WordPerfect for Windows 6.0 SGML File Format Additions July 21, 1994

FUNCTIONS ADDED FOR WordPerfect 6.1
Tabs for this document are every 4 spaces

The minor version number for WP6.1 documents is 1 (was 0 for WP6.0) So the first 16 bytes of the header are (NOTE when "<" and ">" surround a format description the size of the data is a byte. "[" and "]" mean a size of data short (2 bytes). "{" and "}" means size is a long value (4 bytes)).
FormatDescription <0xFF><"W"><"P"><"C"> -1, WPC {Pointer to Document Area} Long pointer to area (absolute offset from Beginning of file <Product type> WPCorp Product # (WordPerfect = 1) <File type> WPCorp file type (WordPerfect document = 0x0A) <Major version> 2 = 6.x file <Minor version> 0 = 6.0 file
1 = 6.1 file [Encryption] if nonzero then document is encrypted [pointer to index area] Offset of inex header (a value of 0-15 implies an offset of 16) <reserved> beginning of extended file header <reserved> [reserved] {file size} File size, not include pad characters at EOF
Prefix Packet additions
#define PREFIX_NEEDS_SPELLCHECK 0x1d /* regardless of clean/dirty spell gates, entire doc is dirty */ Index exists if Spellcheck is needed. Otherwise it doesn't exist. The data size is zero. The data pointer is unused and invalid if non-zero.
#define PREFIX_FORMATTER_REGION 0x1E /* Region being tracked in formatter for any changes */
Region Packet Header
Region Packet # 1
Region Packet #2
...
/* offsets to describe the header in the region prefix packet */ WUWORD REGION_SIZE 0 /*offset to first region packet information */ WUDWORD REGION_ENTRIES 2 /* # of region entries in this prefix packet */ WUDWORD REGION_MAXCOUNT 6 /* the next region # to allocate */

WUBYTE[] REGION_FILENAME 10 /* name of file this packet created with */

/* offsets which describe each regions information */ WUDWORD REGION_NUMBER 0 /* unique # representing this tracked region */ WUDWORD REGION_BEGPOS 4 /* byte count offset in edit buffer to region on code */ /* distance from beginning of file if positive */ /* -distance from end of file if negative */ WUDWORD REGION_ENDPOS 8 /* byte count offset in edit buffer to region off code */ /* see explaination for REGION_BEGPOS */ WUWORD REGION_TOPPAGE 12 /* page # where start of tracked region starts */ WPU REGION_TOPY 14 /* y position where start of tracked region starts */ WUWORD REGION_BOTTOMPAGE 16 /* page # where end of tracked region ends */ WPU REGION_BOTTOMY 18 /* y position where end of tracked region ends */ WUWORD REGION_FONTFACE 20 /* font face at start of region */ WUWORD REGION_FONTSIZE 22 /* font size at start of region */ WUWORD REGION_ATTRIBUTES 24 /* attributes at start of region */ WUBYTE REGION_COLOR_RED 26 /* red color at start of region */ WUBYTE REGION_COLOR_GREEN 27 /* green color at start of region */ WUBYTE REGION_COLOR_BLUE 28 /* blue color at start of region */ WUBYTE REGION_ACTIVEFLAG 29 /* indicates if changes to region should be sent to interface*/

0 = REGIONNOTACTIVE

1 = REGIONISACTIVE WUBYTE REGION_TYPEFLAG 30 /*type of region being tracked */

1 = OLE

2 = OBEX

#define PREFIX_MIDEAST_FLAGS 0x1f /* MIDEAST specific doc flags */ WUBYTE docFlagsBuf[2]; /* ligatue control , number shape */ /* 1st byte ligature */ /* 0 - None */ /* 1 - Partial ligatures */ /* 2 - Full ligatures */ /* 2nd byte number shaping */ /* 0 - Context */ /* 1 - Force English font */ /* 2 - Force Arabic/Hebrew font */

#define PREFIX_DEFAULT_NATIVE_FONTS 0x35 /* MIDEAST default font and size for dual or multiple fonts */ typedef struct { WUWORD numChildPrefixes; /* # prefixes this prefix points to (may BE more than 1!) */ PREFIX_ID fontPrefixId; /* pfx id of PREFIX_DESIRED_FONT */ WUWORD fontSize; /* point size of font (in 3600s) */ WUBYTE angSetNtv; /* the language set of the initial native font (Hebrew:9, Arabic:13) */ } DEFAULTNATIVEFONTS;

