URL` 是对每个组件唯一的路径。

- 19 在 `using` 语句的括号之间输入下面的语句:

```
// Set the URI for the styles document part (/word/styles.xml).  
  
Uri uriPartTarget = new Uri("/word/styles.xml", UriKind.Relative);
```

在编写此文章的那个时刻, `System.IO.Packaging` 名称空间还不允许您复制或者替换一个已经存在的组件。为了交换组件, 您必须首先删除已经存在的组件, 然后使用相同的 `URL` 创建一个新的组件。注

意删除组件不会影响任何组件的关系。任何关系都完好无损并且仍然应用到一个新的组件。

- 20 在您上个步骤中添加的行后面输入以下代码：

```
// Delete the existing document part (/word/styles.xml).  
  
package.DeletePart(uriPartTarget);
```

在 `package` 中添加一个新的文档组件需要使用一个 URL。在这种情况下，您仅仅需要重新使用相同的 URL 来重新创建样式文档组件。当创建一个 `package` 时需要一个额外的参数：组件的内容类型。当前在文档中使用的内容类型可以在位于文档容器中的 `[Content_Types].xml` 组件中发现。

- 21 在您上个步骤中添加的行后面输入以下代码：

```
// Recreate a new document part for styles(/word/styles.xml). PackagePart  
packagePartReplacement = package.CreatePart(uriPartTarget,  
"application/vnd.openxmlformats-officedocument.wordprocessingml.styles+xml  
");
```

利用新创建的样式组件，最后一步就是从原始的 `styles.xml` 组件中的 XML 复制到新创建的组件中。

`System.IO.Packaging` 名称空间不会在一个 XML 文档中本地的处理 XML，所以在 .NET Framework 中复制内容的方式就是通过使用 `Streams`。

在下面的步骤中，您将添加代码，这些代码以流的形式打开外部样式组件，并且将其写入到新的样式文档组件中。为了复制流，您将调用 `CopyStream` 程序，并且在源和目的流中传输。

- 22 在您上个步骤中添加的行后面输入以下代码：

```
using (FileStream fileStream = new FileStream(stylePath, FileMode.Open,  
FileAccess.Read))  
  
{// Load the new styles.xml  
  
    using a  
stream.CopyStream(fileStream,packagePartReplacement.GetStream());  
  
}
```

- 23 下一步，在 `SwapStylePart` 程序的闭括号后添加 `CopyStream` 程序：

```
private static void CopyStream(Stream source, Stream target)  
  
{  
  
    const int bufferSize = 0x1000;  
  
    const int bufferSize = 1024;
```

```
byte[] buf = new byte[bufSize];

int bytesRead = 0;

while ((bytesRead = source.Read(buf, 0, bufSize)) > 0)

{

    target.Write(buf, 0, (int)bytesRead);

}

source.Close();

target.Close();

}
```

这个过程设置了内存中缓冲区的大小，来自于原始 `styles.xml` 组件的数据流存储在缓冲区中。当有读取字节操作时，这些字节会被写入到新的 `styles.xml` 组件中。To see the application in action, you need to build the project: 要查看程序的效果，您需要生成项目：

24 在生成菜单中，点击生成 `StyleSwapper`。

. **备注：**

如果在生成的时候发生错误，您将看到一个对话框问您是否希望运行上次的生成。点击否，您将看到错误列表中的错误描述。如果您没有看到错误列表，在视图菜单中，点击错误列表。

25 假设没有错误发生，您现在运行代码。但是首先，您可能想要查看当前的文档：

.

26 在 `Word 2007` 中打开 `SampleWordDocument.docx`。

.

27 关闭 `Word 2007` 然后在 `SampleWordDocument.docx` 文件名后添加 `.zip` 扩展名。

.

28 按 `F5`。您将看到 `Windows` 控制台出现，很快您就会看到它消失了。因为这是一个控制台应用程序，

. 它没有用户界面，只是很简单的出现。

29 从文件名中移出 `.zip` 扩展名，然后在 `Word 2007` 中重新打开它。注意文档的样式已经更改为"独特"

. 样式。

下面一个练习阐述在没有 `Word 2007` 情况下您如何使用自定义的应用程序来在 `Office Open XML` 格式文件中执行批量操作。在这个练习中，您将创建一个托管程序，这个程序搜索满足特定标准的文件夹和子文件夹，从而找到正确的文件。应用程序为了术语和包含数据文件的出现来检测文档组件。您可以想象使用这个程序来确定包

