对文件“file02\_office-word03.doc”的分析

用到的工具有：

|  |  |
| --- | --- |
| notepad++ |  |
| ultraedit |  |
| offvis（微软官方的office文档分析工具） |  |
| github oletools |  |
|  |  |

参考文章：

<https://www.cnblogs.com/Renyi-Fan/p/9742165.html>

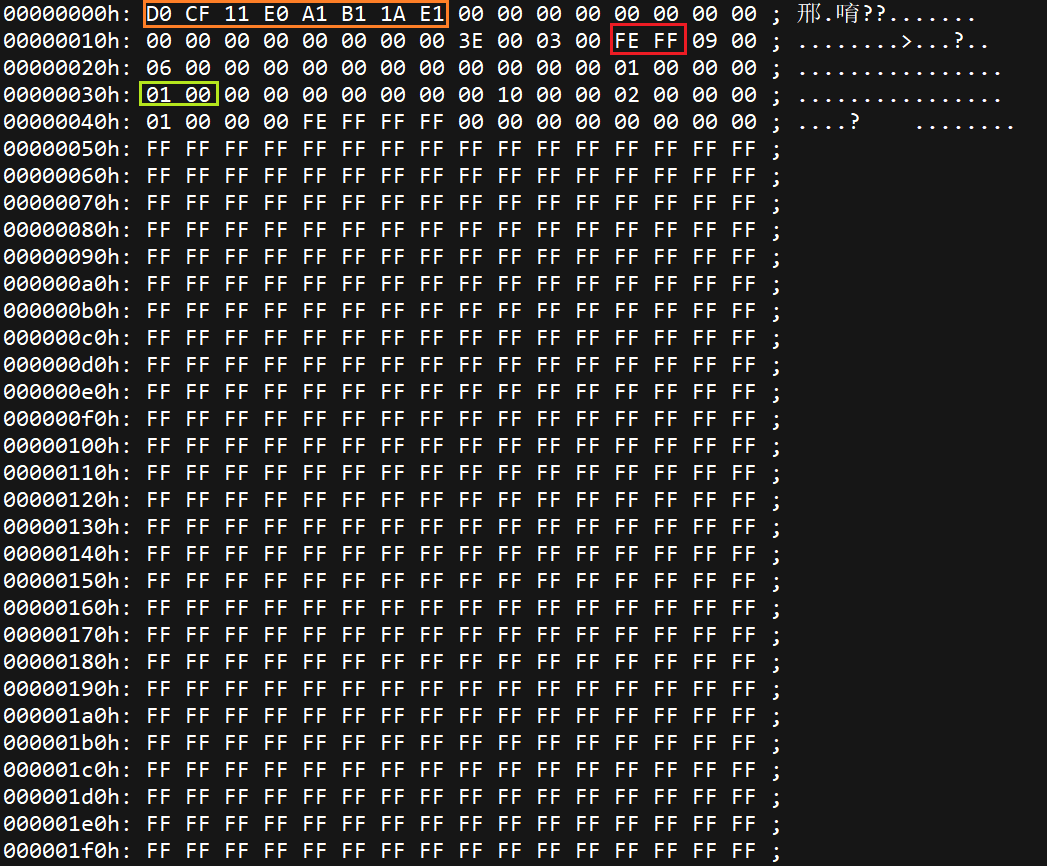
doc文件结构可以到官网下载：

<https://docs.microsoft.com/zh-cn/openspecs/office_file_formats/ms-offfflp/8aea05e3-8c1e-4a9a-9614-31f71e679456>

<https://docs.microsoft.com/en-us/openspecs/office_file_formats/ms-doc/ccd7b486-7881-484c-a137-51170af7cc22>

