

◆ Packet Data Formats

Reserved packets may appear in a document prefix. Most of the strings in the packet data are strings made up of short values. The high byte of the short value is the WP character set number. The low byte of the short value is the character number. The strings are null terminated with a short value unless otherwise specified.

Packet Type 0 (0x00)

Index Entry Is Available or Was Deleted

This does not apply to the index header, which is always the first index in the index block.

Packet Type 1 (0x01)

Document Height

The document height is the height of the entire document from top to bottom.

- ▶ {document height (text mode)} in formatter text lines
{document height (graphics mode)} in WPUs
{document height (page mode)} in 3600ths◀

Packet Type 2 (0x02)

Document-Specific Flags

- ▶ <flag bits for doc>
 - bit 0: 1 = baseline placement for typesetters is active
 - bits 1-2: 00 = font change
 - 01 = character in right margin
 - 10 = character in left margin
 - 11 = character in alternating margins
 - bit 3: 1 = document needs to be formatted
 - bit 4: 1 = document needs to be (re)generated
 - bit 5: 1 = manual display pitch
 - bit 6: 1 = links should be updated on retrieval
 - bit 7: 1 = links codes should not be displayed
- <text/graphics quality>
 - low nibble = text quality
 - high nibble = graphics quality
 - both nibbles:
 - 0 = do not print
 - 1 = draft
 - 2 = medium
 - 3 = high
- [redline character]
- [width of screen character]
- <ink color>
 - 0 = black
 - 1 = full color
 - 2 = solid color
- <merge display flags>
 - 2 (0x02) = display merge codes
 - 4 (0x04) = show displayed codes as icons

- <reserved>
- <default form number>
- [current undo-level number]
- [current range level number]
- <more document flags>
 - bit 0: 1 = hypertext mode is active
 - bit 1: 1 = hiding text marked for hiding
- <reserved for future use>◀

Packet Type 3 (0x03)

Reserved

Packet Type 4 (0x04)

Form Hash Table

► [hash value for each form contained in a document] x ?◀

Packet Type 5 (0x05)

Writing Tools API

► [company ID]
 [tool (or product) ID]
 <tool information> x ?◀

The Spell Checker uses the Tool information field for the document Supplemental Dictionary. The format of the Supplemental Dictionary is a WP short string.

Packet Type 6 (0x06)

Character Map

This structure is the same as used in a MAP file. The file type for a MAP file is 49 (0x31). In a MAP file, this structure begins directly following the 16-byte prefix.

► [size of character mapping packet]
 <map flags>

- bit 7 is set when using 2-byte mappings (WP short string mappings)
- 80 = Japanese mapping
- 81 = Korean mapping
- 82 = Chinese mapping
- 83 = Taiwanese mapping

 [pointer to map name] offset from start of packet
 [pointer to map table] table for converting document characters to WP characters
 [map name] x ? null-terminated WP word string of indeterminate length
 [min value = 1] minimum value of mappable characters
 [max value = 127] maximum value of mappable characters
 [number of mappable chars] maximum – minimum + 1
 [size of map table]
 [document character, WP character]
 Document characters are mapped into the range 1-127. The 1-byte document character to WP character mapping table is essentially a lookup table. It consists of 127 entries, each one word in length. The first byte of the first word represents document character 1 and the second byte represents the WP character that document character 1 is mapped to.◀

Packet Type 7 (0x07)

Native Filename

If this packet does not have any children, the format is:

- ▶ <filename> x ? *or* [filename] x ? null-terminated byte or word string◀

If this packet does have children, the format is the following:

- ▶ [count of child IDs]
[ID 1] ID of child packet
[ID 2] ID of child packet
...
[ID n] ID of last child
[tag 1] tag flag for first child packet
[tag 2] tag flag for second child packet
...
[tag n] last tag flag for last child packet
[filename] x ? null terminated◀

Format for Tag Flags

0x0001 = data being kept internal

0x0010 = data being kept internal (hot link)

0x0011 = data being kept internal (warm link)

0x0012 = data being kept internal (cold link)

Packet Type 8 (0x08)

General WP Text

- ▶ [number of text blocks]
{relative offset of first text block within packet}
{size of 1st text block}
{size of 2nd text block}
...
{size of last text block}
...
<text data for 1st block> x 1st block size
<text data for 2nd block> x 2nd block size
...
<text data for last block> x last block size◀

Packet Type 9 (0x09)

Bookmark Data

- ▶ [reserved]
<bookmark flags>
bits 0-3: reserved
bit 4: 1 = paired bookmark
bit 5: 1 = bookmark is in a substructure
bit 6: 1 = quick bookmark
bit 7: 1 = deleted bookmark
<reserved>
[bookmark name] x ? null-terminated WP short string; max 78 bytes or 38 chars + null◀

Packet Type 10 (0x0A)

Macro Executable

No documentation will be provided for this packet.

Packet Type 11 (0x0B)

Mark Text for Generate Definition

The byte following the PIDs in each of the following structures indicates which type of definition exists in a particular instance of this packet.

Table of Contents Definition

- ▶ [number of child prefix IDs = 2-6]
 - [style PID 1 (type=0x30)]
 - [style PID 2 (type=0x30)]
 - ...
 - [format PID (type=0x62)]
- <ToC definition = 0>
- <number mode>
 - 0 = no number
 - 1 = number follows entry
 - 2 = number in parentheses follows entry
 - 3 = number flush right
 - 4 = number flush right with dot leaders
- [ToC name = 0]◀

Index Definition

- ▶ [number of child prefix IDs = 4]
 - [child PID (level 1) (type=0x30)]
 - [child PID (level 2) (type=0x30)]
 - [concordance filename PID (type=0x07, 0 if none)]
 - [format PID (type=0x62, 0 for document format)]
- <index definition = 2>
- <number mode>
 - 0 = no number
 - 1 = number follows entry
 - 2 = number in parentheses follows entry
 - 3 = number flush right
 - 4 = number flush right with dot leaders
- <definition flags>
 - 1 = reserved
 - 2 = combining of sequential page numbers (default)
- [index name = 0]◀

List Definition

- ▶ [number of child prefix IDs = 3]
 - [style PID (type=0x30)]
 - [format string PID (type=0x62)]
 - [graphics style PID (type=0x11)]
- <list definition = 1>
- <number mode>
 - 0 = no number
 - 1 = number follows entry
 - 2 = number in parentheses follows entry
 - 3 = number flush right

4 = number flush right with dot leaders
<reserved>
[list name] x ? short string format

Table of Authorities Definition

[number of child prefix IDs = 2]
[style PID (type=0x30)]
[format string PID (type=0x62)]
<ToA definition = 4>
<number mode>
0 = no number
1 = number follows entry
2 = number in parentheses follows entry
3 = number flush right
4 = number flush right with dot leaders
<definition flags>
1 = ToA definition, underlining allowed
2 = combining of sequential page numbers (default)
[ToA name] x ?short string format◀

Packet Type 12 (0x0C)

Subdocument

If this packet does not have any children, the format is a short string of the filename for the subdocument. The subdocument is stored externally on disk.

