

## Variable-Length Multi-Byte Functions ♦ Variable-Length Multi-Byte Functions

The codes for variable-length multi-byte functions [208 (0xD0) through 239 (0xEF)] appear twice each time the function is invoked. The first occurrence is the begin gate, and the second occurrence is the end gate. In addition, each begin gate is followed by a subgroup byte, a value of size short (16 bits), and a function flags byte. If the flags byte indicates that there is prefix data associated with the function, a number of prefix ID bytes come next, followed by the prefix index-ID words. Next is a word (16 bits) showing the size of the non-deletable information. Any non-deletable function data is then included, followed by any deletable data. The deletable data format is not documented. It pertains to formatter information, can change at any time, and is rewritten each time a document is retrieved into WordPerfect. You can skip over deletable data by using the size word to move to the next function code. Following the deletable data are a size word and the end gate. The size of the function is the total size of the function including gates.

The structure of variable-length functions is represented as follows (indented fields may or may not exist depending on the flags field):

```
▶          <group>
          <subgroup>
          [size]
          <flags byte>
            <number of prefix IDs>
            [prefix ID 1]
            ...
            [prefix ID n]
          [size of non-deletable information]
          <non-deletable information> x non-deletable size
          <undocumented deletable data> x ?
          [size]
          <group>◀
```

The prefix ID information will exist only if bit 7 of the flags byte is set (see *Flags Byte Information* below). Be aware that prefix ID information may be added to functions at any time. The undocumented deletable data may or may not appear in a file created by WordPerfect. Developers should never create a file with deletable data. Deletable data is noted in the documentation as a signal that an indeterminate length of unspecified data may appear in existing files and must be skipped. Be aware that functions which do not currently contain deletable data may be modified at any time to include deletable data.

### **Flags Byte Information**

The function flags byte is defined as follows:

```
▶          bits 0-2:
            0 = not a reverted, paired, or encased function.
```

1 = Simple paired function. Begin/On codes are even subfunctions and End/Off codes are the next subfunction.  
 2 = Encased/paired function. Begin/On codes are mod 4=0 subfunctions (multiple-of-4 subfunctions) followed immediately by Begin/Off, End/On and End/Off codes numbered consecutively.  
 3 = Encased function. Begin/On codes are even subfunctions and End/Off codes are the next odd subfunction.  
 4 = reserved  
 5 = reserved  
 6 = revert function off.  
 7 = revert function on.

bit 3: reserved - 8 (0x08)  
 bit 4: reserved - 16 (0x10)  
 bit 5: reserved - 32 (0x20)  
 bit 6: 1 = ignore function - 64 (0x40) this function is inactive due to the context of a function enclosing it.  
 bit 7: 1 = prefix IDs follow - 128 (0x80) represented as PRFXID◀

◆ **Function 208 (0xD0) = End-of-Line Group**

The End-of-Line Group subfunctions have EOL orientation.

Subfunctions 0 to 28 (0x1C) of this group are interchangeable with the single-byte function codes 180 (0xB4) to 207 (0xCF). WP6.x uses the deletable area of the multi-byte functions to store formatter data and will change between corresponding codes as needed. A program reading WP6.x documents must handle both the multi-byte and single-byte functions. A program writing documents should use only the single-byte function codes.

An additional level of subfunctions is defined for the End-of-Line function group which will be called *embedded subfunctions*. These embedded subfunctions will be contained between the begin and end gates of an End-of-Line group subfunction. They are not used with any other functions. See *End-of-Line Group Non-Deletable Embedded Subfunctions* below for definitions of the individual embedded subfunctions. Deletable embedded subfunctions exist but are not documented. An application programmer only needs to be aware of the deletable data in order to skip over it in existing documents. Since the corresponding single-byte functions are used to create a document, an application never generates functions in this group.

**Subfunctions: 0 (0x00) - 28 (0x1C)**

Subfunction	Description	Conversion/Search mappings
0 (0x00)	Beginning of File	Ignore
1 (0x01)	Soft End of Line	Space
2 (0x02)	Soft End of Column	Space
3 (0x03)	Soft EOC at EOP	Space

<b>Subfunction</b>	<b>Description</b>	<b>Conversion/Search mappings</b>
4 (0x04)	Hard End of Line	Hard Return
5 (0x05)	Hard EOL at EOC	Hard Return
6 (0x06)	Hard EOL at EOP	Hard Return
7 (0x07)	Hard End of Column	Hard End of Column
8 (0x08)	Hard EOC at EOP	Hard End of Column
9 (0x09)	Hard End of Page	Hard End of Page
10 (0x0A)	Table Cell	Table Cell
11 (0x0B)	Table Row and Cell	Table Row
12 (0x0C)	Table Row at EOC	Table Row
13 (0x0D)	Table Row at EOP	Table Row
14 (0x0E)	Table Row at Hard EOC	Hard Table Row
15 (0x0F)	Table Row at Hard EOC at EOP	Hard Table Row
16 (0x10)	Table Row at Hard EOP	Hard Table Row
17 (0x11)	Table Off	Table Off
18 (0x12)	Table Off at EOC	Table Off
19 (0x13)	Table Off at EOC at EOP	Table Off
20 (0x14)	Deletable Soft EOL	Space
21 (0x15)	Deletable Soft EOC	Space
22 (0x16)	Deletable Soft EOC at EOP	Space
23 (0x17)	Deletable Hard EOL	Hard Return
24 (0x18)	Deletable Hard EOL at EOC	Hard Return
25 (0x19)	Deletable Hard EOL at EOP	Hard Return
26 (0x1A)	Deletable Hard EOC	Hard End of Column
27 (0x1B)	Deletable Hard EOC at EOP	Hard End of Column
28 (0x1C)	Deletable Hard EOP	Hard End of Page