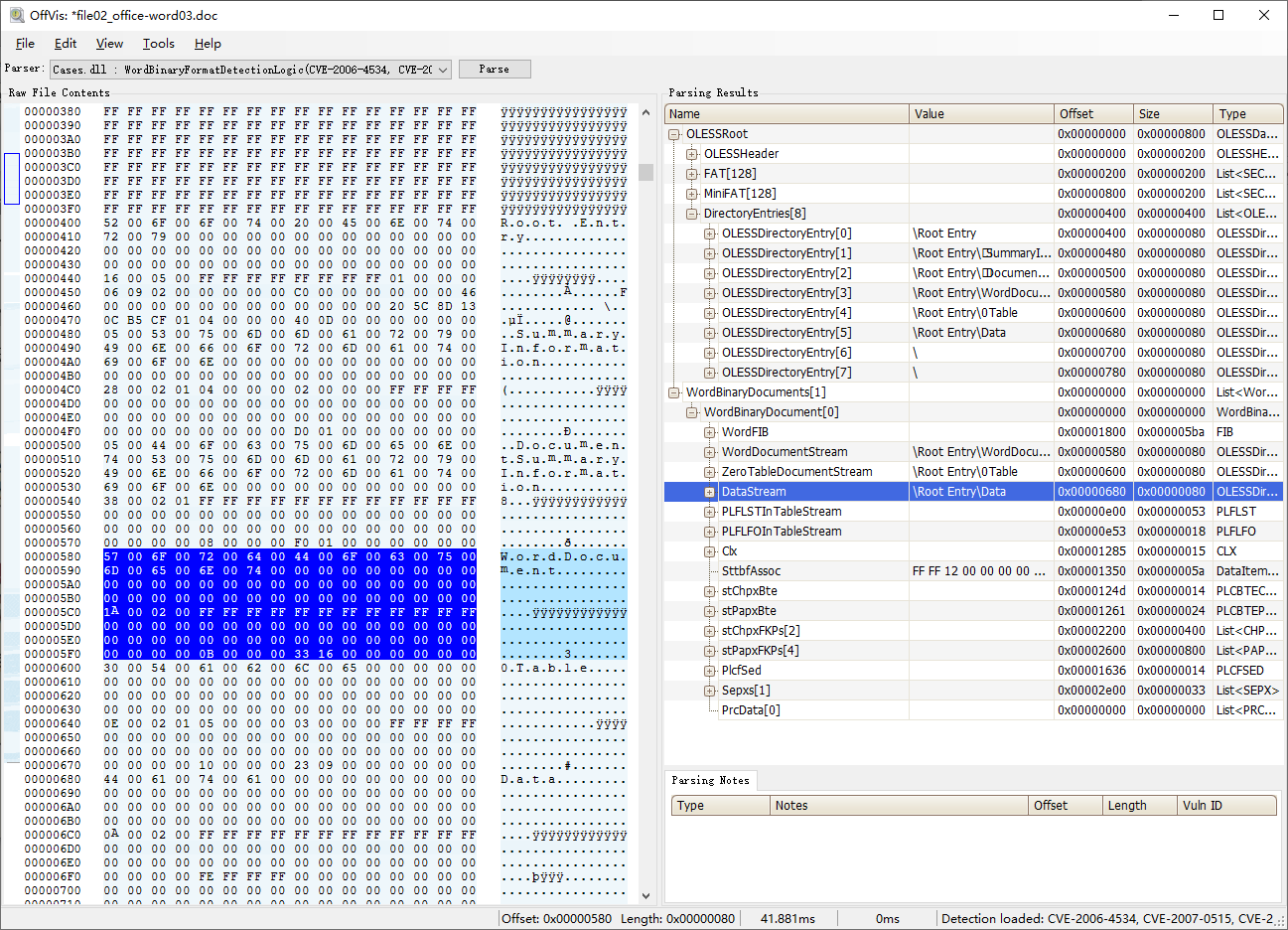
主要是[MS-DOC].pdf

# 文件头



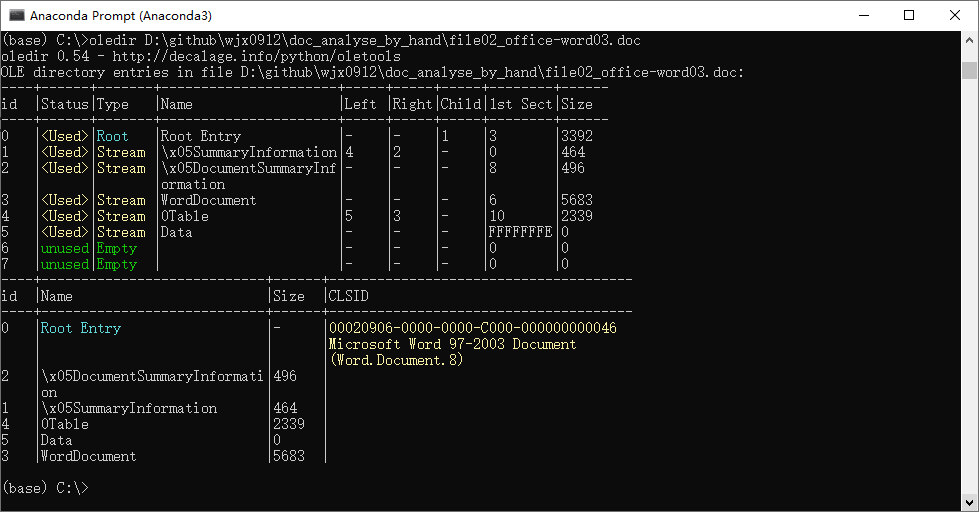
1. Header的前8字节Byte[]，也就是整个文件的前8字节，都是固定的0xD0 0xCF 0x11 0xE0 0xA1 0xB1 0x1A 0xE1，如果不是则说明不是复合文件。
2. 从008H到017H的16字节，是Class Id，不过很多文件都置的0。
3. 从018H到019H的2字节UInt16，是文件格式的次要版本，值：003e。
4. 从01AH到01BH的2字节UInt16，是文件格式的主要版本，值：0003。
5. 从01CH到01DH的2字节UInt16，是固定为0xFE 0xFF，表示文档使用的是Little Endian（低位在前，高位在后）。
6. 从01EH到01FH的2字节UInt16，是Sector大小的幂，默认为9（0x09 0x00），即每个Sector为512字节。
7. 从020H到021H的2字节UInt16，是Mini-Sector大小的幂，默认为6（0x06 0x00），即每个Mini-Sector为64字节。
8. 从022H到023H的2字节UInt16，是预留的，必须置0。
9. 从024H到027H的4字节UInt32，是预留的，必须置0。
10. 从028H到02BH的4字节UInt32，是预留的，必须置0。
11. 从02CH到02FH的4字节UInt32，是FAT的数量，为什么是1？
12. 从030H到033H的4字节UInt32，是Directory开始的SectorID。
13. 从034H到037H的4字节UInt32，是用于事务的，必须置0。
14. 从038H到03BH的4字节UInt32，是最小串（Stream）的最大大小，默认为4096（0x00 0x10 0x00 0x10）。
15. 从03CH到03FH的4字节UInt32，是MiniFAT表开始的SectorID。
