

Office Open XML

**Ecma TC45
Final Draft
Part 4: Markup Language Reference**

October 2006

Warning! This is the “Muscular” version of WordProcessingML. It has been edited for demonstration purposes to show the impact of removing non-normative and duplicative material from the OpenXML proposal. It in no way represents any official opinion or position on OpenXML. Contact: Patrick@durusau.net for further details.

PS: None of the internal links work. Sorry! They did not work in the source I used.

Table of Contents

1		
2	Foreword	iv
3	Introduction	v
4	1. Part Overview	1
5	1.1 WordprocessingML Part Summary	1
6	1.2 SpreadsheetML Part Summary	1
7	1.3 PresentationML Part Summary	3
8	1.4 DrawingML Part Summary	4
9	1.5 Shared Part Summary	5
10	2. WordprocessingML Reference Material	7
11	2.1 Table of Contents	7
12	2.2 Main Document Story	26
13	2.3 Paragraphs and Rich Formatting	27
14	2.4 Tables	79
15	2.5 Custom Markup	106
16	2.6 Sections	119
17	2.7 Styles	129
18	2.8 Fonts	145
19	2.9 Numbering	151
20	2.10 Headers and Footers	160
21	2.11 Footnotes and Endnotes	163
22	2.12 Glossary Document	168
23	2.13 Annotations	171
24	2.14 Mail Merge	195
25	2.15 Settings	206
26	2.16 Fields & Hyperlinks	276
27	2.17 Miscellaneous Topics	328
28	2.18 Simple Types	330
29		

Foreword

This multi-part Standard deals with Office Open XML Format-related technology, and consists of the following parts:

- Part 1: "Fundamentals"
- Part 2: "Open Packaging Conventions"
- Part 3: "Primer"
- **Part 4: "Markup Language Reference" (this document)**
- Part 5: "Markup Compatibility and Extensibility"

This part, Part 4, includes a number of annexes that refer to data files provided in electronic form only.

Introduction

This Part is one piece of a Standard that describes a family of XML schemas, collectively called *Office Open XML*, which define the XML vocabularies for word-processing, spreadsheet, and presentation documents, as well as the packaging of documents that conform to these schemas.

The goal is to enable the implementation of the Office Open XML formats by the widest set of tools and platforms, fostering interoperability across office productivity applications and line-of-business systems, as well as to support and strengthen document archival and preservation, all in a way that is fully compatible with the large existing investments in Microsoft Office documents.

1. Part Overview

This clause is informative.

For convenience, the following subclauses specify the root elements for each part. Full discussion of the use of each part can be found in Part 1 of this Office Open XML Standard.

1.1 WordprocessingML Part Summary

Part	Root Element	Ref.
Alternative Format Import	Not applicable	n/a
Comments	comments	§2.13.4.6
Document Settings	settings	§2.15.1.78
Endnotes	endnotes	§2.11.8
Font Table	fonts	§2.8.2.11
Footer	ft	§2.10.3
Footnotes	footnotes	§2.11.15
Glossary Document	glossaryDocument	§2.12.10
Header	hdr	§2.10.4
Mail Merge Recipient Data	recipients	§2.14.29
Main Document	document	§2.2.3
Numbering Definitions	numbering	§2.9.17
Style Definitions	styles	§2.7.3.18
Web Settings	webSettings	§2.15.2.44

1.2 SpreadsheetML Part Summary

Part	Root Element	Ref.
Calculation Chain	calcChain	§Error! Reference source not found.
Chartsheet	chartsheet	§Error! Reference source not found.
Comments	comments	§Error! Reference source not found.
Connections	connections	§Error! Reference

Part	Root Element	Ref.
		source not found.
Custom Property	Not applicable	n/a
Custom XML Mappings	mapInfo	§Error! Reference source not found.
Dialogsheet	dialogSheet	§Error! Reference source not found.
Drawing	wsDr	§Error! Reference source not found.
External Workbook References	externalLink	§Error! Reference source not found.
Metadata	metadata	§Error! Reference source not found.
Pivot Table	pivotTableDefinition	§Error! Reference source not found.
Pivot Table Cache Definition	pivotCacheDefinition	§Error! Reference source not found.
Pivot Table Cache Records	pivotCacheRecords	§Error! Reference source not found.
Query Table	queryTable	§Error! Reference source not found.
Shared String Table	sst	§Error! Reference source not found.
Shared Workbook Revision Headers	headers	§Error! Reference source not

Part	Root Element	Ref.
		found.
Shared Workbook Revision Log	revisions	\$Error! Reference source not found.
Shared Workbook User Data	users	\$Error! Reference source not found.
Single Cell Table Definitions	singleCells	\$Error! Reference source not found.
Styles	styleSheet	\$Error! Reference source not found.
Table Definition	table	\$Error! Reference source not found.
Volatile Dependencies	volTypes	\$Error! Reference source not found.
Workbook	workbook	\$Error! Reference source not found.
Worksheet	worksheet	\$Error! Reference source not found.

1.3 PresentationML Part Summary

Part	Root Element	Ref.
Comment Authors	cmAuthorLst	\$Error! Reference source not found.
Comments	cmLst	\$Error! Reference source not found.
Handout Master	handoutMaster	\$Error! Reference

Part	Root Element	Ref.
		source not found.
Notes Master	notesMaster	§Error! Reference source not found.
Notes Slide	notes	§Error! Reference source not found.
Presentation	presentation	§Error! Reference source not found.
Presentation Properties	presentationPr	§Error! Reference source not found.
Slide	sld	§Error! Reference source not found.
Slide Layout	sldLayout	§Error! Reference source not found.
Slide Master	sldMaster	§Error! Reference source not found.
Slide Synchronization Data	sldSyncPr	§Error! Reference source not found.
User-Defined Tags	tagLst	§Error! Reference source not found.
View Properties	viewPr	§Error! Reference source not found.

1.4 DrawingML Part Summary

Part	Root Element	Ref.
Chart	chartSpace	§Error!

Part	Root Element	Ref.
		Reference source not found.
Chart Drawing	userShapes	§Error! Reference source not found.
Diagram Colors	colorsDef	§Error! Reference source not found.
Diagram Data	dataModel	§Error! Reference source not found.
Diagram Layout Definition	layoutDef	§Error! Reference source not found.
Diagram Style	styleDef	§Error! Reference source not found.
Theme	officeStyleSheet	§Error! Reference source not found.
Theme Override	themeOverride	§Error! Reference source not found.
Table Styles	tblStyleLst	§Error! Reference source not found.

1.5 Shared Part Summary

Part	Root Element	Ref.
Additional Characteristics	additionalCharacteristics	§Error! Reference source not found.
Audio	Not applicable	n/a
Bibliography	Sources	§Error! Reference source not found.

Part	Root Element	Ref.
		found.
Custom XML Data Storage	Not applicable	n/a
Custom XML Data Storage Properties	datastoreItem	§Error! Reference source not found.
Digital Signature Origin	Not applicable	n/a
Digital Signature XML Signature	Signature	Defined in Part 2
Embedded Control Persistence	Not applicable	n/a
Embedded Object	Not applicable	n/a
Embedded Package	Not applicable	n/a
File Properties, Core	coreProperties	Defined in Part 2
File Properties, Custom	Properties	§Error! Reference source not found.
File Properties, Extended	Properties	§Error! Reference source not found.
Font	Not applicable	n/a
Image	Not applicable	n/a
Printer Settings	Not applicable	n/a
Thumbnail	Not applicable	n/a
Video	Not applicable	n/a

End of informative text.

2. WordprocessingML Reference Material

The subordinate subclauses specify the semantics for the XML markup comprising a WordprocessingML document, as defined by Part 1 of this Standard.

2.1 Table of Contents

This subclause is informative.

2.2 Main Document Story	26
2.2.1 background (Document Background).....	26
2.2.2 body (Document Body).....	27
2.2.3 document (Document)	27
2.3 Paragraphs and Rich Formatting.....	27
2.3.1 Paragraphs.....	28
2.3.1.1 adjustRightInd (Automatically Adjust Right Indent When Using Document Grid)	28
2.3.1.2 autoSpaceDE (Automatically Adjust Spacing of Latin and East Asian Text)	28
2.3.1.3 autoSpaceDN (Automatically Adjust Spacing of East Asian Text and Numbers)	28
2.3.1.4 bar (Paragraph Border Between Facing Pages)	29
2.3.1.5 between (Paragraph Border Between Identical Paragraphs).....	30
2.3.1.6 bidi (Right to Left Paragraph Layout).....	30
2.3.1.7 bottom (Paragraph Border Between Identical Paragraphs)	31
2.3.1.8 cnfStyle (Paragraph Conditional Formatting)	31
2.3.1.9 contextualSpacing (Ignore Spacing Above and Below When Using Identical Styles)	32
2.3.1.10 divId (Associated HTML div ID)	32
2.3.1.11 framePr (Text Frame Properties).....	32
2.3.1.12 ind (Paragraph Indentation)	36
2.3.1.13 jc (Paragraph Alignment)	38
2.3.1.14 keepLines (Keep All Lines On One Page)	38
2.3.1.15 keepNext (Keep Paragraph With Next Paragraph).....	39
2.3.1.16 kinsoku (Use East Asian Typography Rules for First and Last Character per Line)	39
2.3.1.17 left (Left Paragraph Border).....	40
2.3.1.18 mirrorIndents (Use Left/Right Indents as Inside/Outside Indents)	40
2.3.1.19 numPr (Numbering Definition Instance Reference)	41
2.3.1.20 outlineLvl (Associated Outline Level)	41
2.3.1.21 overflowPunct (Allow Punctuation to Extent Past Text Extents)	41
2.3.1.22 p (Paragraph)	41
2.3.1.23 pageBreakBefore (Start Paragraph on Next Page)	43
2.3.1.24 pBdr (Paragraph Borders).....	43
2.3.1.25 pPr (Previous Paragraph Properties)	43
2.3.1.26 pPr (Paragraph Properties)	43
2.3.1.27 pStyle (Referenced Paragraph Style).....	43
2.3.1.28 right (Right Paragraph Border)	44
2.3.1.29 rPr (Run Properties for the Paragraph Mark)	44
2.3.1.30 rPr (Previous Run Properties for the Paragraph Mark)	44
2.3.1.31 shd (Paragraph Shading).....	44
2.3.1.32 snapToGrid (Use Document Grid Settings for Inter-Line Paragraph Spacing)	47
2.3.1.33 spacing (Spacing Between Lines and Above/Below Paragraph)	47

2.3.1.34	suppressAutoHyphens (Suppress Hyphenation for Paragraph)	49
2.3.1.35	suppressLineNumbers (Suppress Line Numbers for Paragraph)	50
2.3.1.36	suppressOverlap (Prevent Text Frames From Overlapping)	50
2.3.1.37	tab (Custom Tab Stop)	50
2.3.1.38	tabs (Set of Custom Tab Stops)	51
2.3.1.39	textAlignment (Vertical Character Alignment on Line)	51
2.3.1.40	textboxTightWrap (Allow Surrounding Paragraphs to Tight Wrap to Text Box Contents)	51
2.3.1.41	textDirection (Paragraph Text Flow Direction)	52
2.3.1.42	top (Paragraph Border Above Identical Paragraphs)	52
2.3.1.43	topLinePunct (Compress Punctuation at Start of a Line)	52
2.3.1.44	widowControl (Allow First/Last Line to Display on a Separate Page)	52
2.3.1.45	wordWrap (Allow Line Breaking At Character Level)	53
2.3.2	Run	53
2.3.2.1	b (Bold)	53
2.3.2.2	bCs (Complex Script Bold)	53
2.3.2.3	bdr (Text Border)	54
2.3.2.4	caps (Display All Characters As Capital Letters)	54
2.3.2.5	color (Run Content Color)	54
2.3.2.6	cs (Use Complex Script Formatting on Run)	55
2.3.2.7	dstrike (Double Strikethrough)	56
2.3.2.8	eastAsianLayout (East Asian Typography Settings)	56
2.3.2.9	effect (Animated Text Effect)	57
2.3.2.10	em (Emphasis Mark)	57
2.3.2.11	emboss (Embossing)	57
2.3.2.12	fitText (Manual Run Width)	58
2.3.2.13	highlight (Text Highlighting)	58
2.3.2.14	i (Italics)	58
2.3.2.15	iCs (Complex Script Italics)	59
2.3.2.16	imprint (Imprinting)	59
2.3.2.17	kern (Font Kerning)	59
2.3.2.18	lang (Languages for Run Content)	60
2.3.2.19	noProof (Do Not Check Spelling or Grammar)	60
2.3.2.20	oMath (Office Open XML Math)	60
2.3.2.21	outline (Display Character Outline)	61
2.3.2.22	position (Vertically Raised or Lowered Text)	61
2.3.2.23	r (Text Run)	62
2.3.2.24	rFonts (Run Fonts)	62
2.3.2.25	rPr (Run Properties)	65
2.3.2.26	rPr (Previous Run Properties)	65
2.3.2.27	rStyle (Referenced Character Style)	65
2.3.2.28	rtl (Right To Left Text)	66
2.3.2.29	shadow (Shadow)	66
2.3.2.30	shd (Run Shading)	66
2.3.2.31	smallCaps (Small Caps)	66
2.3.2.32	snapToGrid (Use Document Grid Settings For Inter-Character Spacing)	67
2.3.2.33	spacing (Character Spacing Adjustment)	67
2.3.2.34	specVanish (Paragraph Mark Is Always Hidden)	67
2.3.2.35	strike (Single Strikethrough)	68
2.3.2.36	sz (Font Size)	68
2.3.2.37	szCs (Complex Script Font Size)	68
2.3.2.38	u (Underline)	68
2.3.2.39	vanish (Hidden Text)	70
2.3.2.40	vertAlign (Subscript/Superscript Text)	70

2.3.2.41	w (Expanded/Compressed Text).....	70
2.3.2.42	webHidden (Web Hidden Text)	71
2.3.3	Run Content.....	71
2.3.3.1	br (Break)	71
2.3.3.2	control (Floating Embedded Control)	72
2.3.3.3	control (Inline Embedded Control)	72
2.3.3.4	cr (Carriage Return)	73
2.3.3.5	dayLong (Date Block - Long Day Format).....	73
2.3.3.6	dayShort (Date Block - Short Day Format).....	73
2.3.3.7	delText (Deleted Text)	73
2.3.3.8	dirty (Invalidated Field Cache)	73
2.3.3.9	drawing (DrawingML Object)	74
2.3.3.10	hps (Phonetic Guide Text Font Size)	74
2.3.3.11	hpsBaseText (Phonetic Guide Base Text Font Size)	74
2.3.3.12	hpsRaise (Distance Between Phonetic Guide Text and Phonetic Guide Base Text)	74
2.3.3.13	lastRenderedPageBreak (Position of Last Calculated Page Break)	74
2.3.3.14	lid (Language ID for Phonetic Guide).....	74
2.3.3.15	monthLong (Date Block - Long Month Format).....	75
2.3.3.16	monthShort (Date Block - Short Month Format).....	75
2.3.3.17	movie (Embedded Video)	75
2.3.3.18	noBreakHyphen (Non Breaking Hyphen Character)	75
2.3.3.19	object (Inline Embedded Object).....	76
2.3.3.20	pgNum (Page Number Block)	76
2.3.3.21	pict (VML Object).....	76
2.3.3.22	ptab (Absolute Position Tab Character)	76
2.3.3.23	rt (Phonetic Guide Text)	77
2.3.3.24	ruby (Phonetic Guide).....	77
2.3.3.25	rubyAlign (Phonetic Guide Text Alignment)	77
2.3.3.26	rubyBase (Phonetic Guide Base Text).....	77
2.3.3.27	rubyPr (Phonetic Guide Properties)	77
2.3.3.28	softHyphen (Optional Hyphen Character)	77
2.3.3.29	sym (Symbol Character).....	78
2.3.3.30	t (Text)	78
2.3.3.31	tab (Tab Character).....	78
2.3.3.32	yearLong (Date Block - Long Year Format)	79
2.3.3.33	yearShort (Date Block - Short Year Format)	79
2.4	Tables.....	79
2.4.1	bidVisual (Visually Right to Left Table)	79
2.4.2	bottom (Table Cell Bottom Margin Exception).....	79
2.4.3	bottom (Table Cell Bottom Border).....	80
2.4.4	bottom (Table Bottom Border).....	80
2.4.5	bottom (Table Cell Bottom Margin Default).....	80
2.4.6	cantSplit (Table Row Cannot Break Across Pages)	81
2.4.7	cnfStyle (Table Cell Conditional Formatting)	81
2.4.8	cnfStyle (Table Row Conditional Formatting).....	81
2.4.9	divId (Associated HTML div ID)	81
2.4.10	gridAfter (Grid Columns After Last Cell)	82
2.4.11	gridBefore (Grid Columns Before First Cell)	82
2.4.12	gridCol (Grid Column Definition)	82
2.4.13	gridSpan (Grid Columns Spanned by Current Table Cell)	83
2.4.14	hidden (Hidden Table Row Marker)	83
2.4.15	hideMark (Ignore End Of Cell Marker In Row Height Calculation)	83

2.4.16hMerge (Horizontally Merged Cell)	84
2.4.17insideH (Table Inside Horizontal Edges Border)	84
2.4.18insideH (Table Cell Inside Horizontal Edges Border)	84
2.4.19insideV (Table Cell Inside Vertical Edges Border)	85
2.4.20insideV (Table Inside Vertical Edges Border)	85
2.4.21jc (Table Alignment Exception)	85
2.4.22jc (Table Row Alignment)	85
2.4.23jc (Table Alignment)	86
2.4.24left (Table Cell Left Border)	86
2.4.25left (Table Cell Left Margin Exception)	86
2.4.26left (Table Cell Left Margin Default)	87
2.4.27left (Table Left Border)	87
2.4.28noWrap (Don't Wrap Cell Content)	87
2.4.29right (Table Cell Right Margin Default)	87
2.4.30right (Table Cell Right Border)	88
2.4.31right (Table Cell Right Margin Exception)	88
2.4.32right (Table Right Border)	88
2.4.33shd (Table Cell Shading)	88
2.4.34shd (Table Shading Exception)	89
2.4.35shd (Table Shading)	89
2.4.36tbl (Table)	90
2.4.37tblBorders (Table Borders Exceptions)	90
2.4.38tblBorders (Table Borders)	90
2.4.39tblCellMar (Table Cell Margin Defaults)	90
2.4.40tblCellMar (Table Cell Margin Exceptions)	90
2.4.41tblCellSpacing (Table Cell Spacing Exception)	90
2.4.42tblCellSpacing (Table Row Cell Spacing)	91
2.4.43tblCellSpacing (Table Cell Spacing Default)	91
2.4.44tblGrid (Table Grid)	91
2.4.45tblGrid (Previous Table Grid)	91
2.4.46tblHeader (Repeat Table Row on Every New Page)	92
2.4.47tblInd (Table Indent from Leading Margin Exception)	92
2.4.48tblInd (Table Indent from Leading Margin)	92
2.4.49tblLayout (Table Layout)	92
2.4.50tblLayout (Table Layout Exception)	93
2.4.51tblLook (Table Style Conditional Formatting Settings)	93
2.4.52tblLook (Table Style Conditional Formatting Settings Exception)	93
2.4.53tblOverlap (Floating Table Allows Other Tables to Overlap)	94
2.4.54tblpPr (Floating Table Positioning)	94
2.4.55tblPr (Table Properties)	96
2.4.56tblPr (Previous Table Properties)	96
2.4.57tblPrEx (Table-Level Property Exceptions)	97
2.4.58tblPrEx (Previous Table-Level Property Exceptions)	97
2.4.59tblStyle (Referenced Table Style)	97
2.4.60tblW (Preferred Table Width Exception)	97
2.4.61tblW (Preferred Table Width)	97
2.4.62tc (Table Cell)	98
2.4.63tcBorders (Table Cell Borders)	98
2.4.64tcFitText (Fit Text Within Cell)	100
2.4.65tcMar (Single Table Cell Margins)	100
2.4.66tcPr (Previous Table Cell Properties)	100
2.4.67tcPr (Table Cell Properties)	100
2.4.68tcW (Preferred Table Cell Width)	100

2.4.69textDirection (Table Cell Text Flow Direction).....	101
2.4.70tl2br (Table Cell Top Left to Bottom Right Diagonal Border)	101
2.4.71top (Table Top Border)	101
2.4.72top (Table Cell Top Margin Default)	101
2.4.73top (Table Cell Top Margin Exception)	101
2.4.74top (Table Cell Top Border)	102
2.4.75tr (Table Row)	102
2.4.76tr2bl (Table Cell Top Right to Bottom Left Diagonal Border)	103
2.4.77trHeight (Table Row Height).....	103
2.4.78trPr (Table Row Properties)	104
2.4.79trPr (Previous Table Row Properties)	104
2.4.80vAlign (Table Cell Vertical Alignment)	104
2.4.81vMerge (Vertically Merged Cell).....	105
2.4.82wAfter (Preferred Width After Table Row)	105
2.4.83wBefore (Preferred Width Before Table Row)	105
2.5 Custom Markup	106
2.5.1 Custom XML and Smart Tags.....	106
2.5.1.1 attr (Custom XML Attribute).....	106
2.5.1.2 attr (Smart Tag Property)	107
2.5.1.3 customXml (Cell-Level Custom XML Element).....	107
2.5.1.4 customXml (Row-Level Custom XML Element)	107
2.5.1.5 customXml (Inline-Level Custom XML Element)	107
2.5.1.6 customXml (Block-Level Custom XML Element).....	108
2.5.1.7 customXmlPr (Custom XML Element Properties).....	108
2.5.1.8 placeholder (Custom XML Element Placeholder Text)	108
2.5.1.9 smartTag (Inline-Level Smart Tag).....	108
2.5.1.10 smartTagPr (Smart Tag Properties)	108
2.5.2 Structured Document Tags.....	108
2.5.2.1 alias (Friendly Name)	109
2.5.2.2 bibliography (Bibliography Structured Document Tag)	109
2.5.2.3 calendar (Date Picker Calendar Type)	109
2.5.2.4 citation (Citation Structured Document Tag)	109
2.5.2.5 comboBox (Combo Box Structured Document Tag).....	109
2.5.2.6 dataBinding (XML Mapping).....	110
2.5.2.7 date (Date Structured Document Tag)	111
2.5.2.8 dateFormat (Date Display Mask).....	111
2.5.2.9 docPart (Document Part Reference)	112
2.5.2.10 docPartCategory (Document Part Category Filter).....	112
2.5.2.11 docPartGallery (Document Part Gallery Filter).....	112
2.5.2.12 docPartList (Document Part Gallery Structured Document Tag)	112
2.5.2.13 docPartObj (Built-In Document Part Structured Document Tag)	113
2.5.2.14 docPartUnique (Built-In Document Part)	113
2.5.2.15 dropDownList (Drop-Down List Structured Document Tag).....	113
2.5.2.16 equation (Equation Structured Document Tag)	113
2.5.2.17 group (Group Structured Document Tag)	114
2.5.2.18 id (Unique ID).....	114
2.5.2.19 lid (Date Picker Language ID).....	114
2.5.2.20 listItem (Combo Box List Item)	114
2.5.2.21 listItem (Drop-Down List Item).....	115
2.5.2.22 lock (Locking Setting).....	115
2.5.2.23 picture (Picture Structured Document Tag)	115
2.5.2.24 placeholder (Structured Document Tag Placeholder Text)	116

2.5.2.25	richText (Rich Text Structured Document Tag)	116
2.5.2.26	rPr (Run Properties For Structured Document Tag Contents)	116
2.5.2.27	rPr (Structured Document Tag End Character Run Properties)	116
2.5.2.28	sdt (Cell-Level Structured Document Tag)	116
2.5.2.29	sdt (Inline-Level Structured Document Tag)	116
2.5.2.30	sdt (Block-Level Structured Document Tag)	116
2.5.2.31	sdt (Row-Level Structured Document Tag)	116
2.5.2.32	sdtContent (Block-Level Structured Document Tag Content)	117
2.5.2.33	sdtContent (Cell-Level Structured Document Tag Content)	117
2.5.2.34	sdtContent (Row-Level Structured Document Tag Content)	117
2.5.2.35	sdtContent (Inline-Level Structured Document Tag Content)	117
2.5.2.36	sdtEndPr (Structured Document Tag End Character Properties)	117
2.5.2.37	sdtPr (Structured Document Tag Properties)	117
2.5.2.38	showingPlcHdr (Current Contents Are Placeholder Text)	117
2.5.2.39	storeMappedDataAs (Custom XML Data Date Storage Format)	118
2.5.2.40	tag (Programmatic Tag)	118
2.5.2.41	temporary (Remove Structured Document Tag When Contents Are Edited)	118
2.5.2.42	text (Plain Text Structured Document Tag)	118
2.6	Sections	119
2.6.1	bidi (Right to Left Section Layout)	119
2.6.2	bottom (Bottom Border)	119
2.6.3	col (Single Column Definition)	119
2.6.4	cols (Column Definitions)	120
2.6.5	docGrid (Document Grid)	121
2.6.6	formProt (Only Allow Editing of Form Fields)	122
2.6.7	left (Left Border)	122
2.6.8	InNumType (Line Numbering Settings)	122
2.6.9	paperSrc (Paper Source Information)	123
2.6.10	pgBorders (Page Borders)	123
2.6.11	pgMar (Page Margins)	124
2.6.12	pgNumType (Page Numbering Settings)	125
2.6.13	pgSz (Page Size)	125
2.6.14	printerSettings (Reference to Printer Settings Data)	126
2.6.15	right (Right Border)	126
2.6.16	rtlGutter (Gutter on Right Side of Page)	127
2.6.17	sectPr (Previous Section Properties)	127
2.6.18	sectPr (Document Final Section Properties)	128
2.6.19	sectPr (Section Properties)	128
2.6.20	textDirection (Text Flow Direction)	128
2.6.21	top (Top Border)	128
2.6.22	type (Section Type)	129
2.6.23	vAlign (Vertical Text Alignment on Page)	129
2.7	Styles	129
2.7.1	Style Inheritance	130
2.7.2	Style Hierarchy	130
2.7.3	General Style Properties	131
2.7.3.1	aliases (Alternate Style Names)	131
2.7.3.2	autoRedefine (Automatically Merge User Formatting Into Style Definition)	131
2.7.3.3	basedOn (Parent Style ID)	131
2.7.3.4	hidden (Hide Style From User Interface)	132
2.7.3.5	latentStyles (Latent Style Information)	132
2.7.3.6	link (Linked Style Reference)	133

2.7.3.7	locked (Style Cannot Be Applied)	134
2.7.3.8	lsdException (Latent Style Exception)	134
2.7.3.9	name (Primary Style Name)	135
2.7.3.10	next (Style For Next Paragraph)	135
2.7.3.11	personal (E-Mail Message Text Style)	135
2.7.3.12	personalCompose (E-Mail Message Composition Style)	135
2.7.3.13	personalReply (E-Mail Message Reply Style)	136
2.7.3.14	qFormat (Primary Style)	136
2.7.3.15	rsid (Revision Identifier for Style Definition)	136
2.7.3.16	semiHidden (Hide Style From Main User Interface)	137
2.7.3.17	style (Style Definition)	137
2.7.3.18	styles (Style Definitions)	138
2.7.3.19	uiPriority (Optional User Interface Sorting Order)	138
2.7.3.20	unhideWhenUsed (Remove Semi-Hidden Property When Style Is Used)	139
2.7.4	Document Defaults	139
2.7.4.1	docDefaults (Document Default Paragraph and Run Properties)	139
2.7.4.2	pPr (Paragraph Properties)	139
2.7.4.3	pPrDefault (Default Paragraph Properties)	139
2.7.4.4	rPr (Run Properties)	139
2.7.4.5	rPrDefault (Default Run Properties)	140
2.7.5	Table Styles	140
2.7.5.1	pPr (Table Style Conditional Formatting Paragraph Properties)	141
2.7.5.2	rPr (Table Style Conditional Formatting Run Properties)	142
2.7.5.3	tblPr (Table Style Conditional Formatting Table Properties)	142
2.7.5.4	tblPr (Style Table Properties)	142
2.7.5.5	tblStyleColBandSize (Number of Columns in Column Band)	142
2.7.5.6	tblStylePr (Style Conditional Table Formatting Properties)	142
2.7.5.7	tblStyleRowBandSize (Number of Rows in Row Band)	143
2.7.5.8	tcPr (Style Table Cell Properties)	143
2.7.5.9	tcPr (Table Style Conditional Formatting Table Cell Properties)	144
2.7.5.10	trPr (Table Style Conditional Formatting Table Row Properties)	144
2.7.5.11	trPr (Style Table Row Properties)	144
2.7.6	Numbering Styles	144
2.7.7	Paragraph Styles	144
2.7.7.1	Numbering in Paragraph Styles	144
2.7.7.2	pPr (Style Paragraph Properties)	145
2.7.8	Run (Character) Styles	145
2.7.8.1	rPr (Run Properties)	145
2.8	Fonts	145
2.8.1	Font Embedding	145
2.8.2	Elements	145
2.8.2.1	altName (Alternate Names for Font)	146
2.8.2.2	charset (Character Set Supported By Font)	146
2.8.2.3	embedBold (Bold Style Font Style Embedding)	147
2.8.2.4	embedBoldItalic (Bold Italic Font Style Embedding)	147
2.8.2.5	embedItalic (Italic Font Style Embedding)	147
2.8.2.6	embedRegular (Regular Font Style Embedding)	148
2.8.2.7	embedSystemFonts (Embed Common System Fonts)	148
2.8.2.8	embedTrueTypeFonts (Embed TrueType Fonts)	148
2.8.2.9	family (Font Family)	148
2.8.2.10	font (Properties for a Single Font)	148
2.8.2.11	fonts (Font Table Root Element)	149

2.8.2.12 notTrueType (Raster or Vector Font)	149
2.8.2.13 panose1 (Pansose-1 Typeface Classification Number)	149
2.8.2.14 pitch (Font Pitch)	149
2.8.2.15 saveSubsetFonts (Subset Fonts When Embedding)	149
2.8.2.16 sig (Supported Unicode Subranges and Code Pages)	150
2.9 Numbering.....	151
2.9.1 abstractNum (Abstract Numbering Definition)	152
2.9.2 abstractNumId (Abstract Numbering Definition Reference)	152
2.9.3 ilvl (Numbering Level Reference)	152
2.9.4 isLgl (Display All Levels Using Arabic Numerals)	152
2.9.5 legacy (Legacy Numbering Level Properties)	153
2.9.6 lvl (Numbering Level Override Definition)	153
2.9.7 lvl (Numbering Level Definition)	154
2.9.8 lvlJc (Justification)	155
2.9.9 lvlOverride (Numbering Level Definition Override)	155
2.9.10lvlPicBulletId (Picture Numbering Symbol Definition Reference)	155
2.9.11lvlRestart (Restart Numbering Level Symbol)	156
2.9.12lvlText (Numbering Level Text)	156
2.9.13multiLevelType (Abstract Numbering Definition Type)	156
2.9.14name (Abstract Numbering Definition Name)	157
2.9.15nsid (Abstract Numbering Definition Identifier)	157
2.9.16num (Numbering Definition Instance)	157
2.9.17numbering (Numbering Definitions)	157
2.9.18numFmt (Numbering Format)	157
2.9.19numId (Numbering Definition Instance Reference)	158
2.9.20numIdMacAtCleanup (Last Reviewed Abstract Numbering Definition)	158
2.9.21numPicBullet (Picture Numbering Symbol Definition)	158
2.9.22numStyleLink (Numbering Style Reference)	159
2.9.23pict (Picture Numbering Symbol Properties)	159
2.9.24pPr (Numbering Level Associated Paragraph Properties)	159
2.9.25pStyle (Paragraph Style's Associated Numbering Level)	159
2.9.26rPr (Numbering Symbol Run Properties)	159
2.9.27start (Starting Value)	159
2.9.28startOverride (Numbering Level Starting Value Override)	160
2.9.29styleLink (Numbering Style Definition)	160
2.9.30suff (Content Between Numbering Symbol and Paragraph Text)	160
2.9.31tmpl (Numbering Template Code)	160
2.10Headers and Footers	160
2.10.1evenAndOddHeaders (Different Even/Odd Page Headers and Footers)	161
2.10.2footerReference (Footer Reference)	161
2.10.3ftr (Footer)	162
2.10.4hdr (Header)	162
2.10.5headerReference (Header Reference)	162
2.10.6titlePg (Different First Page Headers and Footers)	163
2.11Footnotes and Endnotes	163
2.11.1continuationSeparator (Continuation Separator Mark)	164
2.11.2endnote (Endnote Content)	164
2.11.3endnote (Special Endnote List)	164
2.11.4endnotePr (Document-Wide Endnote Properties)	164
2.11.5endnotePr (Section-Wide Endnote Properties)	164
2.11.6endnoteRef (Endnote Reference Mark)	165

