**Summer Studentship Research Project - 2016/2017**

Analysing DOC files for patterns to identify which MSWord product version created the file

**By**

William Sanson

**Supervisors:**

Nigel Stanger, Brendon Woodford

University of Otago

**Project Overview**

The aim of this project was to find a way of identifying which version of MSWord a specific DOC file was created with. Most of the software that currently exists seems to only look at the container file with which the Word data streams are part of. This information is stored in directory sectors that could potentially be altered, so it is not completely reliable. Another way of identifying the specific version would be to look at the data streams created by Word and look for patterns or features that are specific to that version.

I first tried using n-grams to analyse the difference in the histograms between two different versions of a file. This did not very successful as the number of n-grams increases somewhat exponentially as the length of the n-gram increased. The larger n-gram (32 bytes) that would have been the most useful had around 15 million different combination for an average sized file.

Instead of the n-grams, I focused on looking at the File Information Block (FIB) of the WordDocument stream. It contains information about the creation of the file including the product version that created it. However, even after periodically updating each version of MSWord, when the security updates and/or service packages were applied there was no obvious change to the FIB, and nothing to identify that an update had been applied.

Files that were used during this project have been included with this file. These include:

* VBScripts for automating Word and the creation of random DOC files with certain features (tables, paragraphs, pictures etc.)
* various Word documents for MSWord versions 1997, 2000, 2002, 2003, 2007, 2010, 2013, 2016, as well as some Macintosh MSWord files and pre-MSWord 1997 versions.
* Python scripts for identifying the MSWord versions that created the DOC files.
* Also included are the main PDF for understanding the MS-DOC file format.

**Software and Applications Used**

***Python*** - Basic scripts for reading byte sequences  
***Notepad****++* - Used for editing VBScripts as well as viewing files in hex (required hex add-on)  
***010******Editor*** (Free month trial) - Used for comparing files in hex, compare variable length byte sequences   
***Office******Visualizer******Tool*** – Allows you to view a side by side comparison of the raw hex value on one side and the parsed values with their corresponding data structure.

**Features of the Microsoft Word Binary File Format**

MSWord uses a Compound File Binary Format (**CFBF**) as its underlying container file which combines multiple “virtual streams” into one file. It is basically a FAT filesystem and is split into sectors and contains an allocation table referencing these sectors. The CFBF uses different types of sectors (Fat, Directory, Minifat) which can be placed anywhere in the file. The exception to this is the CFBF Header which always starts at offset zero in the file, and is 512 bytes long.

The Directory sectors contain information about the different streams in the file and give information about when these streams where created, modified, and a reference to the location for the start of the stream.

When a file is parsed using the Office Visualizer Tool (OVT), it has two main sections the OLESSROOT which contains the information about the FAT tables, sectors, and file container properties. The other section is the WordBinaryDocument. This section contains information about the streams generated from the Word Document. The two main streams that are generated when a word document is created is the WordDocument Stream and the OneTableDocument Stream (1Table stream or 0Table stream).

The WordDocument Stream is the main stream and begins right after the CFBF header. The WordDocument Stream starts with a File Information Block (**FIB**) which contains a reference to all other information in the file. Two integer values represent these references, one which represents the location (structure names are prefixed with ‘**fc’**), and one which represents the size (names prefixed with ‘**lcb’**).

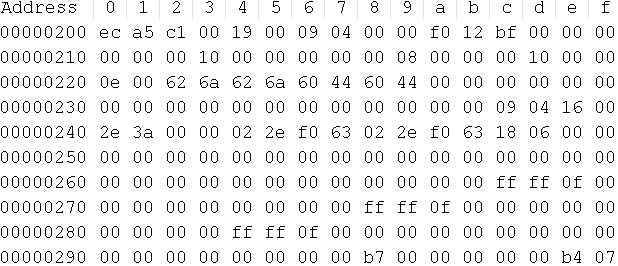
**File Information Block(FIB)**

The FIB defines the start of the Word file after the CFBF header and consists of four substructures. The first being the FIB header, which contains information about the files creation. The other sections are arrays that contain information about the language, time of creation, size and length of document/sections, and the size and location of the data structures.

The following table highlights the important/usable information found in the FIB subsections.