▶ [subdocument filename] x ? null terminated◀

If this packet has children, the format is the following:

▶ [number of child IDs]
[ID 1] ID of child packet containing subdocument information
[ID 2] ID of child packet containing subdocument information
...
[ID count] last child ID for subdocument information
[tag 1] tag flag for first child packet
[tag 2] tag flag for second child packet
...
[tag count] tag flag for last child packet
[subdocument filename] x ?◀

Format for Tag Flags

0x0001 = data is part of this file
0x0010 = data is part of this file (hot link)
0x0011 = data is part of this file (warm link)
0x0012 = data is part of this file (cold link)

Packet Type 13 (0x0D)

Table of Authorities Mark

The full form Table of Authorities mark consists of this parent packet and a General WP Text child packet.

- ▶ [number of child prefix IDs = 1]
[child PID (type=0x08)]
[short form name] x ? WP word string
[ToA section name] x ? WP word string◀

The short form Table of Authorities mark consists of the short form name contained within this packet.

- ▶ [short form name] x ? WP word string◀

Packet Type 14 (0x0E)

Index Mark

- ▶ [index name] x ? null terminated
[heading] x ? null terminated
[subheading] x ? null terminated◀

Packet Type 15 (0x0F)

Cross-Reference Target

- ▶ [target name] x ? null terminated◀

Packet Type 16 (0x10)

NeXT Total Page Count

No documentation will be provided for this packet.

Packet Type 17 (0x11)

Counters Data

- ▶ [name or ID] x 16 There will always be 32 bytes set aside for the name, but if the counter is a system counter, then this field is an ID number instead of a name. Both the name and the ID number are null terminated with a word null and the remaining bytes are random.
 <number of counter levels valid (1-5)> actual number of levels used
 <default numbering method for level 1>
 <default numbering method for level 2>
 <default numbering method for level 3>
 <default numbering method for level 4>
 <default numbering method for level 5>
 [level name 1] x 16 null-terminated word strings, begin with two spaces ([2000] [2000])
 [level name 2] x 16 null-terminated word strings, begin with two spaces ([2000] [2000])
 [level name 3] x 16 null-terminated word strings, begin with two spaces ([2000] [2000])
 [level name 4] x 16 null-terminated word strings, begin with two spaces ([2000] [2000])
 [level name 5] x 16 null-terminated word strings, begin with two spaces ([2000] [2000])◀

Packet Type 18 (0x12)

Extended Document Summary

The extended summary data group occurs for up to 100 times, one for each field defined in the extended summary:

- ▶ [size] byte length of data group
[tag] field ID of the extended summary field
[type] field data type
bit 0: 1 = single line, data length is limited to 255 words

bit 1: 1 = multi-line, data length is limited to 1023 words
 bit 2: 1 = date, see 10-byte date structure below.
 bit 3: 1 = read only (document may be locked by another user)
 [name] x ? null-terminated word string, optional
 [data] x ? null-terminated word string or 10-byte date field◀

In the following table * denotes fields which are also defined in the WP 5.0, 5.1 or 5.2 Summary packet.

Valid predefined extended summary fields:		
Tag	Name	Type
1	Abstract*	Multi-line
2	Account*	Single line
3	Address	Multi-line
4	Attachments	Single line
5	Author*	Single line
6	Authorization	Single line
7	Bill To	Multi-line
8	Blind Copy	Single line
9	Carbon Copy	Single line
10	Category	Single line
11	Checked By	Single line
12	Client	Single line
13	Comments	Multi-line
14	Creation Date*	Date
15	Date Completed	Date
16	Department	Single line
17	Descriptive Name*	Single line
18	Descriptive Type*	Single line
19	Destination	Single line
20	Disposition	Single line
21	Division	Single line
22	Document number	Single line
23	Editor	Single line
24	Forward To	Single line
25	Group	Single line

Valid predefined extended summary fields:		
26	Keywords*	Single line
27	Language	Single line
28	Mail Stop	Single line
29	Matter	Single line
30	Office	Single line
31	Owner	Single line
32	Project	Single line
33	Publisher	Single line
34	Purpose	Multi-line
35	Received From	Single line
36	Recorded By	Single line
37	Recorded Date	Date
38	Reference	Single line
39	Revision Date*	Date and read only
40	Revision Notes	Multi-line
41	Revision number	Single line
42	Section	Single line
43	Security	Single line
44	Source	Single line
45	Status	Single line
46	Subject*	Single line
47	Telephone number	Single line
48	Typist*	Single line
49	Version Date	Date
50	Version Notes	Multi-line
51	Version number	Single line

10-Byte Date Structure

[year]
 <month>
 <day>
 <hour>
 <minute>
 <second>

<day of week> not implemented
<time zone> not implemented
<unused>

User-defined extended summary fields have a tag number in the range of 101 to 65535 and must have a field name.

Predefined fields have tag numbers in the range of 1 to 100 and may have a name, but for multilingual purposes, the name displayed in all interfaces comes from a translatable resource file.

Tag number 39, Revision Date, is a special case where the date is extracted from the DOS file date.

Packet Type 19 (0x13)

Macro Labels

No documentation will be provided for this packet.

Packet Type 20 (0x14)

Macro Global Labels

No documentation will be provided for this packet.

Packet Type 21 (0x15)

Macro Translation Table

No documentation will be provided for this packet.

Packet Type 22 (0x16)

Macro Debug Information

No documentation will be provided for this packet.

Packet Type 23 (0x17)

Hypertext (Bookmark Name)

► [bookmark name] x ? null terminated◄

Packet Type 24 (0x18)

Box Text Filename

► [box text filename] x ? null terminated◄

If this packet ever has child IDs, the format will be that of the Native Filename packet type 0x07.

Packet Type 25 (0x19)

Equation Text Filename

► [equation text filename] x ? null terminated◄

If this packet ever has child IDs, the format will be that of the Native Filename

packet type 0x07.

☺ ♦♣ Packet Type 26 (0x1A) ☹♦

Envelope Address Document

The structure of this packet is the same as packet type 8 (0x08), General WP Text.