2.11.7endnoteReference (Endnote Reference).....	165
2.11.8endnotes (Document Endnotes)	165
2.11.9footnote (Special Footnote List)	165
2.11.10 footnote (Footnote Content)	166
2.11.11 footnotePr (Document-Wide Footnote Properties).....	166
2.11.12 footnotePr (Section-Wide Footnote Properties).....	166
2.11.13 footnoteRef (Footnote Reference Mark)	166
2.11.14 footnoteReference (Footnote Reference)	166
2.11.15 footnotes (Document Footnotes)	166
2.11.16 noEndnote (Suppress Endnotes In Document)	166
2.11.17 numFmt (Footnote Numbering Format)	167
2.11.18 numFmt (Endnote Numbering Format)	167
2.11.19 numRestart (Footnote and Endnote Numbering Restart Location).....	167
2.11.20 numStart (Footnote and Endnote Numbering Starting Value)	167
2.11.21 pos (Footnote Placement).....	167
2.11.22 pos (Endnote Placement).....	168
2.11.23 separator (Footnote/Endnote Separator Mark)	168
2.12Glossary Document.....	168
2.12.1behavior (Entry Insertion Behavior)	168
2.12.2behaviors (Entry Insertion Behaviors)	168
2.12.3category (Entry Categorization).....	169
2.12.4description (Description for Entry)	169
2.12.5docPart (Glossary Document Entry)	169
2.12.6docPartBody (Contents of Glossary Document Entry)	169
2.12.7docPartPr (Glossary Document Entry Properties)	169
2.12.8docParts (List of Glossary Document Entries)	169
2.12.9gallery (Gallery Associated With Entry)	169
2.12.10 glossaryDocument (Glossary Document Root Element).....	170
2.12.11 guid (Entry ID)	170
2.12.12 name (Category Associated With Entry)	170
2.12.13 name (Entry Name)	170
2.12.14 style (Associated Paragraph Style Name).....	171
2.12.15 type (Entry Type).....	171
2.12.16 types (Entry Types).....	171
2.13Annotations.....	171
2.13.1Inline Annotations	172
2.13.2"Cross Structure" Annotations	172
2.13.3Property Annotations	172
2.13.4Comments	172
2.13.4.1 annotationRef (Comment Information Block).....	172
2.13.4.2 comment (Comment Content)	172
2.13.4.3 commentRangeEnd (§2.13.4.3);	173
2.13.4.4 commentRangeStart (§2.13.4.4);	174
2.13.4.5 commentReference (Comment Content Reference Mark)	174
2.13.4.6 comments (Comments Collection)	174
2.13.5Revisions.....	174
2.13.5.1 cellDel (Table Cell Deletion).....	175
2.13.5.2 cellIns (Table Cell Insertion).....	175
2.13.5.3 cellMerge (Vertically Merged/Split Table Cells)	175
2.13.5.4 customXmlDelRangeEnd (Custom XML Markup Deletion End).....	176
2.13.5.5 customXmlDelRangeStart (Custom XML Markup Deletion Start)	176
2.13.5.6 customXmlInsRangeEnd (Custom XML Markup Insertion End).....	177

2.13.5.7 customXmlInsRangeStart (§2.13.5.7);	177
2.13.5.8 customXmlMoveFromRangeEnd (Custom XML Markup Move Source End)	178
2.13.5.9 customXmlMoveFromRangeStart (Custom XML Markup Move Source Start)	178
2.13.5.10 customXmlMoveToRangeEnd (Custom XML Markup Move Destination Location End)	179
2.13.5.11 customXmlMoveToRangeStart (Custom XML Markup Move Destination Location Start)	179
2.13.5.12 del (Deleted Run Content)	180
2.13.5.13 del (Deleted Paragraph)	180
2.13.5.14 del (Deleted Table Row)	180
2.13.5.15 del (Deleted Math Control Character)	180
2.13.5.16 ins (Inserted Table Row)	181
2.13.5.17 ins (Inserted Math Control Character)	181
2.13.5.18 ins (Inserted Paragraph)	181
2.13.5.19 ins (Inserted Numbering Properties)	181
2.13.5.20 ins (Inserted Run Content)	181
2.13.5.21 moveFrom (Move Source Run Content)	182
2.13.5.22 moveFrom (Move Source Paragraph)	182
2.13.5.23 moveFromRangeEnd (Move Source Location Container - End)	182
2.13.5.24 moveFromRangeStart (Move Source Location Container - Start)	183
2.13.5.25 moveTo (Move Destination Paragraph)	184
2.13.5.26 moveTo (Move Destination Run Content)	184
2.13.5.27 moveToRangeEnd (Move Destination Location Container - End)	185
2.13.5.28 moveToRangeStart (Move Destination Location Container - Start)	185
2.13.5.29 numberingChange (Previous Numbering Field Properties)	186
2.13.5.30 numberingChange (Previous Paragraph Numbering Properties)	187
2.13.5.31 pPrChange (Revision Information for Paragraph Properties)	189
2.13.5.32 rPrChange (Revision Information for Run Properties)	189
2.13.5.33 rPrChange (Revision Information for Run Properties on the Paragraph Mark)	190
2.13.5.34 sectPrChange (Revision Information for Section Properties)	190
2.13.5.35 tblGridChange (Revision Information for Table Grid Column Definitions)	190
2.13.5.36 tblPrChange (Revision Information for Table Properties)	191
2.13.5.37 tblPrExChange (Revision Information for Table-Level Property Exceptions)	191
2.13.5.38 tcPrChange (Revision Information for Table Cell Properties)	191
2.13.5.39 trPrChange (Revision Information for Table Row Properties)	192
2.13.6 Bookmarks	192
2.13.6.1 bookmarkEnd (Bookmark End)	192
2.13.6.2 bookmarkStart (Bookmark Start)	192
2.13.7 Range Permissions	193
2.13.7.1 permEnd (Range Permission End)	193
2.13.7.2 permStart (Range Permission Start)	193
2.13.8 Spelling & Grammar	195
2.13.8.1 proofErr (Proofing Error Anchor)	195
2.14 Mail Merge	195
2.14.1 active (Record Is Included in Mail Merge)	196
2.14.2 activeRecord (Record Currently Displayed In Merged Document)	196
2.14.3 addressFieldName (Column Containing E-mail Address)	197
2.14.4 checkErrors (Mail Merge Error Reporting Setting)	197
2.14.5 colDelim (Column Delimiter for Data Source)	197
2.14.6 column (Index of Column Containing Unique Values for Record)	198
2.14.7 column (Index of Column Being Mapped)	198
2.14.8 connectString (Data Source Connection String)	198
2.14.9 dataSource (Data Source File Path)	199
2.14.10 dataType (Data Source Type)	199

2.14.11	destination (Merged Document Destination)	199
2.14.12	doNotSuppressBlankLines (Remove Blank Lines from Merged Documents)	200
2.14.13	dynamicAddress (Use Country-Based Address Field Ordering)	200
2.14.14	fHdr (First Row of Data Source Contains Column Names)	200
2.14.15	fieldMapData (External Data Source to Merge Field Mapping)	201
2.14.16	headerSource (Header Definition File Path)	201
2.14.17	lid (Merge Field Name Language ID)	201
2.14.18	linkToQuery (Query Contains Link to External Query File)	201
2.14.19	mailAsAttachment (Merged Document To E-Mail Attachment)	201
2.14.20	mailMerge (Mail Merge Settings)	202
2.14.21	mailSubject (Merged E-mail or Fax Subject Line)	202
2.14.22	mainDocumentType (Source Document Type)	202
2.14.23	mappedName (Predefined Merge Field Name)	203
2.14.24	name (Data Source Name for Column)	203
2.14.25	odso (Office Data Source Object Settings)	203
2.14.26	query (Query For Data Source Records To Merge)	203
2.14.27	recipientData (Reference to Inclusion/Exclusion Data for Data Source)	204
2.14.28	recipientData (Data About Single Data Source Record)	204
2.14.29	recipients (Inclusion/Exclusion Data for Data Source)	204
2.14.30	src (ODSO Data Source File Path)	205
2.14.31	table (Data Source Table Name)	205
2.14.32	type (Merge Field Mapping)	205
2.14.33	type (ODSO Data Source Type)	205
2.14.34	udl (UDL Connection String)	206
2.14.35	uniqueTag (Unique Value for Record)	206
2.14.36	viewMergedData (View Merged Data Within Document)	206
2.15	Settings	206
2.15.1	Document Settings	207
2.15.1.1	activeWritingStyle (Grammar Checking Settings)	207
2.15.1.2	alignBordersAndEdges (Align Paragraph and Table Borders with Page Border)	207
2.15.1.3	alwaysMergeEmptyNamespace (Do Not Mark Custom XML Elements With No Namespace As Invalid)	208
2.15.1.4	alwaysShowPlaceholderText (Use Custom XML Element Names as Default Placeholder Text)	208
2.15.1.5	attachedSchema (Attached Custom XML Schema)	208
2.15.1.6	attachedTemplate (Attached Document Template)	209
2.15.1.7	autoCaption (Single Automatic Captioning Setting)	209
2.15.1.8	autoCaptions (Automatic Captioning Settings)	209
2.15.1.9	autoFormatOverride (Allow Automatic Formatting to Override Formatting Protection Settings)	209
2.15.1.10	autoHyphenation (Automatically Hyphenate Document Contents When Displayed)	209
2.15.1.11	bookFoldPrinting (Book Fold Printing)	210
2.15.1.12	bookFoldPrintingSheets (Number of Pages Per Booklet)	210
2.15.1.13	bookFoldRevPrinting (Reverse Book Fold Printing)	210
2.15.1.14	bordersDoNotSurroundFooter (Page Border Excludes Footer)	211
2.15.1.15	bordersDoNotSurroundHeader (Page Border Excludes Header)	211
2.15.1.16	caption (Single Caption Type Definition)	211
2.15.1.17	captions (Caption Settings)	213
2.15.1.18	characterSpacingControl (Character-Level Whitespace Compression)	214
2.15.1.19	clickAndTypeStyle (Paragraph Style Applied to Automatically Generated Paragraphs)	214
2.15.1.20	clrSchemeMapping (Theme Color Mappings)	214
2.15.1.21	consecutiveHyphenLimit (Maximum Number of Consecutively Hyphenated Lines)	216
2.15.1.22	decimalSymbol (Radix Point for Field Code Evaluation)	216
2.15.1.23	defaultTableStyle (Default Table Style for Newly Inserted Tables)	216
2.15.1.24	defaultTabStop (Distance Between Automatic Tab Stops)	217

2.15.1.25displayBackgroundShape (Display Background Objects When Displaying Document)	217
2.15.1.26displayHorizontalDrawingGridEvery (Distance between Horizontal Gridlines).....	217
2.15.1.27displayVerticalDrawingGridEvery (Distance between Vertical Gridlines)	217
2.15.1.28documentProtection (Document Editing Restrictions)	218
2.15.1.29documentType (Document Classification)	225
2.15.1.30docVar (Single Document Variable).....	226
2.15.1.31docVars (Document Variables)	226
2.15.1.32doNotAutoCompressPictures (Do Not Automatically Compress Images)	226
2.15.1.33doNotDemarcateInvalidXml (Do Not Show Visual Indicator For Invalid Custom XML Markup)	226
2.15.1.34doNotDisplayPageBoundaries (Do Not Display Visual Boundary For Header/Footer or Between Pages)	226
2.15.1.35doNotEmbedSmartTags (Remove Smart Tags When Saving).....	227
2.15.1.36doNotHyphenateCaps (Do Not Hyphenate Words in ALL CAPITAL LETTERS).....	227
2.15.1.37doNotIncludeSubdocsInStats (Do Not Include Content in Text Boxes, Footnotes, and Endnotes in Document Statistics)	227
2.15.1.38doNotShadeFormData (Do Not Show Visual Indicator For Form Fields).....	227
2.15.1.39doNotTrackFormatting (Do Not Track Formatting Revisions When Tracking Revisions)	227
2.15.1.40doNotTrackMoves (Do Not Use Move Syntax When Tracking Revisions)	228
2.15.1.41doNotUseMarginsForDrawingGridOrigin (Do Not Use Margins for Drawing Grid Origin)	228
2.15.1.42doNotValidateAgainstSchema (Do Not Validate Custom XML Markup Against Schemas)	228
2.15.1.43drawingGridHorizontalOrigin (Drawing Grid Horizontal Origin Point)	228
2.15.1.44drawingGridHorizontalSpacing (Drawing Grid Horizontal Grid Unit Size)	229
2.15.1.45drawingGridVerticalOrigin (Drawing Grid Vertical Origin Point)	229
2.15.1.46drawingGridVerticalSpacing (Drawing Grid Vertical Grid Unit Size).....	229
2.15.1.47forceUpgrade (Upgrade Document on Open)	229
2.15.1.48formsDesign (Structured Document Tag Placeholder Text Should be Resaved).....	230
2.15.1.49gutterAtTop (Position Gutter At Top of Page).....	230
2.15.1.50hdrShapeDefaults (Default Properties for VML Objects in Header and Footer)	230
2.15.1.51hideGrammaticalErrors (Do Not Display Visual Indication of Grammatical Errors)	230
2.15.1.52hideSpellingErrors (Do Not Display Visual Indication of Spelling Errors).....	230
2.15.1.53hyphenationZone (Hyphenation Zone)	231
2.15.1.54ignoreMixedContent (Ignore Mixed Content When Validating Custom XML Markup)	231
2.15.1.55linkStyles (Automatically Update Styles From Document Template).....	231
2.15.1.56listSeparator (List Separator for Field Code Evaluation).....	231
2.15.1.57mirrorMargins (Mirror Page Margins)	232
2.15.1.58noLineBreaksAfter (Custom Set of Characters Which Cannot End a Line)	232
2.15.1.59noLineBreaksBefore (Custom Set Of Characters Which Cannot Begin A Line).....	232
2.15.1.60noPunctuationKerning (Never Kern Punctuation Characters).....	233
2.15.1.61printFormsData (Only Print Form Field Content)	233
2.15.1.62printFractionalCharacterWidth (Print Fractional Character Widths)	233
2.15.1.63printPostScriptOverText (Print PostScript Codes With Document Text).....	233
2.15.1.64printTwoOnOne (Print Two Pages Per Sheet)	233
2.15.1.65proofState (Spelling and Grammatical Checking State)	234
2.15.1.66readModelInkLockDown (Freeze Document Layout).....	234
2.15.1.67removeDateAndTime (Remove Date and Time from Annotations)	235
2.15.1.68removePersonalInformation (Remove Personal Information from Document Properties).....	235
2.15.1.69revisionView (Visibility of Annotation Types).....	235
2.15.1.70rsid (Single Session Revision Save ID)	236
2.15.1.71rsidRoot (Original Document Revision Save ID).....	237
2.15.1.72rsids (Listing of All Revision Save ID Values)	237
2.15.1.73saveFormsData (Only Save Form Field Content)	237
2.15.1.74saveInvalidXml (Allow Saving Document As XML File When Custom XML Markup Is Invalid).....	237
2.15.1.75savePreviewPicture (Generate Thumbnail For Document On Save)	238
2.15.1.76saveThroughXslt (Custom XSL Transform To Use When Saving As XML File).....	238

2.15.1.77saveXmlDataOnly (Only Save Custom XML Markup).....	238
2.15.1.78settings (Document Settings)	239
2.15.1.79shapeDefaults (Default Properties for VML Objects in Main Document)	239
2.15.1.80showEnvelope (Show E-Mail Message Header)	239
2.15.1.81showXMLTags (Show Visual Indicators for Custom XML Markup Start/End Locations)	239
2.15.1.82smartTagType (Supplementary Smart Tag Information)	239
2.15.1.83strictFirstAndLastChars (Use Strict Kinsoku Rules for Japanese Text)	240
2.15.1.84styleLockQFSet (Prevent Replacement of Styles Part)	240
2.15.1.85styleLockTheme (Prevent Modification of Themes Part)	240
2.15.1.86stylePaneFormatFilter (Suggested Filtering for List of Document Styles)	240
2.15.1.87stylePaneSortMethod (Suggested Sorting for List of Document Styles)	241
2.15.1.88summaryLength (Percentage of Document to Use When Generating Summary)	242
2.15.1.89themeFontLang (Theme Font Languages)	242
2.15.1.90trackRevisions (Track Revisions to Document)	242
2.15.1.91updateFields (Automatically Recalculate Fields on Open)	243
2.15.1.92useXSLTWhenSaving (Save Document as XML File through Custom XSL Transform)	243
2.15.1.93view (Document View Setting)	243
2.15.1.94writeProtection (Write Protection)	243
2.15.1.95zoom (Magnification Setting)	245
2.15.2Web Page Settings.....	245
2.15.2.1 allowPNG (Allow PNG as Graphic Format)	246
2.15.2.2 blockQuote (Data for HTML blockquote Element)	246
2.15.2.3 bodyDiv (Data for HTML body Element).....	246
2.15.2.4 bottom (Bottom Border for HTML div).....	246
2.15.2.5 color (Frameset Splitter Color)	246
2.15.2.6 div (Information About Single HTML div Element).....	247
2.15.2.7 divBdr (Set of Borders for HTML div).....	247
2.15.2.8 divs (Information about HTML div Elements).....	247
2.15.2.9 divsChild (Child div Elements Contained within Current div)	247
2.15.2.10doNotOrganizeInFolder (Do Not Place Supporting Files in Subdirectory)	248
2.15.2.11doNotRelyOnCSS (Do Not Rely on CSS for Font Face Formatting)	248
2.15.2.12doNotSaveAsSingleFile (Recommend Web Page Format over Single File Web Page Format)	248
2.15.2.13doNotUseLongFileNames (Do Not Use File Names Longer than 8.3 Characters)	248
2.15.2.14encoding (Output Encoding When Saving as Web Page)	249
2.15.2.15flatBorders (Frameset Splitter Border Style)	249
2.15.2.16frame (Single Frame Properties).....	249
2.15.2.17frameLayout (Frameset Layout)	249
2.15.2.18frameset (Root Frameset Definition).....	249
2.15.2.19frameset (Nested Frameset Definition).....	250
2.15.2.20framesetSplitbar (Frameset Splitter Properties)	250
2.15.2.21left (Left Border for HTML div)	250
2.15.2.22linkedToFile (Maintain Link to Existing File)	250
2.15.2.23marBottom (Bottom Margin for HTML div).....	250
2.15.2.24marH (Top and Bottom Margin for Frame)	251
2.15.2.25marLeft (Left Margin for HTML div).....	251
2.15.2.26marRight (Right Margin for HTML div)	251
2.15.2.27marTop (Top Margin for HTML div).....	252
2.15.2.28marW (Left and Right Margin for Frame)	252
2.15.2.29name (Frame Name).....	252
2.15.2.30noBorder (Do Not Display Frameset Splitters)	253
2.15.2.31noResizeAllowed (Frame Cannot Be Resized)	253
2.15.2.32optimizeForBrowser (Disable Features Not Supported by Target Web Browser).....	253
2.15.2.33pixelsPerInch (Pixels per Inch for Graphics/Images)	254

2.15.2.34	relyOnVML (Utilize VML When Saving as Web Page)	254
2.15.2.35	right (Right Border for HTML div)	254
2.15.2.36	saveSmartTagsAsXml (Save Smart Tag Data in XML Property Bag)	255
2.15.2.37	scrollbar (Scrollbar Display Option)	255
2.15.2.38	sourceFileName (Source File for Frame)	255
2.15.2.39	sz (Frame Size)	256
2.15.2.40	sz (Nested Frameset Size)	256
2.15.2.41	targetScreenSz (Target Screen Size for Web Page)	256
2.15.2.42	top (Top Border for HTML div)	257
2.15.2.43	w (Frameset Splitter Width)	257
2.15.2.44	webSettings (Web Page Settings)	257
2.15.3	Compatibility Settings	257
2.15.3.1	adjustLineHeightInTable (Add Document Grid Line Pitch To Lines in Table Cells)	257
2.15.3.2	alignTablesRowByRow (Align Table Rows Independently)	258
2.15.3.3	allowSpaceOfSameStyleInTable (Allow Contextual Spacing of Paragraphs in Tables)	258
2.15.3.4	applyBreakingRules (Use Legacy Ethiopic and Amharic Line Breaking Rules)	258
2.15.3.5	autofitToFirstFixedWidthCell (Allow Table Columns To Exceed Preferred Widths of Constituent Cells)	258
2.15.3.6	autoSpaceLikeWord95 (Emulate Word 95 Full-Width Character Spacing)	259
2.15.3.7	balanceSingleByteDoubleByteWidth (Balance Single Byte and Double Byte Characters)	259
2.15.3.8	cachedColBalance (Use Cached Paragraph Information for Column Balancing)	259
2.15.3.9	compat (Compatibility Settings)	260
2.15.3.10	convMailMergeEsc (Treat Backslash Quotation Delimiter as Two Quotation Marks)	260
2.15.3.11	displayHangulFixedWidth (Always Use Fixed Width for Hangul Characters)	260
2.15.3.12	doNotAutofitConstrainedTables (Do Not AutoFit Tables To Fit Next To Wrapped Objects)	260
2.15.3.13	doNotBreakConstrainedForcedTable (Don't Break Table Rows Around Floating Tables)	261
2.15.3.14	doNotBreakWrappedTables (Do Not Allow Floating Tables To Break Across Pages)	261
2.15.3.15	doNotExpandShiftReturn (Don't Justify Lines Ending in Soft Line Break)	261
2.15.3.16	doNotLeaveBackslashAlone (Convert Backslash To Yen Sign When Entered)	262
2.15.3.17	doNotSnapToGridInCell (Do Not Snap to Document Grid in Table Cells with Objects)	262
2.15.3.18	doNotSuppressIndentation (Do Not Ignore Floating Objects When Calculating Paragraph Indentation)	262
2.15.3.19	doNotSuppressParagraphBorders (Do Not Suppress Paragraph Borders Next To Frames)	262
2.15.3.20	doNotUseEastAsianBreakRules (Do Not Compress Compressible Characters When Using Document Grid)	263
2.15.3.21	doNotUseHTMLParagraphAutoSpacing (Use Fixed Paragraph Spacing for HTML Auto Setting)	263
2.15.3.22	doNotUseIndentAsNumberingTabStop (Ignore Hanging Indent When Creating Tab Stop After Numbering)	263
2.15.3.23	doNotVertAlignCellWithSp (Don't Vertically Align Cells Containing Floating Objects)	264
2.15.3.24	doNotVertAlignInTxbx (Ignore Vertical Alignment in Textboxes)	264
2.15.3.25	doNotWrapTextWithPunct (Do Not Allow Hanging Punctuation With Character Grid)	264
2.15.3.26	footnoteLayoutLikeWW8 (Emulate Word 6.x/95/97 Footnote Placement)	264
2.15.3.27	forgetLastTabAlignment (Ignore Width of Last Tab Stop When Aligning Paragraph If It Is Not Left Aligned)	265
2.15.3.28	growAutofit (Allow Tables to AutoFit Into Page Margins)	265
2.15.3.29	layoutRawTableWidth (Ignore Space Before Table When Deciding If Table Should Wrap Floating Object)	265
2.15.3.30	layoutTableRowsApart (Allow Table Rows to Wrap Inline Objects Independently)	266
2.15.3.31	lineWrapLikeWord6 (Emulate Word 6.0 Line Wrapping for East Asian Text)	266
2.15.3.32	mwSmallCaps (Emulate Word 5.x for the Macintosh Small Caps Formatting)	266
2.15.3.33	noColumnBalance (Do Not Balance Text Columns within a Section)	266
2.15.3.34	noExtraLineSpacing (Do Not Center Content on Lines With Exact Line Height)	267
2.15.3.35	noLeading (Do Not Add Leading Between Lines of Text)	267
2.15.3.36	noSpaceRaiseLower (Do Not Increase Line Height for Raised/Lowered Text)	267
2.15.3.37	noTabHangInd (Do Not Create Custom Tab Stop for Hanging Indent)	267
2.15.3.38	printBodyTextBeforeHeader (Print Body Text before Header/Footer Contents)	268
2.15.3.39	printColBlack (Print Colors as Black And White without Dithering)	268
2.15.3.40	selectFldWithFirstOrLastChar (Select Field When First or Last Character Is Selected)	268
2.15.3.41	shapeLayoutLikeWW8 (Emulate Word 97 Text Wrapping Around Floating Objects)	268

2.15.3.42showBreaksInFrames (Display Page/Column Breaks Present in Frames).....	269
2.15.3.43spaceForUL (Add Additional Space Below Baseline For Underlined East Asian Text)	269
2.15.3.44spacingInWholePoints (Only Expand/Condense Text By Whole Points)	269
2.15.3.45splitPgBreakAndParaMark (Always Move Paragraph Mark to Page after a Page Break)	270
2.15.3.46subFontBySize (Increase Priority Of Font Size During Font Substitution)	270
2.15.3.47suppressBottomSpacing (Ignore Exact Line Height for Last Line on Page).....	270
2.15.3.48suppressSpacingAtTopOfPage (Ignore Minimum Line Height for First Line on Page).....	270
2.15.3.49suppressSpBfAfterPgBrk (Do Not Use Space Before On First Line After a Page Break)	271
2.15.3.50suppressTopSpacing (Ignore Minimum and Exact Line Height for First Line on Page).....	271
2.15.3.51suppressTopSpacingWP (Emulate WordPerfect 5.x Line Spacing)	271
2.15.3.52swapBordersFacingPages (Swap Paragraph Borders on Odd Numbered Pages)	272
2.15.3.53truncateFontHeightsLikeWP6 (Emulate WordPerfect 6.x Font Height Calculation)	272
2.15.3.54uiCompat97To2003 (Disable Features Incompatible With Earlier Word Processing Formats).....	272
2.15.3.55ulTrailSpace (Underline All Trailing Spaces)	272
2.15.3.56underlineTabInNumList (Underline Following Character Following Numbering)	273
2.15.3.57useAltKinsokuLineBreakRules (Use Alternate Set of East Asian Line Breaking Rules).....	273
2.15.3.58useAnsiKerningPairs (Use ANSI Kerning Pairs from Fonts).....	273
2.15.3.59useFELayout (Do Not Bypass East Asian/Complex Script Layout Code)	274
2.15.3.60useNormalStyleForList (Do Not Automatically Apply List Paragraph Style To Bulleted/Numbered Text)	274
2.15.3.61usePrinterMetrics (Use Printer Metrics To Display Documents).....	274
2.15.3.62useSingleBorderforContiguousCells (Use Simplified Rules For Table Border Conflicts).....	274
2.15.3.63useWord2002TableStyleRules (Emulate Word 2002 Table Style Rules)	275
2.15.3.64useWord97LineBreakRules (Emulate Word 97 East Asian Line Breaking)	275
2.15.3.65wpJustification (Emulate WordPerfect 6.x Paragraph Justification)	275
2.15.3.66wpSpaceWidth (Space width).....	276
2.15.3.67wrapTrailSpaces (Line Wrap Trailing Spaces)	276
2.16Fields & Hyperlinks	276
2.16.1Syntax	276
2.16.2XML representation.....	278
2.16.3Formulas and expressions	279
2.16.3.1 Constants	279
2.16.3.2 Bookmarks.....	279
2.16.3.3 Operators.....	279
2.16.3.4 Functions	280
2.16.3.5 Table cell references.....	281
2.16.4Field formatting.....	281
2.16.4.1 Date and time formatting.....	281
2.16.4.2 Numeric formatting.....	282
2.16.4.3 General formatting	283
2.16.5Field definitions	287
2.16.5.1 ADDRESSBLOCK	288
2.16.5.2 ADVANCE	288
2.16.5.3 ASK.....	289
2.16.5.4 AUTHOR.....	289
2.16.5.5 AUTONUM.....	289
2.16.5.6 AUTONUMLGL	290
2.16.5.7 AUTONUMOUT.....	290
2.16.5.8 AUTOTEXT.....	291
2.16.5.9 AUTOTEXTLIST	291
2.16.5.10BARCODE	291
2.16.5.11BIBLIOGRAPHY	292
2.16.5.12BIDIOUTLINE	292

2.16.5.13CITATION	292
2.16.5.14COMMENTS	293
2.16.5.15COMPARE	293
2.16.5.16CREATEDATE	293
2.16.5.17DATABASE	294
2.16.5.18DATE	295
2.16.5.19DOCPROPERTY	295
2.16.5.20DOCVARIABLE	297
2.16.5.21EDITTIME	297
2.16.5.22EQ	297
2.16.5.23FILENAME	299
2.16.5.24FILESIZE	299
2.16.5.25FILLIN	299
2.16.5.26FORMCHECKBOX	300
2.16.5.27FORMDROPDOWN	300
2.16.5.28FORMTEXT	300
2.16.5.29GOTOBUTTON	300
2.16.5.30GREETINGLINE	301
2.16.5.31HYPERLINK	301
2.16.5.32IF 302	
2.16.5.33INCLUDEPICTURE	302
2.16.5.34INCLUDETEXT	302
2.16.5.35INDEX	303
2.16.5.36INFO	304
2.16.5.37KEYWORDS	304
2.16.5.38LASTSAVEDBY	304
2.16.5.39LINK	305
2.16.5.40LISTNUM	305
2.16.5.41MACROBUTTON	306
2.16.5.42MERGEFIELD	306
2.16.5.43MERGEREC	307
2.16.5.44MERGESEQ	307
2.16.5.45NEXT	307
2.16.5.46NEXTIF	307
2.16.5.47NOTEREF	307
2.16.5.48NUMCHARS	308
2.16.5.49NUMPAGES	308
2.16.5.50NUMWORDS	308
2.16.5.51PAGE	308
2.16.5.52PAGEREF	309
2.16.5.53PRINT	309
2.16.5.54PRINTDATE	309
2.16.5.55PRIVATE	310
2.16.5.56QUOTE	310
2.16.5.57RD	310
2.16.5.58REF	310
2.16.5.59REVNUM	311
2.16.5.60SAVEDATE	311
2.16.5.61SECTION	311
2.16.5.62SECTIONPAGES	312
2.16.5.63SEQ	312
2.16.5.64SET	312
2.16.5.65SKIPIF	313

2.16.5.66	STYLEREF	313
2.16.5.67	SUBJECT	314
2.16.5.68	SYMBOL	314
2.16.5.69	TA.....	314
2.16.5.70	TC.....	315
2.16.5.71	TEMPLATE.....	315
2.16.5.72	TIME.....	315
2.16.5.73	TITLE	316
2.16.5.74	TOA	316
2.16.5.75	TOC	317
2.16.5.76	USERADDRESS	318
2.16.5.77	USERINITIALS	318
2.16.5.78	USERNAME	318
2.16.5.79	XE.....	318
2.16.6	calcOnExit (Recalculate Fields When Current Field Is Modified).....	319
2.16.7	checkBox (Checkbox Form Field Properties)	319
2.16.8	checked (Checkbox Form Field State)	319
2.16.9	ddlList (Drop-Down List Form Field Properties)	319
2.16.10	default (Default Text Box Form Field String).....	320
2.16.11	default (Default Drop-Down List Item Index).....	320
2.16.12	default (Default Checkbox Form Field State)	320
2.16.13	dellnstrText (Deleted Field Code)	320
2.16.14	enabled (Form Field Enabled)	321
2.16.15	entryMacro (Script Function to Execute on Form Field Entry).....	321
2.16.16	exitMacro (Script Function to Execute on Form Field Exit).....	321
2.16.17	ffData (Form Field Properties)	321
2.16.18	fldChar (Complex Field Character)	322
2.16.19	fldData (Custom Field Data)	322
2.16.20	fldData (Custom Field Data)	323
2.16.21	fldSimple (Simple Field).....	323
2.16.22	format (Text Box Form Field Formatting)	323
2.16.23	helpText (Associated Help Text).....	324
2.16.24	hyperlink (Hyperlink).....	324
2.16.25	instrText (Field Code)	326
2.16.26	listEntry (Drop-Down List Entry)	326
2.16.27	maxLength (Text Box Form Field Maximum Length)	326
2.16.28	name (Form Field Name)	326
2.16.29	result (Drop-Down List Selection)	327
2.16.30	size (Checkbox Form Field Size)	327
2.16.31	sizeAuto (Automatically Size Form Field)	327
2.16.32	statusText (Associated Status Text)	327
2.16.33	textInput (Text Box Form Field Properties).....	328
2.16.34	type (Text Box Form Field Type)	328
2.17	Miscellaneous Topics	328
2.17.1	Text Box Content	328
2.17.1.1	txbxContent (Rich Text Box Content Container)	328
2.17.2	Subdocuments.....	328
2.17.2.1	subDoc (Anchor for Subdocument Location)	329
2.17.3	External Content Import.....	329
2.17.3.1	altChunk (Anchor for Imported External Content).....	329
2.17.3.2	altChunkPr (External Content Import Properties)	329
2.17.3.3	matchSrc (Keep Source Formatting on Import).....	330

2.17.4Roundtripping Alternate Content.....	330
2.18Simple Types.....	330
2.18.1ST_AlgorithmClass (Cryptographic Algorithm Classes).....	330
2.18.2ST_AlgorithmType (Cryptographic Algorithm Types).....	330
2.18.3ST_AnnotationVMerge (Table Cell Vertical Merge Revision Type)	331
2.18.4ST_Border (Border Styles)	331
2.18.5ST_BrClear (Line Break Text Wrapping Restart Location)	374
2.18.6ST_BrType (Break Types)	376
2.18.7ST_CalendarType (Calendar Types)	376
2.18.8ST_CaptionPos (Automatic Caption Positioning Values)	377
2.18.9ST_ChapterSep (Chapter Separator Types)	377
2.18.10 ST_CharacterSpacing (Character-Level Whitespace Compression Settings)	378
2.18.11 ST_Cnf (Conditional Formatting Bitmask)	378
2.18.12 ST_ColorSchemeIndex (Theme Color Reference)	379
2.18.13 ST_CombineBrackets (Two Lines in One Enclosing Character Type)	380
2.18.14 ST_CryptProv (Cryptographic Provider Types)	380
2.18.15 ST_DecimalNumber (Decimal Number Value)	381
2.18.16 ST_DisplacedByCustomXml (Location of Custom XML Markup Displacing an Annotation)	381
2.18.17 ST_DocGrid (Document Grid Types)	381
2.18.18 ST_DocPartBehavior (Insertion Behavior Types)	382
2.18.19 ST_DocPartGallery (Entry Gallery Types)	383
2.18.20 ST_DocPartType (Entry Types)	385
2.18.21 ST_DocProtect (Document Protection Types)	385
2.18.22 ST_DocType (Document Classification Values)	386
2.18.23 ST_DropCap (Text Frame Drop Cap Location)	386
2.18.24 ST_EdGrp (Range Permission Editing Group)	387
2.18.25 ST_EdnPos (Endnote Positioning Location)	387
2.18.26 ST_EighthPointMeasure (Measurement in Eighths of a Point)	388
2.18.27 ST_Em (Emphasis Mark Type)	388
2.18.28 ST_FFHelpTextVal (Help Text Value)	389
2.18.29 ST_FFName (Form Field Name Value)	389
2.18.30 ST_FFStatusTextVal (Status Text Value)	389
2.18.31 ST_FFTextType (Text Box Form Field Type Values)	389
2.18.32 ST_FldCharType (Complex Field Character Type)	390
2.18.33 ST_FontFamily (Font Family Value)	390
2.18.34 ST_FrameLayout (Frameset Layout Order)	390
2.18.35 ST_FrameScrollbar (Frame Scrollbar Visibility)	391
2.18.36 ST_FtnEdn (Footnote or Endnote Type)	391
2.18.37 ST_FtnPos (Footnote Positioning Location)	392
2.18.38 ST_Guid (128-Bit GUID)	392
2.18.39 ST_HAnchor (Horizontal Anchor Location)	392
2.18.40 ST_HdrFtr (Header or Footer Type)	393
2.18.41 ST_HeightRule (Height Rule)	394
2.18.42 ST_HexColor (Color Value)	394
2.18.43 ST_HexColorAuto ('Automatic' Color Value)	394
2.18.44 ST_HexColorRGB (Hexadecimal Color Value)	394
2.18.45 ST_HighlightColor (Text Highlight Colors)	395
2.18.46 ST_Hint (Font Type Hint)	396
2.18.47 ST_HpsMeasure (Measurement in Half-Points)	397
2.18.48 ST_InfoTextType (Help or Status Text Type)	397
2.18.49 ST_Jc (Horizontal Alignment Type)	397
2.18.50 ST_Lang (Language Reference)	399