含特定客户名称的文档，或者是包含特定产品名称的文档。您也可以在应用程序中添加额外的逻辑，使用其他的术语来替换现有的，本质上就是在不使用应用程序的地情况下复制 Word 2007 中的查找-替换特性。

根据关键字搜索一组 Office Open XML 格式文件

1. 启动 Visual Studio 2005.
2. 在文件菜单中，点击新建项目。
3. 在新建项目对话框中，从左边的项目类型树型视图中，选择 Visual C#.然后选择 Windows 应用程序，更改项目的名称为 KeywordSearch，然后点击确定。Visual Studio 创建了项目中所有的文件。
4. 在解决方案资源管理器中，右键点击 Form1.cs，然后点击查看设计器。
5. 在 Form1.cs [Design] 标签中，将下面的控件添加到窗体中，然后设置它们的属性，以至于窗体类似于图 5。

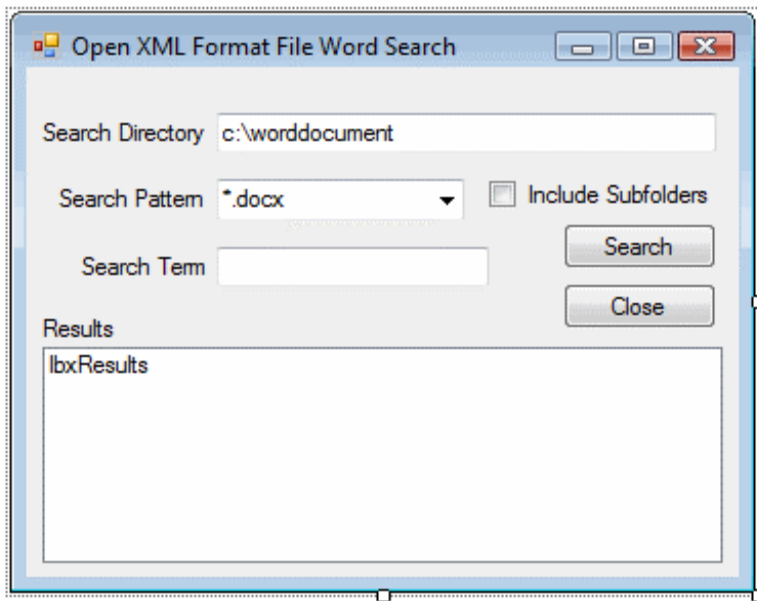


图 5. KeywordSearch 窗体

表 2. 用于 Office Open XML 格式文档关键字搜索窗体的控件列表

类型	属性
Label	Text: Search Directory
TextBox	Name: txtPath
	Text: C:\WordDocuments 注意这是搜索开始的默认目录。
Label	Text: Search Pattern
ComboBox	Name: cboMask
	Items (Collection): *.docx *.docm

	Text: *.docx
CheckBox	Name: ckbSubfolders
	Text: Include Subfolders
Label	Text: Search Term
TextBox	Name: txtTerm
Button	Name: btnSearch
	Text: Search
Button	Name: btnClose
	Text: Close
Label	Text: Results
ListBox	Name: lbxResults

6. 在解决方案资源管理器中，右键单击 **Form1.cs**，然后点击查看代码。

7. 在代码窗口中，在已存在的 `using` 语句后面添加下面的语句：

```
using System.Xml; using System.IO; using System.IO.Packaging;
```

要使用 `System.IO.Packaging` 名称空间，您需要添加一个到 `WindowsBase.dll` 库的引用，它位于 .NET Framework 3.0 SDK (WinFX) 中：

8. 在项目中，点击添加引用。

9. 在浏览标签中的查找框中，搜索 `WindowsBase.dll` 文件名。根据此文章发布的时间，文件位于 `drive\Program Files\Reference Assemblies\Microsoft\WinFx\v3.0` 目录中。

10 点击确定。

11 在 `public partial class Form1 : Form` 的开括 (`{`) 号后面，添加下面的类变量：

```
FileInfo[] tempfiles; List<FileInfo> files = new List<FileInfo>();
```

下一步，添加代码到搜索按钮中：

12 在 `Form1.cs [Design]` 标签中，双击搜索按钮添加 `Click` 事件。

13 在 `btnSearch_Click` 过程的两个括号之间，添加下面的代码：

```
Boolean match = false; // Ensure that the user added a search term.
```

```

if (txtTerm.Text == "")
{
    MessageBox.Show("Don't forget the search term.");
}

List<FileInfo> returnedFiles; // Get the starting directory.

DirectoryInfo dir = new DirectoryInfo(txtPath.Text); // Get the list of files.
returnedFiles = GetDirFiles(dir); // Loop through the file list.

foreach(FileInfo file in returnedFiles)
{
    match = GetDocPart(file);
}

if(!match)
{ // No matching files were found.

    lblResults.Items.Add("No matches.");
}