Packet Type 27 (0x1B)

Comment/Annotation

► [number of PIDs = 1]
[text PID (type=0x08)]
<annotation flag> 0 = comment, 1 = annotation
date/time:
[year]
<month>
<day>
<hour>
<minute>
<second>
<day of week> not implemented
<time zone> not implemented
<unused>
<annotation color (RGSB)> x 4
[user initials] x ? null-terminated word string
[user name] x ? null-terminated word string◀

Packet Type 28 (0x1C)

WP Hidden Outline Text

Used in WordPerfect for DOS but not in WordPerfect for Windows. The packet structure is the same as packet type 8 (0x08), General WP Text.

Packet Type 29 (0x1D)

Prefix Needs Spell-Check

The entire document is dirty regardless of clean/dirty spell gates. An index exists only if a spell-check is needed. The data size is zero. The data pointer is unused and invalid if non-zero.

Packet Type 30 (0x1E)

Formatter Region

Region being tracked in the formatter for any changes. This packet contains a region packet header followed by information for each region packet.

Header information in the Formatter Region packet:

► [offset to first region packet information]
{number of region entries in this prefix packet}
{the next region number to allocate}
<name of file this packet created with> x ?◀

Region information for each region packet:

► {unique number representing this tracked region}
{byte count offset in edit buffer to region on code}

distance from beginning of file if positive
 distance from end of file if negative
 {byte count offset in edit buffer to region off code}
 distance from beginning of file if positive
 distance from end of file if negative
 [page number where start of tracked region starts]
 [y position where start of tracked region starts]
 [page number where end of tracked region ends]
 [y position where end of tracked region ends]
 [font face at start of region]
 [font size at start of region]
 [attributes at start of region]
 <color at start of region (RGB)> x 3
 <region active flag> indicates if changes to region should be sent to the interface
 0 = region is not active
 1 = region is active
 <type of region being tracked>
 1 = OLE
 2 = OBEX◀

Packet Type 31 (0x1F)

Mideast Flags Prefix

Mideast-specific document flags.

► <ligature control>
 0 = none
 1 = partial ligatures
 2 = full ligatures
 <number shape>
 0 = context
 1 = force English font
 2 = force Arabic/Hebrew font◀

Packet Type 32 (0x20)

Font Typeface Descriptor Pool

This packet contains all font descriptors used in the document.

In the SET file the font typeface descriptor pool only contains the default descriptor.

► Matched Font Descriptor (closest available to desired font)
 [average character width (PSU)]
 [ascender height (PSU)]
 [x height (PSU)]
 [descender height (PSU)]
 [italic adjust]
 [primary family ID] see *Primary Font Family ID Values* below
 <scripting system> 1 = European classification
 <primary character set> character set numbers 0-15
 <width (aspect ratio)>
 0 = don't know
 16-31 = ultra condensed
 32-47 = extra compressed
 48-63 = compressed

64-79 = extra condensed
80-95 = condensed (compact)
96-111 = semi condensed (narrow)
112-127 = normal
128-143 = semi expanded
144-159 = expanded (extended)
160-175 = extra expanded
176-191 = ultra expanded
192-223 = double wide
224-254 = triple wide
255 = reserved

<weight>

0 = don't know
1-7 = ultra thin
8-15 = ultra light
16-23 = extra thin
24-31 = thin
32-47 = extra light
48-63 = light
64-79 = demi light
80-87 = semi light
88-95 = book
96-103 = regular (plain)
104-111 = Roman
112-127 = medium
128-135 = demi bold
136-143 = semi bold
144-159 = bold
160-175 = extra bold
176-191 = heavy
192-207 = extra heavy
208-223 = black
224-239 = extra black
240-247 = ultra heavy
248-254 = ultra black
254 = max weight
255 = reserved

<attributes>

bit 0: 1 = italic
bit 1: 1 = outline
bit 2: 1 = shadow
bit 3: 1 = small caps

<general characteristics>

bits 0-1: usage category
0 = don't know
1 = text
2 = display
3 = both
bits 2-3: general serif
0 = don't know
1 = sans serif
2 = serif
3 = script
bits 4-5: spacing
0 = don't know
1 = proportional
2 = monospace

<classification>

bits 0-3: serif type

- 0 = don't know
- 1 = bracket
- 2 = cupped
- 3 = round bracket
- 4 = line
- 5 = slab
- 6 = wedge
- 7 = half
- 8 = sans square
- 9 = sans round
- 10 = sans flare
- 11 = nonconnect
- 12 = joining
- 13 = black serif
- 14 = swash

<fill byte = 0>

In .PRS file:

- 0 = built in
- 1 = cartridge slot
- 2 = memory (quantity units are in kilobytes)
- 3 = memory (quantity units are in font slots)
- 4 = print wheel

In .DRS file:

- 128 (0x80) = WP greeked font
- 129 (0x81) = WP menu
- 130 (0x82) = WP vector
- 131 (0x83) = WP outline
- 132 (0x84) = reserved
- 133 (0x85) = reserved
- 134 (0x86) = Speedo
- 135 (0x87) = bullet (not yet internal)
- 138 (0x8A) = ATM
- 139 (0x8B) = TrueType
- 140 (0x8C) = reserved
- 141 (0x8D) = TrueType
- 142 (0x8E) = reserved
- 143 (0x8F) = reserved
- 255 (0xFF) = no soft font driver loaded

- 16 (0x10) = .PRS file type
- 20 (0x14) = .DRS file type

[typeface name length in bytes]

[typeface name] x 58? (word string; max size = 116 bytes)

The typeface name is made up for four separate null-terminated word strings:

- 1st string = typeface family (such as Times or Swiss)
- 2nd string = attributes (such as Bold, Italic, or Bold Italic)
- 3rd string = name prefix (group abbreviation, used for sorting in fonts and cartridge menus [such as CG])
- 4th string = name extension. This is a symset string, an orientation string, or a string to simply distinguish this font from other fonts (such as scalable).◀

Primary Font Family ID Values

The Primary Font Family ID consists of the family ID (high byte) and the subfamily ID (low byte).