2.18.51	ST_LangCode (Two Digit Hexadecimal Language Code)	399
2.18.52	ST_LevelSuffix (Content Between Numbering Symbol and Paragraph Text)	406
2.18.53	ST_LineNumberRestart (Line Numbering Restart Position)	406
2.18.54	ST_LineSpacingRule (Line Spacing Rule)	406
2.18.55	ST_Lock (Locking Types)	407
2.18.56	ST_LongHexNumber (Four Digit Hexadecimal Number Value)	408
2.18.57	ST_MacroName (Script Subroutine Name Value)	408
2.18.58	ST_MailMergeDataType (Mail Merge Data Source Type Values)	408
2.18.59	ST_MailMergeDest (Merged Document Destination Types)	408
2.18.60	ST_MailMergeDocType (Source Document Types)	409
2.18.61	ST_MailMergeOdsoFMDFieldType (Merge Field Mapping Types)	410
2.18.62	ST_MailMergeSourceType (Mail Merge ODSO Data Source Types)	410
2.18.63	ST_Merge (Merged Cell Type)	411
2.18.64	ST_MultiLevelType (Numbering Definition Type)	411
2.18.65	ST_NumberFormat (Numbering Format)	412
2.18.66	ST_OnOff (On/Off Value)	417
2.18.67	ST_PageBorderDisplay (Page Border Display Options)	418
2.18.68	ST_PageBorderOffset (Page Border Positioning Base)	418
2.18.69	ST_PageBorderZOrder (Page Border Z-Order)	418
2.18.70	ST_PageOrientation (Page Orientation)	419
2.18.71	ST_Panose (Panose-1 Number)	419
2.18.72	ST_Pitch (Font Pitch Value)	419
2.18.73	ST_PixelsMeasure (Measurement in Pixels)	419
2.18.74	ST_PointMeasure (Measurement in Points)	420
2.18.75	ST_Proof (Proofing State Values)	420
2.18.76	ST_ProofErr (Proofing Error Type)	420
2.18.77	ST_PTabAlignment (Absolute Position Tab Alignment)	420
2.18.78	ST_PTabLeader (Absolute Position Tab Leader Character)	421
2.18.79	ST_PTabRelativeTo (Absolute Position Tab Positioning Base)	422
2.18.80	ST_RestartNumber (Footnote/Endnote Numbering Restart Locations)	422
2.18.81	ST_RubyAlign (Phonetic Guide Text Alignment)	422
2.18.82	ST_SdtDateMappingType (Date Storage Format Types)	423
2.18.83	ST_SectionMark (Section Type)	424
2.18.84	ST_Shd (Shading Patterns)	424
2.18.85	ST_ShortHexNumber (Two Digit Hexadecimal Number Value)	431
2.18.86	ST_SignedHpsMeasure (Signed Measurement in Half-Points)	431
2.18.87	ST_SignedTwipsMeasure (Signed Measurement in Twentieths of a Point)	431
2.18.88	ST_String (String)	431
2.18.89	ST_StyleType (Style Types)	431
2.18.90	ST_TabJc (Custom Tab Stop Type)	432
2.18.91	ST_TabTlc (Custom Tab Stop Leader Character)	433
2.18.92	ST_TargetScreenSz (Target Screen Sizes for Generated Web Pages)	434
2.18.93	ST_TblLayoutType (Table Layout Type)	434
2.18.94	ST_TblOverlap (Table Overlap Setting)	436
2.18.95	ST_TblStyleOverrideType (Conditional Table Style Formatting Types)	436
2.18.96	ST_TblWidth (Table Width Units)	437
2.18.97	ST_TextAlignment (Vertical Text Alignment Types)	437
2.18.98	ST_TextboxTightWrap (Lines To Tight Wrap Within Text Box)	438
2.18.99	ST_TextDirection (Text Flow Direction)	438
2.18.100	ST_TextEffect (Animated Text Effects)	439
2.18.101	ST_TextScale (Text Expansion/Compression Percentage)	440
2.18.102	ST_Theme (Theme Font)	440
2.18.103	ST_ThemeColor (Theme Color)	441

2.18.104 ST_TwipsMeasure (Measurement in Twentieths of a Point).....	442
2.18.105 ST_UcharHexNumber (Two Digit Hexadecimal Number Value)	442
2.18.106 ST_Underline (Underline Patterns)	442
2.18.107 ST_UnsignedDecimalNumber (Unsigned Decimal Number Value)	444
2.18.108 ST_VAnchor (Vertical Anchor Location)	444
2.18.109 ST_VerticalAlignRun (Vertical Positioning Location)	444
2.18.110 ST_VerticalJc (Vertical Alignment Type).....	445
2.18.111 ST_View (Document View Values)	446
2.18.112 ST_Wrap (Text Wrapping around Text Frame Type).....	446
2.18.113 ST_XAlign (Horizontal Alignment Location)	447
2.18.114 ST_YAlign (Vertical Alignment Location)	447
2.18.115 ST_Zoom (Magnification Preset Values)	448

End of informative text.

2.2 Main Document Story

A WordprocessingML document consists of a compilation of two types of information:

- Properties
- Stories

In WordprocessingML, *stories* are defined as each unique region of content within a document into which the user can type. The most important story in a WordprocessingML document is the main document story, which contains the primary contents of the document. The main document story in WordprocessingML is stored inside the **body** element.

2.2.1 background (Document Background)

This element specifies the background information for this document. This background shall be displayed on all pages of the document, behind all other document content.

The child elements of the **background** element are in the Vector Markup Language (VML) namespace, which allows any valid VML effect to be applied to the document's background.

For solid color fill backgrounds, however, the attributes on this element allow for the specification of use of any valid RGB or theme color value (the latter a reference to the document's themes part).

Attributes	Description
color (Background Color)	<p>Specifies the color for the background of the document.</p> <p>This color may either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the background color as appropriate.</p> <p>If the background specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value.</p> <p>The possible values for this attribute are defined by the ST_HexColor simple type (§2.18.42).</p>
themeColor (Background Theme Color)	<p>Specifies a theme color to be applied to the current background.</p> <p>The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows color information to be set centrally in the document.</p> <p>If the color attribute is specified, its value shall be ignored in favor of the color resulting from the use of this attribute with any appropriate themeTint and themeShade attribute value calculations applied.</p> <p>The possible values for this attribute are defined by the ST_ThemeColor simple type (§2.18.103).</p>
themeShade (Border	Specifies the shade value applied to the supplied theme color (if any) for this background.

Attributes	Description
Theme Color Shade)	<p>If the themeShade is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to this border.</p> <p>The themeShade value is stored as a hex encoding of the shade value (from 0–255) applied to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $L' = L * \text{Shade}_{\text{percentage}}$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
themeTint (Border Theme Color Tint)	<p>Specifies the tint value applied to the supplied theme color (if any) for this background.</p> <p>If the themeTint is supplied, then it is applied to the RGB value of the theme color (from the theme part) to determine the final color applied to the document's background.</p> <p>The themeTint value is stored as a hex encoding of the tint value (from 0–255) applied to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $L' = L * \text{Tint}_{\text{pct}} + (1 - \text{Tint}_{\text{pct}})$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>

2.2.2 body (Document Body)

This element specifies the contents of the body of the document - the main document editing surface.

The document body contains what is referred to as block-level markup - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

2.2.3 document (Document)

This element specifies the contents of a main document part in a WordprocessingML document.

2.3 Paragraphs and Rich Formatting

The basis of a WordprocessingML document is its actual text contents. Those text contents can be stored in many contexts (tables, text boxes, etc.), but the most basic form of text contents in WordprocessingML is the paragraph, specified using the p element (§2.3.1.22).

Within the paragraph, all rich formatting at the paragraph level is stored within the pPr element (§2.3.1.25; §2.3.1.26).

[Note: Some examples of paragraph properties are alignment, border, hyphenation override, indentation, line spacing, shading, text direction, and widow/orphan control. end note]

Within the paragraph, text is grouped into one or more runs, represented by the r element (§2.3.2.23), which define a region of text with a common set of properties.

Just as a paragraph can have rich formatting, so too can a run. All of the elements inside an **r** element have their properties controlled by a corresponding optional **rPr** run properties element (§2.7.8.1; §2.3.2.26). *[Note: Some examples of run properties are bold, underlined, or visible. end note]*

Within runs, run content is the set of possible objects and characters which can be displayed in the document.

2.3.1 Paragraphs

The most basic unit of block-level content within a WordprocessingML document, *paragraphs* are stored using the **p** element (§2.3.1.22). A paragraph defines a distinct division of content with a WordprocessingML document which begins on a new line.

A paragraph's properties are specified via the **pPr** element (§2.3.1.25; §2.3.1.26). *[Note: Some examples of paragraph properties are alignment, border, hyphenation override, indentation, line spacing, shading, text direction, and widow/orphan control. end note]*

2.3.1.1 **adjustRightInd** (Automatically Adjust Right Indent When Using Document Grid)

This element specifies whether the right indent shall be automatically adjusted for the given paragraph when a document grid has been defined for the current section using the **docGrid** element (§2.6.5), modifying of the current right indent used on this paragraph.

[Note: This setting is used in order to ensure that the line breaking for that paragraph is not determined by the width of the final character on the line. end note]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, its value is assumed to be true.

Attributes	Description
val (On/Off Value)	Specifies a binary value for the property defined by the parent XML element. A value of on, 1, or true specifies that the property shall be explicitly applied. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted. A value of off, 0, or false specifies that the property shall be explicitly turned off. The possible value for this attribute are defined by ST_OnOff simple type (§2.18.66).

2.3.1.2 **autoSpaceDE** (Automatically Adjust Spacing of Latin and East Asian Text)

This element specifies whether inter-character spacing shall automatically be adjusted between regions of Latin text and regions of East Asian text in the current paragraph. These regions shall be determined by the Unicode character values of the text content within the paragraph.

[Note: This property is used to ensure that the spacing between regions of Latin text and adjoining East Asian text is sufficient on each side such that the Latin text can be easily read within the East Asian text. end note]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, its value is assumed to be true.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.3 **autoSpaceDN** (Automatically Adjust Spacing of East Asian Text and Numbers)

This element specifies whether inter-character spacing shall automatically be adjusted between regions of numbers and regions of East Asian text in the current paragraph. These regions shall be determined by the Unicode character values of the text content within the paragraph.

[Note: This property is used to ensure that the spacing between regions of numbers and adjoining East Asian text is sufficient on each side such that the numbers can be easily read within the East Asian text. end note]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, its value is assumed to be true.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.4 **bar** (Paragraph Border Between Facing Pages)

This element specifies the border which may be displayed on the inside edge of the paragraph when the parent's section settings specify that the section shall be printed using mirrored margins using the **mirrorMargins** element (§2.15.1.57).

[Note: This information is present in the WordprocessingML for the purposes of legacy document format compatibility, and it may be removed and/or ignored as required. *end note*]

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no bar border shall be applied to the current paragraph.

Attributes	Description
color (Border Color)	<p>Specifies the color for this border.</p> <p>This color may either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the border color as appropriate.</p> <p>If the border style (the val attribute) specifies the use of an art border, this attribute is ignored. As well, if the border specifies the use of a theme color via the themeColor attribute, this value is superseded by the theme color value.</p> <p>The possible values for this attribute are defined by the ST_HexColor simple type (§2.18.42).</p>
frame (Create Frame Effect)	<p>Specifies whether the specified border should be modified to create a frame effect by reversing the border's appearance from the edge nearest the text to the edge furthest from the text.</p> <p>If this attribute is omitted, then the border is not given any frame effect.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
shadow (Border Shadow)	<p>Specifies whether this border should be modified to create the appearance of a shadow.</p> <p>For the right and bottom borders, this is accomplished by duplicating the border below and right of the normal border location. For the right and top borders, this is accomplished by moving the order down and to the right of its original location.</p> <p>If this attribute is omitted, then the border is not given the shadow effect.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
space (Border Spacing Measurement)	<p>Specifies the spacing offset that shall be used to place this border on the parent object.</p> <p>When a document has a page border that is relative to the page edges (using a value of page in the offsetFrom attribute on pgBorders (§2.6.10)), it shall specify the distance between the edge of the page and the beginning of this border in points.</p> <p>When a document has a page border that is relative to the text extents (using a value of text in the offsetFrom attribute on pgBorders (§2.6.10)), or any other border type, it shall specify the distance between the edge of the object and the beginning of this border in points.</p> <p>The possible values for this attribute are defined by the ST_PointMeasure simple type (§2.18.74).</p>

Attributes	Description
sz (Border Width)	<p>Specifies the width of the current border.</p> <p>If the border style (val attribute) specifies a line border, the width of this border is specified in measurements of eighths of a point, with a minimum value of two (one-fourth of a point) and a maximum value of 96 (twelve points). Any values outside this range may be reassigned to a more appropriate value.</p> <p>If the border style (val attribute) specifies an art border, the width of this border is specified in measurements of points, with a minimum value of one and a maximum value of 31. Any values outside this range may be reassigned to a more appropriate value.</p> <p>The possible values for this attribute are defined by the ST_EighthPointMeasure simple type (§2.18.26).</p>
themeColor (Border Theme Color)	See §2.2.1
themeShade (Border Theme Color Shade)	See §2.2.1
themeTint (Border Theme Color Tint)	See §2.2.1
val (Border Style)	<p>Specifies the style of border used on this object.</p> <p>This border can either be an art border (a repeated image along the borders - only valid for page borders) or a line border (a line format repeated along the borders) - see the simple type definition for a description of each border style.</p> <p>The possible values for this attribute are defined by the ST_Border simple type (§2.18.4).</p>

2.3.1.5 **between** (Paragraph Border Between Identical Paragraphs)

This element specifies the border which shall be displayed between each paragraph in a set of paragraphs which have the same set of paragraph border settings.

To determine if any two adjoining paragraphs should have a **between** border or an individual **top** and **bottom** border, the set of borders on the two adjoining paragraphs are compared. If the border information on those two paragraphs is identical for all possible paragraphs borders, then the **between** border is displayed. Otherwise, each paragraph shall use its **bottom** and **top** border, respectively. If this border specifies a space attribute, that value is ignored - this border is always located at the bottom of each paragraph with an identical following paragraph, taking into account any space after the line pitch.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no between border shall be applied between identical paragraphs.

Attributes	Description
See §2.3.1.4	

2.3.1.6 **bidirectional** (Right to Left Paragraph Layout)

This element specifies that this paragraph shall be presented using a right to left direction. This property only affects the set of paragraph-level properties, and shall not affect the layout of text within the contents of this paragraph.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.7 **bottom** (Paragraph Border Between Identical Paragraphs)

This element specifies the border which shall be displayed below a set of paragraphs which have the same set of paragraph border settings.

To determine if any two adjoining paragraphs shall have an individual **top** and **bottom** border or a **between** border, the set of borders on the two adjoining paragraphs are compared. If the border information on those two paragraphs is identical for all possible paragraphs borders, then the **between** border is displayed. Otherwise, the final paragraph shall use its **bottom** border and the following paragraph shall use its **top** border, respectively. If this border specifies a space attribute, that value determines the space after the bottom of the text (ignoring any space below) which should be left before this border is drawn, specified in points.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no between border shall be applied below identical paragraphs.

Attributes	Description
See 2.3.1.4	

2.3.1.8 **cnfStyle** (Paragraph Conditional Formatting)

This element specifies the set of conditional table style formatting properties which have been applied to this paragraph, if this paragraph is contained within a table cell. [Note: This property is an optimization which may be used by consumers to determine if a given property on a paragraph is the result of the table style properties vs. direct formatting on the paragraph itself. *end note*]

If this property is specified on a paragraph which is not contained within a table cell, then its contents shall be ignored when reading the contents of the document.

Attributes	Description
val (Conditional Formatting Bit Mask)	<p>Specifies the set of conditional formatting properties that have been applied to this object.</p> <p>These properties are expressed using a string serialization of a binary bitmask for each of the following properties (reading from the first character position right):</p> <ul style="list-style-type: none"> • First Row - Is this the first row of the table? • Last Row - Is this the last row of the table? • First Column - Does this belong to the first column of the table? • Last Column - Does this belong to the last column of the table? • Band 1 Vertical - Does this belong to a column which should receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered columns (e.g. 1,3,5,...) • Band 2 Vertical - Does this belong to a column which should receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered columns (e.g. 2,4,6,...) • Band 1 Horizontal - Does this receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered rows (e.g. 1,3,5,...) • Band 2 Horizontal - Does this receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered rows (e.g. 2,4,6,...) • NE Cell - Is this part of the top-right corner of the table? • NW Cell - Is this part of the top-left corner of the table? • SE Cell - Is this part of the bottom-right corner of the table? • SW Cell - Is this part of the bottom-left corner of the table?

Attributes	Description
	For each of these properties, a value of 1 in the specified character position in the string means that the value is true, a value of 0 means false. All values must be specified. The possible values for this attribute are defined by the ST_Cnf simple type (§2.18.11).

2.3.1.9 contextualSpacing (Ignore Spacing Above and Below When Using Identical Styles)

This element specifies that any space specified before or after this paragraph, specified using the **spacing** element (§2.3.1.33), should not be applied when the preceding and following paragraphs are of the same paragraph style, affecting the top and bottom spacing respectively.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then spacing is not ignored. If it is present, then the spacing above or below on this paragraph is subtracted from the spacing which would have been present if contextual spacing was off, never going below zero.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.10 divId (Associated HTML div ID)

This element specifies that this paragraph should be located within the specified HTML **div** tag when this document is saved in HTML format. This ID is then used to look up the associated div stored in the **divs** (§2.15.2.8) element. [Note: This element is used to preserve the fidelity of existing HTML documents when saved in the WordprocessingML format. *end note*].

If the paragraph does not specify this element, then any **div** referenced by the previous paragraph is closed, and this paragraph shall not belong to any **div** when saved as HTML. If this specified id does not exist in the collection of **divs** the current document, then any **div** referenced by the previous paragraph is closed, and this paragraph shall not belong to any **div** when saved as HTML.

Attributes	Description
val (Decimal Number Value)	Specifies that the contents of this attribute will contain a decimal number. The contents of this decimal number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.3.1.11 framePr (Text Frame Properties)

This element specifies information about the current paragraph with regard to *text frames*. *Text frames* are paragraphs of text in a document which are positioned in a separate region or frame in the document, and can be positioned with a specific size and position relative to non-frame paragraphs in the current document.

The first piece of information specified by the **framePr** element is that the current paragraph is actually part of a text frame in the document. This information is specified simply by the presence of the **framePr** element in paragraph's properties. If the **framePr** element is omitted, the paragraph shall not be part of any text frame in the document.

The second piece of information concerns the set of paragraphs which are part of the current text frame in the document. This is determined based on the attributes on the **framePr** element. If the set of attribute values specified on two adjacent paragraphs is identical, then those two paragraphs shall be considered to be part of the same text frame and rendered within the same frame in the document.

The positioning of the frame relative to the properties stored on its attribute values shall be calculated relative to the next paragraphs in the document which is itself not part of a text frame.

Attributes	Description
anchorLock (Lock Frame Anchor to Paragraph)	<p>Specifies that the frame shall always remain in the same logical position relative to the non-frame paragraphs which precede and follow it in this document.</p> <p>This means that consumers which modify this document shall ensure that this text frame remains directly above the non-frame paragraph which it is currently above, by adjusting the frame's positioning properties as needed as the paragraph is moved throughout the document rather than moving the frame's logical location within the paragraphs in the document, if that would be more appropriate.</p> <p>If this attribute is omitted, then this frame shall not have a locked anchor position. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
dropCap (Drop Cap Frame)	<p>Specifies that the current frame contains a drop cap to be located at the beginning of the next non-frame paragraph in the document. Its contents shall be used to specify how that drop cap should be positioned relative to that paragraph.</p> <p>If this attribute is omitted, then this frame shall not be considered a drop cap frame.</p> <p>[Note: Although a drop cap is simply a text frame, this element is used to determine how the cap should be positioned relative to the following non-frame paragraph in relative terms (see possible values), rather than relying on absolute sizing. <i>end note</i>] The possible values for this attribute are defined by the ST_DropCap simple type (§2.18.23).</p>
h (Frame Height)	<p>Specifies the frame's height.</p> <p>This height is expressed in twentieths of a point.</p> <p>If this attribute is omitted, then its value shall be assumed to be 0.</p> <p>The meaning of the value of the h attribute is defined based on the value of the hRule attribute for this text frame as follows:</p> <ul style="list-style-type: none"> • If the value of hRule is <code>auto</code>, then the frame's height should be automatically determined based on the height of its contents. This value is ignored. • If the value of hRule is <code>atLeast</code>, then the frame's height should be at least the value of this attribute. <p>If the value of hRule is <code>exact</code>, then the frame's height should be exactly the value of this attribute. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
hAnchor (Frame Horizontal Positioning Base)	<p>Specifies the base object from which the horizontal positioning in the x attribute should be calculated.</p> <p>A text frame may be horizontally positioned relative to:</p> <ul style="list-style-type: none"> • The vertical edge of the page before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs) • The vertical edge of the text margin before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs) • The vertical edge of the text margin for the column in which the anchor paragraph is located <p>If this attribute is omitted, then its value shall be assumed to be <code>page</code>.</p>

Attributes	Description
	The possible values for this attribute are defined by the ST_HAnchor simple type (§2.18.39).
hRule (Frame Height Type)	<p>Specifies the meaning of the height specified for this frame.</p> <p>The meaning of the value of the h attribute is defined based on the value of the hRule attribute for this text frame as follows:</p> <ul style="list-style-type: none"> • If the value of hRule is <code>auto</code>, then the frame's height should be automatically determined based on the height of its contents. The h value is ignored. • If the value of hRule is <code>atLeast</code>, then the frame's height should be at least the value the h attribute. • If the value of hRule is <code>exact</code>, then the frame's height should be exactly the value of the h attribute. <p>If this attribute is omitted, then its value shall be assumed to be <code>auto</code>.</p> <p>The possible values for this attribute are defined by the ST_HeightRule simple type (§2.18.41).</p>
hSpace (Horizontal Frame Padding)	<p>Specifies the minimum distance which shall be maintained between the current text frame and any non-frame text which has been allowed to flow around this object when the wrap attribute on this text frame is set to <code>around</code>.</p> <p>This distance is expressed in twentieths of a point.</p> <p>If the wrap value is not set to <code>around</code>, this value shall be ignored. If this attribute is omitted, its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
lines (Drop Cap Vertical Height in Lines)	<p>Specifies the number of lines in the non-frame paragraph to which this text frame is anchored which should be used to calculate the drop cap's height.</p> <p>If the current frame is not a drop cap (the parent framePr element does not have the dropCap attribute), this value is ignored. If the current text frame is a dropped cap and this attribute is present, then any other vertical positioning information shall be ignored.</p> <p>If this attribute is omitted, then its value shall be considered to be 1.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
vAnchor (Frame Vertical Positioning Base)	<p>Specifies the base object from which the horizontal positioning in the y attribute should be calculated.</p> <p>A text frame may be horizontally positioned relative to:</p> <ul style="list-style-type: none"> • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the text margin before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) <p>If this attribute is omitted, then its value shall be assumed to be <code>page</code>.</p> <p>The possible values for this attribute are defined by the ST_VAnchor simple type (§2.18.108).</p>
vSpace (Vertical Frame Padding)	Specifies the minimum distance which shall be maintained between the current text frame and any non-frame text which is above or below this text frame.

Attributes	Description
	<p>This distance is expressed in twentieths of a point.</p> <p>If this attribute is omitted, its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
w (Frame Width)	<p>Specifies the exact value for this text frame's width.</p> <p>This value is specified in twentieths of a point.</p> <p>When this attribute is present, the text frame shall be rendered to the exact width specified. If this attribute is omitted, the text frame width shall be automatically determined by the maximum line width of the content within the text frame.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
wrap (Text Wrapping Around Frame)	<p>Specifies the type of text wrapping which should be allowed around the contents of this text frame. This attribute determines if non-frame text shall be allowed to flow around the contents of this frame.</p> <p>If this attribute is omitted, its value shall be assumed to be around.</p> <p>The possible values for this attribute are defined by the ST_Wrap simple type (§2.18.112).</p>
x (Absolute Horizontal Position)	<p>Specifies an absolute horizontal position for the text frame. This absolute position is specified relative to the horizontal anchor specified by the hAnchor attribute for this text frame.</p> <p>This value is expressed in twentieths of a point. If it is positive, then the text frame is positioned after the anchor object in the direction of horizontal text flow in this document. If it is negative, then the text frame is positioned before the anchor object in the direction of horizontal text flow in this document.</p> <p>If the xAlign attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>
xAlign (Relative Horizontal Position)	<p>Specifies a relative horizontal position for the text frame. This relative position is specified relative to the horizontal anchor specified by the hAnchor attribute for this text frame.</p> <p>If omitted, this attribute is not specified and the value of the x attribute determines the absolute horizontal position of the text frame. If specified, the position for this attribute supersedes any value which is specified in the x attribute, and that value is ignored.</p> <p>The possible values for this attribute are defined by the ST_XAlign simple type (§2.18.113).</p>
y (Absolute Vertical Position)	<p>Specifies an absolute vertical position for the text frame. This absolute position is specified relative to the vertical anchor specified by the vAnchor attribute for this text frame.</p> <p>This value is expressed in twentieths of a point. If it is positive, then the text frame is positioned after the anchor object in the direction of vertical text flow in this document. If it is negative, then the text frame is positioned before the anchor object in the direction of vertical text flow in this document.</p> <p>If the yAlign attribute is also specified, then this value is ignored. If this attribute is omitted, then</p>

Attributes	Description
	its value shall be assumed to be 0. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).
yAlign (Relative Vertical Position)	Specifies a relative vertical position for the text frame. This relative position is specified relative to the vertical anchor specified by the vAnchor attribute for this text frame. If omitted, this attribute is not specified and the value of the y attribute determines the absolute horizontal position of the text frame. If specified, the position for this attribute supersedes any value which is specified in the y attribute, and that value is ignored, unless the vAnchor is set to text, in which case any relative positioning is not allowed, and is itself ignored. The possible values for this attribute are defined by the ST_YAlign simple type (§2.18.114).

2.3.1.12 **ind** (Paragraph Indentation)

This element specifies the set of indentation properties applied to the current paragraph.

Indentation settings are overridden on an individual basis - if any single attribute on this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If any single attribute on this element is never specified in the style hierarchy, then no indentation of that type is applied to the paragraph.

Attributes	Description
firstLine (Additional First Line Indentation)	Specifies the additional indentation which shall be applied to the first line of the parent paragraph. This additional indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph. The firstLine and hanging attributes are mutually exclusive, if both are specified, then the firstLine value is ignored. If the firstLineChars attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be zero (if needed). The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
firstLineChars (Additional First Line Indentation in Character Units)	Specifies the additional indentation which shall be applied to the first line of the parent paragraph. This additional indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph. It is specified in one hundredths of a character unit. The firstLineChars and hangingChars attributes are mutually exclusive, if both are specified, then the firstLineChars value is ignored. If the firstLine attribute is also specified, then this value supersedes its other value. If this attribute is omitted, then its value shall be assumed to be zero (if needed). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
hanging (Indentation Removed from First Line)	Specifies the indentation which shall be removed from the first line of the parent paragraph, by moving the indentation on the first line back towards the beginning of the direction of text flow. This indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph. The firstLine and hanging attributes are mutually exclusive, if both are specified, then the firstLine value is ignored. If the hangingChars attribute is also specified, then this value is

Attributes	Description
	<p>ignored. If this attribute is omitted, its value shall be assumed to be zero (if needed). The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
hangingChars (Indentation Removed From First Line in Character Units)	<p>Specifies the indentation which shall be removed from the first line of the parent paragraph, by moving the indentation on the first line back towards the beginning of the direction of text flow.</p> <p>This indentation is specified relative to the paragraph indentation which is specified for all other lines in the parent paragraph.</p> <p>It is specified in one hundredths of a character unit.</p> <p>The firstLineChars and hangingChars attributes are mutually exclusive, if both are specified, then the firstLine value is ignored. If the hanging attribute is also specified, then its value is superseded by this value. If this attribute is omitted, its value shall be assumed to be zero (if needed). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
left (Left Indentation)	<p>Specifies the indentation which shall be placed between the left text margin for this paragraph and the left edge of that paragraph's content in a left to right paragraph, and the right text margin and the right edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§2.3.1.18) is specified for this paragraph, then this indent is used for the inside page edge - the right page edge for odd numbered pages and the left page edge for even numbered pages.</p> <p>If this attribute is omitted, its value shall be assumed to be zero.</p> <p>All other values for this element are relative to the text margin, Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. This value may be superseded for the first line only via use of the firstLine or hanging attributes. As well, if the leftChars attribute is specified, then this value is ignored. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>
leftChars (Left Indentation in Character Units)	<p>Specifies the indentation which shall be placed between the left text margin for this paragraph and the left edge of that paragraph's content in a left to right paragraph, and the right text margin and the right edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§2.3.1.18) is specified for this paragraph, then this indent is used for the inside page edge - the right page edge for odd numbered pages and the left page edge for even numbered pages.</p> <p>This value is specified in hundredths of a character unit.</p> <p>If this attribute is omitted, its value shall be assumed to be zero.</p> <p>All other values for this element are relative to the text margin, Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. This value may be superseded for the first line only via use of the firstLine or hanging attributes. As well, if the left attribute is specified, then its value is ignored, and is superseded by this value. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>

Attributes	Description
right (Right Indentation)	<p>Specifies the indentation which shall be placed between the right text margin for this paragraph and the right edge of that paragraph's content in a left to right paragraph, and the left text margin and the left edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§2.3.1.18) is specified for this paragraph, then this indent is used for the outside page edge - the left page edge for odd numbered pages and the right page edge for even numbered pages.</p> <p>If this attribute is omitted, its value shall be assumed to be zero.</p> <p>All other values for this element are relative to the text margin, Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. As well, if the rightChars attribute is specified, then this value is ignored.</p> <p>The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>
rightChars (Right Indentation in Character Units)	<p>Specifies the indentation which shall be placed between the right text margin for this paragraph and the right edge of that paragraph's content in a left to right paragraph, and the left text margin and the left edge of that paragraph's text in a right to left paragraph. If the mirrorIndents property (§2.3.1.18) is specified for this paragraph, then this indent is used for the outside page edge - the left page edge for odd numbered pages and the right page edge for even numbered pages.</p> <p>This value is specified in hundredths of a character unit.</p> <p>If this attribute is omitted, its value shall be assumed to be zero.</p> <p>All other values for this element are relative to the right text margin, Negative values are defined such that the text is moved past the text margin, positive values move the text inside the text margin. As well, if the right attribute is specified, then its value is ignored, and is superseded by this value.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>

2.3.1.13 **jc** (Paragraph Alignment)

This element specifies the paragraph alignment which shall be applied to text in this paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no alignment is applied to the paragraph.

Attributes	Description
val (Alignment Type)	<p>Specifies the justification which should be applied to the parent object within a document.</p> <p>The possible values (see below) for this attribute are always specified relative to the page, and do not change semantic from right-to-left and left-to-right documents.</p> <p>The possible values for this attribute are defined by the ST_Jc simple type (§2.18.49).</p>

2.3.1.14 **keepLines** (Keep All Lines On One Page)

This element specifies that when rendering this document in a page view, all lines for this page are maintained on a single page whenever possible.

This means that if the contents of the current paragraph would normally span across two pages due to the placement of the paragraph's text, all lines in this paragraph shall be moved onto the next page to ensure they are displayed together. If this

is not possible because all lines in the paragraph would exceed a single page in any case, then lines in this paragraph shall start on a new page, with page breaks as needed afterwards.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.15 **keepNext** (Keep Paragraph With Next Paragraph)

This element specifies that when rendering this document in a paginated view, the contents of this paragraph are at least partly rendered on the same page as the following paragraph whenever possible.

This means that if the contents of the current paragraph would normally be completely rendered on a different page than the following paragraph (because only one of the two paragraphs would fit on the remaining space on the first page), then both paragraphs shall be rendered on a single page. This property can be chained between multiple paragraphs to ensure that all paragraphs are rendered on a single page without any intervening page boundaries. If this is not possible the entire set of paragraphs that are grouped together using this property would exceed a single page in any case, then the set of "keep with next" paragraphs shall start on a new page, with page breaks as needed afterwards.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.16 kinsoku (Use East Asian Typography Rules for First and Last Character per Line)

This element specifies whether East Asian typography and line-breaking rules shall be applied to text in this paragraph to determine which characters may begin and end each line. This property only applies to Chinese PRC, Chinese Taiwan, and Japanese text in this paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall be applied to Chinese PRC, Chinese Taiwan, and Japanese text in this paragraph.