#define PREFIX_ENGLISH_AFC 0x36 /* FAREAST document use English AFC for Asian fonts */ typedef struct { WBOOL UseEnglishAFCFlag; /* use English AFC for double byte fonts */ } USEENGLISHAFC;

#define PREFIX_TABLE_STYLE 0x69 /* Table Style prefix packet (6.1) */

Notes: various "flags" are defined below the layout definition colors are RGB with a percent shading req = required, always written opt = optional (see condition, usually a flag bit is set)

TABLE STYLE PREFIX AREA:

Type #bytes Req/Opt Description WUWORD 2 req # of prefix ids to write (n) WUWORD 2*n opt prefix ids:

Table Border (flags & TABLETMPLT_TBLBORD) Table Default Lines (flags & TABLETMPLT_TBLDEFLNS) Table Border Override Line IDs (see TblBordIds.flags below) Default Line Override IDs (see DefBordIds.flags below) Last Row Line IDs (see LRowBordIds.flags below) First Column Line IDs (see FColBordIds.flags below) Last Column Line IDs (see LColBordIds.flags below) Header Rows Line IDs (check each hrow->HRowBordIds.flags) Table Fill ID (flags & TABLETMPLT_TBLFILL) Table Alt. Fill ID (flags & TABLETMPLT_TBLALTFILL) Last Row Fill ID (flags & TABLETMPLT_LROWFILL) First Column Fill ID (flags & TABLETMPLT_FCOLFILL) Last Column Fill ID (flags & TABLETMPLT_LCOLFILL) Header Rows Fill IDs (check each hrow->flags & HROWDATA_FILLID) WUWORD 2 req size of following data WUWORD 2 req size of name info following WUBYTE 1 req name/library prefix id flags WUBYTE 1 req library management flags Mixed Var req table style name (negative number if predefined) WUWORD 2 req size of rest of data WUWORD 2 req flags for borders and fills prefix IDs WUWORD 2 req override flags for borders, fills, and formats WUWORD 2 req number of header rows WUWORD 2req starting header row number (zero based) WUBYTE 1 req alternating fill flag (0=no alt. fill, 1=rows, 2=columns, 3=both) WUWORD 2 req no. times to do primary fill WUWORD 2 req no. times to do alternating fill WUWORD 2 opt Table Border Override Line IDs flag (flags & TABLETMPLT_TBLBORDIDS) Mixed 6opt Table Border Overrides (see BordOver) (overrideFlags & TABLETMPLT_TBLBORDOVER) WUWORD 2 opt Table Default Line Override flags (see IdsFlag) (flags & TABLETMPLT_DEFBORD) Mixed 6 opt Table Default Line Overrides (see BordOver) (overrideFlags & TABLETMPLT_DEFBORDOVER)

TABLE STYLE PREFIX AREA LAYOUT (continued)

Type #bytes Req/Opt Description Mixed 10 opt Table Fill Overrides (see FillOver) (overrideFlags & TABLETMPLT_TBLFILLOVER) Mixed 10 opt Table Alt. Fill Overrides (see FillOver) (overrideFlags & TABLETMPLT_ALTFILLOVER) Mixed 11 opt Table Format Data (see FmtData) (overrideFlags & TABLETMPLT_TBLFMT) Mixed 7 opt Drop Shadow and Overrides (see DropShadow) (flags & TABLETMPLT_DROPSHADOW)

WUWORD 2 opt Last Row Border IDs flag (see IdsFlag) (flags & TABLETMPLT_LROWBORD) Mixed 6 opt Last Row Border Overrides (see BordOver) (overrideFlags & TABLETMPLT_LROWBORDOVER) Mixed 10 opt Last Row Fill Overrides (see FillOver) (overrideFlags & TABLETMPLT_LROWFILLOVER) Mixed 11 opt Last Row Format Data (see FmtData) (overrideFlags & TABLETMPLT_LROWFMT)

WUWORD 2 opt First Column Border IDs flag (see IdsFlag) (flags & TABLETMPLT_FCOLBORD) Mixed 6 opt First Column Border Overrides (see BordOver) (overrideFlags & TABLETMPLT_FCOLBORDOVER) Mixed 10 opt First Column Fill Overrides (see FillOver) (overrideFlags & TABLETMPLT_FCOLFILLOVER) Mixed 11 opt First Column Format Data (see FmtData) (overrideFlags & TABLETMPLT_FCOLFMT)