0x0100 Family: ITCBenguiat	
►0x0101 = Bellini	0x0117 = Stratford
0x0102 = Erasmus	0x0118 = ITCtiffany
0x0103 = HollandseMediaeval	0x0119 = ITCBenguiat
0x0104 = Pastonchi	0x011A = Seagull
0x0105 = Windsor	0x011B = ITCSouvenir
0x0106 = Kennerley	0x011C = Vendome
0x0107 = Raleigh	0x011D = ItalianOldStyle
0x0108 = SurreyOldStyle	0x011E = ITCItalia
0x0109 = Trajanus	0x011F = JensonOldStyle
0x010A = Verona	0x0120 = Guardi
0x010B = WorchesterRound	0x0121 = Cantoria
0x010C = DellaRobbia	0x0122 = ITCBerkleyOldStyle
0x010D = BauerText	0x0123 = ITCGiovanni
0x010E = Cloister	0x0124 = Lavenham
0x010F = Jenson	0x0125 = ITCVeljovic
0x0110 = SchneidlerOldStyle	0x0126 = HampshireOldStyle
0x0111 = Centaur	0x0127 = ITCEsprit
0x0112 = HorleyOldStyle	0x0128 = Hadriano
0x0113 = Deepdene	0x0129 = ITCWeidemann
0x0114 = Brighton	0x012A = Minion
0x0115 = ITCClearface	0x012B = Seneca
0x0116 = Lutetia	0x012C = Carmina◄
0x0900 Family: TimesRoman	
►0x0901 = Bembo	0x090E = ITCStoneSerif
0x0902 = BernhardModern	0x090F = Life
0x0903 = GoudyOldStyle	0x0910 = Minister
0x0904 = Plantin	0x0911 = TimesRoman
0x0905 = Poliphilus	0x0912 = Calisto
0x0906 = Trajan	0x0913 = Adroit
0x0907 = Leamington	0x0914 = ITCSlimbach
0x0908 = Missal	0x0915 = Administer
0x0909 = TrumpMediaeval	0x0916 = Caxton
0x090A = Berling	0x0917 = Eldorado
0x090B = Hiroshige	0x0918 = Symposia
0x090C = Albertina	0x0919 = Artcraft
0x090D = Emerson	0x091A = ZAntiqua◄

0x1100 Family: ITCGaramond	
►0x1101 = BinnyOldStyle	0x1119 = QuadrigaAntiqua
0x1102 = Bookman	0x111A = Sabon
0x1103 = Bulmer	0x111B = Barbou
0x1104 = Caslon540	0x111C = Fournier
0x1105 = Amerigo	0x111D = Garamond
0x1106 = ChiswellOldFace	0x111E = Spectrum
0x1107 = Fontana	0x111F = Weiss
0x1108 = Granjon	0x1120 = Concorde
0x1109 = Imprint	0x1121 = ITCCushing
0x110A = Monticello	0x1122 = Dante
0x110B = OldStyleNo2	0x1123 = Gazette
0x110C = GarthGraphic	0x1124 = ITCIsbell
0x110D = Scenario	0x1125 = Romulus
0x110E = Ehrhardt	0x1126 = Rotation
0x110F = Galliard	0x1127 = VanDijck
0x1110 = Aldus	0x1128 = Janson
0x1111 = Aster	0x1129 = Roundfunk
0x1112 = Baskerville	0x112A = Charter
0x1113 = RotisSerif	0x112B = Criterion
0x1114 = Congress	0x112C = Ellington
0x1115 = Lectura	0x112D = HollandSeminar
0x1116 = Palatino	0x112E = ITCGamma
0x1117 = PopplPontifex	0x112F = ITCLeawood
0x1118 = TimesEuropa	0x1130 = Pasquale◄
0x1900 Family: CenturySchoolbook	
►0x1901 = Joanna	0x1911 = Impressum
0x1902 = Maximus	0x1912 = Melior
0x1903 = Breughel	0x1913 = Dominante
0x1904 = Excelsior	0x1914 = Menhart
0x1905 = Perpetua	0x1915 = Paragon
0x1906 = Primer	0x1916 = Cochin
0x1907 = Nofret	0x1917 = Electra
0x1908 = CenturySchoolBook	0x1918 = Iridium
0x1909 = Nimrod	0x1919 = ITCZapfInternational
0x190A = Cheltenham	0x191A = Accolade
0x190B = Comenius	0x191B = Cardinal
0x190C = Sorbonne	0x191C = Claridge

0x190D = Bramley	0x191D = Else
0x190E = GoudyModern	0x191E = ITCJamille
0x190F = Athenaeum	0x191F = MonoTypeEngravers
0x1910 = Diotima	0x1920 = ITCStoneInformal◀
0x2100 Family: Bodoni	
►0x2101 = Auriga	0x210A = Walbaum
0x2102 = ITCZapfBook	0x210B = ITCFenice
0x2103 = BauerBodoni	0x210C = Modern
0x2104 = Corvinus	0x210D = ScotchRoman
0x2105 = Albion42	0x210E = Centennial
0x2106 = Caledonia	0x210F = Versailles
0x2107 = Fairfield	0x2110 = Wilke
0x2108 = DeVinne	0x2111 = Inflex
0x2109 = Madison	0x2112 = ITCDidi◀
0x2900 Family: ITCLubalinGraph	
►0x2901 = ASGallatin	0x290C = AntiqueNo5
0x2902 = Beton	0x290D = ITCPacella
0x2903 = Calvert	0x290E = ShadowAntiqua
0x2904 = Candida	0x290F = Aachen
0x2905 = City	0x2910 = PMNCaecilia
0x2906 = Glypha	0x2911 = available17
0x2907 = ITCLubalinGraph	0x2912 = Clarendon
0x2908 = Memphis	0x2913 = EgyptienneF
0x2909 = Rockwell	0x2914 = Egizio
0x290A = Serifa	0x2915 = Fortune
0x290B = Stymie	0x2916 = ITCAmericanTypewriter◀
0x3100 Family: Albertus	
►0x3101 = Albertus	0x310E = Biltmore
0x3102 = Americana	0x310F = ITCLSCBook
0x3103 = Flange	0x3110 = Meridien
0x3104 = FrenchOldStyle	0x3111 = ITCTiepolo
0x3105 = ITCFrizQuadrata	0x3112 = RotisSemiserif
0x3106 = Icone	0x3113 = ITCSerifGothic
0x3107 = ITCKorinna	0x3114 = available20
0x3108 = ITCNewText	0x3115 = available21
0x3109 = ITCNovarese	0x3116 = available22
0x310A = PopplLaudatio	0x3117 = Footlight