16. 从040H到043H的4字节UInt32，是MiniFAT表的数量。
17. 从044H到047H的4字节UInt32，是DIFAT开始的SectorID。
18. 从048H到04BH的4字节UInt32，是DIFAT的数量。
19. 从04CH到1FFH的436字节UInt32[]，是前109块FAT表的SectorID。

用OffVis打开并解析：



文件大小是12288字节，总共24个sector（24\*512=12288）

oletools分析DirectoryEntries：



# 简单描述doc

word文件(.doc)包含：text, tables, fields, pictures, embedded XML markup, and other document content

Word二进制文件的基本单位是字符。 包括字母，数字和标点符号，还包括格式化字符，例如段落标记，单元格结束标记，换行符或分节符。锚字符，例如脚注参考字符，图片锚和注释锚。字符通过**CP**索引（Character Position），一个**PLC**表示一系列**CP**。

# 格式

字符，段落，节，表和图片的格式被指定自己的格式，它们与默认格式不同。使用Prl表示单个属性 或者Sprm（仅仅包含修改信息）。例如，sprmCFBold修改文本的粗体格式，而sprmPDxaLeft修改段落的逻辑左缩进。

文本，段落和表格的最终属性集来自样式的层次结构和直接应用的Prl元素（例如，通过用户选择一些文本并在用户界面中单击加粗按钮）。样式允许以紧凑的方式指定复杂的属性集。

# 表格

表格由一组段落组成，这些段落应用了一组特定的属性。 有一些特殊字符表示表单元格的结尾和表行的结尾，但是没有任何字符表示表单元格的开头或整个表的结尾。 表可以嵌套在其他表中。

# 图片

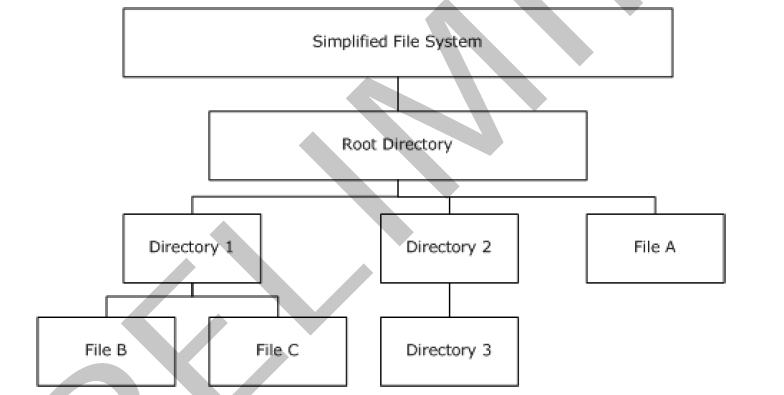
图片可以是内联的或者浮动的。内联图片的sprmCFSpec属性为1，sprmCPicLocation指定图片位置。浮动图片的值是一个锚。