The structure for the End-of-Line function group is shown below. This format is unique in that the non-deletable data area also contains deletable data.



<208 (0xD0)> <0 to 28 (0x1C)> [size = variable] <flags = 0 or PRFXID>

If the prefix ID bit is set, the following information exists:

<number of PIDs>

[cell top line template PID (type=0x42)]

[cell left line template PID (type=0x42)]

[cell bottom line template PID (type=0x42)]  
 [cell right line template PID (type=0x42)]  
 [cell fill template PID (type=0x43)]  
 [size of deletable and non-deletable subfunctions]  
 [size of deletable subfunction data]  
 <deletable subfunctions> x ?  
 <non-deletable subfunctions> x ?    may be absent if no actions are performed  
 [size] <0xD0>◀

## Function 208 (0xD0)

### End-of-Line Group Non-Deletable Embedded Subfunctions

The End-of-Line Group embedded subfunctions have EOL orientation.

The End-of-Line Group non-deletable embedded subfunctions are only found within the non-deletable portion of the End-of-Line Group functions: function 208 (0xD0) including subfunctions 0 (0x00) through 28 (0x1C). Some of the embedded subfunctions described below are fixed length and some are variable length.

## 0x80

### Row Information Embedded Subfunction

▶ <128 (0x80)> (size = 5)  
 <row flags>  
   bit 0:    0 = multi-line text (wrapping)  
           1 = single line of text (no wrap)  
   bit 1:    0 = automatic height  
           1 = fixed height  
   bit 2:    0 = not a header row  
           1 = this is a header row  
 [row height if fixed (WPU)]  
 <128 (0x80)>◀

## 0x81

### New Cell Formula Embedded Subfunction

▶ <129 (0x81)> [size = variable]  
 [length of formula]  
 <tokenized formula> x length of formula (see *Table Formulas* for list of formula codes)  
 [length]  
 <129 (0x81)>◀

## 0x82

### New Top Gutter Spacing Embedded Subfunction

▶ <130 (0x82)> (size = 4)  
 [new top gutter spacing (WPU)]  
 <130 (0x82)>◀

## 0x83

### New Bottom Gutter Spacing Embedded Subfunction

▶ <131 (0x83)> (size = 4)  
 [new bottom gutter spacing (WPU)]

<131 (0x83)>◀

## 0x84

### Cell Information Embedded Subfunction



<132 (0x84)> (size = 9)

<flag>

- bit 0: 1 = use cell attributes
- bit 1: 1 = use cell justification
- bits 2-5: not used
- bit 6: 1 = ignore in calculations
- bit 7: 1 = cell is locked

<justification>

- bits 0-2: justification
  - 0 = left
  - 1 = full
  - 2 = center
  - 3 = right
  - 4 = all (kinto waritsuke)
  - 5 = decimal align

<alignment>

- bits 0-1: vertical alignment
  - 0 = top
  - 1 = center
  - 2 = bottom
  - 3 = full

[attribute word 1]

- bit 0: 1 = extra large
- bit 1: 1 = very large
- bit 2: 1 = large
- bit 3: 1 = small print
- bit 4: 1 = fine print
- bit 5: 1 = superscript
- bit 6: 1 = subscript
- bit 7: 1 = outline
- bit 8: 1 = italics
- bit 9: 1 = shadow
- bit 10: 1 = redline
- bit 11: 1 = double underline
- bit 12: 1 = bold
- bit 13: 1 = strikethrough
- bit 14: 1 = underline
- bit 15: 1 = small caps

[attribute word 2]

- bit 0: 1 = blink
- bit 1: 1 = reverse video

<132 (0x84)>◀

## 0x85

### Cell Spanning Information Embedded Subfunction



<133 (0x85)> (size = 4)

<number of cells spanned horizontally> bit 7 is set if spanned from left  
<number of cells spanned vertically> bit 7 is set if spanned from above

<133 (0x85)>◀

## 0x86

### Cell Fill Colors Embedded Subfunction

▶ <134 (0x86)> (size = 10)  
<foreground color (RGSB)> x 4  
<background color (RGSB)> x 4  
<134 (0x86)>◀

