Table Formatting

DIaLOGIKa/makz 15. April 2008

Contents

Introduction

This document explains how tables are stored in a binary .doc file and how they are formatted. It will not explain the formatting of text in tables because this is related to *paragraph* and *run* formatting.

This guide implies a basic knowledge of paragraph formatting and the binary .doc file format in general and should be used as a supplement to the official specification.

Basic Structure of a Table

Basically, tables are used to structure text like paragraphs or line breaks, consequently, they are stored in a similar way as paragraphs.

The several rows and cells of a table are separated by a special ASCII character in the text stream of the document. The ASCII code of this character is 7 and the character escape code is *\a*.

The first cell of the table is formatted by a PAPX with *fInTable* set to 1. The end of each row is marked by an additional *\a* character that is formatted by a PAPX with *fInTable* and *fTtp* set to 1.

Example

The following part of a document is used to demonstrate the format of a table in the binary .doc file format:

Above

Cell1	Cell2	
Cell4	Cell5	Cell6

Below

This example is stored in the text part of the binary document as follows:

A	b	o	v	e	C	e	l	l	1	\a	C	e	l	l	2	\a	\a	\a	C
e	l	l	4	\a	C	e	l	l	5	\a	C	e	l	l	6	\a	\a	B	e
l	o	w																	

The following PAPX are stored for this table:

From character position	To character position	fInTable	fTtp
0	4	0	0
5	10	1	0
11	16	1	0
17	17	1	0
18	18	1	1
19	24	1	0
25	30	1	0
31	36	1	0
37	37	1	1
38	42	0	0

Actual Table Formatting

Tables are formatted row-wise. That means that every row has its own formatting which stores information about the formatting of the row and corresponding cells in the row.

This information is stored in a PAPX that is assigned to the end character of the row and contains special table SPRMs. This is why such a PAPX is called TAPX (Table Property Exceptions).

Properties which are related to the whole table, such as indentation, alignment or text wrapping are stored in the TAPX of each row.

Table Definition SPRM

A TAPX holds several SPRMs which modify single properties of a table, table row or table cell, but one of these TAPX SPRMs is very special and is stored in likely every TAPX.

This SPRM is called *sprmTDefTable* and has operation code 0xD608. The special thing about it is that it stores most part of the table cell formatting and, thus, more than one property.

As with many other variable length SPRMs the first two bytes contain the operation code and the next two bytes contain the length of the following structure. The structure itself starts with the count of cells which are affected. This is the count of cells in the row. This count is called *itcMac*.

The next bytes are the left and right boundaries of the cells. These boundaries are coordinates relative to the left page margin and specify the begin and/or end of a cell. There are *itcMac* + 1 boundaries in the SPRM, each boundary is a 16 bit integer value.

Following are several blocks of cell formatting. Each block has a length of 20 bytes and stores formatting information of one cell. So there are *itcMac* formatting blocks in the SPRM.

opcode 2 bytes	length 2 bytes	itcMac 1 byte	Boundary 1 2 bytes	…	Boundary n 2 bytes	Formatting 1 20 bytes	…	Formatting n 20 bytes

The 20 bytes of cell formatting are separated in a 16 bit integer containing flags, followed by a 16 bit integer standing for the width of the cell. After that there are four 4 byte BRC structures for the four borders of the cell.

Flag Values

The 16 bit integer value contains 11 properties which can be retrieved by unmasking the flag value with a bitmask.

Property	Bitmask
fFirstMerged	0x0001
fMerged	0x0002
fVertical	0x0004
fBackward	0x0008
fRotateFont	0x0010
fVertMerge	0x0020
fVertRestart	0x0040

vertAlign	0x0180
ftsWidth	0x0E00
fFitText	0x1000
fNoWrap	0x2000

Example

The following hex dump is the *sprmTDefTable* of a TAPX with 2 cells:

3 boundaries

itcMac

Flags of cell 1

Width of cell 1

```
(0000)  08 D6 2F 00 02 94 FF 53 03 60 13 00 06 BF 03 00
(0010)  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
(0020)  06 0D 10 00 00 00 00 00 00 00 00 00 00 00 00 00
(0030)  00 00 00
```

Right BRC of cell 2

Top BRC of cell 2

Left BRC of cell 2

Bottom BRC of cell 2