```

在这个代码中，声明了 **Boolean** 变量用于显示是否搜索发现了任何匹配的内容。下一步，如果文本框是空的，代码将提示用户输入搜索的关键词。在窗体中的其它文本框都有默认值。下一步，定义一个 **List** 类来包含从搜索中返回的文件。**List** 类提供了一个数组，在此数组中可以动态的提升大小。变量 **dirs** 指向了搜索开始的目录。然后指向 **GetDirFiles** 过程的调用将在目录中返回文件列表。接下来代码将循环查询返回的文件集，调用 **GetDocPart** 过程来为搜索关键字检测每一个文件。如果没有匹配的选项，一个消息将被添加到列表框中。

14 添加 GetDirFiles 程序:

```

public List<FileInfo> GetDirFiles(DirectoryInfo dir)
{ // Get all files for the current directory.

    Object selectedItem = cboMask.SelectedItem;

    tempfiles = dir.GetFiles(selectedItem.ToString()); // Add these files to the
returned file list.

```

```

foreach (FileInfo file1 in tempfiles)

{

    files.Add(file1);

} // Search subfolders if requested.

if (ckbSubfolders.Checked)

{ // Get subfolders for the current directory.

    DirectoryInfo[] dirs = dir.GetDirectories("*.");

    foreach (DirectoryInfo directory in dirs)

    {

        GetDirFiles(directory);

    }

}

return files;

}

```

这个程序设置 `selectedItem` 变量等于显示在 `combo` 框中的搜索样式。下一步，在当前的目录中调用 `GetFiles` 方法来返回所有符合搜索样式的文件。返回的文件被添加到文件列表中。如果一个用户选择了 `Include Subfolders` 复选框，每个当前目录的子目录通过递归行式调用 `GetDirFiles` 程序（带有每个子文件夹名称的）而被搜索到。最后文件列表就被返回到调用 (`btnSearch_Click`) 程序中。

- 15 在 `GetDirFiles` 程序后面添加下面的代码：

```

private Boolean GetDocPart(FileInfo file)

{ // Retrieve the start part for the input file.

    Boolean fileFound = false;

    const String documentRelationshipType =
"http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument";

    const String dcPropertiesSchema = "http://purl.org/dc/elements/1.1/"; //
Open the package with read access.

```

```

using (Package myPackage = Package.Open(file.DirectoryName + "\\\" +
file.Name, FileMode.Open, FileAccess.Read))

{ // Get the main document part (document.xml).

    foreach (PackageRelationship relationship in
myPackage.GetRelationshipsByType(documentRelationshipType))

        { // There should be only one document part in the package.

            Uri documentUri = PackUriHelper.ResolvePartUri(new Uri("/",
UriKind.Relative), relationship.TargetUri);

            PackagePart documentPart = myPackage.GetPart(documentUri);

            NameTable nt = new NameTable();

            XmlNamespaceManager nsmgr = new XmlNamespaceManager(nt);

            nsmgr.AddNamespace("w", dcPropertiesSchema);

            XmlDocument doc = new XmlDocument(nt);

            doc.Load(documentPart.GetStream());

            if (doc.OuterXml.IndexOf(txtTerm.Text) >= 1)

                {

                    lbxResults.Items.Add(file.DirectoryName + "\\\" + file.Name);

                    fileFound = true;

                    }// There is only one document part, so exit the loop. break;

                }

        }

    if (lbxResults.Items.Count >0)

        {

            return true;

        }

    else

```

```
{  
  
    return false;  
  
}  
  
}
```

此程序初始时分别设置变量值为文档关系类型中的名称空间和文档属性的架构。接着文档作为 `Package` 对象被打开。在 `System.IO.Packaging` 名称空间中，您可以通过组件之间的关系（定义在关系组件中），通过包含组件结构路径的 URL 来访问 `package` 中各种各样的组件。用于图像组件的一个 URL 示例就是 `\word\media\picture.jpg`。如果您知道到 2007 Office 系统文档中特定文档组件的 URL，您可以直接的访问，编辑或者删除组件。在 `GetDocPart` 程序中，通过调用 `Package` 对象中的 `GetPart` 方法，然后传输到组件 URL 中从而返回 `document.xml` 组件。

下一步，`XmlDocument` 对象指向了一个 XML 文件，此文件包含一个或多个到不同名称空间的引用。和所有的 XML 文件一样，XML 解析器通过带有名称空间前缀的名字来访问文档中大量的元素和属性。