0x87

### Cell Line Color Embedded Subfunction

▶ <135 (0x87)> (size = 6)  
<color (RGSB)> x 4  
<135 (0x87)>◀

0x88

### Cell Number Type Embedded Subfunction

▶ <136 (0x88)> (size = 5)  
[number type]  
bits 0-3: number of digits to display after decimal point or date index information  
bit 4: 1 = display with commas  
bit 5: rounding  
0 = use full precision in calculations  
1 = use displayed precision  
bits 6-7: negative numbers  
0 = '-' (use minus sign)  
1 = ( ) (use parentheses)  
2 = CR/DR (use credit/debit symbols)  
3 = not defined  
bits 8-11: standard formats  
0 = general  
1 = integer  
2 = fixed  
3 = percent  
4 = currency  
5 = accounting  
6 = commas  
7 = scientific  
8 = date  
9 = text  
bit 12: 1 = text  
bit 13: 1 = set currency symbol on  
bits 14-15: notation  
0 = floating point  
1 = scientific  
2 = fixed  
3 = not defined  
<currency type>  
bits 0-6: index of the desired currency, default = 0  
bit 7: alignment  
<currency type flags>  
bits 0-3: number of digits for decimal  
bit 7:  
0 = currency symbol after  
1 = currency symbol before  
<136 (0x88)>◀

## 0x89

### Cell Floating Point Number Embedded Subfunction

- ▶ <137 (0x89)> (size = 11)  
<IEEE format floating point number> x 8  
<floating point source>  
<137 (0x89)>◀

The IEEE value is an 8-byte floating point value. The format of the 8 bytes is separated into the following bit fields going from the high order bit to the low order bit.

- ▶ bit 63: Sign bit. 0 = positive value, 1 = negative value  
bits 62-52: Exponent [if exponent is 0x3FF, the value is 1 (fraction value)]. The decimal point is to the left of bit 51, and to the right of bit 52 of the fractional bits. If this value is non-zero, there is an imaginary bit 52 that is set to 1. If this value is zero, the value for the floating point number is zero.  
bits 51-0: Fractional value in base 2 number system. Binary values are used in the following chart since they are base 2.◀

Floating Point Examples			
Sign Bit	Exponent Bits	Fraction Bits	Result
1	0x03FF	100111b	-1.100111b
0	0x03FE	100111b	.1100111b
0	0x0400	100111b	11.00111b
1	0x040A	100111b	-110011100000.0b

If the exponent bits are non-zero, it is assumed that there is an extra bit set to 1 (imaginary bit 52) to the left bit 51 of the fractional value. If the exponent bits are zero, the value for the floating point number is zero.

When you work on a computer that uses reverse-order format (Intel\* processor chips), you need to reverse these 8 bytes to address the bits correctly.

### Explanation of the Result in the Last Example Above

The fractional bits are as follows, moving from the highest-order fractional bit to the lowest-order fractional bit:

bit 51 = 1  
bit 50 = 0  
bit 49 = 0  
bit 48 = 1  
bit 47 = 1  
bit 46 = 1  
bits 45-0 = 0

This is the value 100111b. However, since the exponent value is non-zero there is also an imaginary bit 52 that is set to 1. The value becomes 1100111. Depending on the exponent, the decimal point floats right or left, adding zeros as needed. For the last example the exponent is 0x040A. The decimal is placed 11 places to the right of bit 51, and the result becomes 110011100000.0b.

**0x8A**

**Reserved**

**0x8B**

**Cell Prefix Flag Embedded Subfunction**

- ▶ <139 (0x8B)> (size = 3)  
<flags to indicate if default is zero or none>  
<139 (0x8B)>◀

**0x8C**

**Cell Recalculation Error Number Embedded Subfunction**

- ▶ <140 (0x8C)> (size = 4)  
[recalculation error number]  
<140 (0x8C)>◀

**0x8D**

**Don't End a Paragraph Style for this Hard Return**

- ▶ <141 (0x8D)> (size = 1)◀

◆ **Function 209 (0xD1) = Page Group**

The Page Group subfunctions have page orientation.

**0xD100**

**Top Margin Set**

- ▶ <209 (0xD1)> <0 (0x00)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[top margin (WPU)] distance from top edge of paper to text  
<undocumented deletable data> x ?  
[size] <0xD1>◀

**0xD101**

**Bottom Margin Set**

- ▶ <209 (0xD1)> <1 (0x01)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[bottom margin (WPU)] distance from bottom edge of paper to text (WPU)  
<undocumented deletable data> x ?  
[size] <0xD1>◀

**0xD102**

## Suppress Page Characteristics

- ▶ <209 (0xD1)> <2 (0x02)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<suppress code>
  - bit 0: suppress page numbering
  - bit 1: suppress current page numbering and print page number at bottom center
  - bit 2: suppress header A
  - bit 3: suppress header B
  - bit 4: suppress footer A
  - bit 5: suppress footer B
  - bit 6: suppress watermark A
  - bit 7: suppress watermark B<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD103