If these rules are set on the current paragraph, then the following rules are applied to the all first and last characters in the paragraph except the first and last character in the paragraph. By default, the following settings are used for **kinsoku** paragraphs:

Chinese (Simplified)

Cannot start a line:

!%)...;>?]}€°·~—||'...%°'°C:、。〃>》「」】〕】〃ㄣㄤㄥㄥㄥ}}! "% ') , . : ; ? [` | } ~ ∅

Cannot end a line:

\$({\{£¥.“” < ≪ 「 『 【 [[”_{\\$} (. 「 { £ ¥

Chinese (Traditional)

Cannot start a line:

!) , ; ?] } , . : ; ?] | } ,

Cannot end a line:

([{f¥^{“”} < 《 「 『 【 (" ∪ ∩ ⊂ ⊆ ≅ ^ ⊤ ⊥ } {

Japanese

Cannot start a line:

!%) ,,:?]]{°'''%0'''°C、。々>》「」】) °>ゞ・ \ゞ!%) , . : ; ?] }。└、°°¢

Cannot end a line:

\$([\{£¥““ < 《 「 『 【 (\$ ([{ 「 £ ¥

Korean

Cannot start a line:

!%),,,:;?]}c°°°°°C>》」』】〕!%) , . : ; ?] } ¢

Cannot end a line:

\$([\{£¥““ < 《 「 『 【 (\$ ([{ £ ¥₩

If the **strictFirstAndLastChars** property (§2.15.1.83) is set in the Document Settings part, then the following settings supersede the defaults for Japanese:

Cannot start a line:

!%),,,:;?]}c°°°°°C、。々>》」』】）あいうえおっやゆよわ “ ° > ゾアイウエオツヤユヨワカケ・ー\ゝ

!%) , . : ; ?] } 。 ー・アイウエオツヤユヨワカケ・ー\ゝ ¢

Cannot end a line:

\$([\{£¥““ < 《 「 『 【 (\$ ([{ 「 £ ¥

If the **noLineBreaksBefore** property (§2.15.1.59) is set in the Document Settings part, then the characters it specifies cannot begin a line for the specified language. If the **noLineBreaksAfter** property (§2.15.1.58) is set in the Document Settings part, then the characters it specifies cannot end a line for the specified language. In both cases, those settings shall supersede the defaults specified above.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.17 **left** (Left Paragraph Border)

This element specifies the border which shall be displayed on the left side of the page around the specified paragraph. To determine if any two adjoining paragraphs should have a **left** border which spans the full line height or not, the left border shall be drawn between the **top** border or **between** border at the top (whichever would be rendered for the current paragraph), and the **bottom** border or **between** border at the bottom (whichever would be rendered for the current paragraph).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no left border shall be applied.

Attributes	Description
See 2.3.1.4	

2.3.1.18 **mirrorIndents** (Use Left/Right Indents as Inside/Outside Indents)

This element specifies whether the paragraph indents should be interpreted as mirrored indents. When this element is present, the left indent shall become the inside indent and the right indent shall become the outside indent.

If the **mirrorIndents** property is specified for this paragraph, then the inside page edge is the right page edge for odd numbered pages and the left page edge for even numbered pages. Conversely, the outside page edge is the left page edge for odd numbered pages and the right page edge for even numbered pages.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.19 numPr (Numbering Definition Instance Reference)

This element specifies that the current paragraph references a *numbering definition instance* in the current document. The presence of this element specifies that the paragraph will inherit the properties specified by the numbering definition in the **num** element (§2.9.16) at the level specified by the level specified in the **lvl** element (§2.9.7) and shall have an associated number positioned before the beginning of the text flow in this paragraph. When this element appears as part of the paragraph formatting for a paragraph style, then any numbering level defined using the **ilvl** element shall be ignored, and the **pStyle** element (§2.9.25) on the associated abstract numbering definition shall be used instead.

Child Elements
ilvl (§2.9.3); ins (§2.13.5.19); numberingChange (§2.13.5.30); numId (§2.9.19);

2.3.1.20 outlineLvl (Associated Outline Level)

This element specifies the *outline level* which shall be associated with the current paragraph in the document. The *outline level* specifies an integer which defines the level of the associated text. This level shall not affect the appearance of the text in the document, but shall be used to calculate the TOC field (§2.16.5.75) if the appropriate field switches have been set, and may be used by consumers to provide additional application behavior.

The outline level of text in the document (specified using the **val** attribute) may be from 0 to 9, where 9 specifically indicates that there is no outline level specifically applied to this paragraph. If this element is omitted, then the outline level of the content is assumed to be 9 (no level).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.3.1.21 overflowPunct (Allow Punctuation to Extent Past Text Extents)

This element specifies that punctuation characters which appear at the end of text in any line in this paragraph shall be allowed to extend one character past the text extents (indents/margins) as needed in order to ensure that they are not displayed as hanging punctuation. *Hanging punctuation* is defined as punctuation which appears on a different line than the text which it logically would appear with.

Omitting this element sets its value to true.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.22 p (Paragraph)

This element specifies a paragraph of content in the document.

The contents of a paragraph in a WordprocessingML document shall consist of any combination of the following four types of content:

- Paragraph properties
- Annotations (bookmarks, comments, revisions)
- Custom markup
- Run level content (fields, hyperlinks, runs)
-

Attributes	Description
rsidDel (Revision Identifier for Paragraph Deletion)	Specifies a unique identifier used to track the editing session when the paragraph was deleted from the main document.

Attributes	Description
	<p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidP (Revision Identifier for Paragraph Properties)	<p>This attribute specifies a unique identifier used to track the editing session when the paragraph's properties were last modified in this document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidR (Revision Identifier for Paragraph)	<p>This attribute specifies a unique identifier used to track the editing session when the paragraph was added to the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidRDefault (Default Revision Identifier for Runs)	<p>This attribute specifies a unique identifier used for all runs in this paragraph which do not explicitly declare an rsidR attribute. This attribute allows consumers to optimize the locations where rsid* values are written in this document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>

Attributes	Description
rsidRPr (Revision Identifier for Paragraph Glyph Formatting)	<p>This attribute specifies a unique identifier used to track the editing session when the glyph character representing the paragraph mark was last modified in the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>

2.3.1.23 **pageBreakBefore** (Start Paragraph on Next Page)

This element specifies that when rendering this document in a paginated view, the contents of this paragraph are rendered on the start of a new page in the document.

This means that if the contents of the current paragraph would normally be rendered on the middle of a page in the host document, then the paragraph shall be rendered on a new page as if the paragraph was preceded by a page break in the WordprocessingML contents of the document. This property supersedes any use of the **keepNext** property, so that if any paragraph wishes to be on the same page as this paragraph, they will still be separated by a page break.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this property shall not be applied.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.24 **pBdr** (Paragraph Borders)

This element specifies the borders for the parent paragraph. Each child element shall specify a specific type of border (left, right, bottom, top, and between).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no paragraph borders shall be applied.

2.3.1.25 **pPr** (Previous Paragraph Properties)

This element specifies a set of paragraph properties which shall be attributed to a revision by a particular author and at a particular time. This element contains the set of properties which have been tracked as a specific set of revisions by one author.

2.3.1.26 **pPr** (Paragraph Properties)

This element specifies a set of paragraph properties which shall be applied to the contents of the parent paragraph after all style/numbering/table properties have been applied to the text. These properties are defined as *direct formatting*, since they are directly applied to the paragraph and supersede any formatting from styles.

2.3.1.27 **pStyle** (Referenced Paragraph Style)

This element specifies the style ID of the paragraph style which shall be used to format the contents of this paragraph. This formatting is applied at the following location in the *style hierarchy*:

- Document defaults

- Table styles
- Numbering styles
- Paragraph styles (this element)
- Character styles
- Direct Formatting

This means that all properties specified in the **style** element (§2.7.3.17) with a styleId which corresponds to the value in this element's val attribute are applied to the paragraph at the appropriate level in the hierarchy.

If this element is omitted, or it references a style which does not exist, then no paragraph style shall be applied to the current paragraph. As well, this property is ignored if the paragraph properties are part of a paragraph style.

Attributes	Description
val (String Value)	Specifies that its contents will contain a string. The contents of this string are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.3.1.28 **right** (Right Paragraph Border)

This element specifies the border which shall be displayed on the right side of the page around the specified paragraph. To determine if any two adjoining paragraphs should have a **right** border which spans the full line height or not, the right border shall be drawn between the **top** border or **between** border at the top (whichever would be rendered for the current paragraph), and the **bottom** border or **between** border at the bottom (whichever would be rendered for the current paragraph).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no right border shall be applied.

Attributes	Description
See 2.3.1.4	

2.3.1.29 **rPr** (Run Properties for the Paragraph Mark)

This element specifies the set of run properties applied to the glyph used to represent the physical location of the paragraph mark for this paragraph. This paragraph mark, being a physical character in the document, can be formatted, and therefore must be capable of representing this formatting like any other character in the document.

If this element is not present, the paragraph mark is unformatted, as with any other run of text.

2.3.1.30 **rPr** (Previous Run Properties for the Paragraph Mark)

This element specifies a set of run properties applied to the glyph used to represent the physical location of the paragraph mark for this paragraph which shall be attributed to a revision by a particular author and at a particular time. This element contains the set of properties which have been tracked as a specific set of revisions by one author.

2.3.1.31 **shd** (Paragraph Shading)

This element specifies the shading applied to the contents of the paragraph.

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no paragraph shading shall be applied.

Attributes	Description
color (Shading Pattern Color)	<p>Specifies the color used for any foreground pattern specified for this shading using the val attribute.</p> <p>This color may either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the foreground shading color as appropriate.</p> <p>If the shading style (the val attribute) specifies the use of no shading format or is omitted, then this property has no effect. Also, if the shading specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value.</p> <p>If this attribute is omitted, then its value shall be assumed to be auto. The possible values for this attribute are defined by the ST_HexColor simple type (§2.18.42).</p>
fill (Shading Background Color)	<p>Specifies the color used for the background for this shading.</p> <p>This color may either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the background shading color as appropriate.</p> <p>If this attribute is omitted, then its value shall be assumed to be auto. If the shading specifies the use of a theme color via the themeFill attribute, then this value is superseded by the theme color value.</p> <p>The possible values for this attribute are defined by the ST_HexColor simple type (§2.18.42).</p>
themeColor (Shading Pattern Theme Color)	<p>Specifies a theme color which should be applied to any foreground pattern specified for this shading using the val attribute.</p> <p>The specified theme color is a reference to one of the predefined theme colors, located in the document's themes part, which allows for color information to be set centrally in the document.</p> <p>If this element is omitted, then no theme color is applied, and the color attribute shall be used to determine the shading pattern color. The possible values for this attribute are defined by the ST_ThemeColor simple type (§2.18.103).</p>
themeFill (Shading Background Theme Color)	<p>Specifies a theme color which should be applied to the background for this shading.</p> <p>The specified theme color is a reference to one of the predefined theme colors, located in the document's themes part, which allows for color information to be set centrally in the document.</p> <p>If this element is omitted, then no theme color is applied, and the color attribute shall be used to determine the shading background color. The possible values for this attribute are defined by the ST_ThemeColor simple type (§2.18.103).</p>
themeFillShade (Shading Background Theme Color Shade)	<p>Specifies the shade value applied to the supplied theme color (if any) for this shading color.</p> <p>If the themeFillShade is supplied, then it is applied to the RGB value of the themeFill color (from the theme part) to determine the final color applied to this border.</p> <p>The themeFillShade value is stored as a hex encoding of the shade value (from 0 to 255) applied</p>

Attributes	Description
	<p>to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $L' = L * \text{Shade}_{\text{percentage}}$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
themeFillTint (Shading Background Theme Color Tint)	<p>Specifies the tint value applied to the supplied theme color (if any) for this shading instance.</p> <p>If the themeFillTint is supplied, then it is applied to the RGB value of the themeFill color (from the theme part) to determine the final color applied to this border.</p> <p>The themeFillTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $L' = L * \text{Tint}_{\text{pct}} + (1 - \text{Tint}_{\text{pct}})$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
themeShade (Shading Pattern Theme Color Shade)	<p>Specifies the shade value applied to the supplied theme color (if any) for this shading color.</p> <p>If the themeShade is supplied, then it is applied to the RGB value of the themeColor color (from the theme part) to determine the final color applied to this border.</p> <p>The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> • Convert the color to the HSL color format (values from 0 to 1) • Modify the luminance factor as follows: $L' = L * \text{Shade}_{\text{percentage}}$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
themeTint (Shading Pattern Theme Color Tint)	<p>Specifies the tint value applied to the supplied theme color (if any) for this shading instance.</p> <p>If the themeTint is supplied, then it is applied to the RGB value of the themeColor color (from the theme part) to determine the final color applied to this border.</p> <p>The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border.</p>

Attributes	Description
	<p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> Convert the color to the HSL color format (values from 0 to 1) Modify the luminance factor as follows: $L' = L * \text{Tint}_{\text{pct}} + (1 - \text{Tint}_{\text{pct}})$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
val (Shading Pattern)	<p>Specifies the pattern which shall be used to lay the pattern color over the background color for this paragraph shading.</p> <p>This pattern consists of a mask which is applied over the background shading color to get the locations where the pattern color should be shown. Each of these possible masks are shown in the simple type values referenced below.</p> <p>The possible values for this attribute are defined by the ST_Shd simple type (§2.18.84).</p>

2.3.1.32 **snapToGrid** (Use Document Grid Settings for Inter-Line Paragraph Spacing)

This element specifies whether the current paragraph should use the document grid lines per page settings defined in the **docGrid** element (§2.6.5) when laying out the contents in the paragraph. This setting determines whether the additional line pitch specified in the document grid shall be added to each line in this paragraph as specified by the document grid. If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall use the document grid to lay out text when a document grid is defined for this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.33 **spacing** (Spacing Between Lines and Above/Below Paragraph)

This element specifies the inter-line and inter-paragraph spacing which shall be applied to the contents of this paragraph when it is displayed by a consumer.

If this element is omitted on a given paragraph, each of its values is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall have no spacing applied to its lines, or above and below its contents.

When determining the spacing between any two paragraphs, a consumer shall use the maximum of the inter-line spacing in each paragraph, the spacing after the first paragraph and the spacing before the second paragraph to determine the net spacing between the paragraphs.

Attributes	Description
after (Spacing Below Paragraph)	<p>Specifies the spacing that should be added after the last line in this paragraph in the document in absolute units.</p> <p>If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall have no spacing applied below its contents.</p>

Attributes	Description
	<p>If the afterLines attribute or the afterAutoSpacing attribute is also specified, then this attribute value is ignored.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
afterAutospacing (Automatically Determine Spacing Below Paragraph)	<p>Specifies whether a consumer shall automatically determine the spacing after this paragraph based on its contents.</p> <p>This automatic spacing shall match the spacing which would be applied to the paragraph in an HTML document where no explicit spacing before/after is specified.</p> <p>If this attribute is specified, then any value in after or afterLines is ignored, and the spacing is automatically determined by the consumer.</p> <p>If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then automatic spacing is turned off (not applied). The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
afterLines (Spacing Below Paragraph in Line Units)	<p>Specifies the spacing that should be added after the last line in this paragraph in the document in line units.</p> <p>The value of this attribute is specified in one hundredths of a line.</p> <p>If the afterAutoSpacing attribute is also specified, then this attribute value is ignored. If this setting is never specified in the style hierarchy, then its value shall be zero (if needed). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
before (Spacing Above Paragraph)	<p>Specifies the spacing that should be added above the first line in this paragraph in the document in absolute units.</p> <p>If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall have no spacing applied above its contents.</p> <p>If the beforeLines attribute or the beforeAutoSpacing attribute is also specified, then this attribute value is ignored.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
beforeAutospacing (Automatically Determine Spacing Above Paragraph)	<p>Specifies whether a consumer shall automatically determine the spacing before this paragraph based on its contents.</p> <p>This automatic spacing shall match the spacing which would be applied to the paragraph in an HTML document where no explicit spacing before/after is specified.</p> <p>If this attribute is specified, then any value in before or beforeLines is ignored, and the spacing is automatically determined by the consumer.</p> <p>If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting</p>

Attributes	Description
	is never specified in the style hierarchy, then automatic spacing is turned off (not applied). The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
beforeLines (Spacing Above Paragraph IN Line Units)	<p>Specifies the spacing that should be added before the first line in this paragraph in the document in line units.</p> <p>The value of this attribute is specified in one hundredths of a line.</p> <p>If the beforeAutoSpacing attribute is also specified, then this attribute value is ignored. If this setting is never specified in the style hierarchy, then its value shall be zero (if needed). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
line (Spacing Between Lines in Paragraph)	<p>This attribute specifies the amount of vertical spacing between lines of text within this paragraph.</p> <p>If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no line spacing shall be applied to lines within this paragraph.</p> <p>If the value of the lineRule attribute is either atLeast or exactly, then the value of this attribute shall be interpreted as twentieths of a point. When the value of the lineRule attribute is either exactly, the text shall be positioned as follows within that line height:</p> <ul style="list-style-type: none"> • When the line height is too small, the text shall be positioned at the bottom of the line (i.e. clipped from the top down) • When the line height is too large, the text shall be centered in the available space. <p>If the value of the lineRule attribute is auto, then the value of the line attribute shall be interpreted as 240ths of a line, in the manner described by the simple type's values. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>
lineRule (Type of Spacing Between Lines)	<p>Specifies how the spacing between lines is calculated as stored in the line attribute.</p> <p>If this attribute is omitted, then it shall be assumed to be of a value auto if a line attribute value is present.</p> <p>If the value of this attribute is either atLeast or exactly, then the value of the line attribute shall be interpreted as twentieths of a point, in the manner described by the simple type's values.</p> <p>If the value of this attribute is auto, then the value of the line attribute shall be interpreted as 240ths of a line, in the manner described by the simple type's values. The possible values for this attribute are defined by the ST_LineSpacingRule simple type (§2.18.54).</p>

2.3.1.34 **suppressAutoHyphens** (Suppress Hyphenation for Paragraph)

This element specifies whether any hyphenation shall be performed on this paragraph by the consumer when requested using the **autoHyphenation** element (§2.15.1.10) in the document's settings. This element specifies whether the current paragraph should be exempted from any hyphenation which is applied by the consumer on this document.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the

default hyphenation settings for the document, as specified in the **autoHyphenation** element, shall apply to the contents of this paragraph.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.35 **suppressLineNumbers** (Suppress Line Numbers for Paragraph)

This element specifies whether line numbers shall be calculated for lines in this paragraph by the consumer when line numbering is requested using the **InNumType** element (§2.6.8) in the paragraph's parent section settings. This element specifies whether the current paragraph's lines should be exempted from line numbering which is applied by the consumer on this document, not just suppressing the display of the numbering, but removing these lines from the line numbering calculation.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the default line number settings for the section, as specified in the **InNumType** element shall apply to each line of this paragraph.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.36 **suppressOverlap** (Prevent Text Frames From Overlapping)

This element specifies whether a text frame which intersects another text frame at display time shall be allowed to overlap the contents of the other text frame. If a text frame cannot overlap other text frames, it shall be repositioned when displayed to prevent this overlap as needed.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then overlap shall be allowed between a text frame which intersects another text frame displayed at the same location.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.37 **tab** (Custom Tab Stop)

This element specifies a single custom tab stop within a set of custom tab stops applied as part of a set of customized paragraph properties in a document.

Attributes	Description
leader (Tab Leader Character)	<p>Specifies the character which shall be used to fill in the space created by a tab which ends at this custom tab stop. This character shall be repeated as required to completely fill the tab spacing generated by the tab character.</p> <p>If this attribute is omitted, then no tab leader character shall be used. The possible values for this attribute are defined by the ST_TabTlc simple type (§2.18.91).</p>
pos (Tab Stop Position)	<p>Specifies the position of the current custom tab stop with respect to the current page margins.</p> <p>Negative values are valid and move the tab stop into the current page margin the specified amount. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>

Attributes	Description
val (Tab Stop Type)	<p>Specifies the type of custom tab stop, which determines the behavior of the tab stop and the alignment which shall be applied to text entered at the current custom tab stop.</p> <p>The value of clear is unique and specifies that this tab stop shall be removed when the document is next edited by a consumer which supports rendering the document contents. The possible values for this attribute are defined by the ST_TabJc simple type (§2.18.90).</p>

2.3.1.38 **tabs** (Set of Custom Tab Stops)

This element specifies a sequence of custom tab stops which shall be used for any tab characters in the current paragraph. If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no custom tab stops shall be used for this paragraph.

As well, this property is additive - tab stops at each level in the style hierarchy are added to each other to determine the full set of tab stops for the paragraph. A hanging indent specified via the hanging attribute on the **ind** element (§2.3.1.12) shall also always implicitly create a custom tab stop at its location.

2.3.1.39 **textAlignment** (Vertical Character Alignment on Line)

This element specifies the vertical alignment of all text on each line displayed within a paragraph. If the line height (before any added spacing) is larger than one or more characters on the line, all characters will be aligned to each other as specified by this element.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the vertical alignment of all characters on the line shall be automatically determined by the consumer.

Attributes	Description
val (Vertical Character Alignment Position)	<p>Specifies the type of vertical alignment which shall be used to align the characters on each line in the current paragraph.</p> <p>The possible values for this attribute are defined by the ST_TextAlignment simple type (§2.18.97).</p>

2.3.1.40 **textboxTightWrap** (Allow Surrounding Paragraphs to Tight Wrap to Text Box Contents)

This element specifies whether, for paragraphs in a text box, the surrounding text shall be allowed to overlap with the empty text box boundaries and tight wrap to the extents of the text within the text box.

This element shall only be read for paragraphs which are contained within a text box (have a **txbxContent** ancestor), ignored otherwise.

If the parent text box does not meet the following three criteria, then this property has no effect:

- The text box wrapping must be set to **tight**
- The text box border must not be set
- The text box shading must not be set

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then paragraphs in a text box have no tight wrapping overrides, and text shall wrap to the extents of the text box.

Attributes	Description
val (Lines to Tight Wrap to Paragraph Extents)	<p>Specifies the lines in the parent paragraph which shall allow the text to be tight wrapped to the paragraph (and not the text box) extents when displaying the document.</p> <p>The possible values for this attribute are defined by the ST_TextboxTightWrap simple type (§2.18.98).</p>

2.3.1.41 **textDirection** (Paragraph Text Flow Direction)

This element specifies the direction of the text flow for this paragraph.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the paragraph shall inherit the text flow settings from the parent section.

Attributes	Description
val (Direction of Text Flow)	Specifies the direction of the text flow for this object. The possible values for this attribute are defined by the ST_TextDirection simple type (§2.18.99).

2.3.1.42 **top** (Paragraph Border Above Identical Paragraphs)

This element specifies the border which shall be displayed above a set of paragraphs which have the same set of paragraph border settings.

To determine if any two adjoining paragraphs shall have an individual **top** and **bottom** border or a **between** border, the set of borders on the two adjoining paragraphs are compared. If the border information on those two paragraphs is identical for all possible paragraphs borders, then the **between** border is displayed. Otherwise, the final paragraph shall use its **bottom** border and the following paragraph shall use its **top** border, respectively. If this border specifies a space attribute, that value determines the space above the text (ignoring any spacing above) which should be left before this border is drawn, specified in points.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then no between border shall be applied above identical paragraphs.

Attributes	Description
See 2.3.1.4	

2.3.1.43 **topLinePunct** (Compress Punctuation at Start of a Line)

This element specifies whether punctuation shall be compressed when it appears as the first character in a line, allowing subsequent characters on the line to be move in accordingly.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then punctuation shall not be compressed in this paragraph, even when it appears at the start of a line.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.44 **widowControl** (Allow First/Last Line to Display on a Separate Page)

This element specifies whether a consumer shall prevent a single line of this paragraph from being displayed on a separate page from the remaining content at display time by moving the line onto the following page.

When displaying a paragraph in a page, it is sometimes the case that the first line of that paragraph would display as the last line on one page, and all subsequent lines would display on the following page. This property ensures that a consumer shall move the single line to the following page as well to prevent having one line on its own page. As well, if a single line appears at the top of a page, a consumer shall move the preceding line onto the following page as well, to prevent a single line from being displayed on a separate page.

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this paragraph shall prevent a single line from being shown on a separate page whenever it would normally occur.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.1.45 wordWrap (Allow Line Breaking At Character Level)

This element specifies whether a consumer shall break Latin text which exceeds the text extents of a line by breaking the word across two lines (breaking on the character level) or by moving the word to the following line (breaking on the word level).

If this element is omitted on a given paragraph, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then this paragraph shall break Latin words at the word level, not the character level when it is displayed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2 Run

The next level of the document hierarchy is the *run*, which defines a region of text with a common set of properties, represented by the *r* element (§2.3.2.23). An *r* element allows the producer to specify a single set of formatting properties, applying the same information to all the contents of the run.

Just as a paragraph can have properties, so too can a run. All of the elements inside an *r* element have their properties controlled by a corresponding optional **rPr** run properties element (§2.7.8.1; §2.3.2.26), which must be the first child of the *r* element. In turn, the **rPr** element is a container for a set of property elements that are applied to the rest of the children of the *r* element. [Note: The elements inside the **rPr** container element allow the consumer to control whether the content in the following run content is bold, underlined, or visible, for example. *end note*]

2.3.2.1 b (Bold)

This element specifies whether the bold property shall be applied to all non-complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to *false* (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to *true* or *false* shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then bold shall not be applied to non-complex script characters.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.2 bCs (Complex Script Bold)

This element specifies whether the bold property shall be applied to all complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to *false* (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to *true* or *false* shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then bold shall not be applied to complex script characters.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.3 **bdr** (Text Border)

This element specifies information about the border applied to the text in the current run.

The first piece of information specified by the **bdr** element is that the current shall have a border when displayed. This information is specified simply by the presence of the **bdr** element in run's properties.

The second piece of information concerns the set of runs which share the current run border. This is determined based on the attributes on the **bdr** element. If the set of attribute values specifies on two adjacent runs is identical, then those two runs shall be considered to be part of the same run border group and rendered within the same set of borders in the document.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then no run border shall be applied to the text in this run.

Attributes	Description
See 2.3.1.4	

2.3.2.4 **caps** (Display All Characters As Capital Letters)

This element specifies that any lowercase characters in this text run shall be formatted for display only as their capital letter character equivalents. This property does not affect any non-alphabetic character in this run, and does not change the Unicode character for lowercase text, only the method in which it is displayed.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to **false** (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to **true** or **false** shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the characters are not formatted as capital letters.

This element shall not be present with the **smallCaps** (§2.3.2.31) property on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.5 **color** (Run Content Color)

This element specifies the color which shall be used to display the contents of this run in the document.

This color may be explicitly specified, or set to allow the consumer to automatically choose an appropriate color based on the background color behind the run's content.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the characters are set to allow the consumer to automatically choose an appropriate color based on the background color behind the run's content.

Attributes	Description
themeColor (Run Content Theme Color)	Specifies a theme color which should be applied to the current run. The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows for color information to be set centrally in the document.

Attributes	Description
	<p>If the themeColor attribute is specified, then the val attribute is ignored for this run. The possible values for this attribute are defined by the ST_ThemeColor simple type (§2.18.103).</p>
themeShade (Run Content Theme Color Shade)	<p>Specifies the shade value applied to the supplied theme color (if any) for this run's contents.</p> <p>If the themeShade is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run.</p> <p>The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border.</p> <p>Given a input red, green, or blue color value C (from 0-255), an output color value of C' (from 0-255), and a shade value S (from 0-100), the shade is applied as follows:</p> $C' = \left(1 - \frac{S}{100}\right) C$ <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
themeTint (Run Content Theme Color Tint)	<p>Specifies the tint value applied to the supplied theme color (if any) for this run's contents.</p> <p>If the themeTint is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run.</p> <p>The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border.</p> <p>Given a input red, green, or blue color value C (from 0-255), an output color value of C' (from 0-255), and a tint value T (from 0-100), the tint is applied as follows:</p> $C' = \left(1 - \frac{T}{100}\right) (255 - C) + C$ <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
val (Run Content Color)	<p>Specifies the color for this run.</p> <p>This color may either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the run color as appropriate.</p> <p>If the run specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value.</p> <p>The possible values for this attribute are defined by the ST_HexColor simple type (§2.18.42).</p>

2.3.2.6 cs (Use Complex Script Formatting on Run)

This element specifies whether the contents of this run shall be treated as complex script text regardless of their Unicode character values when determining the formatting for this run.

This means that a consumer shall use the complex script formatting applied to the run when determining the resulting formatting properties.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run contents are set to complex script based on the Unicode character positions of the content.

Attributes	Description
------------	-------------

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.7 **dstrike** (Double Strikethrough)

This element specifies that the contents of this run shall be displayed with two horizontal lines through each character displayed on the line.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then double strikethrough shall not be applied to the contents of this run.

This element shall not be present with the **strike** (§2.3.2.35) property on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.8 **eastAsianLayout** (East Asian Typography Settings)

This element specifies any East Asian typography settings which shall be applied to the contents of the run.

The specific typography settings represented by this element include the *two lines in one* and *horizontal in vertical* text options.

The *two lines in one* setting specifies that the characters in this run should be written out on a single line in the document by creating two sub-lines within the regular line, and laying out this text equally between those sub lines.

The *horizontal in vertical* setting specifies that characters in this run should be rendered with a 90 degree rotation to the left from all other contents of the line when displayed in the document, while keeping the text on the same line as all other text in the paragraph.

Attributes	Description
combine (Two Lines in One)	Specifies whether the contents of the current run should be combined into one line using the two lines in one logic described above in the parent element. If this attribute is omitted, then this run shall not be displayed on two sub lines. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
combineBrackets (Display Brackets Around Two Lines in One)	Specifies that the two lines in one text should be enclosed within a pair of brackets when displayed. This attribute's values determine the bracket style to put around combined text. If this attribute is not specified, then no brackets shall be placed around this content when displayed in the document. If the combine attribute is not specified, then this attribute is ignored. The possible values for this attribute are defined by the ST_CombineBrackets simple type (§2.18.13).
id (East Asian Typography Run ID)	Specifies a unique ID which shall be used to link multiple runs containing eastAsianLayout element to each other to ensure that their contents are correctly displayed in the document. This means that multiple runs which are broken apart due to differences in formatting can be identified as belonging to the same grouping in terms of eastAsianLayout properties, although they are separated into multiple runs of text. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
vert (Horizontal in Vertical (Rotate Text))	Specifies that characters in this run should be rendered with a 270 degree rotation to the left from all other contents of the line when displayed in the document as described above.

Attributes	Description
	If this attribute is omitted, then the contents of this run shall not be rotated with respect to the normal text flow. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
vertCompress (Compress Rotated Text to Line Height)	Specifies whether the rotated text shall be compressed at display time in order to ensure that it fits into the existing line height without increasing the overall height of the line. If the vert attribute is not specified, then this attribute is ignored. If this attribute is omitted, then text shall not be compressed in order to fit into the existing height of the line when it is rotated. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).

2.3.2.9 **effect** (Animated Text Effect)

This element specifies an animated text effect which should be displayed when rendering the contents of this run. This effect is rendered around the extents of the text in the run in the same location as a run border with zero pixels of padding would be rendered (if such a run border was present).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then no text effect shall be applied to the contents of this run.

Attributes	Description
val (Animated Text Effect Type)	Specifies the type of animated text effect which shall be applied to this text run. The possible values for this attribute are defined by the ST_TextEffect simple type (§2.18.100).

2.3.2.10 **em** (Emphasis Mark)

This element specifies the emphasis mark which shall be displayed for each non-space character in this run. An *emphasis mark* is an additional character that is rendered above or below the main character glyph as specified by the contents of the val attribute.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then no emphasis mark shall be added to each character in the contents of this run.

Attributes	Description
val (Emphasis Mark Type)	Specifies the emphasis mark type used for each character in this run. The possible values for this attribute are defined by the ST_Em simple type (§2.18.27).

2.3.2.11 **emboss** (Embossing)

This element specifies that the contents of this run should be displayed as if embossed, which makes text appear as if it is raised off the page in relief.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to *false* (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to *true* or *false* shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then embossing shall not be applied to the contents of this run.

This element shall not be present with either the **imprint** (§2.3.2.16) or **outline** (§2.3.2.21) properties on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.12 **fitText** (Manual Run Width)

This element specifies that the contents of this run shall not be automatically displayed based on the width of its contents, rather its contents shall be resized to fit the width specified by the **val** attribute. This expansion/contraction shall be performed by equally increasing/decreasing the size of each character in this run's contents when displayed. If this element is omitted, then the contents of this run shall be displayed based on the size of its contents.

Attributes	Description
id (Fit Text Run ID)	<p>Specifies a unique ID which shall be used to link multiple contiguous runs containing fitText elements to each other to ensure that their contents are correctly merged into the specified width in the document.</p> <p>This means that multiple runs which are broken apart due to differences in formatting can be identified as belonging to the same grouping in terms of fitText properties, although they are multiple runs of text in the WordprocessingML.</p> <p>If the runs are not contiguous, then the id attribute is ignored, and the runs are not linked.</p> <p>If this attribute is omitted, then this run has no id and shall not be linked with any other run in the parent paragraph.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
val (Value)	<p>This attribute specifies the exact width of space which this run shall be fit into when displayed in the document.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>

2.3.2.13 **highlight** (Text Highlighting)

This element specifies a highlighting color which is applied as a background behind the contents of this run. If this run has any background shading specified using the **shd** element (§2.3.2.30), then the background shading shall be superseded by the highlighting color when the contents of this run are displayed. If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then text highlighting shall not be applied to the contents of this run.

Attributes	Description
val (Highlighting Color)	<p>Specifies the color of the text highlighting which shall be applied to the contents of this run.</p> <p>The possible values for this attribute are defined by the ST_HighlightColor simple type (§2.18.45).</p>

2.3.2.14 **i** (Italics)

This element specifies whether the italic property should be applied to all non-complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to **false** (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to **true** or **false** shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then italics shall not be applied to non-complex script characters.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.15 iCs (Complex Script Italics)

This element specifies whether the italic property should be applied to all complex script characters in the contents of this run when displayed in a document.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to `false` (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to `true` or `false` shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then italics shall not be applied to complex script characters.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.16 imprint (Imprinting)

This element specifies that the contents of this run should be displayed as if imprinted, which makes text appear to be imprinted or pressed into page (also referred to as 'engrave').

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to `false` (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to `true` or `false` shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then imprinting shall not be applied to the contents of this run.

This element shall not be present with either the **emboss** (§2.3.2.11) or **outline** (§2.3.2.21) properties on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.17 kern (Font Kerning)

This element specifies whether font kerning shall be applied to the contents of this run. If it is specified, then kerning shall be automatically adjusted when displaying characters in this run as needed.

The `val` attribute specifies the smallest font size which shall have its kerning automatically adjusted if this setting is specified. If the font size in the **sz** element (§2.3.2.36) is smaller than this value, then no font kerning shall be performed.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then font kerning shall not be applied to the contents of this run.