0x310B = ITCQuorum	0x3118 = ITCElan
0x310C = Romana	0x3119 = ITCUsherwood
0x310D = ITCBarcelona	0x311A = Copperplate◀
0x3900 Family: Courier	
►0x3901 = Courier	0x3902 = PrestigeElite◀
0x4100 Family: Helvetica	
►0x4101 = Abadi	0x4116 = Helvetica
0x4102 = AkzidenzGrotesk	0x4117 = HollandTitle
0x4103 = AlpinGothic	0x4118 = Imago
0x4104 = AntiqueOlive	0x4119 = Impact
0x4105 = Avenir	0x411A = Insignia
0x4106 = CGSymphony	0x411B = ITCAvantGardeGothic
0x4107 = DINSchriften	0x411C = ITCKabel
0x4108 = DirectoryGothic	0x411D = ITCMixage
0x4109 = Flyer	0x411E = Ambassador
0x410A = Folio	0x411F = ITCStoneSans
0x410B = Formata	0x4120 = TradeGothic
0x410C = FranklinGothic	0x4121 = Metro
0x410D = Frontiera	0x4122 = NeuzeitS
0x410E = Frutiger	0x4123 = NewsGothic
0x410F = Futura	0x4124 = Placard
0x4110 = GillSans	0x4125 = VAGRounded
0x4111 = Gothic	0x4126 = RotisSansSerif
0x4112 = Grotesque	0x4127 = Sans
0x4113 = CenturyGothic	0x4128 = SassoonPrimary
0x4114 = Harry	0x4129 = Syntax
0x4115 = Headline	0x412A = Uncia◀
0x4900 Family: ITCRonda	
►0x4901 = ITCBauhaus	0x4904 = ITCRonda
0x4902 = BlippoBlack	0x4905 = Wexford
0x4903 = Churchward70	0x4906 = ITCBusorama◀
0x5100 Family: Microstyle	
►0x5101 = Doric	0x5106 = MagnumGothic
0x5102 = Heldustry	0x5107 = Microstyle
0x5103 = ITCBolt	0x5108 = Neographik
0x5104 = ITCeras	0x5109 = RussellSquare
0x5105 = available5	0x510A = Serpentine◀

0x5900 Family: Optima	
►0x5901 = BakerSignet	0x5908 = Optima
0x5902 = BlueJack	0x5909 = Lisbon
0x5903 = Chelmsford	0x590A = RotisSemisans
0x5904 = Peignot	0x590B = Shannon
0x5905 = ITCPanache	0x590C = Triplett
0x5906 = ITCSymbol	0x590D = Video
0x5907 = Omega	◄
0x6100 Family: LetterGothic	
►0x6101 = available1	0x6105 = OCRB
0x6102 = LetterGothic	0x6106 = Orator
0x6103 = LinePrinter	0x6107 = Presentation
0x6104 = OCRA	◄
0x6900 Family: Brush	
►0x6901 = AshleyScript	0x6908 = FreeStyleScript
0x6902 = BiffoScript	0x6909 = Kaufmann
0x6903 = BrophyScript	0x690A = MonoLineScript
0x6904 = BrushScript	0x690B = BickleyScript
0x6905 = CascadeScript	0x690C = PresentScript
0x6906 = Chaplin	0x690D = Swing
0x6907 = Charme	0x690E = Bronx◄
0x7100 Family: ParkAvenue	
►0x7101 = AristonLightScript	0x710D = ITCIsadora
0x7102 = Basilica	0x710E = ITCZapfChancery
0x7103 = Bellevue	0x710F = Jasper
0x7104 = BertholdScript	0x7110 = Liberty
0x7105 = Boulevard	0x7111 = LisbonCursive
0x7106 = Amazone	0x7112 = MediciScript
0x7107 = CarmineTango	0x7113 = MurrayHill
0x7108 = CommercialScript	0x7114 = ParkAvenue
0x7109 = Coronet	0x7115 = Stuyvesant
0x710A = Corsiva	0x7116 = UniversityRoman
0x710B = Diskus	0x7117 = Musketeer
0x710C = FloridianScript	0x7118 = TypoUpright◄
0x7900 Family: SignetRoundhand	
►0x7901 = AbelCursive	0x790D = NuptialScript

0x7902 = Aristocrat	0x790E = OldFashionScript
0x7903 = CitadelScript	0x790F = OriginalScript
0x7904 = DorchesterScript	0x7910 = PalaceScript
0x7905 = FlemishScript	0x7911 = PopplResidenz
0x7906 = FlorentineScript	0x7912 = Balmoral
0x7907 = FrenchScript	0x7913 = ShelleyAllegroScript
0x7908 = Isabella	0x7914 = ShelleyAndanteScript
0x7909 = KunstlerScript	0x7915 = SignetRoundhand
0x790A = LinoScript	0x7916 = VenitianScript
0x790B = LuciaScript	0x7917 = Vivaldi
0x790C = MahoganyScript	◀
0x8100 Family: Tekton	
►0x8101 = Banco	0x810B = Matura
0x8102 = Bison	0x810C = Mercurius
0x8103 = ITCGrizzly	0x810D = NewBerolina
0x8104 = DomCasual	0x810E = Mistral
0x8105 = Hebrastyle	0x810F = Pepita
0x8106 = Herculenum	0x8110 = Reporter
0x8107 = ITCFlora	0x8111 = Tekton
0x8108 = Forte	0x8112 = Jiffy
0x8109 = Klang	0x8113 = Visigoth
0x810A = Lithos	◀
0x8900 Family: Uncial	
►0x8901 = AbbottOldStyle	0x890A = available10
0x8902 = Amigo	0x890B = available11
0x8903 = Auriol	0x890C = Libra
0x8904 = Camelot	0x890D = available13
0x8905 = Carolina	0x890E = Omnia
0x8906 = Clairvaux	0x890F = Ondine
0x8907 = AuraScript	0x8910 = available16
0x8908 = Euclid	0x8911 = PostAntiqua
0x8909 = Feinen	0x8912 = Uncial◀
0x9100 Family: OldEnglish	
►0x9101 = Basque	0x910A = BlackMoor
0x9102 = CloisterBlack	0x910B = OldEnglish
0x9103 = DucDeBerry	0x910C = Agincourt
0x9104 = EngraversText	0x910D = SanMarco

0x9105 = FetteFraktur	0x910E = WeddingText
0x9106 = GoudyText	0x910F = WilhelmKlingsporGotisch
0x9107 = AmericanText	0x9110 = YorkShire
0x9108 = LinoText	0x9111 = AlteSchwabacher
0x9109 = LondonText	◀
0x9900 Family: ITCZapfDingbats	
►0x9901 = BorderPi	0x990B = Sonata
0x9902 = BundesbahnPi	0x990C = StarTrekPi
0x9903 = Carta	0x990D = Symbol
0x9904 = Cheq	0x990E = WoodTypeOrnaments
0x9905 = EuropeanPi	0x990F = LineDraw
0x9906 = HelveticaFractions	0x9910 = BarCodes
0x9907 = ITCZapfDingbats	0x9911 = KeyCaps
0x9908 = LinoTypeDecorationPi	0x9912 = MonoTypeSorts
0x9909 = LinoTypeGamePi	0x9913 = MTExtra
0x990A = LinoTypeHolidayPi	0x9914 = Fences◀
0xA100 Family: Broadway	
►0xA101 = Braggadocio	0xA10A = ITCLSCManhattan
0xA102 = Brittanic	0xA10B = ITCTomsRoman
0xA103 = Broadway	0xA10C = McCollough
0xA104 = Falstaff	0xA10D = Binner
0xA105 = Annlie	0xA10E = Normande
0xA106 = GlobeGothic	0xA10F = Parisian
0xA107 = ITCBernase	0xA110 = Stencil
0xA108 = ITCFatFace	0xA111 = Trooper
0xA109 = ITCFirenze	0xA112 = Bremen◀
0xA900 Family: Computer	
►0xA901 = Computer	0xA907 = PrinterNo1
0xA902 = Countdown	0xA908 = PrinterNo2
0xA903 = Data70	0xA909 = Zeitgeist
0xA904 = Digital	0xA90A = LCD
0xA905 = ITCMachine	0xA90B = OrbitB
0xA906 = Amelia	◀
0xB100 Family: PTBarnum	
►0xB101 = Playbill	0xB105 = Figaro
0xB102 = BrandingIron	0xB106 = Hidalgo
0xB103 = available3	0xB107 = PTBarnum