### Page Number Position

- ▶ <209 (0xD1)> <3 (0x03)> [size = variable] <flags = PRFXID>
  - <number of PIDs = 1>
  - [page numbering font PID (type=0x55)] 0 if use flag is 0
  - [size of non-deletable information = variable]
  - [hash of matched typeface descriptor]
  - <use flag>
    - 0 = use document initial values for font, point size, attributes and colors
    - 1 = use values in function
  - [copy of page numbering font PID]
  - [point size (3600ths)]
  - <page number position>
    - 0 = none
    - 1 = top left
    - 2 = top center
    - 3 = top right
    - 4 = top outside left and right
    - 5 = bottom left
    - 6 = bottom center
    - 7 = bottom right
    - 8 = bottom outside left and right
    - 9 = top inside left and right
    - 10 = bottom inside left and right
  - [matched font index in font list]
  - [matched point size of font (3600ths)]
  - [attribute word 1]
    - bit 0: 1 = extra large
    - bit 1: 1 = very large
    - bit 2: 1 = large
    - bit 3: 1 = small print
    - bit 4: 1 = fine print
    - bit 5: 1 = superscript
    - bit 6: 1 = subscript
    - bit 7: 1 = outline
    - bit 8: 1 = italics
    - bit 9: 1 = shadow
    - bit 10: 1 = redline
    - bit 11: 1 = double underline
    - bit 12: 1 = bold
    - bit 13: 1 = strikeout

bit 14: 1 = underline  
bit 15: 1 = small caps  
[attribute word 2]  
bit 0: 1 = blink  
bit 1: 1 = reverse video  
<color (RGSB)> x 4  
[page number height] calculated  
<new page position override> optional (see <page number position> above)  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD104

### Center Current Page Top to Bottom

▶ <209 (0xD1)> <4 (0x04)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<center current page state>  
bit 0: 0 = off, 1 = center current page  
bits 1-7: undefined  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD105

### Center Page Top to Bottom

▶ <209 (0xD1)> <5 (0x05)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<center page state>  
bit 0: 0 = off, 1 = center page  
bits 1-7: undefined  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD106

### Widow/Orphan On/Off

▶ <209 (0xD1)> <6 (0x06)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<widow/orphan state>  
bit 0: 0 = off, 1 = on  
bits 1-7: undefined  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD107

### Set Space Between Footnotes

▶ <209 (0xD1)> <7 (0x07)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[space between footnotes (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD108

### Set Space Between Endnotes

▶ <209 (0xD1)> <8 (0x08)> [size = variable] <flags = 0>

[size of non-deletable information = 2]  
[space between endnotes (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD109

### Footnote Minimum Amount to Keep Together

▶ <209 (0xD1)> <9 (0x09)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[footnote minimum amount (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

### 0xD10A Endnote Minimum Amount to Keep Together

▶ <209 (0xD1)> <10 (0x0A)> [size = variable] <flags = 0>  
[size of non-deletable information = 2][endnote minimum amount (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD1>◀ ☹ ♦

## 0xD10B

### Footnote Number Restart on Each Page

▶ <209 (0xD1)> <11 (0x0B)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<restart footnote number state>  
bit 0: 0 = off, 1 = restart number of each page  
bits 1-7: undefined  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD10C

### Footnote Continued Message

▶ <209 (0xD1)> <12 (0x0C)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<footnote continued state>  
bit 0: 0 = off, 1 = output footnote continued message  
bits 1-7: undefined  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD10D

### Footnote Text Position

▶ <209 (0xD1)> <13 (0x0D)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<footnote text position>  
bit 0: 0 = bottom of page, 1 = after text  
bits 1-7: undefined  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD10E

### Footnote Separator Line

- ▶ <209 (0xD1)> <14 (0x0E)> [size = variable] <flags = PRFXID>  
 <number of PIDs = 1>  
 [line information PID (type=0x40)]  
 [size of non-deletable information = 9]  
 [space above line (WPU)]  
 [space below line (WPU)]  
 <alignment>  
 0 = left  
 1 = full  
 2 = center  
 3 = right  
 4 = set position  
 [width of line, if not full]  
 [position of start of line, if alignment = set position]  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD10F

### Binding Width

- ▶ <209 (0xD1)> <15 (0x0F)> [size = variable] <flags = 0>  
 [size of non-deletable information = 3]  
 <binding flag>  
 bits 0-1:  
 0 = left  
 1 = right  
 2 = top  
 3 = bottom  
 bits 2-7: undefined  
 [binding width (WPU)]  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD110

### Page Border

- ▶ <209 (0xD1)> <16 (0x10)> [size = variable] <flags = PRFXID>  
 <number of PIDs (minimum 2)>  
 [border template PID (type=0x44)] required, may be 0  
 [fill template PID (type=0x43)] required, may be 0  
 If number of PIDs > 2, then any border line template PIDs appear in the following order as indicated by the border override flags.  
 [left side line template PID (type=0x42)]  
 [right side line template PID (type=0x42)]  
 [top side line template PID (type=0x42)]  
 [bottom side line template PID (type=0x42)]  
 [separator side line template PID (type=0x42)]  
 [size of non-deletable information = variable]  
 <reserved> x 14  
 [total size of override data] not including this word  
 [border override flags]  
 See *Specific Format of Border Override Flags and Data* below.  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

### General Format of Border Override Flags and Data

The border may have associated override flags. The override flags are a 16-bit entity (short value) and each bit corresponds to data in the border definition.