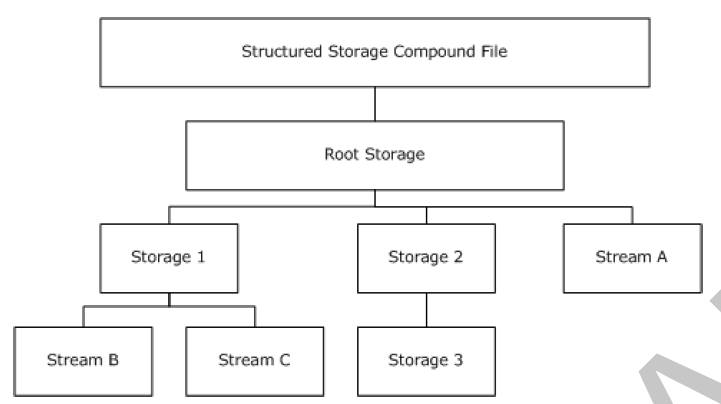
# MS-CFB

搜索MS-CFB.doc，微软官网可以找到。

doc文件属于OLE复合文件的一种。所谓OLE复合文件，是将文本图片表格等等对象复合到一个文件中，它的规范就是MS-CFB。

在一个文档中存储多种对象（比如包含了图片文件）时，就会遇到挑战。复合文件通过在文件内实现简化的文件系统来提供解决方案。结构化存储定义了如何将单个文件视为两种类型的对象（存储对象storage和流对象stream）的分层集合，它们分别充当目录和文件。结构化存储的目的是减少与将单独对象存储在平面文件中相关的性能损失和开销。



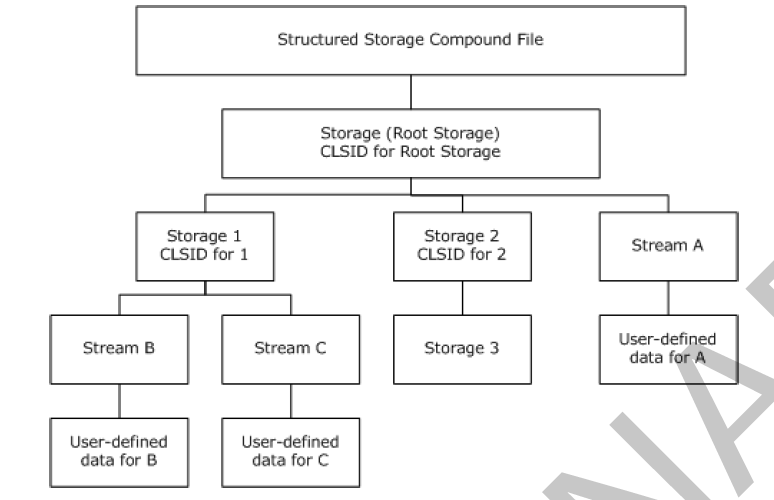


存储对象类似于文件系统目录可以包含其他目录和文件一样，存储对象也可以包含其他存储对象和流对象。也像目录一样，存储对象跟踪子存储对象以及嵌套在其下的流对象的位置和大小。

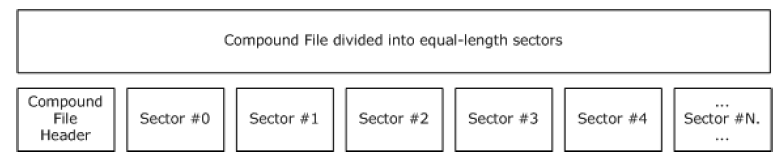
流对象类似于文件的传统概念。像文件一样，流包含用户定义的数据，这些数据存储为连续的字节序列。

层次结构由父对象/子对象关系定义。流对象不能包含子对象。存储对象可以包含流对象或其他存储对象，每个流对象都具有一个在其父存储对象的子对象中唯一标识它的名称。

根存储对象没有父对象。根存储对象也没有名称；因为名称用于标识子对象，所以不需要根存储对象的名称，并且文件格式不提供其表示形式。



复合文件可以这样读取：



最前面的Compound File Header也就是FIB（file info block）；

# CFB文件结构

xxx