WUWORD 2 opt Last Column Border IDs flag (see IdsFlag) (flags & TABLETMPLT_LCOLBORD) Mixed 6 opt Last Column Border Overrides (see BordOver) (overrideFlags & TABLETMPLT_LCOLBORDOVER) Mixed 10 opt Last Column Fill Overrides (see FillOver) (overrideFlags & TABLETMPLT_LCOLFILLOVER) Mixed 11 opt Last Column Format Data (see FmtData) (overrideFlags & TABLETMPLT_LCOLFMT)

FOR EACH HEADER ROW: (optional) WUWORD 2 req headerRowFlags (see below) WUWORD 2 opt line IDs flags (hrow->flags & HROWDATA_BORDIDS) Mixed 6 opt border overrides (hrow->flags & HROWDATA_BORDOVER) Mixed 10 opt fill overrides (hrow->flags & HROWDATA_FILLOVER) Mixed 11 opt format data (hrow->flags & HROWDATA_TABLEFMT)

TABLE STYLE PREFIX AREA LAYOUT NOTES:

flags: PRESENCE FLAGS

TABLETMPLT_TBLBORD 0x0001 table border id TABLETMPLT_TBLDEFLNS 0x0002 table default lines id TABLETMPLT_TBLFILL 0x0004 table fill id TABLETMPLT_TBLALTFILL 0x0008 table alt. fill id TABLETMPLT_TBLBORDIDS 0x0010 table border override lines ids TABLETMPLT_DEFBORD 0x0020 default line override ids TABLETMPLT_LROWBORD 0x0040 last row line ids TABLETMPLT_LROWFILL 0x0080 last row fill id TABLETMPLT_FCOLBORD 0x0100 first column line ids TABLETMPLT_FCOLFILL 0x0200 first column fill id TABLETMPLT_LCOLBORD 0x0400 last column line ids ABLETMPLT_LCOLFILL 0x0800 last column fill id TABLETMPLT_DROPSHADOW 0x1000 drop shadow

overrideFlags: PRESENCE OVERRIDE FLAGS TABLETMPLT_TBLBORDOVER 0x0001 table border overrides TABLETMPLT_DEFBORDOVER 0x0002 default line overrides TABLETMPLT_TBLFILLOVER 0x0004 table fill overrides TABLETMPLT_ALTFILLOVER 0x0008 table alt. fill overrides TABLETMPLT_LROWBORDOVER 0x0010 last row line overrides TABLETMPLT_LROWFILLOVER 0x0020 last row fill overrides TABLETMPLT_FCOLBORDOVER 0x0040 first column line overrides TABLETMPLT_FCOLFILLOVER 0x0080 first column fill overrides TABLETMPLT_LCOLBORDOVER 0x0100 last column line overrides TABLETMPLT_LCOLFILLOVER 0x0200 last column fill overrides TABLETMPLT_TBLFMT 0x0400 table format overrides TABLETMPLT_LROWFMT 0x0800 last row format overrides TABLETMPLT_FCOLFMT 0x1000 first column format overrides TABLETMPLT_LCOLFMT 0x2000 last column format overrides

IdsFlags: PRESENCE FLAGS BORDIDS_LEFT 0x0001 Left Line Prefix ID BORDIDS_RIGHT 0x0002 Right Line Prefix ID BORDIDS_TOP 0x0004 Top Line Prefix ID BORDIDS_BOTTOM 0x0008 Bottom Line Prefix ID BORDIDS_INSIDE 0x0010 Inside/Separator Line Prefix ID

headerRowFlags: PRESENCE FLAGS HROWDATA_BORDIDS 0x01 border/line ids HROWDATA_FILLID 0x02 fill id HROWDATA_BORDOVER 0x04 border overrides HROWDATA_FILLOVER 0x08 fill overrides HROWDATA_TABLEFMT 0x10 format data

BordOver: BORDER/LINE OVERRIDE DATA WUWORD 2 use flags

0x0001	red
0x0002	green
0x0004	blue
0x0008	shade

WUBYTE 1 Red value WUBYTE 1 Green value WUBYTE 1 Blue value WUBYTE 1 Shade value

FillOver: FILL OVERRIDE DATA WUWORD 2 use flags

0x0001 foreground

0x0002 background WUBYTE 1 Foreground Red WUBYTE 1 Foreground Green WUBYTE 1 Foreground Blue WUBYTE 1 Foreground Shade WUBYTE 1 Background Red WUBYTE 1 Background Green WUBYTE 1 Background Blue WUBYTE 1 Background Shade