Attributes	Description
val (Half Point Measurement)	Specifies a positive measurement specified in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element.

Attributes	Description
	The possible values for this attribute are defined by the ST_HpsMeasure simple type (§2.18.47).

2.3.2.18 lang (Languages for Run Content)

This element specifies the languages which shall be used to check spelling and grammar (if requested) when processing the contents of this run.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the languages for the contents of this run shall be automatically determined based on their contents using any method desired.

Attributes	Description
bidl (Complex Script Language)	Specifies the language which shall be used when processing the contents of this run which use complex script characters, as determined by the Unicode character values of the run content. If this attribute is omitted, then the languages for the contents of this run using complex script characters shall be automatically determined based on their contents using any appropriate method. The possible values for this attribute are defined by the ST_Lang simple type (§2.18.50).
eastAsia (East Asian Language)	Specifies the language which shall be used when processing the contents of this run which use East Asian characters, as determined by the Unicode character values of the run content. If this attribute is omitted, then the languages for the contents of this run using East Asian characters shall be automatically determined based on their contents using any appropriate method. The possible values for this attribute are defined by the ST_Lang simple type (§2.18.50).
val (Latin Language)	Specifies the language which shall be used to check spelling and grammar (if requested) when processing the contents of this run which use Latin characters, as determined by the Unicode character values of the run content. If this attribute is omitted, then the languages for the contents of this run using Latin characters shall be automatically determined based on their contents using any appropriate method. The possible values for this attribute are defined by the ST_Lang simple type (§2.18.50).

2.3.2.19 noProof (Do Not Check Spelling or Grammar)

This element specifies that the contents of this run shall not report any errors when the document is scanned for spelling and grammar. [Note: It is entirely at the consumer's/producer's discretion whether this is done by not checking the region for spelling and grammar, or simply by suppressing the results. *end note*]

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then spelling and grammar error shall not be suppressed on the contents of this run.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.20 oMath (Office Open XML Math)

This element specifies that this run contains WordprocessingML which shall be handled as though it was Office Open XML Math.

[Rationale: Like other run properties may be applied to the glyph representing the paragraph mark, it is possible to create an Office Open XML Math equation on an empty paragraph as well. Since that paragraph mark must be defined by

WordprocessingML, it is not possible to store the paragraph using the Office Open XML Math markup. Instead, this run property is stored on the paragraph mark's run properties to indicate that the paragraph mark is part of an Office Open XML Math equation. For example, the first paragraph below is stored as Office Open XML Math:

The paragraph must be a **p** (§2.3.1.22) element, but that would mean the data loss of the Math markup when saving as a WordprocessingML package. In order to prevent that data loss, this property stores the Math property as a run property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then this run shall not be treated as Office Open XML Math.

This property may be applied to any run, but that should only introduce the semantic that the run is math in the user interface, and shall not change the appearance of the text.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.21 **outline** (Display Character Outline)

This element specifies that the contents of this run should be displayed as if they have an outline, by drawing a one pixel wide border around the inside and outside borders of each character glyph in the run.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to **false** (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to **true** or **false** shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then outline shall not be applied to the contents of this run.

This element shall not be present with either the **emboss** (§2.3.2.11) or **imprint** (§2.3.2.16) properties on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.22 **position** (Vertically Raised or Lowered Text)

This element specifies the amount by which text shall be raised or lowered for this run in relation to the default baseline of the surrounding non-positioned text. This allows the text to be repositioned without altering the font size of the contents. If the **val** attribute is positive, then the parent run shall be raised above the baseline of the surrounding text by the specified number of half-points. If the **val** attribute is negative, then the parent run shall be lowered below the baseline of the surrounding text by the specified number of half-points.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the text shall not be raised or lowered relative to the default baseline location for the contents of this run.

Attributes	Description
val (Signed Half-Point Measurement)	Specifies a positive or negative measurement in half-points (1/144 of an inch). The contents of this attribute value are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_SignedHpsMeasure simple type (§2.18.86).

2.3.2.23 **r** (Text Run)

This element specifies a run of content in the parent field, hyperlink, custom XML element, structured document tag, smart tag, or paragraph.

The contents of a run in a WordprocessingML document shall consist of any combination of run content.

Attributes	Description
rsidDel (Revision Identifier for Run Deletion)	<p>Specifies a unique identifier used to track the editing session when the run was deleted from the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidR (Revision Identifier for Run)	<p>Specifies a unique identifier used to track the editing session when the run was added to the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidRPr (Revision Identifier for Run Properties)	<p>Specifies a unique identifier used to track the editing session when the run properties were last modified in the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>

2.3.2.24 **rFonts** (Run Fonts)

This element specifies the fonts which shall be used to display the text contents of this run. Within a single run, there may be up to four types of content present which shall each be allowed to use a unique font:

- ASCII

- High ANSI
- Complex Script
- East Asian

The use of each of these fonts shall be determined by the Unicode character values of the run content, unless manually overridden via use of the **cs** element (§2.3.2.6).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the text shall be displayed in any default font which supports each type of content.

Attributes	Description
ascii (ASCII Font)	<p>Specifies a font which shall be used to format all characters in the ASCII range (0 - 127) within the parent run.</p> <p>If the asciiTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports ASCII content.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
asciiTheme (ASCII Theme Font)	<p>Specifies a theme font which shall be used to format all characters in the ASCII range (0 - 127) within the parent run. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part, which allows for font information to be set centrally in the document.</p> <p>If the ascii attribute is also specified, then that attribute shall be ignored and this value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the ascii attribute.</p> <p>The possible values for this attribute are defined by the ST_Theme simple type (§2.18.102).</p>
cs (Complex Script Font)	<p>Specifies a font which shall be used to format all characters in a complex script Unicode range within the parent run.</p> <p>If the csTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports complex script content.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
csTheme (Complex Script Theme Font)	<p>Specifies a theme font which shall be used to format all characters in a complex script Unicode range within the parent run. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part, which allows for font information to be set centrally in the document.</p> <p>If the cs attribute is also specified, then that attribute shall be ignored and this value shall be used instead.</p>

Attributes	Description
	<p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the cs attribute.</p> <p>The possible values for this attribute are defined by the ST_Theme simple type (§2.18.102).</p>
eastAsia (East Asian Font)	<p>Specifies a font which shall be used to format all characters in an East Asian Unicode range within the parent run.</p> <p>If the eastAsiaTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports East Asian content.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
eastAsiaTheme (East Asian Theme Font)	<p>Specifies a theme font which shall be used to format all characters in an East Asian Unicode range within the parent run. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part, which allows for font information to be set centrally in the document.</p> <p>If the eastAsia attribute is also specified, then that attribute shall be ignored and this value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the eastAsia attribute.</p> <p>The possible values for this attribute are defined by the ST_Theme simple type (§2.18.102).</p>
hAnsi (High ANSI Font)	<p>Specifies a font which shall be used to format all characters in a Unicode range within the parent run which does not fall into one of the three categories defined above, which is called the <i>high ANSI</i> range in WordprocessingML.</p> <p>If the hAnsiTheme attribute is also specified, then this attribute shall be ignored and that value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in any default font which supports high ANSI content.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
hAnsiTheme (High ANSI Theme Font)	<p>Specifies a theme font which shall be used to format all characters in a Unicode range within the parent run which does not fall into one of the three categories defined above, which is called the <i>high ANSI</i> range in WordprocessingML. This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part, which allows for font information to be set centrally in the document.</p> <p>If the hAnsi attribute is also specified, then that attribute shall be ignored and this value shall be used instead.</p> <p>If this attribute is not present, the default value is to leave the formatting applied at previous level in the <i>style hierarchy</i>. If this attribute is never applied in the style hierarchy, then the text shall be displayed in the font specified by the hAnsi attribute.</p>

Attributes	Description
	The possible values for this attribute are defined by the ST_Theme simple type (§2.18.102).
hint (Font Content Type)	<p>Specifies the font type which shall be used to format any ambiguous characters in the current run.</p> <p>There are certain characters which are not explicitly stored in the document, and may be mapped into multiple categories of the four mentioned above. This attribute shall be used to arbitrate that conflict, and determine how ambiguities in this run shall be handled. [<i>Note</i>: This is primarily used to handle the formatting on the paragraph mark glyph, and other characters that are not stored as text in the WordprocessingML document. <i>end note</i>]</p> <p>If this attribute is omitted, then this ambiguity may be resolved by any means available. The possible values for this attribute are defined by the ST_Hint simple type (§2.18.46).</p>

2.3.2.25 **rPr** (Run Properties)

This element specifies a set of run properties which shall be applied to the contents of the parent run after all style formatting has been applied to the text. These properties are defined as *direct formatting*, since they are directly applied to the run and supersede any formatting from styles.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults
- Table styles
- Numbering styles
- Paragraph styles
- Character styles
- Direct formatting (this element)
-

2.3.2.26 **rPr** (Previous Run Properties)

This element specifies a set of run properties which shall be attributed to a revision by a particular author and at a particular time. This element contains the set of properties which have been tracked as a specific set of revisions by one author.

2.3.2.27 **rStyle** (Referenced Character Style)

This element specifies the style ID of the character style which shall be used to format the contents of this paragraph.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults
- Table styles
- Numbering styles
- Paragraph styles
- Character styles (this element)
- Direct Formatting

This means that all properties specified in the **style** element (§2.7.3.17) with a styleId which corresponds to the value in this element's val attribute are applied to the run at the appropriate level in the hierarchy.

If this element is omitted, or it references a style which does not exist, then no character style shall be applied to the current paragraph. As well, this property is ignored if the run properties are part of a character style.

Attributes	Description
val (String Value)	See 2.3.1.27

2.3.2.28 **rtl** (Right To Left Text)

This element specifies that the alignment and reading order for this run shall be right to left. This setting determines the way in which the run contents are presented in the document when punctuation characters are part of the run's contents. When this property is specified, each part of the run between a punctuation mark shall be laid out right to left on the line. If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then right to left alignment shall not be applied to the contents of this run.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.29 **shadow** (Shadow)

This element specifies that the contents of this run shall be displayed as if each character has a shadow, displayed beneath the text and to its right.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to **false** (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to **true** or **false** shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then shadowing shall not be applied to the contents of this run.

This element shall not be present with either the **emboss** (§2.3.2.11) or **imprint** (§2.3.2.16) properties on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.30 **shd** (Run Shading)

Like paragraph shading, this element specifies the shading applied to the contents of the run.

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then run shading shall not be applied to the contents of this run.

Attributes	Description
See 2.3.1.31	

2.3.2.31 **smallCaps** (Small Caps)

This element specifies that all small letter characters in this text run shall be formatted for display only as their capital letter character equivalents in a font size two points smaller than the actual font size specified for this text. This property does not affect any non-alphabetic character in this run, and does not change the Unicode character for lowercase text, only the method in which it is displayed. If this font cannot be made two point smaller than the current size, then it shall be displayed as the smallest possible font size in capital letters.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to `false` (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to `true` or `false` shall set the absolute state of the resulting property. If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the characters are not formatted as capital letters. This element shall not be present with the **caps** (§2.3.2.4) property on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.32 **snapToGrid** (Use Document Grid Settings For Inter-Character Spacing)

This element specifies whether the current run should use the document grid characters per line settings defined in the **docGrid** element (§2.6.5) when laying out the contents in this run. This setting determines whether the additional character pitch specified in the document grid shall be added to each character in this run as specified by the document grid. If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run shall use the document grid setting to lay out text when a document grid is defined for the parent section.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.33 **spacing** (Character Spacing Adjustment)

This element specifies the amount of character pitch which shall be added or removed after each character in this run before the following character is rendered in the document. This property has an effect equivalent to the additional character pitched added by a document grid applied to the contents of a run. If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run shall not have any additional character pitch applied to any character in its contents.

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	Specifies a value whose contents shall contain a positive whole number, whose contents consist of a positive or negative measurement in twentieths of a point (equivalent to 1/1440th of an inch). The contents of this measurement shall be interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).

2.3.2.34 **specVanish** (Paragraph Mark Is Always Hidden)

This element specifies that the given run shall always behave as if it is hidden, even when hidden text is being displayed in the current document.

This property shall only be used to specify that a paragraph mark shall never be used to break the end of a paragraph for display, even if it is being shown on the document, as would be the case if a regularly hidden paragraph was not being displayed in the document. [Note: This property was typically used to ensure that a paragraph style can be applied to a part of a paragraph, and still appear as in the Table of Contents (which in previous word processors would ignore the use of the style if it were being used as a character style. *end note*] If this element is applied to any other run, it may be ignored.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run properties for the paragraph mark shall not always be treated as if hidden.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.35 **strike** (Single Strikethrough)

This element specifies that the contents of this run shall be displayed with a single horizontal line through the center of the line.

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to `false` (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to `true` or `false` shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then strikethrough shall not be applied to the contents of this run. This element shall not be present with the **dstrike** (§2.3.2.7) property on the same run, since they are mutually exclusive in terms of appearance.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.36 **sz** (Font Size)

This element specifies the font size which shall be applied to all non complex script characters in the contents of this run when displayed.

If this element is not present, the default value is to leave the value applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then any appropriate font size may be used for non complex script characters.

Attributes	Description
val (Half Point Measurement)	See 2.3.2.17

2.3.2.37 **szCs** (Complex Script Font Size)

This element specifies the font size which shall be applied to all complex script characters in the contents of this run when displayed.

If this element is not present, the default value is to leave the value applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then any appropriate font size may be used for complex script characters.

Attributes	Description
val (Half Point Measurement)	See 2.3.2.17

2.3.2.38 **u** (Underline)

This element specifies that the contents of this run should be displayed along with an underline appearing directly below the character height (less all spacing above and below the characters on the line).

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then an underline shall not be applied to the contents of this run.

Attributes	Description
color (Underline Color)	<p>Specifies the color for the underlining on this run.</p> <p>This color may either be presented as a hex value (in RRGGBB format), or auto to allow a consumer to automatically determine the underline color as appropriate.</p> <p>If the underline specifies the use of a theme color via the themeColor attribute, then this value is superseded by the theme color value.</p> <p>The possible values for this attribute are defined by the ST_HexColor simple type (§2.18.42).</p>
themeColor (Underline Theme Color)	<p>Specifies a theme color which should be applied to the current underline.</p> <p>The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows for color information to be set centrally in the document.</p> <p>If the themeColor attribute is specified, then the color attribute is ignored for this underline.</p> <p>The possible values for this attribute are defined by the ST_ThemeColor simple type (§2.18.103).</p>
themeShade (Underline Theme Color Shade)	<p>Specifies the shade value applied to the supplied theme color (if any) for this underline.</p> <p>If the themeShade is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this underline.</p> <p>The themeShade value is stored as a hex encoding of the shade value (from 0 to 255) applied to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> Convert the color to the HSL color format (values from 0 to 1) Modify the luminance factor as follows: $L' = L * \text{Shade}_{\text{percentage}}$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
themeTint (Underline Theme Color Tint)	<p>Specifies the tint value applied to the supplied theme color (if any) for this underline's contents.</p> <p>If the themeTint is supplied, then it is applied to the RGB value of the theme color to determine the final color applied to this run.</p> <p>The themeTint value is stored as a hex encoding of the tint value (from 0 to 255) applied to the current border.</p> <p>Given an RGB color defined as three hex values in RRGGBB format, the shade is applied as follows:</p> <ul style="list-style-type: none"> Convert the color to the HSL color format (values from 0 to 1) Modify the luminance factor as follows: $L' = L * \text{Tint}_{\text{pct}} + (1 - \text{Tint}_{\text{pct}})$ <p>Convert the resultant HSL color to RGB</p> <p>The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).</p>
val (Underline Style)	Specifies the pattern which shall be used to create the underline applied beneath the text in this

Attributes	Description
	run. Each of these possible patterns are shown in the simple type referenced below. The possible values for this attribute are defined by the ST_Underline simple type (§2.18.106).

2.3.2.39 **vanish** (Hidden Text)

This element specifies whether the contents of this run shall be hidden from display at display time in a document. [Note: The setting should affect the normal display of text, but an application may have settings to force hidden text to be displayed. *end note*]

This formatting property is a *toggle property*, which specifies that its behavior differs between its use within a style definition and its use as direct formatting. When used as part of a style definition, setting this property shall toggle the current state of that property as specified up to this point in the hierarchy (i.e. applied to not applied, and vice versa). Setting it to `false` (or an equivalent) shall result in the current setting remaining unchanged. However, when used as direct formatting, setting this property to `true` or `false` shall set the absolute state of the resulting property.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then this text shall not be hidden when displayed in a document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.2.40 **vertAlign** (Subscript/Superscript Text)

This element specifies the alignment which shall be applied to the contents of this run in relation to the default appearance of the run's text. This allows the text to be repositioned as subscript or superscript without altering the font size of the run properties.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the text shall not be subscript or superscript relative to the default baseline location for the contents of this run.

Attributes	Description
val (Subscript/Superscript Value)	Specifies the type of vertical alignment applied to the contents of the current run. The possible values for this attribute are defined by the ST_VerticalAlignRun simple type (§2.18.109).

2.3.2.41 **w** (Expanded/Compressed Text)

This element specifies the amount by which each character shall be expanded or when the character is rendered in the document. This property has an of stretching or compressing each character in the run, as opposed to the **spacing** element (§2.3.2.33) which expands/compresses the text by adding additional character pitch but not changing the width of the actual characters displayed on the line.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then the run shall be displayed at 100% of its normal width.

Attributes	Description
val (Text Expansion/Compression Value)	Specifies that the percentage by which the contents of this run shall be expanded or compressed with respect to its normal (100%) character width. If this attribute is omitted, then the contents of this run shall be displayed at 100% of its normal size. The possible values for this attribute are defined by the ST_TextScale simple type (§2.18.101).

2.3.2.42 **webHidden** (Web Hidden Text)

This element specifies whether the contents of this run shall be hidden from display at display time in a document when the document is being displayed in a web page view. [*Note*: The setting should affect the normal display of text in a web page view, but an application may have settings to force hidden text to be displayed. *end note*] As well, this setting should not affect a normal paginated view of the document.

If this element is not present, the default value is to leave the formatting applied at previous level in the *style hierarchy*. If this element is never applied in the style hierarchy, then this text shall not be hidden when displayed in a document in a web page view.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.3 Run Content

The final level of the document hierarchy is *run content*, which is defined as the set of elements which can be contained as the contents of a particular run in a document.

[*Note*: Types of run content in WordprocessingML include:

- Text
- Field Codes
- DrawingML objects
- VML objects
- Fields

end note]

2.3.3.1 **br** (Break)

This element specifies that a break shall be placed at the current location in the run content. A *break* is a special character which is used to override the normal line breaking that would be performed based on the normal layout of the document's contents.

The behavior of this break character (the location where text shall be restarted after this break) shall be determined by its type and clear attribute values, described below.

Attributes	Description
clear (Restart Location For Text Wrapping Break)	<p>Specifies the location which shall be used as the next available line when the break's type attribute has a value of <code>textWrapping</code>. This property only affects the restart location when the current run is being displayed on a line which does not span the full text extents due to the presence of a floating object (see possible values for details).</p> <p>If this break is not of type <code>textWrapping</code>, then this attribute shall be ignored. If this attribute is omitted, then its value shall be assumed to be <code>none</code> if needed.</p> <p>The possible values for this attribute are defined by the <code>ST_BrClear</code> simple type (§2.18.5).</p>
type (Break Type)	<p>Specifies the type of the current break. The break type determines the next location where text shall be placed after this manual break is applied to the text contents (see possible values for details).</p> <p>If this attribute is omitted, then it shall be assumed to be of type <code>textWrapping</code>.</p> <p>The possible values for this attribute are defined by the <code>ST_BrType</code> simple type (§2.18.6).</p>

2.3.3.2 control (Floating Embedded Control)

This element specifies that the parent VML object is a representation of an embedded control at the current location in the document. This element shall be used to associate the VML data with the appropriate embedded control settings and properties when the document is displayed.

If the embedded control is not present, cannot be loaded due to application settings, or is not supported, then the VML data shall be used to provide an image representation of the control at the appropriate location in the document.

Attributes	Description
id (Embedded Control Properties Relationship Reference) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID for the relationship which contains the properties for this embedded control. This property bag is contained in a separate part within the Word Open XML package. The relationship explicitly targeted by this attribute shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/control or the document shall be considered non-conformant. If this attribute is omitted, then the embedded control shall be given no property bag when instantiated. The possible values for this attribute are defined by the ST_RelationshipId simple type (\$Error! Reference source not found.).
name (Unique Name for Embedded Control)	Specifies a unique name for this embedded control. This name shall be unique across all controls in this document. The possible values for this attribute are defined by the ST_String simple type (\$2.18.88).
shapeid (Associated VML Data Reference)	Specifies the shape ID for a shape which shall be used to define the presentation and location of this embedded control within the document if the control is floating using the VML syntax. [Note: This positioning data is sufficient to display the control in any case where: <ul style="list-style-type: none"> • The embedded control is not on the current machine • Embedded controls are disabled • Embedded controls of this type are not supported <i>end note</i> This shape ID reference is resolved by looking for a VML shape element (\$Error! Reference source not found.) whose id attribute matches the value specified within this attribute. If no such shape exists, then the control shall be rendered inline in the document content at the current run content location. If this attribute is omitted, then this embedded control shall be displayed inline in the current location in the parent run. The possible values for this attribute are defined by the ST_String simple type (\$2.18.88).

2.3.3.3 control (Inline Embedded Control)

This element specifies that the parent embedded object is a representation of an embedded control at the current location in the document. This element shall be used to associate the appropriate embedded control settings and properties when the document is displayed.

If the embedded control is not present, cannot be loaded due to application settings, or is not supported, then a suitable placeholder image shall be used to provide a representation of the presence of an embedded control at the appropriate location in the document.

Attributes	Description
------------	-------------

Attributes	Description
See 2.3.3.2	

2.3.3.4 **cr** (Carriage Return)

This element specifies that a carriage return shall be placed at the current location in the run content. A *carriage return* is the equivalent of Unicode character 000D, and is used to end the current line of text in WordprocessingML.

The behavior of a carriage return in run content shall be identical to a break character with null type and clear attributes, which shall end the current line and find the next available line on which to continue.

2.3.3.5 **dayLong** (Date Block - Long Day Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs) .

[Note: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

The long day format date block shall use a date picture of DDDD, retrieving the long day format for the primary editing language.

2.3.3.6 **dayShort** (Date Block - Short Day Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs) .

[Note: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

The short day format date block shall use a date picture of DD, retrieving the short day format for the primary editing language.

2.3.3.7 **delText** (Deleted Text)

This element specifies that this run contains literal text which shall be displayed in the document. The **delText** element shall be used for all text runs which are part of a region of text that is contained in a deleted region using the **del** element (§2.13.5.12).

Attributes	Description
space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	Specifies how white space should be handled for the contents of this element using the W3C space preservation rules. The possible values for this attribute are defined by the <code>type</code> in the <code>namespace</code> .

2.3.3.8 **dirty** (Invalidated Field Cache)

This element specifies that the field has been changed and the results shall be updated on open in a conforming consumer.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.3.3.9 **drawing** (DrawingML Object)

This element specifies that a DrawingML object is located at this position in the run's contents. The layout properties of this DrawingML object are specified using the WordprocessingML Drawing syntax (**Error! Reference source not found.**).

2.3.3.10 **hps** (Phonetic Guide Text Font Size)

This element specifies the font size which shall be applied to the phonetic guide text in the contents of this run when displayed.

If this element disagrees with the run properties on the phonetic guide text **rt** element (§2.3.3.23), then those properties shall be ignored and this element shall determine the size of the phonetic guide text.

Attributes	Description
val (Half Point Measurement)	See 2.3.2.17

2.3.3.11 **hpsBaseText** (Phonetic Guide Base Text Font Size)

This element specifies the font size which shall be applied to the base text of this phonetic guide text when displayed. If this element disagrees with the run properties on the phonetic guide base text **rubyBase** element (§2.3.3.26), then this property shall be ignored and the **sz** element (§2.3.2.36) in that run shall determine the size of the phonetic guide base text.

Attributes	Description
val (Half Point Measurement)	See 2.3.2.17

2.3.3.12 **hpsRaise** (Distance Between Phonetic Guide Text and Phonetic Guide Base Text)

This element specifies the distance which shall be left between the phonetic guide base text and the phonetic guide text when this phonetic guide text is displayed.

Attributes	Description
val (Half Point Measurement)	See 2.3.2.17

2.3.3.13 **lastRenderedPageBreak** (Position of Last Calculated Page Break)

This element specifies that this position delimited the end of a page when this document was last saved by an application which paginates its content.

[*Guidance*: This element shall be used by applications to specify the locations of page breaks within a document when it is saved as WordprocessingML, in order to allow other applications (e.g. assistive software) to utilize this information when reading the document. *end guidance*]

2.3.3.14 **lid** (Language ID for Phonetic Guide)

This element specifies the language which shall be for this phonetic guide.

Attributes	Description
val (Language Code)	Specifies an ISO 639-1 letter code or 4 digit hexadecimal code for a specific language. This code is interpreted in the context of the parent XML element. The possible values for this attribute are defined by the ST_Lang simple type (§2.18.50).

2.3.3.15 **monthLong** (Date Block - Long Month Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs).

[*Note:* The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

The long month format date block shall use a date picture of MMMM, retrieving the long month format for the primary editing language.

2.3.3.16 **monthShort** (Date Block - Short Month Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs).

[*Note:* The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

The short month format date block shall use a date picture of MM, retrieving the short month format for the primary editing language.

2.3.3.17 **movie** (Embedded Video)

This element specifies a location within a document where the specified parent image shall be treated as a static placeholder for an embedded movie. The specified movie file's contents should be displayed when requested at this location in the document. The location of the embedded movie to be displayed when supported shall be specified by the relationship whose Id attribute matches the id attribute on this element.

If the relationship type of the relationship specified by this element is not

<http://schemas.openxmlformats.org/officeDocument/2006/movie>, or is not present, then the document shall be considered non-conformant. If an application cannot process external content of the content type specified by the targeted part, then it may be ignored.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	Specifies the relationship ID to a specified part. The specified relationship shall match the type required by the parent element: <ul style="list-style-type: none"> • http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element • http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element • http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element The possible values for this attribute are defined by the ST_RelationshipId simple type (\$Error! Reference source not found.).

2.3.3.18 **noBreakHyphen** (Non Breaking Hyphen Character)

This element specifies that a non breaking hyphen character shall be placed at the current location in the run content. A *non breaking hyphen* is the equivalent of Unicode character 002D (the hyphen-minus), however it shall not be used as a valid line breaking character for the current line of text when displaying this WordprocessingML content.

The behavior of a non breaking hyphen in run content shall be to display using the same glyph as the hyphen-minus character, however without being a valid line breaking position (unlike the hyphen-minus character).

2.3.3.19 **object** (Inline Embedded Object)

This element specifies that an embedded object is located at this position in the run's contents. The layout properties of this embedded object are specified using the VML syntax (§Error! Reference source not found.).

Attributes	Description
dxaOrig (Original Image Width)	Specifies the original (natural) width of the image representation of the current control within the document. Some vector image formats do not store a native size within their format, and this attribute shall only be used in those cases to store this information, so that the image may be appropriately restored as needed. If this element is excluded, then the natural size of the image as stored in its format shall be used. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
dyaOrig (Original Image Height)	Specifies the original (natural) height of the image representation of the current control within the document. Some vector image formats do not store a native size within their format, and this attribute shall only be used in those cases to store this information, so that the image may be appropriately restored as needed. If this element is excluded, then the natural size of the image as stored in its format shall be used. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).

2.3.3.20 **pgNum** (Page Number Block)

This element specifies the presence of a page number block at the current location in the run content. A *page number block* is a non-editable region of text which shall display the current page using ascending decimal numbers. [Note: The page number block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the PAGENUM field is used in its place. *end note*]

A page number block shall be displayed using ascending decimal numbers, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

2.3.3.21 **pict** (VML Object)

This element specifies that an object is located at this position in the run's contents. The layout properties of this object are specified using the VML syntax (§Error! Reference source not found.).

2.3.3.22 **ptab** (Absolute Position Tab Character)

This element specifies that an absolute position tab character shall be placed at the current location in the run content. An *absolute position tab* is a character which is used to advance the position on the current line of text when displaying this WordprocessingML content, using the following logic:

Regardless of any number of custom tab stops defined using the **tabs** element (§2.3.1.38), the absolute position tab character shall advance to the position specified by its alignment and relativeTo attributes. The resulting end position of the tab character shall not be affected by the addition of any custom tab stops or changes to the value of the **defaultTabStop** element (§2.15.1.24).

If the alignment location specified by the positional tab cannot be found on the current line, because the starting location is past that point, then the tab character shall advance to that location on the next available line in the document.

Attributes	Description
alignment (Positional Tab Stop Alignment)	Specifies the location of the positional tab stop on the line, as well as the alignment which shall be applied to text subsequent to the current positional tab stop.

Attributes	Description
	The possible values for this attribute are defined by the ST_PTabAlignment simple type (§2.18.77).
leader (Tab Leader Character)	Specifies the character which shall be used to fill in the space created by a positional tab. This character shall be repeated as required to completely fill the tab spacing generated by the positional tab character. The possible values for this attribute are defined by the ST_PTabLeader simple type (§2.18.78).
relativeTo (Positional Tab Base)	Specifies the extents which shall be used to calculate the absolute positioning of this positional tab character. The possible values for this attribute are defined by the ST_PTabRelativeTo simple type (§2.18.79).

2.3.3.23 **rt** (Phonetic Guide Text)

This element specifies the presence of the guide text within a phonetic guide at the current location in the document. The contents of the guide text run are specified in the child **r** element (§2.3.2.23).

2.3.3.24 **ruby** (Phonetic Guide)

This element specifies the presence of a phonetic guide at the current location in the document. A *phonetic guide* (often called ruby text) is a run of content with base text which appears at the normal baseline location for text in this run, with phonetic guide text displayed above it in the document. The resulting construct is called a phonetic guide as it is typically used to map words in one language to another phonetically.

The base text is stored in the **rubyBase** element (§2.3.3.26) and the guide text is stored in the **rt** element (§2.3.3.23).

2.3.3.25 **rubyAlign** (Phonetic Guide Text Alignment)

This element specifies the alignment setting which shall be used to determine the placement of phonetic guide text with respect to the base text when this phonetic guide is displayed.

Attributes	Description
val (Phonetic Guide Text Alignment Value)	Specifies the type of alignment to be applied to the phonetic guide text. The possible values for this attribute are defined by the ST_RubyAlign simple type (§2.18.81).

2.3.3.26 **rubyBase** (Phonetic Guide Base Text)

This element specifies the presence of the base text within a phonetic guide at the current location in the document. The contents of the base text run are specified in the child **r** element (§2.3.2.23).

2.3.3.27 **rubyPr** (Phonetic Guide Properties)

This element specifies a set of properties which determine the behavior and appearance of a phonetic guide within the document.

2.3.3.28 **softHyphen** (Optional Hyphen Character)

This element specifies that an optional hyphen character shall be placed at the current location in the run content. An *optional hyphen* is a character which may be used as a valid line breaking character for the current line of text when displaying this WordprocessingML content, using the following logic:

- When this character is not the character which is used to break the line, then it shall not change the normal display of text (it shall have zero width)
- When this character is the character used to break the line, it shall display using the hyphen-minus character within the display of text

[Note: This character is typically used to mark locations where a word may optionally be hyphenated without causing the hyphen character to be displayed unnecessarily. *end note*]

2.3.3.29 **sym** (Symbol Character)

This element specifies the presence of a symbol character at the current location in the run's content. A *symbol character* is a special character within a run's content which does not use any of the run fonts specified in the **rFonts** element (§2.3.2.24) (or by the style hierarchy).

Instead, this character shall be determined by pulling the character with the hexadecimal value specified in the **char** attribute from the font specified in the **font** attribute.

Attributes	Description
char (Symbol Character Code)	<p>Specifies the hexadecimal code for the Unicode character value of the symbol.</p> <p>When this value is stored in the char attribute, it may be stored in either of the following two formats:</p> <p>Directly in its Unicode character value from the font glyph</p> <p>In a Unicode character value created by adding F000 to the actual character value, shifting the character value of this character into the Unicode private use area.</p> <p>[Note: The use of the latter syntax allows for interoperability with legacy word processing formats, as they used this technique to store the fact that a particular character or set of characters came from a font which was not Unicode compliant, and therefore any font matching performed on this range (if the specified font was not present) would be undesirable, as the resulting glyphs and their appearance could not be predicted. <i>end note</i>]</p> <p>The possible values for this attribute are defined by the ST_ShortHexNumber simple type (§2.18.85).</p>
font (Symbol Character Font)	<p>Specifies a font which shall be used to format this symbol character.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>

2.3.3.30 **t** (Text)

This element specifies that this run contains literal text which shall be displayed in the document. The **t** element shall be used for all text runs which is not:

- Part of a region of text that is contained in a deleted region using the **del** element (§2.13.5.12)
- Part of a region of text that is contained within a field code

Attributes	Description
<p>space (Content Contains Significant Whitespace)</p> <p>Namespace: http://www.w3.org/XML/1998/namespace</p>	See 2.3.3.7

2.3.3.31 **tab** (Tab Character)

This element specifies that a tab character shall be placed at the current location in the run content. An *tab* is a character which is used to advance the position on the current line of text when displaying this WordprocessingML content, using the following logic:

- When there are one or more custom tab stops defined using the **tabs** element (§2.3.1.38), then the tab character shall advance to the next custom tab stop location which is further along than the starting location of the tab
- When there are no custom tab stops which are further than the current position in the line, the tab character shall advance to the nearest multiple of the **defaultTabStop** element (§2.15.1.24) width value.

2.3.3.32 **yearLong** (Date Block - Long Year Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs).

[*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

The long year format date block shall use a date picture of YYYY, retrieving the long year format for the primary editing language.

2.3.3.33 **yearShort** (Date Block - Short Year Format)

This element specifies the presence of a date block at the current location in the run content. A *date block* is a non-editable region of text which shall display the current date filtered through the specified date picture (see following paragraphs).

[*Note*: The date block is a legacy construct used for compatibility with older word processors, and should not be produced unless it was consumed while reading a document – it is recommended that the DATE field is used in its place. *end note*]

A date block shall be displayed using the primary editing language of the host application, regardless of the languages specified in the parent run's **lang** property (§2.3.2.18).