0xB104 = available4	◀
0xB900 Family: Ponderosa	
►0xB901 = AldousVertical	0xB90A = Lafayette
0xB902 = Arcadia	0xB90B = LatinExtraCondensed
0xB903 = Birch	0xB90C = Mesquite
0xB904 = Compacta	0xB90D = Onyx
0xB905 = Eccentric	0xB90E = Ponderosa
0xB906 = Industria	0xB90F = Runic
0xB907 = Ironwood	0xB910 = TipTop
0xB908 = ITCLSCCondensed	0xB911 = Willow
0xB909 = Juniper	◀
0xC100 Family: Madrone	
►0xC101 = BlackOak	0xC103 = Poplar
0xC102 = Madrone	0xC104 = Thunderbird◀
0xC900 Family: Hobo	
►0xC901 = AntikvaMargaret	0xC911 = KochAntiqua
0xC902 = ArnoldBocklin	0xC912 = Milton
0xC903 = Belwe	0xC913 = Neptune
0xC904 = BlackfriarsRoman	0xC914 = Packard
0xC905 = BrynMawr	0xC915 = Pretorian
0xC906 = Charleston	0xC916 = Raphael
0xC907 = Columbus	0xC917 = Richmond
0xC908 = CooperBlack	0xC918 = Skyjald
0xC909 = Cottonwood	0xC919 = Tango
0xC90A = Croydon	0xC91A = Thalia
0xC90B = Eckmann	0xC91B = Belshaw
0xC90C = Hobo	0xC91C = LoType
0xC90D = Hogarth	0xC91D = Parsons
0xC90E = ITCGorilla	0xC91E = Romic
0xC90F = ITCGrouch	0xC91F = BitstreamArrus
0xC910 = ITCHonda	◀
0xD100 Family: Revue	
►0xD101 = AdLib	0xD108 = Pump
0xD102 = Becket	0xD109 = Revue
0xD103 = Kino	0xD10A = Sintex1
0xD104 = MotterTektura	0xD10B = Stop
0xD105 = Neuland	0xD10C = Tabasco

0xD106 = Octopuss	0xD10D = Balloon
0xD107 = Othello	◀
0xD900 Family: AestheticOrnamented	
►0xD901 = Abramesque	0xD919 = RaffiaInitials
0xD902 = AestheticOrnamented	0xD91A = RayShaded
0xD903 = BedRock	0xD91B = RomantiquesNo5
0xD904 = BlockUp	0xD91C = Sapphire
0xD905 = Chevalier	0xD91D = Sinaloa
0xD906 = Fatima	0xD91E = SnowCap
0xD907 = Flintstones	0xD91F = TalbotsRockyMountian
0xD908 = Flirt	0xD920 = Tonal
0xD909 = Florist	0xD921 = Trocadero
0xD90A = Fontanesi	0xD922 = UncleSam
0xD90B = FrysOrnamented	0xD923 = ViaFaceDonBlack
0xD90C = Gallia	0xD924 = Zip
0xD90D = GoodVibrations	0xD925 = Shadows
0xD90E = ITCNeon	0xD926 = Cameos
0xD90F = Karnak	0xD927 = StripedPatterns
0xD910 = Kismet	0xD928 = HatchPatterns
0xD911 = LettresOrnees	0xD929 = Engraved
0xD912 = Lexington	0xD92A = Outlines
0xD913 = Lilith	0xD92B = BabyTeeth
0xD914 = Nymphic	0xD92C = Bottleneck
0xD915 = OldGlory	0xD92D = Davida
0xD916 = Picadilly	0xD92E = OldDreadfulNo7
0xD917 = Pinball	0xD92F = Shotgun
0xD918 = Quentin	◀

Packet Type 33 (0x21)

Font PS Table

One document file packet is used for each PS (proportional space) table.

In a SET file, this packet contains the font PS table for the default initial font.

-
- <PS status flag>
 - 0 = byte characters
 - 1 = word characters
 - <PS type flag>
 - 0 = PS table
 - 1 = monospaced bitmap

[pointers to the 16 character sets] x 16
[character widths/bitmap] x ? variable length◀

If the type is PS table, the following block of data will exist for each character set from 1 through 16.

► set <count of characters in this set> (starting at 0) same as the number of the last character in the set
[width of nth character] x count+1 (n = 0 to count)◀

If the type is monospaced bitmap, the following block of data will exist for each character set from 1 through 16.

► set <count of characters in this set> (starting at 0) same as the number of the last character in the set
<bitmap of character set> x (count/8)+1 The bits in the bitmap in physical order indicate the existence (1) or absence (0) of the corresponding character in the character set. For example, if the high bit of first byte is 1 then character 0 exists. If the low bit of the first bit is 0, then character 7 is omitted.◀

The actual spacing used by a monospaced font may be calculated by the formula:

► nominal character width = (average character width * current point size) / 300◀

The current point size is found in the Font Face Change function code (function 0xD4, subfunction 0x1A) in units of 3600ths and must be converted into 1200ths before it is used in the formula. The average character width is found in the Font Typeface Descriptor Pool packet (packet type 0x20) in units of 1200ths. The nominal character width is the spacing used by a monospaced font and is also in units of 1200ths.

Packet Type 34 (0x22)

Font List

In a SET File, this packet contains the default initial font.