For each bit in the border override flags, there is corresponding data. The size of the data depends on what the override bit represents. The order of the data depends on the order of the override bits, with one exception, noted below. For example, if bits 15 and 14 in the override flag are both set, the data corresponding to bit 15 appear earlier in the override data area than the data corresponding to bit 14. The exception is for border or fill template overrides. Because the data are prefix IDs, the data appear in the prefix ID area at the beginning of the function. The order of the prefix IDs corresponds to the bit order of the override flags.

If an override bit is set in the border override flags, a mask appears defining which bits within the corresponding data are overridden followed by the actual override data. For example, assume a data byte has two fields, the first field covering bits 0-3 and the second field covering bits 4-7. Assume the first field (bits 0-3) is to be overridden but the second field (bits 4-7) is not. The corresponding override bit will be set in the border override flags word. The data corresponding to this override bit will be a mask byte (value 0x0F) and a data byte, of which only the bottom 4 bits matter.

### Specific Format of Border Override Flags and Data

Bit 15: Border counter data

If the border information override bit is set in the border override flags, the following data appear:

[total size of border template override data] not including this word  
[border template override flags]

The following data appear according to bits set in the border template override flags:

Bit 15: Border PID flags. Corresponding data:

<mask>

<data>

Bit 14: Border general flags data. Corresponding data:

<mask>

<data>

Bit 13: Corner radius data. Corresponding data:

[corner radius]

Bit 12: Inside spacing data. Corresponding data:

[left inside spacing]

[right inside spacing]

[top inside spacing]

[bottom inside spacing]

Bit 11: Outside spacing data. Corresponding data:

[left outside spacing]

[right outside spacing]

[top outside spacing]

[bottom outside spacing]

Bit 10: Drop shadow data. Corresponding data:

<drop shadow flag>

[drop shadow spacing]

<shadow color (RGSB)> x 4

Bit 9: Border color data. Corresponding data:

<border color (RGSB)> x 4

Bit 8-0: currently undefined  
 Bit 14: Fill data override.  
 If the fill information override bit is set in the border override flags, the following data appear:  
 [total size of fill template override data] not including this word  
 [fill template override flags]  
 The following data appear according to bits set in the fill template override flags:  
 Bit 15: Fill colors override. Corresponding data:  
     <foreground/start color (RGBA)> x 4  
     <background/start color (RGBA)> x 4  
 Bit 14-0: currently undefined  
 Bits 13-0: currently undefined◀

## 0xD111

### Form

▶ <209 (0xD1)> <17 (0x11)> [size = variable] <flags = 0>  
 [size of non-deletable information = 82]  
 <matched form hash table index>  
 [matched form hash value]  
 [length (WPU)]  
 [width (WPU)]  
 <type>  
 <orientation>  
     0 = portrait  
     1 = landscape  
 <type name length>  
 [type name] x 36 word string  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD112

### Form Labels

▶ <209 (0xD1)> <18 (0x12)> [size = variable] <flags = 0>  
 [size of non-deletable information = 100]  
 [width (WPU)]  
 [length (WPU)]  
 <number of columns per page>  
 <number of rows per page>  
 [label width (WPU)]  
 [label length (WPU)]  
 [left offset - top-left corner (WPU)]  
 [top offset - top-left corner (WPU)]  
 [distance between label columns (WPU)]  
 [distance between label rows (WPU)]  
 [label left margin (WPU)]  
 [label right margin (WPU)]  
 [label top margin (WPU)]  
 [label bottom margin (WPU)]  
 [label description] x 37 word string  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD113

### Double-Sided Printing

- ▶ <209 (0xD1)> <19 (0x13)> [size = variable] <flags = 0>  
 [size of non-deletable information = 1]  
 <double-sided printing mode>  
     bit 0:     0 = off, 1 = on  
     bit 1:     0 = long edge duplexing, 1 = short edge duplexing  
     bits 2-7: undefined  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD114

### Logical Pages

- ▶ <209 (0xD1)> <20 (0x14)> [size = variable] <flags = 0>  
 [size of non-deletable information = 2]  
 <number of columns per page>  
 <number of rows per page>  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD115

### Delayed Codes

- ▶ <209 (0xD1)> <21 (0x15)> [size = variable] <flags = PRFXID>  
     <number of PIDs = 1>  
     [delayed codes PID (type=0x08)]  
 [size of non-deletable information = 4]  
 [hash of codes]  
 [number of pages to delay]  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

The Page Delay feature gives the user the option of setting the page formatting for subsequent pages. The user may specify at the beginning of the document (or any other place) how the page formatting will appear one or more pages later.