The short year format date block shall use a date picture of YY, retrieving the short year format for the primary editing language.

2.4 Tables

Another type of block-level content in WordprocessingML, a *table* is a set of paragraphs (and other block-level content) arranged in *rows* and *columns*. Tables in WordprocessingML are defined via the **tbl** element, which is analogous to the HTML <table> tag. The table element specifies the location of a table present in the document.

A **tbl** element (§2.4.36) has two elements that define its properties:

- **tblPr** (§2.4.55), which defines the set of table-wide properties (such as style and width)
- **tblGrid** (§2.4.44), which defines the grid layout of the table.

A **tbl** element can also contain an arbitrary non-zero number of rows, where each row is specified with a **tr** element (§2.4.75). Each **tr** element can contain an arbitrary non-zero number of cells, where each cell is specified with a **tc** element (§2.4.62).

2.4.1 **bidirectional** (Visually Right to Left Table)

This element specifies that the cells with this table shall be visually represented in a right to left direction. This element also affects the application of all table-level properties.

When this property is specified, then the ordering of all cells (and table-level properties) in this table shall be applied to the table assuming that the table is a normal left to right table, but the table cells shall be displayed in a right to left direction.

If this element is omitted, then the table shall not be presented right to left.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.2 **bottom** (Table Cell Bottom Margin Exception)

This element specifies the amount of space which shall be left between the bottom extent of the cell contents and the border of a specific table cell within a table. This setting shall override the table cell bottom margin definition specified by the **bottom** element contained within the table properties (§2.4.5).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If omitted, then this table cell shall use the bottom cell margins defined in the **bottom** element contained within the table properties (§2.4.5).

Attributes	Description
type (Table Width Type)	Specifies the units of the width property being defined by the parent element's w attribute. This property is used to define various properties of a table, including: cell spacing, preferred width, and table margins. If this attribute is omitted, then its value shall be assumed to be dxa (twentieths of a point). The possible values for this attribute are defined by the ST_TblWidth simple type (§2.18.96).
w (Table Width Value)	Specifies the value of the width property being defined by the parent element. This property is used to define various properties of a table, including: cell spacing, preferred widths, and table margins. If this attribute is omitted, then its value shall be assumed to be 0 . The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.4.3 **bottom** (Table Cell Bottom Border)

This element specifies the border which shall be displayed at the bottom of the current table cell. The appearance of this table cell border in the document shall be determined by the following settings:

- If the net **tblCellSpacing** element value (§2.4.41;§2.4.42;§2.4.43) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the bottom of this table cell shall not have a cell border, and its border may use the table's border settings as appropriate.

Attributes	Description
See 2.3.1.4	

2.4.4 **bottom** (Table Bottom Border)

This element specifies the border which shall be displayed at the bottom of the current table. The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the bottom of this table shall have the border specified by the associated table style. If no bottom border is specified in the style hierarchy, then this table shall not have a bottom border.

Attributes	Description
See 2.3.1.4	

2.4.5 **bottom** (Table Cell Bottom Margin Default)

This element specifies the amount of space which shall be left between the bottom extent of the cell contents and the border of all table cells within the parent table (or table row). This setting may be overridden by the table cell bottom margin definition specified by the **bottom** element contained within the table cell's properties (§2.4.2).

This value is specified in the units applied via its **type** attribute. Any width value of type **pct** or **auto** for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a bottom margin is never specified in the style hierarchy, then this table shall have no bottom cell padding by default (excepting individual cell overrides).

Attributes	Description
See 2.4.2	

2.4.6 cantSplit (Table Row Cannot Break Across Pages)

This element specifies whether the contents within the current cell shall be rendered on a single page. When displaying the contents of a table cell (such as the table cells in this specification), it is possible that a page break would fall within the contents of a table cell, causing the contents of that cell to be displayed across two different pages. If this property is set, then all contents of a table row shall be rendered on the same page by moving the start of the current row to the start of a new page if necessary. If the contents of this table row cannot fit on a single page, then this row shall start on a new page and flow onto multiple pages as necessary.

If this element is not present, the default behavior is dictated by the setting in the associated table style. If this property is not specified in the style hierarchy, then this table row shall be allowed to split across multiple pages.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.7 cnfStyle (Table Cell Conditional Formatting)

This element specifies the set of conditional table style formatting properties which have been applied to this table cell. [Note: This property is an optimization which is used by consumers to determine if a given property on a table cell is the result of the table style conditional formatting properties vs. direct formatting on the table cell itself. It specifies the components of the conditional formatting in the table style applied to this cell, so that the table's conditional formatting can be applied after the document is displayed without having the table style properties override the style hierarchy. *end note*]

If this element is omitted, then its value shall be assumed to be zero for all entries in the bit mask.

Attributes	Description
val (Conditional Formatting Bit Mask)	See 2.3.1.8

2.4.8 cnfStyle (Table Row Conditional Formatting)

This element specifies the set of conditional table style formatting properties which have been applied to this table row. [Note: This property is an optimization which is used by consumers to determine if a given property on a table row is the result of the table style conditional formatting properties vs. direct formatting on the table cell itself. It specifies the components of the conditional formatting in the table style applied to this cell, so that the table's conditional formatting can be applied after the document is displayed without having the table style properties override the style hierarchy. *end note*]

If this element is omitted, then its value shall be assumed to be zero for all entries in the bit mask.

Attributes	Description
val (Conditional Formatting Bit Mask)	See 2.3.1.8

2.4.9 divId (Associated HTML div ID)

This element specifies the HTML div information which is associated with the current table row. This information, stored in the Web Settings part, is used to associate one or more table rows with a particular HTML div element. [Note: This property is used when saving an HTML document into the WordprocessingML format in order to prevent a loss of all HTML div information, so that the document can later be saved back into HTML format and have the stored information replaced, since the HTML div can store formatting properties on arbitrary regions. *end note*]

In order to determine the associated HTML `div` properties, the value of the `val` attribute on this element is used to look up an associated **div** element (§2.15.2.8) whose `id` attribute matches this value.

If this table row does not have a **div** element present, then this table row shall not have any associated HTML `div` information. If this element is present, but the `val` attribute specifies an `id` value which does not have an associated **div** element, then this element is ignored.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.4.10 `gridAfter` (Grid Columns After Last Cell)

This element specifies the number of grid columns in the parent table's table grid (§2.4.44; §2.4.45) which shall be left after the last cell in the table row.

If this element conflicts with the remaining size of the document grid after all table cells in this row have been added to the grid, then it shall be ignored. If this element is not specified, then its value shall be assumed to be zero grid units.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.4.11 `gridBefore` (Grid Columns Before First Cell)

This element specifies the number of grid columns in the parent table's table grid (§2.4.44; §2.4.45) which shall be skipped before the contents of this table row (its table cells) are added to the parent table. [Note: This property is used to specify tables whose leading edge (left for left-to-right tables, right for right-to-left tables) does not start at the first grid column (the same shared edge). *end note*]

If this element is omitted, then its value shall be assumed to be zero grid units. If this element's value is larger than the size of the table grid, then the value shall be ignored and the first cell in the row may span the full table grid (i.e. the second cell, if one exists, should start at the last shared edge in the table).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.4.12 `gridCol` (Grid Column Definition)

This element specifies the presence and details about a single grid column within a table grid. A *grid column* is a logical column in a table used to specify the presence of a shared vertical edge in the table. When table cells are then added to this table, these shared edges (or grid columns, looking at the column between those shared edges) determine how table cells are placed into the table grid.

If the table grid does not match the requirements of one or more rows in the table (i.e. it does not define enough grid columns), then the grid may be redefined as needed when the table is processed.

Attributes	Description
w (Grid Column Width)	<p>Specifies the width of this grid column.</p> <p>[Note: This value does not solely determine the actual width of the resulting grid column in the document. When the table is displayed in a document, these widths determine the initial width of each grid column, which may then be overridden by:</p> <ul style="list-style-type: none"> The table layout algorithm (§2.4.49; §2.4.50) applied to the current table row(s)

Attributes	Description
	<ul style="list-style-type: none"> The preferred widths of specific cells which are part of that grid column as the table is displayed (which is an input to the algorithm above) <i>end note]</i> This value is specified in twentieths of a point. If this attribute is omitted, then the last saved width of the grid column is assumed to be zero. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).

2.4.13 gridSpan (Grid Columns Spanned by Current Table Cell)

This element specifies the number of grid columns in the parent table's table grid which shall be spanned by the current cell. This property allows cells to have the appearance of being merged, as they span vertical boundaries of other cells in the table.

If this element is omitted, then the number of grid units spanned by this cell shall be assumed to be one. If the number of grid units specified by the val attribute exceeds the size of the table grid, then the table grid shall be augmented as needed to create the number of grid columns required.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.4.14 hidden (Hidden Table Row Marker)

This element specifies that the glyph representing the end character of current table row shall not be displayed in the current document.

[*Note*: This setting is used to hide the end of row glyph in order to ensure that the entire table row is hidden and not displayed in the document, as if any part of the row is visible, the row is displayed. *end note*]

[*Note*: Applications may have settings which allow hidden content to be displayed, in which case this content may be visible - this property is not meant to supersede that setting. *end note*]

If this element is omitted, then this table row shall not be hidden in the document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.15 hideMark (Ignore End Of Cell Marker In Row Height Calculation)

This element specifies whether the end of cell glyph shall influence the height of the given table row in the table. If it is specified, then only printing characters in this cell shall be used to determine the row height.

[*Rationale*: Typically, the height of a table row is determined by the height of all glyphs in all cells in that row, including the non-printing end of cell glyph characters. However, if these characters are not formatted, they are always created with the document default style properties. This means that the height of a table row cannot ever be reduced below the size of the end of cell marker glyph without manually formatting each paragraph in that run.

In a typical document, this behavior is desirable as it prevents table rows from 'disappearing' if they have no content. However, if a table row is being used as a border (for example, by shading its cells or putting an image in them), then this behavior makes it impossible to have a virtual border that is reasonably small without formatting each cell's content directly. This setting specifies that the end of cell glyph shall be ignored for this cell, allowing it to collapse to the height of its contents without formatting each cell's end of cell marker, which would have the side effect of formatting any text ever entered into that cell. *end rationale*]

If this element is omitted, then the end of cell marker shall be included in the determination of the height of this row.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.16 hMerge (Horizontally Merged Cell)

This element specifies that this cell is part of a horizontally merged set of cells in a table. The **val** attribute on this element determines how this cell is defined with respect to the previous cell in the table (i.e. does this cell continue the horizontal merge or start a new merged group of cells).

[Note: This property is maintained for compatibility with legacy word processing documents which defined tables in this manner. Whenever possible, this form or horizontal merges should not be produced, and should be translated to the appropriate **gridSpan** (§2.4.13) settings on the table cells instead. *end note*]

If this element is omitted, then this cell shall not be part of any horizontally merged grouping of cells, and any horizontal merge group in the preceding cells shall be closed.

Attributes	Description
val (Horizontal Merge Type)	Specifies how the table cell is part of a horizontally merged region. This determine whether the cell should join onto an existing grouping of merged cells if any exist, or start a new group of merged cells. Refer to the simple type definition for a full description of each type. If this attribute is omitted, its value shall be assumed to be <code>continue</code> . The possible values for this attribute are defined by the <code>ST_Merge</code> simple type (§2.18.63).

2.4.17 insideH (Table Inside Horizontal Edges Border)

This element specifies the border which shall be displayed on all horizontal table cell borders which are not on an outmost edge of the parent table (all horizontal borders which are not the topmost or bottommost border). The appearance of this table cell border in the document shall be determined by the following settings:

- The display of the border on interior edges is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the inside horizontal borders of this table shall have the border specified by the associated table style. If no inside horizontal edge border is specified in the style hierarchy, then this table shall not have an inside horizontal edge border.

Attributes	Description
See 2.3.1.4	

2.4.18 insideH (Table Cell Inside Horizontal Edges Border)

This element specifies the border which shall be displayed on all interior horizontal edges of the current group of table cells.

[Note: Although individual table cells have no concept of an internal edge, which would render this property useless in most cases, it is used to determine the cell borders to apply to a specific group of cells as part of table conditional formatting in a table style, for example, the inside horizontal edges on the set of cells in the first column. *end note*]

The appearance of this table cell border in the document shall be determined by the following settings:

- If the net **tblCellSpacing** element value (§2.4.41;§2.4.42;§2.4.43) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the specified conditional formatting on the table shall not change the current set of internal edge borders on its set of table cells (i.e. their current setting shall remain unchanged).

Attributes	Description
------------	-------------

Attributes	Description
See 2.3.1.4	

2.4.19 **insideV (Table Cell Inside Vertical Edges Border)**

This element specifies the border which shall be displayed on all interior vertical edges of the current group of table cells. [Note: Although individual table cells have no concept of an internal edge, which would render this property useless in most cases, it is used to determine the cell borders to apply to a specific group of cells as part of table conditional formatting in a table style, for example, the inside vertical edges on the set of cells in the header row. *end note*]

The appearance of this table cell border in the document shall be determined by the following settings:

- If the net **tblCellSpacing** element value (§2.4.41;§2.4.42;§2.4.43) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the specified conditional formatting on the table shall not change the current set of internal edge borders on its set of table cells (i.e. their current setting shall remain unchanged).

Attributes	Description
See 2.3.1.4	

2.4.20 **insideV (Table Inside Vertical Edges Border)**

This element specifies the border which shall be displayed on all vertical table cell borders which are not on an outmost edge of the parent table (all horizontal borders which are not the leftmost or rightmost border). The appearance of this table cell border in the document shall be determined by the following settings:

- The display of the border on interior edges is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the inside vertical borders of this table shall have the border specified by the associated table style. If no inside vertical edge border is specified in the style hierarchy, then those cells in this table shall not have an inside vertical edge border.

Attributes	Description
See 2.3.1.4	

2.4.21 **jc (Table Alignment Exception)**

This element specifies the alignment of the set of rows which are part of the current table properties exception list with respect to the text margins in the current section. When a table is placed in a WordprocessingML document that does not have the same width as the margins, this property is used to determine how the table is positioned with respect to those margins. The interpretation of property is reversed if the parent table is right to left using the **bidirectional** element (§2.4.1). If this property is omitted on a table, then the justification shall be determined by the default set of table properties on the parent table.

Attributes	Description
val (Alignment Type)	See 2.3.1.13

2.4.22 **jc (Table Row Alignment)**

This element specifies the alignment of a single row in the parent table with respect to the text margins in the current section. When a table is placed in a WordprocessingML document that does not have the same width as the margins, this

property is used to determine how a specific row in that table is positioned with respect to those margins. The interpretation of property is reversed if the parent table is right to left using the `bidiVisual` element (§2.4.1). If this property is omitted on a table, then the justification shall be determined by the default set of table properties on the parent table.

Attributes	Description
val (Alignment Type)	See 2.3.1.13

2.4.23 `jc` (Table Alignment)

This element specifies the alignment of the current table with respect to the text margins in the current section. When a table is placed in a WordprocessingML document that does not have the same width as the margins, this property is used to determine how the table is positioned with respect to those margins. The interpretation of property is reversed if the parent table is right to left using the `bidiVisual` element (§2.4.1).

If this property is omitted on a table, then the justification shall be determined by the associated table style. If this property is not specified in the style hierarchy, then the table shall be left justified with zero indentation from the leading margin (the left margin in a left-to-right table or the right margin in a right-to-left table).

Attributes	Description
val (Alignment Type)	See 2.3.1.13

2.4.24 `left` (Table Cell Left Border)

This element specifies the border which shall be displayed on the left side of the current table cell. The appearance of this table cell border in the document shall be determined by the following settings:

- If the net **tblCellSpacing** element value (§2.4.41;§2.4.42;§2.4.43) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the left side of this table cell shall not have a cell border, and its border may use the table's border settings as appropriate.

Attributes	Description
See 2.3.1.4	

2.4.25 `left` (Table Cell Left Margin Exception)

This element specifies the amount of space which shall be left between the left extent of the current cell contents and the left border of a specific individual table cell within a table. This setting shall override the table cell bottom margin definition specified by the **left** element contained within the table properties (§2.4.26).

This value is specified in the units applied via its type attribute. Any width value of type `pct` or `auto` for this element shall be ignored.

If omitted, then this table cell shall use the bottom cell margins defined in the **left** element contained within the table properties (§2.4.26).

Attributes	Description
See 2.4.2	

2.4.26 left (Table Cell Left Margin Default)

This element specifies the amount of space which shall be left between the left extent of the cell contents and the left border of all table cells within the parent table (or table row). This setting may be overridden by the table cell bottom margin definition specified by the **left** element contained within the table cell's properties (§2.4.26).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a left margin is never specified in the style hierarchy, this table shall have 115 twentieths of a point (0.08 inches) left cell padding by default (excepting individual cell overrides).

Attributes	Description
See 2.4.2	

2.4.27 left (Table Left Border)

This element specifies the border which shall be displayed at the left edge of the current table. The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the left edge of this table shall have the border specified by the associated table style. If no left border is specified in the style hierarchy, then this table shall not have a left border.

Attributes	Description
See 2.3.1.4	

2.4.28 noWrap (Don't Wrap Cell Content)

This element specifies how this table cell shall be laid out when the parent table is displayed in a document. This setting only affects the behavior of the cell when the **tblLayout** for this row (§2.4.49; §2.4.50) is set to use the auto algorithm.

This setting shall be interpreted in the context of the **tcW** element (§2.4.68) as follows:

- If the table cell width has a type attribute value of **fixed**, then this element specifies that that this table cell shall never be smaller than that fixed value when other cells on the line are not at their absolute minimum width.
- If the table cell width has a type attribute value of **pct** or **auto**, then this element specifies that when running the auto fit algorithm, the contents of that this table cell shall be treated as though they have no breaking characters (the contents should be treated as a single contiguous non-breaking string)

If this element is omitted, then cell content shall be allowed to wrap (the cell may be shrunk as needed if it is a fixed preferred width value, and the contents shall be treated as having breaking characters if it is a percentage or automatic width value).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.29 right (Table Cell Right Margin Default)

This element specifies the amount of space which shall be present between the right extent of the cell contents and the right border of all table cells within the parent table (or table row). This setting may be overridden by the table cell bottom margin definition specified by the **right** element contained within the table cell's properties (§2.4.31).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a right margin is never specified in the style hierarchy, this table shall have 115 twentieths of a point (0.08 inches) left cell padding by default (excepting individual cell overrides).

Attributes	Description
See 2.4.2	

2.4.30 right (Table Cell Right Border)

This element specifies the border which shall be displayed on the right side of the current table cell. The appearance of this table cell border in the document shall be determined by the following settings:

- If the net **tblCellSpacing** element value (§2.4.41;§2.4.42;§2.4.43) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the right side of this table cell shall not have a cell border, and its border may use the table's border settings as appropriate.

Attributes	Description
See 2.3.1.4	

2.4.31 right (Table Cell Right Margin Exception)

This element specifies the amount of space which shall be present between the right extent of the current cell's text contents and the right border of a specific individual table cell within a table. This setting shall override the table cell bottom margin definition specified by the **right** element contained within the table properties (§2.4.29).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If omitted, then this table cell shall use the bottom cell margins defined in the **right** element contained within the table properties (§2.4.29).

Attributes	Description
See 2.4.2	

2.4.32 right (Table Right Border)

This element specifies the border which shall be displayed at the right edge of the current table. The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the right edge of this table shall have the border specified by the associated table style. If no right border is specified in the style hierarchy, then this table shall not have a right border.

Attributes	Description
See 2.3.1.4	

2.4.33 shd (Table Cell Shading)

This element specifies the shading which shall be applied to the extents of the current table cell. Similarly to paragraph shading, this shading shall be applied to the contents of the cell up to the cell borders, regardless of the presence of text.

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted, then the cell shading shall be determined by the table-level or table-level exception cell shading settings (§2.4.34;§2.4.35) for the current table.

Attributes	Description
See 2.3.1.31	

2.4.34 shd (Table Shading Exception)

This element specifies the shading which shall be applied to all cells in the current row as part of a set of table-level property exceptions. Similarly to paragraph shading, this shading shall be applied to the contents of the tab up to the table borders, regardless of the presence of text - unlike cell shading, table shading shall include any cell padding. This property shall be superseded by any cell-level shading on any cell in this row (§2.4.33).

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted, then the cell shading shall be determined by the table-level cell shading settings (§2.4.35) for the current table.

Attributes	Description
See 2.3.1.31	

2.4.35 shd (Table Shading)

This element specifies the shading which shall be applied to the extents the current table. Similarly to paragraph shading, this shading shall be applied to the contents of the tab up to the table borders, regardless of the presence of text - unlike cell shading, table shading shall include any cell padding. This property shall be superseded by any cell-level shading via any table-level property exceptions (§2.4.34); or on any cell in this row (§2.4.33).

This shading consists of three components:

- Background Color
- (optional) Pattern
- (optional) Pattern Color

The resulting shading is applied by setting the background color behind the paragraph, then applying the pattern color using the mask supplied by the pattern over that background.

If this element is omitted, then the cells within this table shall have the shading specified by the associated table style. If no cell shading is specified in the style hierarchy, then the cells in this table shall not have any cell shading (i.e. they shall be transparent).

Attributes	Description
See 2.3.1.31	

2.4.36 **tbl (Table)**

This element specifies the contents of a table present in the document. A *table* is a set of paragraphs (and other block-level content) arranged in *rows* and *columns*. Tables in WordprocessingML are defined via the **tbl** element, which is analogous to the HTML table tag.

2.4.37 **tblBorders (Table Borders Exceptions)**

This element specifies the set of borders for the edges of the parent table row via a set of table-level property exceptions, using the six border types defined by its child elements.

If the cell spacing for any row is non-zero as specified using the **tblCellSpacing** element (§2.4.41; §2.4.42; §2.4.43), then there is no border conflict and the table-level exception border shall be displayed.

If the cell spacing is zero, then there is a conflict, which shall be resolved as follows:

- If there is a cell border, then the cell border shall be displayed
- If there is no cell border, then the table-level exception border shall be displayed

If this element is omitted, then this table shall have the borders specified by the associated table level borders (§2.4.38).

2.4.38 **tblBorders (Table Borders)**

This element specifies the set of borders for the edges of the current table, using the six border types defined by its child elements.

If the cell spacing for any row is non-zero as specified using the **tblCellSpacing** element (§2.4.41; §2.4.42; §2.4.43), then there is no border conflict and the table border (or table-level exception border, if one is specified) shall be displayed.

If the cell spacing is zero, then there is a conflict, which shall be resolved as follows:

- If there is a cell border, then the cell border shall be displayed
- If there is no cell border but there is a table-level exception border on this table row, then the table-level exception border shall be displayed
- If there is no cell or table-level exception border, then the table border shall be displayed

If this element is omitted, then this table shall have the borders specified by the associated table style. If no borders are specified in the style hierarchy, then this table shall not have any table borders.

2.4.39 **tblCellMar (Table Cell Margin Defaults)**

This element specifies the default cell margin settings for all cells in the current table. These setting may be overridden by the table cell margin definition specified by the **tcMar** element contained within the table cell's properties (§2.4.65) or by a set of table-level property exceptions (§2.4.40).

If this element is omitted, then it shall inherit the table cell margins from the associated table style. If table margins are never specified in the style hierarchy, then each margin shall use its default margin size (see child element definitions).

2.4.40 **tblCellMar (Table Cell Margin Exceptions)**

This element specifies a set of cell margins for all cells in the parent table row via a set of table-level property exceptions. These settings may be overridden by the table cell margin definition specified by the **tcMar** element contained within the table cell's properties (§2.4.40).

If this element is omitted, then it shall inherit the table cell margins from the table-level cell margins (§2.4.39).

2.4.41 **tblCellSpacing (Table Cell Spacing Exception)**

This element specifies a table cell spacing exception for all cells in the parent table row as part of a set of table-level property exceptions. If specified, this element specifies the minimum amount of space which shall be left between all cells in the parent row after including the width of the table borders in the calculation. This setting shall be superseded by the row cell spacing value (§2.4.42). It is important to note that table-level cell spacing shall be added outside of the text margins, which shall be aligned with the innermost starting edge of the text extents in a table cell.

This value is specified in the units applied via its type attribute. Any width value of type **pct** or **auto** for this element shall be ignored. If this element is omitted, then the row shall inherit the table cell spacing from the table-level cell spacing setting (§2.4.39), excepting the case of a row level override.

Attributes	Description
See 2.4.2	

2.4.42 **tblCellSpacing (Table Row Cell Spacing)**

This element specifies the default table cell spacing (the spacing between adjacent cells and the edges of the table) for all cells in the parent row. If specified, this element specifies the minimum amount of space which shall be left between all cells in the table including the width of the table borders in the calculation. It is important to note that row-level cell spacing shall be added inside of the text margins, which shall be aligned with the innermost starting edge of the text extents in a cell without row-level indentation or cell spacing. Row-level cell spacing shall not increase the width of the overall table.

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then the cells in this row shall inherit the cell spacing from the associated table level properties.

Attributes	Description
See 2.4.2	

2.4.43 **tblCellSpacing (Table Cell Spacing Default)**

This element specifies the default table cell spacing (the spacing between adjacent cells and the edges of the table) for all cells in the parent table. If specified, this element specifies the minimum amount of space which shall be left between all cells in the table including the width of the table borders in the calculation. This setting shall be superseded by a table-level exception (§2.4.41) or the row cell spacing value (§2.4.42) in that order. It is important to note that table-level cell spacing shall be added outside of the text margins, which shall be aligned with the innermost starting edge of the text extents in a table cell.

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then the table shall inherit the table cell spacing from the associated table style. If table cell spacing is never specified in the style hierarchy, no cell spacing shall be added to the parent table.

Attributes	Description
See 2.4.2	

2.4.44 **tblGrid (Table Grid)**

This element specifies the table grid for the current table. The *table grid* is a definition of the set of grid columns which define all of the shared vertical edges of the table, as well as default widths for each of these grid columns. These grid column widths are then used to determine the size of the table based on the table layout algorithm used (§2.4.49;§2.4.50). If the table grid is omitted, then a new grid shall be constructed from the actual contents of the table assuming that all grid columns have a width of 0.

2.4.45 **tblGrid (Previous Table Grid)**

This element specifies a previous table grid state, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table grid settings which were previously in place before a specific set of revisions by one author. The *table grid* is a definition of the set of grid columns which define all of the shared vertical edges of the table, as well as default widths for each of these grid columns. These grid column widths are then used to determine the size of the table based on the table layout algorithm used (§2.4.49;§2.4.50).

2.4.46 **tblHeader (Repeat Table Row on Every New Page)**

This element specifies that the current table row shall be repeated at the top of each new page on which part of this table is displayed. This gives this table row the behavior of a 'header' row on each of these pages.

If this element is omitted, then the table shall inherit the table cell spacing from the associated table style. If table cell spacing is never specified in the style hierarchy, then this table row shall not be repeated on each new page on which the table is displayed. As well, if this row is not contiguously connected with the first row of the table (that is, if this table row is not either the first row, or all rows between this row and the first row are not marked as header rows) then this property shall be ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.47 **tblInd (Table Indent from Leading Margin Exception)**

This element specifies the indentation which shall be added before the leading edge of the set of parent table rows which have this set of table-level property exceptions applied. This indentation should shift the table into the text margin by the specified amount in the document (the left edge in a left-to-right table, and the right edge in a right-to-left table).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then the table shall inherit the table indentation from the associated table level property setting.

If the resulting justification on the parent table row is not left after applying the value of the **jc** element from the three levels of this property (§2.4.21;§2.4.22;§2.4.23), then this property shall be ignored.

Attributes	Description
See 2.4.2	

2.4.48 **tblInd (Table Indent from Leading Margin)**

This element specifies the indentation which shall be added before the leading edge of the current table in the document (the left edge in a left-to-right table, and the right edge in a right-to-left table). This indentation should shift the table into the text margin by the specified amount.

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then the table shall inherit the table indentation from the associated table style. If table indentation is never specified in the style hierarchy, no indentation shall be added to the parent table. If the resulting justification on any table row is not left after applying the value of the **jc** element from the three levels of this property (§2.4.21;§2.4.22;§2.4.23), then this property shall be ignored.

Attributes	Description
See 2.4.2	

2.4.49 **tblLayout (Table Layout)**

This element specifies the algorithm which shall be used to lay out the contents of this table within the document. When a table is displayed in a document, it can either be displayed using a fixed width or autofit layout algorithm (each discussed in the simple type referenced by the val attribute).

If this element is omitted, then the value of this element shall be assumed to be auto.

Attributes	Description
type (Table Layout Setting)	Specifies the algorithm which shall be used to lay out the contents of the parent table (see simple type definition for details on each algorithm used).

Attributes	Description
	The possible values for this attribute are defined by the ST_TblLayoutType simple type (§2.18.93).

2.4.50 tblLayout (Table Layout Exception)

This element specifies the algorithm which shall be used to lay out the contents of all rows with this table within the table which have the set of table-level property exceptions specified by the parent element. When a table is displayed in a document, it can either be displayed using a fixed width or autofit layout algorithm (each discussed in the simple type referenced by the val attribute).

If this element is omitted, then the value of this element shall be assumed to be auto.

Attributes	Description
type (Table Layout Setting)	See 2.4.49

2.4.51 tblLook (Table Style Conditional Formatting Settings)

This element specifies the components of the conditional formatting of the referenced table style (if one exists) which shall be applied to the current table. A table style can specify up to six different optional conditional formats , which then can be applied or omitted from individual tables in the document.

This element's value is hexadecimal code containing a bitmask of options, interpreted as follows:

- 0x0020=Apply first row conditional formatting
- 0x0040=Apply last row conditional formatting
- 0x0080=Apply first column conditional formatting
- 0x0100=Apply last column conditional formatting
- 0x0200=Do not apply row banding conditional formatting
- 0x0400=Do not apply column banding conditional formatting

If omitted, the bitmask of table style options on the current table shall be assumed to be 0000.

Attributes	Description
val (Two Digit Hexadecimal Value)	Specifies a value specified as a two digit hexadecimal number), whose contents are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_ShortHexNumber simple type (§2.18.85).

2.4.52 tblLook (Table Style Conditional Formatting Settings Exception)

This element specifies the components of the conditional formatting of the referenced table style (if one exists) which shall be applied to the set of table rows with the current table-level property exceptions. A table style can specify up to six different optional conditional formats , which then can be applied or omitted from individual table rows in the parent table.

This element's value is hexadecimal code containing a bitmask of options, interpreted as follows:

- 0x0020=Apply first row conditional formatting
- 0x0040=Apply last row conditional formatting
- 0x0080=Apply first column conditional formatting
- 0x0100=Apply last column conditional formatting
- 0x0200=Do not apply row banding conditional formatting
- 0x0400=Do not apply column banding conditional formatting

If omitted, the bitmask of table style options on the current table row shall be assumed to be the value specified on the table-level properties.

Attributes	Description
val (Two Digit Hexadecimal Value)	See 2.4.51

2.4.53 tblOverlap (Floating Table Allows Other Tables to Overlap)

This element specifies whether the current table shall allow other floating tables to overlap its extents when the tables are displayed in a document. If specified, then no adjustment shall be made to prevent tables whose properties would normally cause them to overlap from overlapping when displayed. If turned off, then the tables shall be adjusted as needed to prevent them from overlapping when displayed by adjusting the floating table properties as needed.

If this element is omitted on a given table, then this table shall allow other tables to overlap when displayed. If the parent table is not floating via the **tblpPr** element (§2.4.54), then this element shall be ignored.

Attributes	Description
val (Floating Table Overlap Setting)	Specifies whether a floating table shall allow other floating tables in the document to overlap its extents when displayed. The possible values for this attribute are defined by the ST_TblOverlap simple type (§2.18.94).

2.4.54 tblpPr (Floating Table Positioning)

This element specifies information about the current table with regard to floating tables. *Floating tables* are tables in a document which are not part of the main text flow in the document, and are instead absolutely positioned with a specific size and position relative to non-frame content in the current document.

The first piece of information specified by the **tblpPr** element is that the current table is actually a floating table. This information is specified simply by the presence of the **tblpPr** element in table's properties. If the **tblpPr** element is omitted, the table shall not floating in the document.

The second piece of information is the positioning of the table, which is specified by the attribute values stored on the **tblpPr** element. In all absolute positioning cases, the positioning of the table is relative to its top-left corner position. For relative positioning (e.g. center), the positioning of the table is relative to its entire frame.

Note that the table still has a logical position in the file (its location within the block-level elements in the document). This logical location shall be used to calculate the position of the table relative to a paragraph, using the next regular (non-table, non-frame) paragraph in the document.