► [font ID (instance pointer)]
[cache ID of PS table] -1 = no PS table
[top shoulder height (WPU)]
[bottom shoulder height (WPU)]
[printed character width (WPU)]
[optimal space width (WPU)]
[normal space width (WPU)]
[horizontal motion units]
[numerator for fractional units of horizontal motion]
[vertical motion units]
[numerator for fractional units of vertical motion]
[pointer in pool of typeface descriptor (packet type 20)]
<flags >
01 = scalable font
02 = faked small cap font
[point size (3600ths)]

[nominal width (WPU)]
 [hash word of descriptor]
 [afc list (entry numbers, -1 = none)] x 16
 [character afc list] x 16
 [orientation afc list] x 4
 [language afc list] x 4
 [substitute font list] x 6 entry number, -1 = none, -2 = on disk◀

Packet Type 35 (0x23)

Printer Selection

►

[long .PRS file] x 37 null terminated, padded to 74 bytes
 <actual .PRS file name> x 13 null terminated, padded to 13 bytes
 [minimum top margin (WPU)]
 [minimum bottom margin (WPU)]
 [minimum left margin (WPU)]
 [minimum right margin (WPU)]
 [minimum top margin (landscape) (WPU)]
 [minimum bottom margin (landscape) (WPU)]
 [minimum left margin (landscape) (WPU)]
 [minimum right margin (landscape) (WPU)]
 <flags>
 bit 0: available
 bit 1: initialize printer when WP starts
 bit 3: right hyphenation zone disabled
 bits 4-7: orientations supported
 16 (0x10) = portrait
 32 (0x20) = landscape
 64 (0x40) = reverse portrait
 128 (0x80) = reverse landscape
 <port number>
 0 = lpt1
 1 = lpt2
 2 = lpt3
 4 = com1
 5 = com2
 6 = com3
 7 = com4
 -1 = filename or device
 <byte to initialize serial port>
 length (bit 0): 0 = 7, 1 = 8
 stop bit (bit 1): 0 = 1 stop bit, 1 = 2 stop bits
 extended baud rate (bit 2): 0 = no, 1 = yes (19200)
 parity (bits 3-4): 0 = none, 1 = odd, 3 = even
 baud rate (bits 5-7):
 0 = 110
 1 = 150
 2 = 300
 3 = 600
 4 = 1200
 5 = 2400
 6 = 4800
 7 = 9600
 <network printer flag>
 1 (0x01) network printer
 2 (0x02) suppress top of form
 4 (0x04) reserved
 8 (0x08) reserved

16 (0x10) reserved
 32 (0x20) reserved
 64 (0x40) reserved
 128 (0x80) reserved
 <network form number>
 <full pathname for non-port device> x 80
 If port number is 0-7, bytes 1-6 (0 based) of this field are used for hardware printing information.
 <pathname only, for auxiliary printer files> x 67
 <additional flags>
 1 (0x01) = prompt for file name
 2 (0x02) = Xon/Xoff instead of DTR
 4 (0x04) = check initial printer status
 8 (0x08) = printer added to list by setup
 16 (0x10) = configured for color
 32 (0x20) = color-capable printer
 64 (0x40) = extended checking
 128 (0x80) = available
 <output tray flags>
 1 (0x01) = sorting supported
 2 (0x02) = grouping supported
 4 (0x04) = offset jogger supported
 8 (0x08) = offset jogger on
 16 (0x10) = output bin on
 32 (0x20) = output bin supported
 64 (0x40) = no output option
 128 (0x80) = printer supports output options
 [bin numbers] x 2 Up to 32 total bin numbers. Each bit is a bin number. When a bit is set, the corresponding bin is active.
 bit 0 = output bin 1
 bit 1 = output bin 2
 bit 2 = output bin 3
 bit 3 = output bin 4
 bit 4 = output bin 5
 bit 5 = output bin 6
 bit 6 = output bin 7
 bit 7 = output bin 8
 bit 8 = output bin 9
 bit 9 = output bin 10
 bit 10 = output bin 11
 bit 11 = output bin 12
 bit 12 = output bin 13
 bit 13 = output bin 14
 bit 14 = output bin 15
 bit 15 = output bin 16
 bit 16 = output bin 17
 bit 17 = output bin 18
 bit 18 = output bin 19
 bit 19 = output bin 20
 bit 20 = output bin 21
 bit 21 = output bin 22
 bit 22 = output bin 23
 bit 23 = output bin 24
 bit 24 = output bin 25
 bit 25 = output bin 26
 bit 26 = output bin 27
 bit 27 = output bin 28
 bit 28 = output bin 29

bit 29 = output bin 30
 bit 30 = output bin 31
 bit 31 = output bin 32
 [date of .PRS file]
 bits 15-9 = year + 80
 bits 8-5 = month
 bits 4-0 = day (starting at 1)
 [time of .PRS file]
 bits 15-11 = hour (24 hour starting at 0)
 bits 10-5 = minutes
 bits 4-0 = seconds/2 (may not be completely accurate)◀

Packet Type 36 (0x24)

Embedded Printer Command

► <embedded printer command> x ? native byte string, null terminated◀

Packet Type 37 (0x25)

Default Initial Font

► [number of PIDs = 1]
 [initial font descriptor PID (type=0x55)]
 [point size (3600ths)]◀

Packet Type 38 (0x26)

Print Information for Q-Codes

No documentation will be provided for this packet.

Packet Type 39 (0x27)

Output Device Type or Port for Q-Codes

No documentation will be provided for this packet.

Packet Type 40 (0x28)

Machine-Dependent Information

No documentation will be provided for this packet.

Packet Type 41 (0x29)

PS Table Prefix Packet IDs

This packet contains the prefix packet IDs of each font's PS table. There should be one PS table entry for each font in the font list. The format is:

► [number of child PS table IDs]
 [ID 0] index 0 PS table ID (type=0x21)
 [ID 1] index 1 PS table ID
 ...
 [ID n] index n PS table ID◀

Packet Type 42 (0x2A)

Third-Party Data

This packet is always used in conjunction with the Third Party function (0xE10B). The format of this packet is defined by the third-party company.

Packet Type 43 (0x2B)

Form Paper Size/Type Name

No documentation will be provided for this packet.

Packet Type 44 (0x2C)

Subdocument Summary

This is the same format as the extended document summary packet, packet type 18 (0x12).

Packet Type 45 (0x2D)

Subdocument Password

No documentation will be provided for this packet.

Packet Type 46 (0x2E)

Deletable Packet

No documentation will be provided for this packet.