Consider the example of the first page of a document that is printed on letterhead, but all remaining pages are printed on standard paper. At the beginning of the document the user may specify a two-inch top margin to leave room for the letterhead on page one. The user may also specify a one-inch top margin, delayed one page, so all remaining pages have a one-inch top margin. If the document is less than one page, the delayed page format code never takes effect.

## 0xD116

### Start of Delayed Codes

- ▶ <209 (0xD1)> <22 (0x16)> [size = variable] <flags = 3 (encased function)>  
 [size of non-deletable information]  
 <number of IDs and hashes>  
 The following three words are repeated for each count:  
     [ID]  
     [hash]  
     [page number]  
 <undocumented deletable data> x ?  
 [size] <0xD1>◀

## 0xD117

### End of Delayed Codes

- ▶ <209 (0xD1)> <23 (0x17)> [size = variable] <flags = 3 (encased function)>  
[size of non-deletable information = 0]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD118

### Page Number Format (text)

- ▶ <209 (0xD1)> <24 (0x18)> [size = variable] <flags = PRFXID>  
<number of PIDs = 1>  
[PID of the format string (type=0x62)]  
[size of non-deletable information = 0]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD119

### Text Direction (Asia)

- ▶ <209 (0xD1)> <25 (0x19)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
[text direction]  
0 = horizontal  
1 = vertical  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD11A

### Header Separator Distance

- ▶ <209 (0xD1)> <26 (0x1A)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[vertical size (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD11B

### Footer Separator Distance

- ▶ <209 (0xD1)> <27 (0x1B)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[vertical size (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD11C

### Line Count

- ▶ <209 (0xD1)> <28 (0x1C)> [size = variable] <flags = PRFXID>  
<number of PIDs = 1>  
[definition PID]  
[size of non-deletable information = 4]  
[line count (WUWORD)]  
[line height (WPU)]

<undocumented deletable data> x ?  
[size] <0xD1>◀

## 0xD11D

### Logical Page Flow

- ▶ <209 (0xD1)> <29 (0x1D)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[flow direction]  
0 = left to right, top to bottom  
1 = top to bottom, right to left  
2 = right to left, top to bottom  
<undocumented deletable data> x ?  
[size] <0xD1>◀

## ◆ Function 210 (0xD2) = Column Group

The Column Group subfunctions have column orientation.

## 0xD200

### Left Margin Set

- ▶ <210 (0xD2)> <0 (0x00)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[left margin (WPU)] distance from left edge of paper to text  
<undocumented deletable data> x ?  
[size] <0xD2>◀

## 0xD201

### Right Margin Set

- ▶ <210 (0xD2)> <1 (0x01)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[right margin (WPU)] distance from right edge of paper to text  
<undocumented deletable data> x ?  
[size] <0xD2>◀

## 0xD202

### Define Text Columns

- ▶ <210 (0xD2)> <2 (0x02)> [size = variable] <flags = 0>  
[size of non-deletable information]  
<column type>  
bits 0-1:  
0 = newspaper  
1 = newspaper with vertical balance  
2 = parallel  
3 = parallel with protect  
Bit 1 can be used to determine if the columns are newspaper or parallel.  
{spacing between columns (WPSP)}  
<number of columns (0-24)> 0 or 1 = columns off  
The following data exists only if number of columns > 1 and is repeated for each column.  
The width between columns information does not exist for the last column.  
<column n definition>  
bit 0: 0 = fixed point value (WFPF),

1 = fixed width (WPU)  
 [column n width (WFPF or WPU)]  
 <width between columns n and n+1 definition>  
 bit 0: 0 = fixed point value (WFPF),  
 1 = fixed width (WPU)  
 [width between columns n and n+1 (WFPF or WPU)]  
 <undocumented deletable data> x ?  
 [size] <0xD2>◀

If WFPF, column width values are the percentage of the remaining available width on the page after margin widths and spaces between columns are subtracted from the total page count.

## 0xD203

### Column Border

Same format as Page Border function 209 (0xD1) subfunction 16 (0x10).

▶ <210 (0xD2)> <3 (0x03)> [size = variable] <flags=PRFXID>  
 ...  
 [size] <0xD2>◀

## ◆ Function 211 (0xD3) = Paragraph Group

The Paragraph Group subfunctions have paragraph orientation.

## 0xD300

### Set Line Height

▶ <211 (0xD3)> <0 (0x00)> [size = variable] <flags = 0>  
 [size of non-deletable information = 2]  
 [lines per inch (WPU)]  
 0 = automatic line height  
 non-zero = fixed line height  
 <undocumented deletable data> x ?  
 [size] <0xD3>◀

## 0xD301

### Set Line Spacing

▶ <211 (0xD3)> <1 (0x01)> [size = variable] <flags = 0>  
 [size of non-deletable information = 4]  
 {line spacing (WPSP)}  
 <undocumented deletable data> x ?  
 [size] <0xD3>◀

## 0xD302

### Set Left Hotzone

▶ <211 (0xD3)> <2 (0x02)> [size = variable] <flags = 0>  
 [size of non-deletable information = 1]  
 <left hotzone>  
 <undocumented deletable data> x ?