Attributes	Description
bottomFromText (Distance From Bottom of Table to Text)	Specifies the minimum distance which shall be maintained between the current floating table and the top of text in the paragraph which is below this floating table. This distance is expressed in twentieths of a point. If this attribute is omitted, its value shall be assumed to be 0. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
horzAnchor (Table Horizontal Anchor)	Specifies the base object from which the horizontal positioning in the tblpX and/or tblpXSpec attribute should be calculated. A floating table may be horizontally positioned relative to: <ul style="list-style-type: none"> The vertical edge of the page before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs)

Attributes	Description
	<ul style="list-style-type: none"> The vertical edge of the text margin before any runs of text (the left edge for left-to-right paragraphs, the right edge for right-to-left paragraphs) The vertical edge of the text margin for the column in which the anchor paragraph is located <p>If this attribute is omitted, then its value shall be assumed to be page.</p> <p>The possible values for this attribute are defined by the ST_HAnchor simple type (§2.18.39).</p>
leftFromText (Distance From Left of Table to Text)	<p>Specifies the minimum distance which shall be maintained between the current floating table and the edge of text in the paragraph which is to the left of this floating table.</p> <p>This distance is expressed in twentieths of a point.</p> <p>If this attribute is omitted, its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
rightFromText ((Distance From Right of Table to Text))	<p>Specifies the minimum distance which shall be maintained between the current floating table and the edge of text in the paragraph which is to the right of this floating table.</p> <p>This distance is expressed in twentieths of a point.</p> <p>If this attribute is omitted, its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>
tblpX (Absolute Horizontal Distance From Anchor)	<p>Specifies an absolute horizontal position for the floating table. This absolute position is specified relative to the horizontal anchor specified by the horzAnchor attribute for this floating table.</p> <p>This value is expressed in twentieths of a point. If it is positive, then the floating table is positioned after the anchor object in the direction of horizontal text flow in this document. If it is negative, then the floating table is positioned before the anchor object in the direction of horizontal text flow in this document.</p> <p>If the tblpXSpec attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>
tblpXSpec (Relative Horizontal Alignment From Anchor)	<p>Specifies a relative horizontal position for the floating table. This relative position is specified relative to the horizontal anchor specified by the horzAnchor attribute for this floating table.</p> <p>If omitted, this attribute is not specified and the value of the tblpX attribute determines the absolute horizontal position of the floating table. If specified, the position for this attribute supersedes any value which is specified in the tblpX attribute, and that value is ignored.</p> <p>The possible values for this attribute are defined by the ST_XAlign simple type (§2.18.113).</p>
tblpY (Absolute Vertical Distance From Anchor)	<p>Specifies an absolute vertical position for the floating table. This absolute position is specified relative to the vertical anchor specified by the vertAnchor attribute for this floating table.</p> <p>This value is expressed in twentieths of a point. If it is positive, then the floating table is positioned after the anchor object in the direction of vertical text flow in this document. If it is negative, then the floating table is positioned before the anchor object in the direction of vertical text flow in this document.</p>

Attributes	Description
	<p>If the <code>tblpYSpec</code> attribute is also specified, then this value is ignored. If this attribute is omitted, then its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the <code>ST_SignedTwipsMeasure</code> simple type (§2.18.87).</p>
tblpYSpec (Relative Vertical Alignment from Anchor)	<p>Specifies a relative vertical position for the floating table. This relative position is specified relative to the vertical anchor specified by the <code>vertAnchor</code> attribute for this floating table.</p> <p>If omitted, this attribute is not specified and the value of the <code>tblpY</code> attribute determines the absolute horizontal position of the floating table. If specified, the position for this attribute supersedes any value which is specified in the <code>tblpY</code> attribute, and that value is ignored, unless the <code>vertAnchor</code> is set to <code>text</code>, in which case any relative positioning is not allowed, and is itself ignored.</p> <p>The possible values for this attribute are defined by the <code>ST_YAlign</code> simple type (§2.18.114).</p>
topFromText (Distance From Top of Table to Text)	<p>Specifies the minimum distance which shall be maintained between the current floating table and the bottom edge of text in the paragraph which is above this floating table.</p> <p>This distance is expressed in twentieths of a point.</p> <p>If this attribute is omitted, its value shall be assumed to be 0.</p> <p>The possible values for this attribute are defined by the <code>ST_TwipsMeasure</code> simple type (§2.18.104).</p>
vertAnchor (Table Vertical Anchor)	<p>Specifies the base object from which the vertical positioning in the <code>tblpY</code> attribute should be calculated.</p> <p>A floating table may be horizontally positioned relative to:</p> <ul style="list-style-type: none"> • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the text margin before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) • The horizontal edge of the page before any runs of text (the top edge for top-to-bottom sections, the bottom for bottom-to-top sections) <p>If this attribute is omitted, then its value shall be assumed to be <code>page</code>.</p> <p>The possible values for this attribute are defined by the <code>ST_VAnchor</code> simple type (§2.18.108).</p>

2.4.55 `tblPr` (Table Properties)

This element specifies the set of table-wide properties applied to the current table. These properties affect the appearance of all rows and cells within the parent table, but may be overridden by individual table-level exception, row, and cell level properties as defined by each property.

2.4.56 `tblPr` (Previous Table Properties)

This element specifies a previous set of table properties, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table property settings which were previously in place before a specific set of revisions by one author. These properties affect the appearance of all rows and cells within the parent table, but may be overridden by individual table-level exception, row, and cell level properties, as defined by each property.

2.4.57 **tblPrEx (Table-Level Property Exceptions)**

This element specifies a set of table properties which shall be applied to the contents of this row in place of the table properties specified in the **tblPr** element.

[Note: These properties are typically used in cases involving legacy documents, as well as cases where two existing independent tables are merged (in order to prevent the look of the second table from being superseded by the first table). *end note*]

2.4.58 **tblPrEx (Previous Table-Level Property Exceptions)**

This element specifies a previous set of table-level property exceptions, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table-level property exceptions which were previously in place before a specific set of revisions by one author.

2.4.59 **tblStyle (Referenced Table Style)**

This element specifies the style ID of the table style which shall be used to format the contents of this table.

This formatting is applied at the following location in the *style hierarchy*:

- Document defaults
- Table styles (this element)
- Numbering styles
- Paragraph styles
- Character styles
- Direct Formatting

This means that all properties specified in the **style** element (§2.7.3.17) with a styleId which corresponds to the value in this element's val attribute are applied to the table at the appropriate level in the hierarchy.

If this element is omitted, or it references a style which does not exist, then no table style shall be applied to the current table. As well, this property is ignored if the table properties are themselves part of a table style.

Attributes	Description
val (String Value)	See 2.3.1.27

2.4.60 **tblW (Preferred Table Width Exception)**

This element specifies the preferred width for the parent table row via a set of table-level property exceptions. This preferred width is used as part of the table layout algorithm specified by the **tblLayout** element (§2.4.49n; §2.4.50) - full description of the algorithm in the **ST_TblLayout** simple type (§2.18.93).

All widths in a table are considered preferred because:

- The table must satisfy the shared columns as specified by the **tblGrid** element (§2.4.44)
- Two or more widths may have conflicting values for the width of the same grid column
- The table layout algorithm (§2.18.93) may require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type pct for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type auto.

Attributes	Description
See 2.4.2	

2.4.61 **tblW (Preferred Table Width)**

This element specifies the preferred width for this table. This preferred width is used as part of the table layout algorithm specified by the **tblLayout** element (§2.4.49; §2.4.50) - full description of the algorithm in the **ST_TblLayout** simple type (§2.18.93).

All widths in a table are considered preferred because:

- The table must satisfy the shared columns as specified by the **tblGrid** element (§2.4.44)
- Two or more widths may have conflicting values for the width of the same grid column
- The table layout algorithm (§2.18.93) may require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type `pct` for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type `auto`.

Attributes	Description
See 2.4.2	

2.4.62 **tc (Table Cell)**

This element specifies a single cell in a table row, which contains the table's content. Table cells in WordprocessingML are analogous to HTML `td` elements.

A **tc** element has one formatting child element, **tcPr** (§2.4.67), which defines the properties for the cell. Each unique property on the table cell is specified by a child element of this element. As well, a table cell can contain any valid block-level content, which allows for the nesting of paragraphs and tables within table cells.

If a table cell does not include at least one block-level element, then this document shall be considered corrupt.

2.4.63 **tcBorders (Table Cell Borders)**

This element specifies the set of borders for the edges of the current table cell, using the eight border types defined by its child elements.

If the cell spacing for any row is non-zero as specified using the **tblCellSpacing** element (§2.4.41; §2.4.42; §2.4.43), then there is never a border conflict (as the non-zero cell spacing is applied above and beyond each individual cell border's width) and all table, table-level exception, and table cell borders shall be displayed.

If the cell spacing is zero, then there may be a conflict between two adjacent cell borders, which shall be resolved as follows:

1. If either conflicting table cell border is `nil` or `none` (no border), then the opposing border shall be displayed.
2. If a cell border conflicts with a table border, the cell border always wins.
3. Each border shall then be assigned a weight using the following formula, and the border value using this calculation shall be displayed over the alternative border:

$$W_{\text{border}} = \# \text{ of lines in border} * \text{border number}$$

4. The border number shall be determined by this list:

single	1
thick	2
double	3
dotted	4
dashed	5
dotDash	6
dotDotDash	7
triple	8
thinThickSmallGap	9
thickThinSmallGap	10
thinThickThinSmallGap	11
thinThickMediumGap	12
thickThinMediumGap	13
thinThickThinMediumGap	14

thinThickLargeGap	15
thickThinLargeGap	16
thinThickThinLargeGap	17
wave	18
doubleWave	19
dashSmallGap	20
dashDotStroked	21
threeDEmboss	22
threeDEngrave	23
outset	24
inset	25

5. If the borders have an equal weight, than the higher of the two on this precedence list shall win:
 - single
 - thick
 - double
 - dotted
 - dashed
 - dotDash
 - dotDotDash
 - triple
 - thinThickSmallGap
 - thickThinSmallGap
 - thinThickThinSmallGap
 - thinThickMediumGap
 - thickThinMediumGap
 - thinThickThinMediumGap
 - thinThickLargeGap
 - thickThinLargeGap
 - thinThickThinLargeGap
 - wave
 - doubleWave
 - dashSmallGap
 - dashDotStroked
 - threeDEmboss
 - threeDEngrave
 - outset
 - inset
6. If the borders have an identical style, than each border color shall be assigned a brightness value as follows:

$$\text{Brightness} = R + B + 2 * G$$

The color with the smaller brightness value shall win.
7. If the borders have an identical brightness value above, than each border color shall be assigned a new brightness value as follows:

$$\text{Brightness} = B + 2 * G$$

The color with the smaller brightness value shall win.
8. If the borders have an identical brightness value above, than each border color shall be assigned a brightness value as follows:

$$\text{Brightness} = G$$

The color with the smaller brightness value shall win.

9. If the borders have an identical brightness value above, then they are functionally identical, and the first border in reading order should be displayed.

If this element is omitted, then this table shall have the borders specified by the associated table style. If no borders are specified in the style hierarchy, then this table shall not have any table borders.

2.4.64 **tcFitText (Fit Text Within Cell)**

This element specifies that the contents of the current cell shall have their inter-character spacing increased or reduced as necessary to fit the width of the text extents of the current cell. This setting shall behave identically to placing the contents of this paragraph in a run and using the **fitText** element (§2.3.2.12), if the width provided on that element matched the width of the current cell.

If this element is omitted, then the text in this cell shall not be fit to the current cell extents.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.4.65 **tcMar (Single Table Cell Margins)**

This element specifies a set of cell margins for a single table cell in the parent table.

This setting, if present, shall override the table cell margins from the table-level cell margins (§2.4.39).

2.4.66 **tcPr (Previous Table Cell Properties)**

This element specifies a previous set of table cell properties, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table cell property settings which were previously in place before a specific set of revisions by one author. Each unique property is specified by a child element of this element. In any instance where there is a conflict between the table level, table-level exception, or row level properties with a corresponding table cell property, these properties shall overwrite the table or row wide properties.

2.4.67 **tcPr (Table Cell Properties)**

This element specifies the set of properties which shall be applied a specific table cell. Each unique property is specified by a child element of this element. In any instance where there is a conflict between the table level, table-level exception, or row level properties with a corresponding table cell property, these properties shall overwrite the table or row wide properties.

2.4.68 **tcW (Preferred Table Cell Width)**

This element specifies the preferred width for this table cell. This preferred width is used as part of the table layout algorithm specified by the **tblLayout** element (§2.4.49; §2.4.50) - full description of the algorithm in the **ST_TblLayout** simple type (§2.18.93).

All widths in a table are considered preferred because:

- The table must satisfy the shared columns as specified by the **tblGrid** element (§2.4.44)
- Two or more widths may have conflicting values for the width of the same grid column
- The table layout algorithm (§2.18.93) may require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type **pct** for this element shall be calculated relative to the overall width of the table.

If this element is omitted, then the cell width shall be of type **auto**.

Attributes	Description
See 2.4.2	

2.4.69 textDirection (Table Cell Text Flow Direction)

This element specifies the direction of the text flow for this table cell.

If this element is omitted on a given table cell, its value is determined by the setting previously set at any level of the style hierarchy (i.e. that previous setting remains unchanged). If this setting is never specified in the style hierarchy, then the table cell shall inherit the text flow settings from the parent section.

Attributes	Description
val (Direction of Text Flow)	See 2.3.1.41

2.4.70 tl2br (Table Cell Top Left to Bottom Right Diagonal Border)

This element specifies the border which shall be displayed on the top left side to bottom right diagonal within the current table cell.

If this element is omitted, then the top left to bottom right diagonal of this table cell shall not have a cell border, and its border may use the table's border settings as appropriate.

Attributes	Description
See 2.3.1.4	

2.4.71 top (Table Top Border)

This element specifies the border which shall be displayed at the top of the current table. The appearance of this table border in the document shall be determined by the following settings:

- The display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the top of this table shall have the border specified by the associated table style. If no top border is specified in the style hierarchy, then this table shall not have a top border.

Attributes	Description
See 2.3.1.4	

2.4.72 top (Table Cell Top Margin Default)

This element specifies the amount of space which shall be left between the top extent of the cell contents and the top border of all table cells within the parent table. This setting may be overridden by the table cell top margin definition specified by the **top** element contained within the table cell's properties (§2.4.73).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If this element is omitted, then it shall inherit the table cell margin from the associated table style. If a top margin is never specified in the style hierarchy, then this table shall have no top cell padding by default (excepting individual cell overrides).

Attributes	Description
See 2.4.2	

2.4.73 top (Table Cell Top Margin Exception)

This element specifies the amount of space which shall be left between the top extent of the cell contents and the top border of a specific table cell within a table. This setting shall override the table cell top margin definition specified by the **top** element contained within the table properties (§2.4.72).

This value is specified in the units applied via its type attribute. Any width value of type pct or auto for this element shall be ignored.

If omitted, then this table cell shall use the bottom cell margins defined in the **top** element contained within the table properties (§2.4.72).

Attributes	Description
See 2.4.2	

2.4.74 top (Table Cell Top Border)

This element specifies the border which shall be displayed at the top of the current table cell. The appearance of this table cell border in the document shall be determined by the following settings:

- If the net **tblCellSpacing** element value (§2.4.41;§2.4.42;§2.4.43) applied to the cell is non-zero, then the cell border shall always be displayed
- Otherwise, the display of the border is subject to the conflict resolution algorithm defined by the **tcBorders** element (§2.4.63) and the **tblBorders** element (§2.4.37;§2.4.38)

If this element is omitted, then the top of this table cell shall not have a cell border, and its border may use the table's border settings as appropriate.

Attributes	Description
See 2.3.1.4	

2.4.75 tr (Table Row)

This element specifies a single table row, which contains the table's cells. Table rows in WordprocessingML are analogous to HTML **tr** elements.

A **tr** element has one formatting child element, **trPr** (§2.4.78), which defines the properties for the row. Each unique property on the table row is specified by a child element of this element. As well, a table row can contain any valid row-level content, which allows for the use of table cells.

If a table cell does not include at least one child element other than the row properties, then this document shall be considered corrupt.

Attributes	Description
rsidDel (Revision Identifier for Table Row Deletion)	<p>Specifies a unique identifier used to track the editing session when the row was deleted from the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidR (Revision Identifier for Table Row)	<p>Specifies a unique identifier used to track the editing session when the table row was added to the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save</p>

Attributes	Description
	<p>actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidRPr (Revision Identifier for Table Row Glyph Formatting)	<p>Specifies a unique identifier used to track the editing session when the glyph character representing the table row mark was last modified in the main document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidTr (Revision Identifier for Table Row Properties)	<p>Specifies a unique identifier used to track the editing session when the table row's properties were last modified in this document.</p> <p>All rsid* attributes throughout this document with the same value, if present, must indicate that those regions were modified during the same editing session (time between subsequent save actions).</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions to indicate the order of the modifications relative to other modifications in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>

2.4.76 **tr2bl (Table Cell Top Right to Bottom Left Diagonal Border)**

This element specifies the border which shall be displayed on the top right to bottom left diagonal within the current table cell.

If this element is omitted, then the top right to bottom left diagonal of this table cell shall not have a cell border, and its border may use the table's border settings as appropriate.

Attributes	Description
See 2.3.1.4	

2.4.77 **trHeight (Table Row Height)**

This element specifies the height of the current table row within the current table. This height shall be used to determine the resulting height of the table row, which may be absolute or relative (depending on its attribute values).

If omitted, then the table row shall automatically resize its height to the height required by its contents (the equivalent of an hRule value of auto).

Attributes	Description
hRule (Table Row Height Type)	<p>Specifies the meaning of the height specified for this table row.</p> <p>The meaning of the value of the val attribute is defined based on the value of the hRule attribute for this table row as follows:</p> <ul style="list-style-type: none"> • If the value of hRule is <code>auto</code>, then the table row's height should be automatically determined based on the height of its contents. The h value is ignored. • If the value of hRule is <code>atLeast</code>, then the table row's height should be at least the value the h attribute. • If the value of hRule is <code>exact</code>, then the table row's height should be exactly the value of the h attribute. <p>If this attribute is omitted, then its value shall be assumed to be <code>auto</code>. The possible values for this attribute are defined by the ST_HeightRule simple type (§2.18.41).</p>
val (Table Row Height)	<p>Specifies the table row's height.</p> <p>This height is expressed in twentieths of a point.</p> <p>If this attribute is omitted, then its value shall be assumed to be 0.</p> <p>The meaning of the value of the val attribute is defined based on the value of the hRule attribute for this table row as follows:</p> <ul style="list-style-type: none"> • If the value of hRule is <code>auto</code>, then the table row's height should be automatically determined based on the height of its contents. This value is ignored. • If the value of hRule is <code>atLeast</code>, then the table row's height should be at least the value of this attribute. <p>If the value of hRule is <code>exact</code>, then the table row's height should be exactly the value of this attribute. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>

2.4.78 trPr (Table Row Properties)

This element specifies the set of row-level properties applied to the current table row. Each unique property is specified by a child element of this element. These properties affect the appearance of all cells in the current row within the parent table, but may be overridden by individual cell-level properties, as defined by each property.

2.4.79 trPr (Previous Table Row Properties)

This element specifies a previous set of table cell properties, the modifications to which shall be attributed to a revision by a particular author and at a particular time. This element contains the table cell property settings which were previously in place before a specific set of revisions by one author. Each unique property is specified by a child element of this element. These properties affect the appearance of all cells in the current row within the parent table, but may be overridden by individual cell-level properties, as defined by each property.

2.4.80 vAlign (Table Cell Vertical Alignment)

This element specifies the vertical alignment for text within the current table cell. The vertical alignment of this text is determined by the value of the val attribute.

Attributes	Description
val (Vertical Alignment Setting)	Specifies the vertical alignment for text between the top and bottom margins of the parent container (page or table cell). The possible values for this attribute are defined by the ST_VerticalJc simple type (§2.18.110).

2.4.81 vMerge (Vertically Merged Cell)

This element specifies that this cell is part of a vertically merged set of cells in a table. The **val** attribute on this element determines how this cell is defined with respect to the previous cell in the table (i.e. does this cell continue the vertical merge or start a new merged group of cells).

If this element is omitted, then this cell shall not be part of any vertically merged grouping of cells, and any vertically merged group of preceding cells shall be closed. If a vertically merged group of cells do not span the same set of grid columns, then this vertical merge is invalid.

Attributes	Description
val (Vertical Merge Type)	Specifies how the table cell is part of a vertically merged region. This determines whether the cell should join onto an existing grouping of merged cells if any exist, or start a new group of merged cells. Refer to the simple type definition for a full description of each type. If this attribute is omitted, its value shall be assumed to be <code>continue</code> . The possible values for this attribute are defined by the ST_Merge simple type (§2.18.63).

2.4.82 wAfter (Preferred Width After Table Row)

This element specifies the preferred width for the total number of grid columns after this table row as specified in the **gridAfter** element (§2.4.10). This preferred width is used as part of the table layout algorithm specified by the **tblLayout** element (§2.4.49; §2.4.50) - full description of the algorithm in the **ST_TblLayout** simple type (§2.18.93).

All widths in a table are considered preferred because:

- The table must satisfy the shared columns as specified by the **tblGrid** element (§2.4.44)
- Two or more widths may have conflicting values for the width of the same grid column
- The table layout algorithm (§2.18.93) may require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type `pct` for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type `auto`.

Attributes	Description
See 2.4.2	

2.4.83 wBefore (Preferred Width Before Table Row)

This element specifies the preferred width for the total number of grid columns before this table row as specified in the **gridAfter** element (§2.4.10). This preferred width is used as part of the table layout algorithm specified by the **tblLayout** element (§2.4.49; §2.4.50) - full description of the algorithm in the **ST_TblLayout** simple type (§2.18.93).

All widths in a table are considered preferred because:

- The table must satisfy the shared columns as specified by the **tblGrid** element (§2.4.44)
- Two or more widths may have conflicting values for the width of the same grid column
- The table layout algorithm (§2.18.93) may require a preference to be overridden

This value is specified in the units applied via its type attribute. Any width value of type `pct` for this element shall be calculated relative to the text extents of the page (page width excluding margins).

If this element is omitted, then the cell width shall be of type `auto`.

Attributes	Description
See 2.4.2	

2.5 Custom Markup

Within a WordprocessingML document, it is often necessary for specific documents to contain semantic information beyond the presentation information specified by this Office Open XML Standard.

For these scenarios, multiple facilities are provided for the insertion and round-tripping of customer defined semantics within a WordprocessingML document. There are three distinct forms in which customer-defined semantics can be inserted into a WordprocessingML document, each with their own specific intended usage:

- Smart tags
- Custom XML markup
- Structured document tags (content controls)

The elements and attributes which define each of these forms is described in the following clauses.

2.5.1 Custom XML and Smart Tags

The final type of customer-defined semantics which can be embedded in a WordprocessingML document are structured document tags (SDTs).

As shown above, smart tags and custom XML markup each provide a facility for embedding customer-defined semantics into the document: smart tags, via the ability to provide a basic namespace/name for a run or set of runs within a documents; and custom XML markup, via the ability to tag the document with XML elements and attributes specified by any valid XML Schema file.

However, each of these techniques, while they each provide a way to add the desired semantic information, does not provide a way to affect the presentation or interaction within the document. To bridge these two worlds, structured document tags allow both the specification of customer semantics as well as the ability to influence the presentation of that data in the document.

This means that the customer can define the semantics and context of the tag, but can then use a rich set of pre-defined properties to define its behavior and appearance within the WordprocessingML document's presentation.

As shown above, each of the structured document tags in the WordprocessingML file is represented using the **sdt** element. Within a structured document tag, there are two child elements which contain the definition and the content of this SDT. The first of these is the **sdtPr** element, which contains the set of properties specified for this structured document tag. The second is the **sdtContent** element, which contains all the content which is contained within this structured document tag.

2.5.1.1 attr (Custom XML Attribute)

This element specifies a custom XML attribute which shall be located on the parent custom XML element specified via the **customXml** element (§2.5.1.3;§2.5.1.4;§2.5.1.5; §2.5.1.6). The attributes on this element shall be used to specify the contents of the custom XML attribute.

Attributes	Description
name (Name)	Specifies the name of the current custom XML attribute or smart tag property. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
uri (Namespace)	Specifies the namespace URI of the current custom XML attribute or smart tag property. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
val (Value)	Specifies the value of the current custom XML attribute or smart tag property. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.1.2 **attr** (Smart Tag Property)

This element specifies a single smart tag property which shall be located on the parent smart tag, specified via the **smartTag** element (§2.5.1.9). The attributes on this element shall be used to specify the contents of smart tag property.

Attributes	Description
See 2.5.1.1	

2.5.1.3 **customXml** (Cell-Level Custom XML Element)

This element specifies the presence of a custom XML element around a single table cell. The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

Attributes	Description
element (Custom XML Element Name)	Specifies the name of the current custom XML element or smart tag within the document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
uri (Custom XML Element Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.1.4 **customXml** (Row-Level Custom XML Element)

This element specifies the presence of a custom XML element around a single table row. The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

Attributes	Description
element (Custom XML Element Name)	Specifies the name of the current custom XML element or smart tag within the document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
uri (Custom XML Element Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.1.5 **customXml** (Inline-Level Custom XML Element)

This element specifies the presence of a custom XML element around one or more inline level structures (runs, images, fields, etc.) within a paragraph. The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

Attributes	Description
element (Element name)	Specifies the name of the current custom XML element or smart tag within the document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
uri (Custom XML Markup Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.1.6 **customXml** (Block-Level Custom XML Element)

This element specifies the presence of a custom XML element around one or more block level structures (paragraphs, tables, etc.). The attributes on this element shall be used to specify the name and namespace URI of the current custom XML element.

Attributes	Description
element (Custom XML Element Name)	Specifies the name of the current custom XML element or smart tag within the document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
uri (Custom XML Element Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.1.7 **customXmlPr** (Custom XML Element Properties)

This element specifies the set of properties which shall be applied to the parent custom XML element.

2.5.1.8 **placeholder** (Custom XML Element Placeholder Text)

This element specifies the placeholder text which shall be displayed in place of this custom XML element when the contents of this custom XML markup are empty (i.e. there are no runs of text within the current custom XML element). If this custom XML element does contain run content, then this text shall not be displayed.

The val attribute stores the string of text which shall be displayed as the placeholder text. This string may be displayed in any font face/size desired by the hosting application.

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.1.9 **smartTag** (Inline-Level Smart Tag)

This element specifies the presence of a smart tag around one or more inline structures (runs, images, fields, etc.) within a paragraph. The attributes on this element shall be used to specify the name and namespace URI of the current smart tag.

Attributes	Description
element (Smart Tag Name)	Specifies the name of the current custom XML element or smart tag within the document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
uri (Smart Tag Namespace)	Specifies the namespace URI of the current custom XML element or smart tag. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.1.10 **smartTagPr** (Smart Tag Properties)

This element specifies the set of properties which shall be applied to the parent smart tag.

2.5.2 Structured Document Tags

The final type of customer-defined semantics which can be embedded in a WordprocessingML document are structured document tags (SDTs).

As shown above, smart tags and custom XML markup each provide a facility for embedding customer-defined semantics into the document: smart tags, via the ability to provide a basic namespace/name for a run or set of runs within a documents; and custom XML markup, via the ability to tag the document with XML elements and attributes specified by any valid XML Schema file.

However, each of these techniques, while they each provide a way to add the desired semantic information, does not provide a way to affect the presentation or interaction within the document. To bridge these two worlds, structured document tags allow both the specification of customer semantics as well as the ability to influence the presentation of that data in the document.

This means that the customer can define the semantics and context of the tag, but can then use a rich set of pre-defined properties to define its behavior and appearance within the WordprocessingML document's presentation.

As shown above, each of the structured document tags in the WordprocessingML file is represented using the **sdt** element. Within a structured document tag, there are two child elements which contain the definition and the content of this SDT. The first of these is the **sdtPr** element, which contains the set of properties specified for this structured document tag. The second is the **sdtContent** element, which contains all the content which is contained within this structured document tag.

2.5.2.1 **alias** (Friendly Name)

This element specifies the friendly name associated with the current structured document tag. The string representing the friendly name shall be stored on this element's **val** attribute.

If this element is omitted, then no friendly name shall be associated with the given structured document tag.

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.2.2 **bibliography** (Bibliography Structured Document Tag)

This element specifies that the parent structured document tag shall be of type **bibliography**.

This type setting does not require or imply that the contents of the structured document tag shall contain only a field of type **BIBLIOGRAPHY**, it shall only be used to specify that the structured document tag is of this type, which may be used by an application as desired.

2.5.2.3 **calendar** (Date Picker Calendar Type)

This element specifies the type of calendar which shall be displayed for the current date picker structured document tag, if a user interface is present for the structured document tag. The calendar type is stored on this element's **val** attribute.

If this element is omitted, then the calendar type shall be **gregorian**.

Attributes	Description
val (Calendar Type Value)	Specifies a type of calendar, the use of which is determined by the parent XML element. If this attribute is omitted, then the calendar type shall be gregorian . The possible values for this attribute are defined by the ST_CalendarType simple type (§2.18.7).

2.5.2.4 **citation** (Citation Structured Document Tag)

This element specifies that the parent structured document tag shall be of type **citation**.

This type setting does not require or imply that the contents of the structured document tag shall contain only a field of type **CITATION**, it shall only be used to specify that the structured document tag is of this type, which may be used by an application as desired.

2.5.2.5 **comboBox** (Combo Box Structured Document Tag)

This element specifies that the parent structured document tag shall be a combo box when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The child elements of this element specify choices which shall be displayed in a standard drop-down list format
- Formatting applied to any part of this structured document tag's contents shall apply to its entire contents

As well, the structured document tag must satisfy the following restraints or the document shall be considered invalid:

- The contents shall only be contain a single run (one set of formatting properties)

- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

Attributes	Description
lastValue (Combo Box Last Saved Value)	<p>Specifies the value associated with the current display text for the combo box structured document tag.</p> <p>If this structured document tag is not mapped to XML using the dataBinding element (§2.5.2.6), then this attribute shall be ignored. If this structured document tag is mapped to XML, it shall be used to determine whether the current display text in the combo box structured document tag shall be retained when the document is opened, as follows:</p> <ul style="list-style-type: none"> • When the XML mapping is created, the content in the custom XML data is retrieved • If this content has an associated list item (matching its value attribute), then the corresponding display text shall be displayed in the structured document tag <p>If no list item exists, this content shall be matched against the lastValue attribute value. If the values match, the current display text shall be retained. If the values do not match, the current custom XML data content shall be the new display text (since no match exists in the combo box list items)</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>

2.5.2.6 dataBinding (XML Mapping)

This element specifies the information which shall be used to establish a mapping between the parent structured document tag and an XML element stored within a Custom XML Data part in the current WordprocessingML document.

If this element is omitted, then no XML mapping shall be associated with the current structured document tag. If the parent structured document tag is of type rich text or document part gallery, then this property shall be ignored.

If this element is present and the parent structured document tag is not of a rich text type, then the current value of the structured document tag shall be determined by finding the XML element (if any) which is determined by the attributes on this element. If this information does not result in a valid XML element, then the application may use any algorithm desired to find the closest available match. If this information does result in a valid XML element, then the contents of that element shall be used to replace the current run content within the document.

Attributes	Description
prefixMappings (XML Namespace Prefix Mappings)	<p>Specifies the set of prefix mappings which shall be used to interpret the XPath expression specified on the xpath attribute when the XPath expression is evaluated against the custom XML data parts in the current document.</p> <p>This attribute's value shall be specified using the following syntax: xmlns:prefix= 'namespace', where prefix is the namespace prefix to be mapped, and namespace is the namespace to be mapped to the current prefix. Each prefix mapping shall be delimited by one or more whitespace characters in the attribute's contents.</p> <p>If this attribute is omitted, then the prefix mappings specified on each of the custom XML data parts itself shall be used to evaluate the given XPath expression.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
storeItemID (Custom XML Data Storage ID)	<p>Specifies the custom XML data identifier for the custom XML data part which shall be used to evaluate the given XPath expression. The <i>custom XML data identifier</i>, specified using the storeItemID attribute of the dataStoreItem element (§Error! Reference source not found.) on the Custom XML Data Properties part is a string that uniquely identifies a particular custom XML data part in a WordprocessingML document (as multiple parts may have the same namespace for their root element).</p>

Attributes	Description
	<p>If specified, then the XPath expression specified on the xpath attribute shall only be evaluated against the custom XML data part whose properties part has a matching custom XML data identifier. If no custom XML data part exists with a matching identifier, then the XML mapping shall not be connected.</p> <p>If omitted, then the XPath expression shall be evaluated against each custom XML data part in turn until the given XPath expression is resolved to an XML element.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
xpath (XPath)	<p>Specifies the XPath expression which shall be evaluated to find the custom XML node which is mapped to the parent structured document tag. This XPath expression shall be specified using the syntax defined in the XML Path Language (XPath) Version 1.0 specification (see Annex A for bibliographic reference information).</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>

2.5.2.7 **date** (Date Structured Document Tag)

This element specifies that the parent structured document tag shall be a date picker when displayed in the document. This setting specifies that the behavior for this structured document tag shall be as follows:

- The child elements of this element specify how the dates in this structured document tag shall be stored in any mapped custom XML data and displayed in the document
- Formatting applied to any part of this structured document tag's contents shall apply to its entire contents

As well, the structured document tag must satisfy the following restraints or the document shall be considered invalid:

- The contents shall only be contain a single run (one set of formatting properties)
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table cell
-

Attributes	Description
fullDate (Last Known Date in XML Schema DateTime Format)	<p>Specifies the full date and time last entered into the parent structured document tag using the standard XML Schema DateTime syntax.</p> <p>[<i>Note: This cache is used because the date display mask stored on the dateFormat element (§2.5.2.8) may not contain all of the information about the date, which may be needed if the date display mask is later changed. end note</i>]</p> <p>If this attribute is specified, then the current fullDate attribute shall be used to populate the run content of the parent structured document tag by filtering it through the date display mask specified in the dateFormat element, if one is present.</p> <p>If this attribute is omitted, then the current display text shall be maintained when the document is displayed.</p> <p>The possible values for this attribute are defined by the ST_DateTime simple type (§Error! Reference source not found.).</p>

2.5.2.8 **dateFormat** (Date Display Mask)

The element specifies the display format which shall be used to format any date entered into the parent structured document tag in full DateTime format before displaying it in the structured document tag's run content.

If this element is omitted, then the date shall be formatted using the standard date display mask for the language ID specified on the **lid** element (§2.5.2.19) if present, or the language ID of the run contents otherwise. The date display mask specified in the **val** attribute shall be interpreted using the semantics specified in §2.16.4.1 of this Office Open XML Standard.

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.2.9 **docPart** (Document Part Reference)

This element specifies the name of the document part which shall be displayed in the parent structured document tag when its run contents are empty. If this element is specified, then a document part whose **name** element (§2.12.12) specifies a name matching the value of this element, and which belongs to the **bbP1cHdr** type shall be located to be used as the placeholder text for the parent structured document tag.

If no document part is located matching the criteria specified by this element, then five non-breaking spaces shall be used as the default placeholder text.

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.2.10 **docPartCategory** (Document Part Category Filter)

This element specifies the category of document parts which shall be used as the filter when determining the possible choices of document parts which are displayed for insertion into the parent structured document tag. A document part *category* is a sub-classification within a given document part gallery which may be used to further categorize the parts in a given gallery. . The category which shall be used as a filter is stored in this element's **val** attribute.

If this element is omitted, then the parent structured document tag shall display all document parts in the specified gallery regardless their specified category. If this element is present, but no document parts of the specified gallery and category combination are located by the application, then no document parts shall be displayed (i.e. the application shall not fall back to showing document parts in all categories in the specified gallery).