Packet Type 47 (0x2F)

Routing Annotation

► <flags>
 bit 0: set means this document came from the routing system
 bits 1-7: reserved
<user count> the number of users listed in this packet
{user IDs size} the size, in bytes, of the user IDs data area
{work flow ID offset} offset into packet of work flow ID
{work flow filename offset} offset into packet of routed filename
<user data> x ? the following five bytes per user, ordered to match the user IDs
 <user display color (RGBBS)> x 4
 <flag byte>
 bit 0: set means the user has completed his review of the document
 bits 1-7: reserved
[users IDs index] x ? two bytes per user containing the offset of his mail system user ID relative to the beginning of user IDs data (user IDs are stored in route order)
<user IDs data> x ? the user IDs stored as packed strings of 8-bit ANSI characters
[user names index] x ? two bytes per user containing the offset of his long name relative to the beginning of user names data (ordered to match the user IDs)
<user names data> x ? the user long names stored as packed strings of 8-bit ANSI characters (ordered to match the user IDs)
<work flow ID> x ? Ensemble Lite work flow ID passed to WP on the command line (a null-terminated string)
<work flow document name> x ? Ensemble Lite routed document name as returned from the routing system (a null-terminated string)◀

Packet Type 48 (0x30)

Normal Style

► [number of PIDs]
[PID of style to link to (type=0x30)]
 If this is null, look at the Enter flag. If this ID exists, it means that when this style is ended (or when Enter is pressed), the style ID that is linked to begins.
[number of text blocks = 4]
{relative offset of 1st text block}

{paragraph text size}
 {beginning style text size}
 {end style text size}
 {extra style text size}
 <style type>
 1 = open style
 2 = paired style
 3 = paragraph style
 4 = deleted “including codes”
 5 = deleted “leaving codes”
 <style flag>
 1 (0x01) = library style (otherwise local only)
 2 (0x02) = system style
 4 (0x04) = Enter turns style off (ignored if link ID is valid)
 8 (0x08) = Enter turns style back on again (ignored if link ID is valid)
 16 (0x10) = reserved (always set to null when creating a style code)
 32 (0x20) = style includes other style(s)
 64 (0x40) = reserved
 128 (0x80) = reserved
 [style hash]
 [offset from beginning of packet to description]
 [style name] x 12(max) null terminated
 [style description] x ? null terminated
 The following data is the text block part of the style.
 <paragraph-oriented begin information> x paragraph text size
 <other begin information> x beginning style text size
 <end information> x ending style text size
 <extra information> x extra style text size◀

Packet Type 49 (0x31)

Outline Style

Null level IDs indicate that the default outline is used for that level.

►

[child ID count = 8]
 [level 1 ID (type=0x30)]
 [level 2 ID (type=0x30)]
 [level 3 ID (type=0x30)]
 [level 4 ID (type=0x30)]
 [level 5 ID (type=0x30)]
 [level 6 ID (type=0x30)]
 [level 7 ID (type=0x30)]
 [level 8 ID (type=0x30)]
 <style flags>
 low nibble values:
 0x01 = library style
 0x02 = system style
 0x04 = outline does not use paragraph styles
 0x08 = reserved
 high nibble values:
 0x00 = heading paragraph styles
 0x10 = reserved
 0x20 = normal paragraph styles
 0x40 = use legal (non-indented) paragraph styles
 0x60 = reserved
 0x80 = reserved
 0xA0 = reserved

0xC0 = reserved
 0xE0 = this style has been deleted
 [hash of IDs, name, flags] set to null when creating the style
 <numbering methods for printing> x 8
 one byte for each level
 0 = Arabic numbers
 1 = lowercase letters (a-z)
 2 = uppercase letters (A-Z)
 3 = lowercase Roman
 4 = uppercase Roman
 5 = character string (only valid for footnotes and endnotes)
 6 = leading-zero Arabic numbers
 For Asia: (2-byte character required; uses two character cells to PRN)
 0x40 = 2-byte Arabic
 0x41 = 2-byte lowercase letters (a-z)
 0x42 = 2-byte uppercase letters (A-Z)
 0x43 = 2-byte lowercase Roman
 0x44 = 2-byte uppercase Roman
 0x45 = 2-byte character string (only valid for footnotes and endnotes)
 0x46 = 2-byte leading-zero Arabic numbers
 0x80 = kanji
 0x81 = 1-byte katakana (aiueo)
 0x82 = 2-byte katakana (aiueo)
 0x83 = 1-byte katakana (iroha)
 0x84 = 2-byte katakana (iroha)
 0x85 = available
 0x86 = standard kanji with leading-zero
 0xC0 = kanji 2 (ten = juu)
 If the numbering method is character string, the following information exists:
 [character string length]
 [character string] x length in bytes
 [pointer to description]
 [8 pointers to number strings] x 8
 [outline style name] x 13(max) null terminated
 [description] x ? null terminated
 [level 1 num string] x 21(max) null-terminated WP word string
 [level 2 num string] x 21(max) null-terminated WP word string
 [level 3 num string] x 21(max) null-terminated WP word string
 [level 4 num string] x 21(max) null-terminated WP word string
 [level 5 num string] x 21(max) null-terminated WP word string
 [level 6 num string] x 21(max) null-terminated WP word string
 [level 7 num string] x 21(max) null-terminated WP word string
 [level 8 num string] x 21(max) null-terminated WP word string◀

Packet Type 50 (0x32)

Personal Style Library Filename

► [personal style library filename] x ? null terminated◀

If this packet ever has child IDs, the format will be that of the Native Filename packet type 0x07.

Packet Type 51 (0x33)

Shared Style Library Filename

► [shared style library filename] x ? null terminated◀

If this packet ever has child IDs, the format will be that of the Native Filename packet type 0x07.

Packet Type 52 (0x34)

Personal and Shared Library File Date/Time

- ▶ <date structure for personal style library file> x 10
<date structure for shared style library file> x 10◀

10-Byte Date Structure

- ▶ [year]
<month>
<day>
<hour>
<minute>
<second> (unused = 0)
<day of week> (unused = 0)
<time zone> (unused = 0)
<unused> (= 0)◀

Packet Type 53 (0x35)

Default Native Fonts

Mideast default font and size for dual or multiple fonts.

- ▶ [number of child prefix IDs = 1]
[PID of desired font]
<point size of font> in 3600ths
<language set of the initial native font>
9 = Hebrew
13 = Arabic◀

Packet Type 54 (0x36)

English AFC (Fareast)

Use English AFC for Asian fonts.

- ▶ <use English AFC flag> 1 = use English AFC for double byte fonts

Packet Type 55 (0x37) -63 (0x3F)

Reserved

Packet Type 64 (0x40)

Graphics Filename

If this packet does not have any children, the format is:

- ☺ [graphics filename] x ? null terminated◀

If this packet does have children, the format is:

- ▶ [number of child IDs]
[ID 1] (type=0x6F, 0x70, or 0x71) ID of child packet containing graphics data
[ID 2] (type=0x6F, 0x70, or 0x71) ID of child packet containing graphics data
...

[ID n] (type=0x6F, 0x70, or 0x71) ID of last child containing graphics data
[tag 1] tag flag for first child packet
[tag 2] tag flag for second child packet
...
[tag n] last tag flag for last child packet
[graphics filename] x ? null terminated◀

Format for Tag Flags

0x0001 = data being kept internal

0x0010 = data being kept internal (hot link)

0x0011 = data being kept internal (warm link)

0x0012 = data being kept internal (cold link)