[size] <0xD3>◀

## 0xD303

### Set Right Hotzone

- ▶ <211 (0xD3)> <3 (0x03)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<right hotzone>  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD304

### Tab Set

- ▶ <211 (0xD3)> <4 (0x04)> [size = variable] <flags = 0>  
[size of non-deletable information]  
<definition>  
bit 0: 0 = absolute, 1 = relative  
[tab adjust value] set to left margin for relative tabs, ignored for absolute tabs  
<number of tab type/tab position combinations>  
The following three bytes are repeated for each combination:  
<tab type>  
bits 0-3:  
0 = left tab  
1 = centered tab  
2 = right tab  
3 = decimal aligned tab  
4 = vertical bar tab  
bit 4: 0 = no dot leader, 1 = dot leader  
bit 7: 1 = repeated tab  
If bit 7 is set, bits 0-6 contain the repetition count, using the previous tab type and position.  
[tab position or spacing (WPU)] tap spacing for repeated tab  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD305

### Set Justification Mode

- ▶ <211 (0xD3)> <5 (0x05)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<justification mode>  
0 = left  
1 = full  
2 = center  
3 = right  
4 = full all lines (kinto waritsuke)  
5 = reserved (decimal aligned in tables)  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD306

### Set Hyphenation Mode

- ▶ <211 (0xD3)> <6 (0x06)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<hyphenation state>

bit 0: 0 = hyphenation off, 1 = hyphenation on  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD307

### Set Leading Adjustment

▶ <211 (0xD3)> <7 (0x07)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[leading adjustment/value (signed WPU)]  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD308

### Begin Generated Text

▶ <211 (0xD3)> <8 (0x08)> [size = variable] <flags = 1 (simple paired function)>  
[size of non-deletable information = 2]  
[size of generate table (WPU)] used only during generate  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD309

### End of Generated Text

▶ <211 (0xD3)> <9 (0x09)> [size = variable] <flags = 1 (simple paired function)>  
[size of non-deletable information = 2]  
[size of generate table (WPU, used only during generate)]  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD30A

### Set Spacing After Paragraph

▶ <211 (0xD3)> <10 (0x0A)> [size = variable] <flags = 0>  
[size of non-deletable information = 4]  
{spacing after paragraph (WPSP)}  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD30B

### Indent First Line of Paragraph

▶ <211 (0xD3)> <11 (0x0B)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[paragraph indent (signed WPU)]  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD30C

### Left Margin Adjustment

▶ <211 (0xD3)> <12 (0x0C)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[left margin adjustment (signed WPU)]  
<undocumented deletable data> x ?

[size] <0xD3>◀

## 0xD30D

### Right Margin Adjustment

- ▶ <211 (0xD3)> <13 (0x0D)> [size = variable] <flags = 0>  
[size of non-deletable information = 2]  
[right margin adjustment (signed WPU)]  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD30E

### Outline Define

- ▶ <211 (0xD3)> <14 (0x0E)> [size = variable] <flags = PRFXID>  
<number of PIDs = 9>  
[PID of this outline style (type=0x31)]  
[PIDs of paragraph styles associated with this outline (type=0x30)] x 8  
[size of non-deletable information = 11]  
[hash of this definition]  
<8 numbering methods> x 8  
<outline flags>  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD30F

### Paragraph Border

Same format as Page Border function 209 (0xD1) subfunction 16 (0x10).

- ▶ <211 (0xD3)> <15 (0x0F)> [size = variable] <flags=PRFXID>  
...  
[size] <0xD3>◀

## 0xD310

### Define Math Columns

- ▶ <211 (0xD3)> <16 (0x10)> [size = variable] <flags = 0 or PRFXID>  
<number of PIDs = 1>  
[math formulas PID (type=0x63)]  
[size of non-deletable information = 26]  
[calculations PID (just a copy)]  
<math definition> x 24  
Each math definition byte has the following meaning:  
bits 0-2: number of digits in range of 0-4  
bit 3: 0 = use parentheses for negative numbers  
1 = use '-' for negative numbers  
if bit 7 = 0:  
bits 4-6:  
1 = text  
2 = numeric  
3 = total  
if bit 7 = 1:  
bits 4-6:  
0 = calculation, formula 1  
1 = calculation, formula 2  
2 = calculation, formula 3

3 = calculation, formula 4  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD311

### Math On/Off

▶ <211 (0xD3)> <17 (0x11)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<math state>  
bit 0: 0 = off, 1 = on  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD312