**Table 1:** Information in the FIBBase/ FIB header, and the FibRGW97 array, and FibRgLw97 structure.

|  |  |  |  |
| --- | --- | --- | --- |
| **Offset (0x)** | **Name** | **Bitfield Mask (0x)** | **Comments** |
| **0000** | wIdent | ECA5 | Magic Number that indicates that this is a Word Binary File |
| **0002** | nFib |  | FIB version (greater than 101 for Word version 6.0+ |
| **0004** | nProduct |  | Product version that wrote the file |
| **0006** | Lid |  | Language stamp |
| **0012** | Envr | 00 = Word for Windows 01 = Word for Mac | Environment in which file was created |
| **0013** | fMac | 11 or 01 = Last saved in Mac | Environment last saved in. |
| **0022** | wMagicCreated | 6A62 | Creator ID for word. |
| **0044** | 1ProductCreated |  | Build date of the creator program (when it was compiled) |
| **0048** | 1ProductRevisited |  | Build date of the file’s last modifier (word or another program) |



**Fig 1:** The FIB that starts at offset 0x200, directly after the 512 byte CFBF Header.

**Character and Paragraph Formatting Properties**

There are 5 property defining data strucutres. A CHP (Character Properties), PAP (Paragraph Properties), TAP (Table Properties), SEP (Section Properties), and PIC(Picture Desciptor).

Character and paragraph properties are stored compressed as follows:

1. A PAP is generated for each paragraph and a CHP is generated for each character.
2. Each paragraph in a word document inherits a default paragraph style from one of the recoreded paragraph styles in the documents Style Sheet (STSH) data structures.
3. The compressed PAP (PAPX) is created by comparing the differences between the generated PAP and the STSH PAP, and storing these differences as a list of SPRMs (opcodes).

CHPs are stored similarily and use the STSH PAP and CHP to make comparisons and create the CHPX, stored as a list of SPRMs.

The other data structures are not compressed.

Using the Office Visualisor Tool it is possible to identify where the data structures are stored. However, there is no obvious pattern for storing the data structures any differently across the MSWord versions (possibly due to the encoding) other than the amount of stored styles in the documents style sheet. In word 2002 more styles were added for Tables and Lists. For the each PAPX, it contains a **istd** value which indicates which built in style has been applied. Values 107-155 were added in 2002.

**Summary**

Aside from the **istd** value, I have found nothing other than what is available in the FIB to identify which version a document was created with. I have included a basic script that can be used to read values in the FIB to identify which version of word was created with. There are however issues with this because when updates are applied there is no pattern of change to the **nProduct** value (Note: values are stored in Little Endian).

For example the service pack updates for MSWord 2000 nProduct values were as follows:

SP1: 4D20   
SP2: 5B20  
SP3: 0B20

The same occurred for MSWord 2003:

SP1: 7F60  
SP2: 0360  
SP3: 2360

The most significant byte seems to indicate which version of word the file was created with.

**Table 2:** MSWord version and its corresponding nProduct

|  |  |
| --- | --- |
| **Word Version** | **nProduct** (most significant byte) |
| MSWord 1997 (v8.0) | ‘00’ up to ‘09’ |
| MSWord 2000 (v9.0) | ‘20’ up to ‘29’ |
| MSWord 2002 (v10.0) | ‘40’ up to ‘49’ |
| MSWord 2003 (v11.0, 11.1-11.4, 11.6) | ‘60’ up to ‘69’ |
| MSWord 2007 (v12.0) | ‘80’ up to ‘89’ |
| MSWord 2010 (v14.0) | ‘C0’ up to ‘C9’ |
| MSWord 2013 (v15.0) | ‘E0’ up to ‘E9’ |
| MSWord 2016 (v16.0) | ‘00’ up to ‘09’ |

The environment that is was created on (Windows or Mac) and the environment it was last saved on could also prove to be valuable, especially for the older versions of MSWord, where there were differences between the Windows and Mac versions. They are referenced in the FIB header as **Envr** and **fMac** respectively.

One of the included files (check\_doc\_version.py) is basic version checker for identifying the MSWord version used to create the file. It uses the most significant byte of the **nProduct** and checks the **fMac** to see which environment it was last saved in to determine the version. It also looks at offset 0x299 for distinguishing between MSWord 1997 and MSWord 2016, which seems to be have the value ‘B7’ for MSWord 2007-2016.