备注：

没有限定符的元素名称被认为是一个默认名称空间的组件。

这些名称空间的限定符在运行时必须被解析到它们名称空间的引用。为了使此任务轻松并且一致，.NET Framework 包含了提供大量名称空间管理工具的 `XmlNamespaceManager`。其中一个就是 `NameTable` 类。`NameTable` 类在内部存储了属性和元素名称。当一个元素或者属性名称在一个 XML 文档中出现多次时，它只在 `NameTable` 中存储一次。当一个名称空间限定符发生冲突时，它能够使用 `NameTable` 中的字符串进行解析。

在下一个语句中，创建了一个 `XmlDocument` 对象，并且利用文档组件中的内容进行填充。下一步，为了搜索，对象的内容被扫描。注意 `C# IndexOf` 方法与 `Visual Basic InStr` 方法相同。如果找到搜索关键字，目录和文件名被添加到窗体中的列表框中，程序返回 `True`。否则就是找不到匹配选项，程序返回 `False`。

- 16 最后，在 `Form1 Designer` 中，双击关闭按钮然后在程序中添加下面的语句：

```
. Close();
```

测试程序：

- 17 在磁盘 C 或者是您选择的任何位置上，创建一个名为 `WordDocuments` 的文件夹。
.
- 18 复制 `SampleWordDocument.docx` 文档和 `AnotherSampleWordDocument.docx` 文档到
. `WordDocuments` 文件夹中。

- 19 打开 Word 2007 并且创建一个新的文档。
.
- 20 添加一些文本（不要包含 Eagle）并且将文档保存在 WordDocuments 文件夹下，名称为
. SearchSampleDocument.docx 。
- 21 按 F5 运行此项目。
.
- 22 在窗体中，在 Search Directory 框中，输入.docx 文档的位置。
.
- 23 在 Search Term 框中，输入 Eagle，然后点击 Search。
. 包含“Eagle”的两个文档显示在列表框中，如图 6 所示。SearchSampleDocument.docx 文档没有显示。

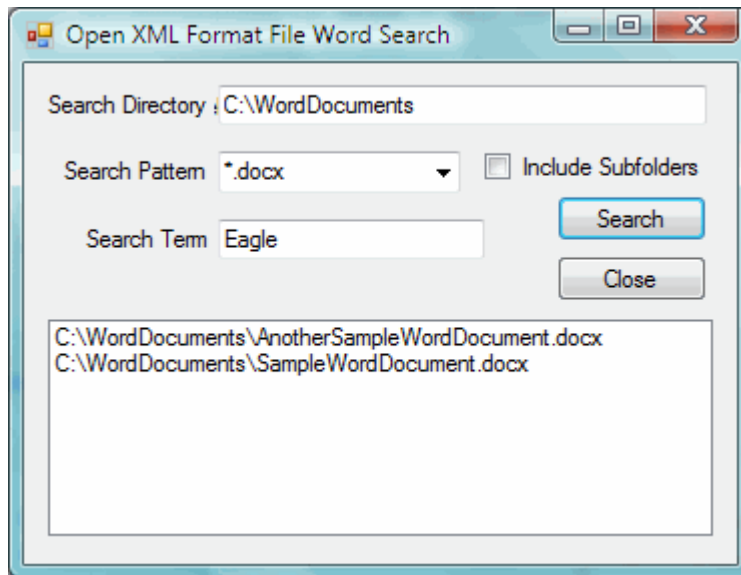


图 6. 运行关键字搜索的结果

- 24 关闭窗口。
.

这就是您需要搜索文档的所有代码，此种情况不需要使用 Word 2007。您需要添加一些小功能：计算搜索文档的个数，添加替换功能，或者搜索其它组件。

[返回页首](#)

结论

在这篇文章中，您熟悉了 Office Open XML 格式文件的结构。您浏览了文件格式，同时演示了如何使用标准的 XML 和 ZIP 技术轻松的访问和编辑 2007 Microsoft Office 文档。您也学习到了如何手动和编写程序的方式操作 Office Open XML 格式文件。通过这些内容，您具备了为组织创建自定义应用程序的基础。

[返回页首](#)

额外资源

更多信息，请查看下面的资源：

- [Microsoft Office Developer Center](#)
- [Microsoft Office Developer Center: What's New for Developers in the 2007 Microsoft Office System](#)
- [Introducing the Office \(2007\) Open XML File Formats](#)
- [Walkthrough: Word 2007 XML Format](#)

[↑返回页首](#)