FmtData: FORMAT DATA WUWORD 2 use flags

0x0001 attributes

0x0002 number type

0x0004 currency index

0x0008 justification

0x0010 join cells in row or col WUDWORD 4 attribute double word (2 WP standard attribute words) WUWORD 2 number type (WP standard) WUBYTE 1 currency symbol index (WP standard) WUWORD 2 justification (WP standard)

DropShadow: DROP SHADOW DATA WUBYTE 1 presence flags

1: No Drop Shadow

1: Upper Left Corner

2: Lower Left Corner

3: Lower Right Corner

4: Upper Right Corner WUWORD 2 drop shadow space WUBYTE 1 Red WUBYTE 1 Green WUBYTE 1 Blue WUBYTE 1 Shade

#define PREFIX_CVT_FONT 0x6e /* conversion font descriptor packet */ Stored are 5.1 and 6.0 fonts used for conversion purposes only.

5.x Equivalent packet 0x30.

When document is converted to 5.x from 6.0 0x30 packet is used to store native 6.0 fonts.

converting from 5.x to 6.0 then 0x6e is used to store native 5.x fonts.

This went out with 6.0a for Windows and 6.0b for DOS.

#define PREFIX_UNDO_STATE 0x77 /* formatter undo state */ Offset Variable name Description

0 undo_levels # of undos in doc

2 redo_levels # of redos in doc (The structures for redos are the same as the UNDOs, but will begin after the last undo index)

(1...n repeats for UNDOs and REDOs /* the next 6 bytes are repeated for each undo level */)
0 undolevel[i] level # at this index
2 undocnt[i] # undos at this level
4 undos[i].token token that triggered this undo (margin change, text, etc...)
/*The next three packets were going added for WP6.1 fonts stuff, but the decision was made to not use them for 6.1. */ #define PREFIX_FONT_PS_TABLE_ENG 0x78 /* Font PS table for WP 6.0 (multiple-use) */ #define PREFIX_FONT_LIST_ENG 0x79 /* list of matched fonts used in document */ #define PREFIX_PS_TABLE_IDS_ENG 0x7a /* ps table prefix id packet */
#define STYLE_UNDO_SWAPPED 0x7b /* undo - swapped prefix packet */ This packet is the same format as packet type 0x30. It is used to undo a style change.
#define PREFIX_OBEX_ID 0x7c /* id string for export of obex data */ dword 0xffffffff region ID that mean whole documents char[] Obex object ID ansistring format NULL termination
NOTE: No changes were made in the prefix packet to support OLE2.0
SGML Prefix Packets Information
Tag with Attributes -- parent packet (0x7D)
[# of attributes] [attribute 1 prefix ID] /* Prefix IDs of attribute values (child packets) */ [attribute 2 prefix ID] .. ..
[attribute n prefix ID]
[tag error flag] /* Error flag for entire tag - see note 1 */
/* Individual attribute info */ [attribute 1 DTD index] /* Index to attribute info in memory */ <attribute 1 error code> /* Error for individual attribute - see note 2 */ [attribute 2 DTD index] <attribute 2 error code> .. ..
[attribute n DTD index] <attribute n error code> -- Note 1: Defines for tag error flag - bitwise flag
0 /* No error */ ERTAGLEN 1 /* TAGLEN is exceeded for the tag */ ERATSPLEN 2 /* Attribute specification length exceeded for the tag */ ERIDGPCNT 4 /* IDGPCNT (# of ids in attribute values) exceeded */ ERENGPCNT 8 /* ENGPCNT (# of entities in att values) exceeded */ -- Note 2: Defines for attribute error code - only bit 7 is bitwie. If bit 7 is set then both LITDEL and ALITDEL is in the attribute value.
0 /* No error */ ER_UNKATT 1 /* Attribute is unknown to DTD */ ER_ATNAM 2 /* Illegal character in attribute name value */ ER_ATNMTK 3 /* Illegal character in attribute name token value */ ER_ATNUM 4 /* Illegal character in attribute number value */ ER_ATNUMTK 5 /* Illegal character in attribute number token value */ ER_NM2LNG 6 /* Value in attribute exceeds NAMELEN limit */