### Line Numbering Definition

▶ <211 (0xD3)> <18 (0x12)> [size = variable] <flags = 0 or PRFXID>  
If the PID bit is set, the following information exists:  
<number of PIDs = 1>  
[line numbering font PID (type=0x55)] (0 if use flag = 0)  
[size of non-deletable information = 25]  
[hash (matched typeface descriptor)]  
<use flag>  
0 = use document initial values for font, point size, attributes and colors  
1 = use values in function  
[copy of line numbering font PID]  
[point size (3600ths)]  
<line numbering state>  
bit 0: 0 = off, 1 = on  
bit 1: 0 = line numbering restarts on each page off  
1 = line numbering restarts on each page state on  
bit 2: 0 = count blank lines off  
1 = count blank lines state on  
bit 3: 0 = numbering in all newspaper columns off  
1 = numbering in all newspaper columns on  
bit 4: 0 = numbering footnote/endnote text lines off  
1 = numbering footnote/endnote text lines on  
bit 7: 0 = absolute position from left edge of page  
1 = relative to left margin position  
[line numbering position (WPU/signed WPU)]  
[first line number to print]  
<line numbering interval>  
[matched font index in font list]  
[matched point size of font (3600ths)]  
[attribute word 1]  
bit 0: 1 = extra large  
bit 1: 1 = very large  
bit 2: 1 = large  
bit 3: 1 = small print  
bit 4: 1 = fine print  
bit 5: 1 = superscript  
bit 6: 1 = subscript  
bit 7: 1 = outline  
bit 8: 1 = italics  
bit 9: 1 = shadow  
bit 10: 1 = redline  
bit 11: 1 = double underline

bit 12: 1 = bold  
bit 13: 1 = strikethrough  
bit 14: 1 = underline  
bit 15: 1 = small caps  
[attribute word 2]  
bit 0: 1 = blink  
bit 1: 1 = reverse video  
<color (RGBA)> x 4  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD313

### Force Odd/Even/New

▶ <211 (0xD3)> <19 (0x13)> [size = variable] <flags = 0>  
[size of non-deletable information = 1]  
<force flag>  
0 = even  
1 = odd  
2 = new  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD314

### Endnotes Print Here

▶ <211 (0xD3)> <20 (0x14)> [size = variable] <flags = 3 (encased function)>  
[size of non-deletable information = 0]  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD315

### Endnotes Print Here End

▶ <211 (0xD3)> <21 (0x15)> [size = variable] <flags = 3 (encased function)>  
[size of non-deletable information = 0]  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD316

### Define Marked Text

▶ <211 (0xD3)> <22 (0x16)> [size] <flags = PRFXID>  
<number of PIDs = 1>  
[definition PID (type=0x0B)]  
[size of non-deletable information = 1]  
<type flag>  
0 = ToC  
1 = list  
2 = index  
3 = ToA  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD317

### Define Drop Cap

► <211 (0xD3)> <23 (0x17)> [size = variable] <flags = PRFXID>  
 <number of PIDs = variable (minimum 3)>  
 [typeface descriptor ID (0 if none)]  
 [border style ID (0 if none)]  
 [fill style ID (0 if none)]  
 [any override PIDs] x ?  
 <number of characters to affect>  
 [text attributes first word data]  
 [text attributes first word mask]  
 [text attributes second word data]  
 [text attributes second word mask]  
 <drop cap color (RGSB)> x 4  
 {drop cap text size (WPSP)}  
 {paragraph vertical adjustment (WPSP)}  
 [kerning (signed WPU, 1200's)]  
 [horizontal position adjustment (65536ths, 0xffff = all in margin)]  
 <flags>  
 bits 0-2: currently undefined  
 bit 3: no wrapping (takes precedence over contour wrapping)  
 bit 4: include descender height in square wrap area  
 bit 5: use shade from function  
 bit 6: use color from function  
 bit 7: contour wrapping  
 [size of override information]  
 [general override flags]  
 <border and fill override data> x ?  
 <undocumented deletable data> x ?  
 [size] <0xD3>◀

## 0xD318

### Paragraph Text Direction

► <211 (0xD3)> <24 (0x18)> [size = variable] <flags = 0>  
 [size of non-deletable information = 1]  
 <flag>  
 0 = left to right text  
 1 = right to left text  
 <undocumented deletable data> x ?  
 [size] <0xD3>◀

## 0xD319

### Asian Wrapping

► <211 (0xD3)> <25 (0x19)> [size = variable] <flags = 0>  
 [size of non-deletable information = 2]  
 [flags]  
 bit 0: Asia wrapping on/off  
 0 = off  
 1 = on  
 bit 1: Asia wrap type  
 0 = wrap  
 1 = squish  
 bit 2: Asia wrap hanging characters  
 0 = off  
 1 = on  
 bit 3: Wrap alphanumerics  
 0 = break alphanumerics at right margin

1 = wrap alphanumerics  
bit 4: AMF processing  
0 = off  
1 = on  
<undocumented deletable data> x ?  
[size] <0xD3>◀

## 0xD31A

### Paragraph Character Count

▶ <211 (0xD3)> <26 (0x1A)> [size = variable] <flags = PRFXID>  
<number of PIDs = 1>  
[definition PID]  
[size of non-deletable information = 4]  
[character count]  
[character width (WPU)]  
<undocumented deletable data> x ?  
[size] <0xD3>◀