ER_2MNYTOK 7 /* Too many tokens in attribute */ ER_LITLEN 8 /* Value in attribute exceeds LITLEN limit */ ER_ATEXMAP 9 /* No mapping for extended character in attribute */ ER_EMPTYAT 10 /* Attribute has empty attribute value literal */ ER_DECLVL 11 /* Token value not in declared value list */ ER_FXDATTR 12 /* Mismatch between DTD fixed value and attr value */ ER_UNKENTA 13 /* Unknown entity type attribute value */ ER_ILLENTA 14 /* Illegal entity type attribute value */ ER_ATENT 15 /* Entity open in attribute value not expanded */ ER_ATCREF 16 /* Char ref open in attribute value not checked */ ER_ATDUP 17 /* Attribute is duplicated in tag */

Attribute Value -- child packet (0x7E)

[number of blocks = 1] {offset to text} {size of text} <literal delimiter> /* 1 = LITDEL, 2 = ALITDEL, 3 = Both present */ [attribute name]* <text>*

LGC Filename (0x7F)

[LGC filename] x ? null terminated

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

External Entity (0x80)

[number of child IDs = 1] [ID 1(type=0x81) ID of child packet for notation attributes]

[data type] /* CDATA, SDATA, etc. */ [entity declaration error]/* Error flag for full declaration */

[number of text strings] [size of entity name] [size of file path (system ID)] [size of public ID] [size of notation name]

[external entity name]* [file path (system ID)]* [public ID]* [notation name]*

Notation with Attributes (0x81) -- identical to "Tag with Attributes" packet (0x7D)
[# of attributes] [attribute 1 prefix ID] /* Prefix IDs of attribute values (child packets) */
[attribute 2 prefix ID] .. ..
[attribute n prefix ID]
[notation error flag] /* Error flag for entire notation - see note 1 */
/* Individual attribute info */ [attribute 1 DTD index] /* Index to attribute info in memory */
<attribute 1 error code> /* Error for individual attribute - see note 2 */ [attribute 2 DTD
index] <attribute 2 error code> .. ..
[attribute n DTD index] <attribute n error code>

Document Function code additions for WP6.1
Single byte function codes
EOL Group Single byte function additions Additions #define YUNKSB 0x98 /* Reserved unknown single byte function */ #define YNOREV 0x99 /* No "revert" for this style */ #define YOPTSPC 0x9A /* Optional space for selection clean u #define YSPCLON 0x9B /* Speller clean on */ #define YSPCLOF 0x9C /* Speller clean off */ #define YJOIN 0x9D /* (MIDEAST) Joiner control */ #define YNJOIN 0x9E /* (MIDEAST) Non-joiner control */ #define YWEAKON 0x9F /* international */ #define YWEAKOFF 0xA0 /* international */ #define YDEAN 0xA1 /* (MIDEAST) end of right to left DECalign */

//EOL Single Byte functions #define YIHRTPG 0xAE /* Invisible/Deletable hard eol/eoc/eop*/ #define YIHRTCL 0xAF /* Invisible/Deletable hard eol/eoc */ #define YIHRT 0xB0 /* Invisible/Deletable hard eol */ #define YISRTPG 0xB1 /* Invisible/Deletable soft eol/eoc/eop*/ #define YISRTCL 0xB2 /* Invisible/Deletable soft eol/eoc */ #define YISRT 0xB3 /* Invisible/Deletable soft eol */

End Of Line Group 0xD0 functions (added in WPwin 6.0)

#define EOL_ISRT 0x1D /* Inv/Del soft eol */ #define EOL_ISRTCL 0x1E /* Inv/Del soft eol/eoc */ #define EOL_ISRTPG 0x1F /* Inv/Del soft eol/eoc/eop */ #define EOL_IHRT 0x20 /* Inv/Del hard eol */ #define EOL_IHRTCL 0x21 /* Inv/Del hard eol/eoc */ #define EOL_IHRTPG 0x22 /* Inv/Del hard eol/eoc/eop */

Page Group (0xD1) additions

pag_linecnt	0xD11C
pag_flow	0xD11D

Paragraph Group (0xD3) addtions

par_dropcap	0xD317

par_tdir	0xD318
par_asiawrap	0xD319
par_charcnt	0xD31A

Character Group (0xD4) additions

chr_attchmark	0xD447
chr_attchtxt	0xD448
chr_atxtopt	0xD449
chr_lig	0xD44A

chr_arabnum	0xD44B
chr_rtlb	0xD44C
chr_rtle	0xD44D
chr_ffacentv	0xD44E
chr_fsizentv	0xD44F

chr_regionon	0xD450
chr_regionof	0xD451
chr_stexb	0xD452
chr_stexe	0xD453

chr_stund 0xD454