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.2.11 **docPartGallery** (Document Part Gallery Filter)

This element specifies the gallery of document parts which shall be used as the filter when determining the possible choices of document parts which are displayed for insertion into the parent structured document tag. A document part *gallery* is a classification of document parts, which may then be subdivided into categories. . The gallery which shall be used is stored in this element's **val** attribute.

If this element is omitted, then the parent structured document tag shall display all document parts in its default gallery. If this element is present, but no document parts of the specified gallery are located by the application, then document parts in the default gallery shall be displayed (i.e. the application shall behave as if the value was omitted).

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.2.12 **docPartList** (Document Part Gallery Structured Document Tag)

This element specifies that the parent structured document tag shall be of a document part gallery type.

This type setting does not require or imply that the contents of the structured document tag shall contain only the exact contents of a document part of the specified gallery and category which is present on the current machine, it shall only be

used to specify that the structured document tag is of this type, which shall be used by an application to present the possible list of choices for insertion into the parent structured document tag.

2.5.2.13 **docPartObj** (Built-In Document Part Structured Document Tag)

This element specifies that the parent structured document tag shall be of a document part type.

This type setting does not require or imply that the contents of the structured document tag shall contain only the exact contents of a document part of the specified gallery and category which is present on the current machine, it shall only be used to specify that the structured document tag is of this type, which shall be used by an application to present the possible list of choices for insertion into the parent structured document tag.

This element differs from the **docPartList** element (§2.5.2.12) in that it may be used to semantically tag a set of block-level objects in a WordprocessingML document without requiring the ability to specify a category and gallery of objects which may be swapped with it via the user interface.

2.5.2.14 **docPartUnique** (Built-In Document Part)

This element specifies that this structured document tag is being used to encapsulate a built-in document part (i.e. this element appears as a child element of the **docPartObj** element).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.5.2.15 **dropDownList** (Drop-Down List Structured Document Tag)

This element specifies that the parent structured document tag shall be a drop-down list when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The contents shall not be editable when displayed by a hosting application regardless of the locking settings
- The child elements of this element specify choices which shall be displayed in a standard drop-down list format

As well, the structured document tag must satisfy the following restraints or the document shall be considered invalid:

- The contents shall only contain a single run (one set of formatting properties)
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

Attributes	Description
lastValue (Drop-down List Last Saved Value)	<p>Specifies the value associated with the current display text for the drop-down list structured document tag.</p> <p>If this structured document tag is not mapped to XML using the dataBinding element (§2.5.2.6), then this attribute shall be ignored. If this structured document tag is mapped to XML, it shall be used to determine whether the current display text in the combo box structured document tag shall be retained when the document is opened, as follows:</p> <ul style="list-style-type: none"> • When the XML mapping is created, the content in the custom XML data is retrieved • If this content has an associated list item (matching its value attribute), then the corresponding display text shall be displayed in the structured document tag <p>If no list item exists, this content shall be matched against the lastValue attribute value. If the values match, the current display text shall be retained. If the values do not match, the current custom XML data content shall be the new display text (since no match exists in the combo box list items)</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>

2.5.2.16 **equation** (Equation Structured Document Tag)

This element specifies that the parent structured document tag shall be of type equation.

This type setting does not require or imply that the contents of the structured document tag shall contain only an equation or associated placeholder text, it shall only be used to specify that the structured document tag is of this type, which may be used by an application as desired.

2.5.2.17 **group** (Group Structured Document Tag)

This element specifies that the parent structured document tag shall be a restricted grouping when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The contents of this structured document tag shall not be editable when displayed by a hosting application regardless of the locking settings. This restriction may be superseded by any structured document tag contained within the group, as each structured document tag specifies the locking properties for its own content.

2.5.2.18 **id** (Unique ID)

This element specifies a unique numerical ID for the parent structured document tag. This ID shall be persisted through multiple sessions (i.e. shall not be changed once specified).

If multiple structured document tags specify the same decimal number value for the id attribute, then the first structured document tag in the document shall maintain this original ID, and all subsequent structured document tags shall have new identifiers assigned to them when the document is opened.

If this element is omitted, then the parent structured document tag shall have a new unique identifier assigned to it when the document is opened.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.5.2.19 **lid** (Date Picker Language ID)

This element specifies the language ID which shall be used for displaying a calendar for the current date picker structured document tag, if a user interface is present for the structured document tag.

If this element is omitted, then the language ID shall be the language ID of the run contents of the parent structured document tag.

Attributes	Description
val (Language Code)	See 2.3.3.14

2.5.2.20 **listItem** (Combo Box List Item)

This element specifies a single list item within the parent combo box structured document tag. Each list item shall be displayed in the list displayed for the parent structured document tag (if a user interface is present).

Attributes	Description
displayText (List Entry Display Text)	<p>Specifies the text to display in the run content (as well as any supplied user interface) in place of the value attribute contents for this drop-down list entry.</p> <p>This value shall be used as follows:</p> <ul style="list-style-type: none"> • If the parent structured document tag is mapped to a custom XML element, the value in that custom XML element shall be mapped the content of the value attribute, and the resulting displayText attribute value (if one is present) shall be displayed in the run content. If no displayText attribute is present, then the value shall be displayed. • If the corresponding entry is selected via a user interface, this value shall be stored in the parent element's run content in the document (this is the value that shall be shown in the

Attributes	Description
	document's WordprocessingML content). If this attribute is omitted, then the content of the value attribute shall be used as the display text for the current list item entry. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
value (List Entry Value)	Specifies the value for the current list item entry. This value shall be used as follows: <ul style="list-style-type: none"> If the parent structured document tag is mapped to a custom XML element, the value in that custom XML element shall be mapped to this value, and the resulting displayText attribute value (if one is present) shall be displayed in the run content. If no displayText attribute is present, then the value shall be displayed. If the corresponding entry is selected via a user interface, this value shall be stored in the parent element's listItem attribute value. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.5.2.21 **listItem** (Drop-Down List Item)

This element specifies a single list item within the parent drop-down list structured document tag. Each list item shall be displayed in the list displayed for the parent structured document tag (if a user interface is present).

Attributes	Description
See 2.5.2.20	

2.5.2.22 **lock** (Locking Setting)

This element specifies the set of behaviors which shall be applied to the contents of the parent structured document tag when the contents of this documents are edited by an application (whether through a user interface or directly). The type of locking applied to the structured document tag is specified via the value of the associated val attribute.

If this element is omitted, then the locking settings implied for the structured document tag shall be as follows:

- If the structured document tag specifies that it is a group via the **group** element (§2.5.2.17), then the contents of the structured document tag shall be editable, but the entire tag may be deleted.
- For all other types, no locking settings shall be applied to the structured document tag.

Attributes	Description
val (Locking Type)	Specifies the type of locking which shall be applied to the parent structured document tag. If this attribute is omitted, this its value shall be assumed to be unlocked (using the defaults stated above). The possible values for this attribute are defined by the ST_Lock simple type (§2.18.55).

2.5.2.23 **picture** (Picture Structured Document Tag)

This element specifies that the parent structured document tag shall be a picture when displayed in the document.

This setting specifies that the behavior for this structured document tag shall be as follows:

- The contents shall always be restricted to a single picture using either the DrawingML (§Error! Reference source not found.) or VML (§Error! Reference source not found.) syntax

As well, the structured document tag must satisfy the following restraints or the document shall be considered invalid:

- The contents shall only be a single picture using either the DrawingML (§Error! Reference source not found.) or VML (§Error! Reference source not found.) syntax
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

2.5.2.24 **placeholder** (Structured Document Tag Placeholder Text)

This element specifies the placeholder text which should be displayed when this structured document tag's run contents are empty, the associated mapped XML element is empty as specified via the **dataBinding** element (§2.5.2.6) or the **showingPlcHdr** element (§2.5.2.38) is set in the structured document tag's properties. The placeholder text which shall be shown is itself specified via the child element **docPart**.

If this element is omitted, then five non-breaking spaces shall be used as the default placeholder text for this structured document tag.

2.5.2.25 **richText** (Rich Text Structured Document Tag)

This element specifies that the parent structured document tag shall be a rich text box when displayed in the document. If no type element (the `xsd:choice` block in the XML Schema fragment for the parent **sdtPr** element) is specified, then the parent structured document tag shall be of type **richText**.

2.5.2.26 **rPr** (Run Properties For Structured Document Tag Contents)

This element specifies the set of run properties which shall be applied to the text entered into the parent structured document tag in replacement of placeholder text. When placeholder text is present in a structured document tag, its formatting is often different than the desired underlying formatting, and this element specifies the formatting which shall be used for non-placeholder text contents when they are initially added to the control.

If this element is not present, the inserted is unformatted, as with any other run of text - it shall not inherit the properties of the placeholder text.

2.5.2.27 **rPr** (Structured Document Tag End Character Run Properties)

This element specifies the set of run properties which shall be applied to the character present to delimit the end of the structured document tag's contents. When these properties are applied, they shall be applied in addition to the run properties specified for the entire structured document tag via the **rPr** element (§2.5.2.26) stored in the tag's main property container.

If this element is not present, the inserted closing tag shall be formatting identically to the start tag.

2.5.2.28 **sdt** (Cell-Level Structured Document Tag)

This element specifies the presence of a structured document tag around a single table cell. The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the **sdtPr** and **sdtContent** elements, respectively.

2.5.2.29 **sdt** (Inline-Level Structured Document Tag)

This element specifies the presence of a structured document tag around one or more inline-level structures (runs, DrawingML objects, fields, etc.) in the current paragraph. The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the **sdtPr** and **sdtContent** elements, respectively.

2.5.2.30 **sdt** (Block-Level Structured Document Tag)

This element specifies the presence of a structured document tag around one or more block-level structures (paragraphs, tables, etc.). The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the **sdtPr** and **sdtContent** elements, respectively.

2.5.2.31 **sdt** (Row-Level Structured Document Tag)

This element specifies the presence of a structured document tag around a single table row. The two child elements of this element shall be used to specify the properties and content of the current structured document tag via the **sdtPr** and **sdtContent** elements, respectively.

2.5.2.32 **sdtContent** (Block-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around one or more block-level structures (paragraphs, tables, etc.). This element's contents shall be treated as a cache of the contents to be displayed in the structured document tag for the following reasons:

- If the structured document tag specifies an XML mapping via the **dataBinding** element (§2.5.2.6), changes to the custom XML data part shall be reflected in the structured document tag as needed
- If the contents of the structured document tag are placeholder text via the **showingPlcHdr** element (§2.5.2.38), then this content may be updated with the placeholder text stored in the Glossary Document part

2.5.2.33 **sdtContent** (Cell-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around a single table cell. This element's contents shall be treated as a cache of the contents to be displayed in the structured document tag for the following reasons:

- If the structured document tag specifies an XML mapping via the **dataBinding** element (§2.5.2.6), changes to the custom XML data part shall be reflected in the structured document tag as needed
- If the contents of the structured document tag are placeholder text via the **showingPlcHdr** element (§2.5.2.38), then this content may be updated with the placeholder text stored in the Glossary Document part

2.5.2.34 **sdtContent** (Row-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around a single table row.

[*Note*: Unlike other types of structured document tags, this type cannot show placeholder text or have mapped XML data, therefore it is never a cache. *end note*]

2.5.2.35 **sdtContent** (Inline-Level Structured Document Tag Content)

This element specifies the last known contents of a structured document tag around one or more inline-level structures (runs, DrawingML objects, fields, etc.). This element's contents shall be treated as a cache of the contents to be displayed in the structured document tag for the following reasons:

- If the structured document tag specifies an XML mapping via the **dataBinding** element (§2.5.2.6), changes to the custom XML data part shall be reflected in the structured document tag as needed
- If the contents of the structured document tag are placeholder text via the **showingPlcHdr** element (§2.5.2.38), then this content may be updated with the placeholder text stored in the Glossary Document part

2.5.2.36 **sdtEndPr** (Structured Document Tag End Character Properties)

This element specifies the properties which shall be applied to the physical character which delimits the end of a structured document tag.

2.5.2.37 **sdtPr** (Structured Document Tag Properties)

This element specifies the set of properties which shall be applied to the parent structured document tag.

2.5.2.38 **showingPlcHdr** (Current Contents Are Placeholder Text)

This element specifies whether the content of the **sdtContent** element (§2.5.2.32; §2.5.2.33; §2.5.2.34; §2.5.2.35) for the parent structured document tag shall be interpreted to contain placeholder text for this structured document tag (as opposed to regular text contents within the structured document tag). If this element is present and set to **true**, this state shall be resumed (showing placeholder text) upon opening this document.

If this element is omitted, then the structured document tag shall not be interpreted to be showing placeholder text when the document is displayed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.5.2.39 **storeMappedDataAs** (Custom XML Data Date Storage Format)

This element specifies the translation which shall be performed on the displayed date in a date picker structured document tag when the current contents are saved into the associated custom XML data via the **dataBinding** element (§2.5.2.6). If this element is omitted, then the value of the associated custom XML element shall be placed into the custom XML data part with no translation.

Attributes	Description
val (Date Storage Type)	Specifies the type of date translation which shall be applied to the parent date picker structured document tag. If this attribute is omitted, this its value shall be assumed to be text . The possible values for this attribute are defined by the ST_SdtDateMappingType simple type (§2.18.82).

2.5.2.40 **tag** (Programmatic Tag)

This element specifies a programmatic tag associated with the current structured document tag. A *programmatic tag* is an arbitrary string which applications may associate with a structured document tag in order to identify it without providing a visible friendly name. The string representing the programmatic tag shall be stored on this element's **val** attribute. If this element is omitted, then no programmatic tag shall be associated with the given structured document tag.

Attributes	Description
val (String Value)	See 2.3.1.27

2.5.2.41 **temporary** (Remove Structured Document Tag When Contents Are Edited)

This element specifies whether the parent structured document tag shall be removed from the WordprocessingML document when the its contents are modified.

[*Note*: This setting is primarily intended for creating structured document tags whose sole purpose is one-time placeholder text, and which should not return once replaced with content. *end note*]

If this element is omitted, then the parent structured document tag shall not be automatically removed when its contents are modified.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.5.2.42 **text** (Plain Text Structured Document Tag)

This element specifies that the parent structured document tag shall be a plain text box when displayed in the document. This setting specifies that the behavior for this structured document tag shall be as follows:

- Formatting applied to any part of this structured document tag's contents shall apply to its entire contents

As well, the structured document tag must satisfy the following restraints or the document shall be considered invalid:

- The contents shall only be contain a single run (one set of formatting properties) with exceptions for soft carriage returns via the **multiLine** attribute on this element
- The contents shall not contain more than a single paragraph or table cell and shall not contain a table row or table

Attributes	Description
multiLine (Allow Soft Line Breaks)	Specifies whether soft line breaks may be added to the contents of this structured document tag when this document is modified. This setting shall not affect the ability of the structured document tag to display existing soft line breaks (which shall be preserved) and shall only affect the ability to add line breaks when the document is modified by an application.

Attributes	Description
	<p>If this attribute is omitted, then the parent plain text structured document control shall not allow soft line breaks to be added to its contents.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>

2.6 Sections

WordprocessingML does not natively store the concept of pages, since it is based on paragraphs and runs (which are laid out on to pages by consumers of this content). However, although there is no concept of storing pages in the WordprocessingML format, it is often necessary to store information about a page or group of pages in a document, in order to store information that is to be used to format the pages on which a set of paragraphs will appear. In WordprocessingML, this information is stored via the use of *sections*.

In WordprocessingML, *sections* are groupings of paragraphs that have a specific set of properties used to define the pages on which the text will appear, as well as other section-level (applying to all paragraphs' appearance) properties.

2.6.1 bidi (Right to Left Section Layout)

This element specifies that this section shall be presented using a right-to-left page direction. This property only affects section-level properties, and does not affect the layout of text within the contents of this section.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.6.2 bottom (Bottom Border)

This element specifies the presentation and display of the page border displayed at the bottom of each page in this section.

When a document has a bottom border that is relative to the page edges (using the offsetFrom attribute on **pgBorders**), it shall span the bottom edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding left or right page border (if one is specified).
- It reaches the edge of the page.

When a document has a bottom border that is relative to the text (using the offsetFrom attribute on **pgBorders**), it shall span only the necessary width to satisfy the requirement of spanning the width of the text.

Attributes	Description
See 2.3.1.4	

2.6.3 col (Single Column Definition)

This element specifies the properties for a single column of text within this section.

The contents of the **col** element are only used to calculate the number and size of columns if the fixedWidth attribute is set to false or omitted.

Attributes	Description
space (Space Before Following Column)	<p>Specifies the spacing (in twentieths of a point) between the current column and the next column. For the last text column in the section, no spacing is allowed after the column, and, if present, any space value is ignored.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type</p>

Attributes	Description
	(§2.18.104).
w (Column Width)	Specifies the width (in twentieths of a point) of this text column. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).

2.6.4 cols (Column Definitions)

This element specifies the set of columns defined for this section in the document.

Based on the presence of the equalWidth attribute, a consumer shall render the columns using:

- If equalWidth is `true`, then the columns are defined using the data stored as attributes of the **cols** element (defined below).
- If equalWidth is `false`, then the columns are defined using the presence and data on each child **col** element (§2.6.3).

Child Elements	Subclause
col (Single Column Definition)	§2.6.3

Attributes	Description
equalWidth (Equal Column Widths)	<p>Specifies whether all text columns in the current section are of equal width.</p> <p>If this attribute is present and its value is set to <code>true</code>, <code>on</code>, or <code>1</code>, then all columns for this text section are of an equal width and are calculated as follows:</p> <ul style="list-style-type: none"> • Take width of page (from margin to margin) • Divide by number of columns specified in <code>num</code> attribute • For each column, leave space after as defined in the <code>space</code> attribute • Remaining width of each column is the text column width. <p>If this attribute is present and its value is set to <code>false</code>, <code>off</code>, or <code>0</code>, then all columns for this text section are of different widths and are defined by each col element as follows:</p> <ul style="list-style-type: none"> • Each col element defines a single column • Each <code>w</code> attribute defines the text column width <p>Each <code>space</code> attribute defines the space after the text column</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
num (Number of Equal Width Columns)	<p>Specifies the number of text columns in the current section.</p> <p>If all columns are not of equal width (the equalWidth attribute is not set), then this element is ignored, and the number of columns is defined by the number of col elements defined under the cols element.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
sep (Draw Line Between Columns)	<p>Specifies if a vertical line is drawn between each of the text columns in this section.</p> <p>If set to <code>true</code>, <code>on</code>, or <code>1</code>, then a vertical line shall be drawn in the center of the spacing between each column in this section.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
space (Spacing Between Equal Width)	Specifies the spacing between text columns in the current section.

Attributes	Description
Columns)	<p>If all columns are not of equal width (the equalWidth attribute is not set), then this element is ignored, and the spacing after columns is defined by the space attribute on each of the col elements defined under the cols element.</p> <p>The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).</p>

2.6.5 docGrid (Document Grid)

This element specifies the settings for the document grid, which enables precise layout of full-width East Asian language characters within a document by specifying the desired number of characters per line and lines per page for all East Asian text content in this section.

If Latin text is interspersed on this line, then it is placed across the number of grid units needed to fit the content, but all other grid positions are unaffected.

Attributes	Description
charSpace (Document Grid Character Pitch)	<p>Specifies the number of characters to be allowed on the document grid for each line in this section.</p> <p>This attribute's value shall be specified by multiplying the difference between the desired character pitch and the character pitch for that character in the font size of the Normal font by 4096.</p> <p>This value shall then be used to add the character pitch for the specified point size to each character in the section [Note: This results in text in the Normal style having a specific number of characters per line. end note]</p> <p>Individual runs of text can override the line pitch information specified for the document grid by specifying that the run text shall not snap to the document grid via the snapToGrid element (§2.3.2.32).</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
linePitch (Document Grid Line Pitch)	<p>Specifies the number of lines to be allowed on the document grid for the current page assuming all lines have equal line pitch applied to them. This line pitch shall not be added to any line which appears within a table cell unless the adjustLineHeightInTable element (§2.15.3.1) is present in the document's compatibility settings.</p> <p>This attribute is specified in twentieths of a point, and defines the pitch for each line of text on this page such that the desired number of single spaced lines of text fits on the current page. Individual paragraphs can override the line pitch information specified for the document grid by either:</p> <ul style="list-style-type: none"> • Specifying an exact line spacing value using the lineRule attribute of value exact on the spacing element (§2.3.1.33). • Specifying that the paragraph text shall not snap to the document grid via the snapToGrid element (§2.3.1.32). <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
type (Document Grid Type)	Specifies the type of the current document grid, which defines the grid behavior.

Attributes	Description
	The grid can define a grid which snaps all East Asian characters to grid positions, but leaves Latin text with its default spacing; a grid which adds the specified character pitch to each character on each row; or a grid which affects only the line pitch for the current section. The possible values for this attribute are defined by the ST_DocGrid simple type (§2.18.17).

2.6.6 formProt (Only Allow Editing of Form Fields)

This element specifies that the contents of the current section shall be protected such that they cannot be edited by a user (if the consumer is displaying the document and allowing the user to make modification) except for the text contained in any form field or embedded control that is part of the current section.

The enforcement of this property is determined by the **documentProtection** element (§2.15.1.28), as it is possible to specify protection without turning it on.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.6.7 left (Left Border)

This element specifies the presentation and display of the page border displayed at the left of each page in this section.

When a document has a left border that is relative to the page edges (using the offsetFrom attribute value of **page** on **pgBorders**), it shall span the left edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding top or bottom page border (if one is specified).
- It reaches the edge of the page.

When a document has a left border that is relative to the text (using the offsetFrom attribute value of **text** on **pgBorders**), it shall span only the necessary width to satisfy the requirement of spanning the width of the text.

Attributes	Description
See 2.3.1.4	

2.6.8 lnNumType (Line Numbering Settings)

This element specifies the settings for line numbering to be displayed before each column of text in this section in the document.

Attributes	Description
countBy (Line Number Increments to Display)	Specifies the line number increments to be displayed in the current document. Although each line has an associated line number, only lines which are an even multiple of this value shall be displayed. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
distance (Distance Between Text and Line Numbering)	Specifies the distance between the text margin and the edge of any line numbers appearing in that section. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).

Attributes	Description
restart (Line Numbering Restart Setting)	Specifies when the line numbering in this section shall be reset to the line number specified by the start attribute's value. The line numbering increments for each line (even if it is not displayed) until it reaches the restart point specified by this element. The possible values for this attribute are defined by the ST_LineNumberRestart simple type (§2.18.53).
start (Line Numbering Starting Value)	Specifies the starting value used for the first line whenever the line numbering is restarted by use of the restart attribute. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.6.9 paperSrc (Paper Source Information)

This element specifies printer-specific settings for the printer tray(s) that shall be used to print different pages in this section in the document.

Attributes	Description
first (First Page Printer Tray Code)	Specifies a printer-specific code that uniquely identifies a specific printer tray to be used to print the first page of this section in the document. A first value of 1 (the default) is specifically used to indicate that the printer shall automatically select the appropriate printer tray based on the printed page size. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
other (Non-First Page Printer Tray Code)	Specifies a printer-specific code that uniquely identifies a specific printer tray to be used to print the each subsequent (non-first) page of this section in the document. An value of 1 (the default) is specifically used to indicate that the printer shall automatically select the appropriate printer tray based on the printed page size. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.6.10 pgBorders (Page Borders)

This element specifies the page borders for each page in this section. Each child element of the **pgBorders** element specifies a specific of border (**left**, **right**, **bottom**, or **top**).

Attributes	Description
display (Pages to Display Page Borders)	Specifies the pages in this section on which the page border shall be printed. If this attribute is omitted, then the page borders shall be displayed on all pages in this section (equivalent to a value of allPages). The possible values for this attribute are defined by the ST_PageBorderDisplay simple type (§2.18.67).
offsetFrom (Page Border Positioning)	Specifies how the relative positioning of the page borders shall be calculated. If the value of this attribute is page , then the space attribute on each page border shall be interpreted as the distance from the edge of the page that shall be left before the page border.

Attributes	Description
	If the value of this attribute is text, then the space attribute on each page border shall be interpreted as the distance from the text margins that shall be left before the page border. The possible values for this attribute are defined by the ST_PageBorderOffset simple type (§2.18.68).
zOrder (Z-Ordering of Page Border)	Specifies whether the page border is positioned above or below intersecting texts and objects in this document. The possible values for this attribute are defined by the ST_PageBorderZOrder simple type (§2.18.69).

2.6.11 pgMar (Page Margins)

This element specifies the page margins for all pages in this section.

Attributes	Description
bottom (Page Bottom Spacing)	Specifies the distance (in twentieths of a point) between the bottom of the text margins for the main document and the bottom of the page for all pages in this section. If the value of bottom is non-negative, then the text will be placed at the greater of: <ul style="list-style-type: none"> The value of bottom The extent of the footer text A negative value indicates that the contents of the main document shall be measured from the bottom of the page extent regardless of the footer for that document, and therefore shall overlap the footer text. The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).
footer (Spacing to Bottom of Footer)	Specifies the distance (in twentieths of a point) from the bottom edge of the page to the bottom edge of the footer. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
gutter (Page Gutter Spacing)	Specifies the page gutter for each page in the current section. The <i>page gutter</i> defines the amount of extra space added to the specified margin, above any existing margin values. [Note: This setting is typically used when a document is being created for binding, in order to ensure that the resulting margins are present after the binding gutter is consumed by the printed matter binding. <i>end note</i>] The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
header (Spacing to Top of Header)	Specifies the distance (in twentieths of a point) from the top edge of the page to the top edge of the header. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
left (Left Margin Spacing)	Specifies the distance (in twentieths of a point) between the left edge of the page and the left edge of the text extents for this document. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
right (Right Margin Spacing)	Specifies the distance (in twentieths of a point) between the right edge of the page and the right edge of the text extents for this document.

Attributes	Description
	The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
top (Top Margin Spacing)	<p>Specifies the distance (in twentieths of a point) between the top of the text margins for the main document and the top of the page for all pages in this section.</p> <p>If the value of top is non-negative, then the text will be placed at the greater of:</p> <ul style="list-style-type: none"> • The value of top <p>The extent of the header text</p> <p>A negative value indicates that the contents of the main document shall be measured from the top of the page extent regardless of the header for that document, and therefore shall overlap the header text.</p> <p>The possible values for this attribute are defined by the ST_SignedTwipsMeasure simple type (§2.18.87).</p>

2.6.12 pgNumType (Page Numbering Settings)

This element specifies the page numbering settings for all page numbers that appear in the contents of the current section.

Attributes	Description
chapSep (Chapter Separator Character)	<p>Specifies the separator character that shall appear between the chapter and page number, if a chapter style has been set for page numbers in this section.</p> <p>If the chapStyle attribute is not present, or its specified heading level does not have an associated numbering format, then this value is ignored, since no chapter number is output by the field.</p> <p>The possible values for this attribute are defined by the ST_ChapterSep simple type (§2.18.9).</p>
chapStyle (Chapter Heading Style)	<p>Specifies the one-based index of the heading style applied to chapter titles in the document which shall be used as chapter headings in all page numbers for this section, by locating the nearest heading of that style and extracting the numbering information.</p> <p>If the specified heading style does not exist in the current section, or does not have a numbering format, then any previous level heading format shall be used as needed as the specified chapter number. If no heading has numbering information and/or is used in the section, then the chapter and chapter separator shall be omitted from the page numbering data.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
fmt (Page Number Format)	<p>Specifies the number format that shall be used for all page numbering in this section.</p> <p>The possible values for this attribute are defined by the ST_NumberFormat simple type (§2.18.65).</p>
start (Starting Page Number)	<p>Specifies the page number that appears on the first page of the section.</p> <p>If this value is omitted, numbering will continue from the highest page number in the previous section.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>

2.6.13 pgSz (Page Size)

This element specifies the properties (size and orientation) for all pages in the current section.

Attributes	Description
code (Printer Paper Code)	Specifies a printer-specific paper code for the paper type, which shall be used by the printer for pages in this section. This code is stored to ensure the proper paper type is chosen if the specified paper size matches the sizes of multiple paper types supported by the current printer. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
h (Page Height)	Specifies the height (in twentieths of a point) for all pages in the current section. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).
orient (Page Orientation)	Specifies the orientation of all pages in this section. This information is used to determine the actual paper size to use on the printer. This implies that the actual paper size width and height are reversed for pages in this section. If this attribute is omitted, then <code>portrait</code> shall be implied. The possible values for this attribute are defined by the ST_PageOrientation simple type (§2.18.70).
w (Page Width)	This attribute indicates the width (in twentieths of a point) for all pages in the current section. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).

2.6.14 printerSettings (Reference to Printer Settings Data)

This element specifies an explicit relationship to a Printer Settings part containing information about the printer settings used for this section.

If this element is omitted, then no additional settings are associated with this section.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.6.15 right (Right Border)

This element specifies the presentation and display of the page border displayed at the right of each page in this section.

When a document has a right border that is relative to the page edges (using the `offsetFrom` attribute value of `page` on **pgBorders**), it shall span the right edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding top or bottom page border (if one is specified)
- It reaches the edge of the page.

When a document has a right border that is relative to the text (using the `offsetFrom` attribute value of `text` on **pgBorders**), it shall only span the necessary width to satisfy the requirement of spanning the width of the text.

Attributes	Description
See 2.3.1.4	

2.6.16 rtlGutter (Gutter on Right Side of Page)

This element specifies that the page gutter shall be placed on the right side of the page for this section only. The *page gutter* defines the amount of extra space added to the specified margin, above any existing margin values. [Note: This setting is typically used when a document is being created for binding, in order to ensure that the resulting margins are present after the binding gutter is consumed by the printed matter binding. *end note*]

If the gutter is set to the side of the page by the omission of the **gutterAtTop** element (§2.15.1.49), then each section's gutter is placed at the left by default, unless that default is overridden by the **rtlGutter** element.

If the **gutterAtTop** element (§2.15.1.49) is specified and true, then each section's gutter is at the top and this setting is ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.6.17 sectPr (Previous Section Properties)

When specified as a child element of **sectPrChange**, the **sectPr** element specifies a set of section properties that were modified when the document was set to track all revisions.

Attributes	Description
rsidDel (Section Deletion Revision ID)	<p>Specifies a unique identifier used to track the <i>editing session</i> when the section mark for this section was deleted from the document.</p> <p>All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session.</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidR (Section Addition Revision ID)	<p>Specifies a unique identifier used to track the <i>editing session</i> when the section mark for this section was added to the document.</p> <p>All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session.</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidRPr (Physical Section Mark Character Revision ID)	<p>Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted.</p> <p>All rsid* attributes throughout this document of an equal value, if present, shall indicate that</p>

Attributes	Description
	<p>those regions were modified during the same editing session.</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>
rsidSect (Section Properties Revision ID)	<p>Specifies a unique identifier used to track the editing session when the physical character representing this section mark was last formatted.</p> <p>All rsid* attributes throughout this document of an equal value, if present, shall indicate that those regions were modified during the same editing session.</p> <p>A producer may choose to increment the revision save ID value to indicate subsequent editing sessions (editing between save actions) to indicate the order of the saves performed.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>

2.6.18 sectPr (Document Final Section Properties)

This element defines the section properties for the final section of the document. [*Note: For any other section the properties are stored as a child element of the paragraph element corresponding to the last paragraph in the given section. end note*]

Attributes	Description
See 2.6.17	

2.6.19 sectPr (Section Properties)

This element defines the section properties for the a section of the document. [*Note: For the last section in the document, the section properties are stored as a child element of the body element. end note*]

Attributes	Description
See 2.6.17	

2.6.20 textDirection (Text Flow Direction)

This element specifies the direction of the text flow for this section.

Attributes	Description
val (Direction of Text Flow)	See 2.3.1.41

2.6.21 top (Top Border)

This element specifies the presentation and display of the page border displayed at the top of each page in this section.

When a document has a top border that is relative to the page edges (using an offsetFrom attribute value of **page** on **pgBorders**), it shall span the top edge of the page at the location defined by its properties, stopping when:

- It intersects with the corresponding left or right page border (if one is specified)
- It reaches the edge of the page.

When a document has a top border that is relative to the text (using the `offsetFrom` attribute value of `text` on `pgBorders`), it shall only span the necessary width to satisfy the requirement of spanning the width of the text.

Attributes	Description
See 2.3.1.4	

2.6.22 `type` (Section Type)

This element specifies the type of the current section. The section type specifies how the contents of the current section shall be placed relative to the previous section.

WordprocessingML supports five distinct types of section breaks:

- *Next page section breaks* (the default if **type** is not specified), which begin the new section on the following page.
- *Odd page section breaks*, which begin the new section on the next odd-numbered page.
- *Even page section breaks*, which begin the new section on the next even-numbered page.
- *Continuous section breaks*, which begin the new section on the following paragraph. This means that continuous section breaks might not specify certain page-level section properties, since they must be inherited from the following section. These breaks, however, can specify other section properties, such as line numbering and footnote/endnote settings.
- *Column section breaks*, which begin the new section on the next column on the page.

Attributes	Description
val (Section Type Setting)	Specifies the type of the current section. The possible values for this attribute are defined by the <code>ST_SectionMark</code> simple type (§2.18.83).

2.6.23 `vAlign` (Vertical Text Alignment on Page)

This element specifies the vertical alignment for text on pages in the current section, relative to the top and bottom margins in the main document story on each page.

Attributes	Description
val (Vertical Alignment Setting)	See 2.4.80

2.7 Styles

Within a WordprocessingML file, *styles* are predefined sets of table, numbering, paragraph, and/or character properties which can be applied to text within the document. This allows the formatting properties to be stored and managed independently from the content, allowing the look of document content to be changed in a single location (e.g. the look of all first-level headings is changed by changing the style with `styleId` `Heading1` rather than looking for and changing each paragraph in the document).

Each style defined within a WordprocessingML document requires a *style definition*. The style definition contains all of the information needed by a consumer to store and display that style within a WordprocessingML document, and is defined using the **style** element. The style definition for any style in WordprocessingML can be divided into three segments. The complete definition of style properties can be found on the reference for the **style** element (§2.7.3.17):

- General style properties
- Style types
- Type specific formatting properties

Each of these three segments are discussed in the following subclauses.

2.7.1 Style Inheritance

In order to compile the complete set of paragraph and character properties specified by any given style (as appropriate), a consumer must follow the rule of style inheritance to determine each property in that set.

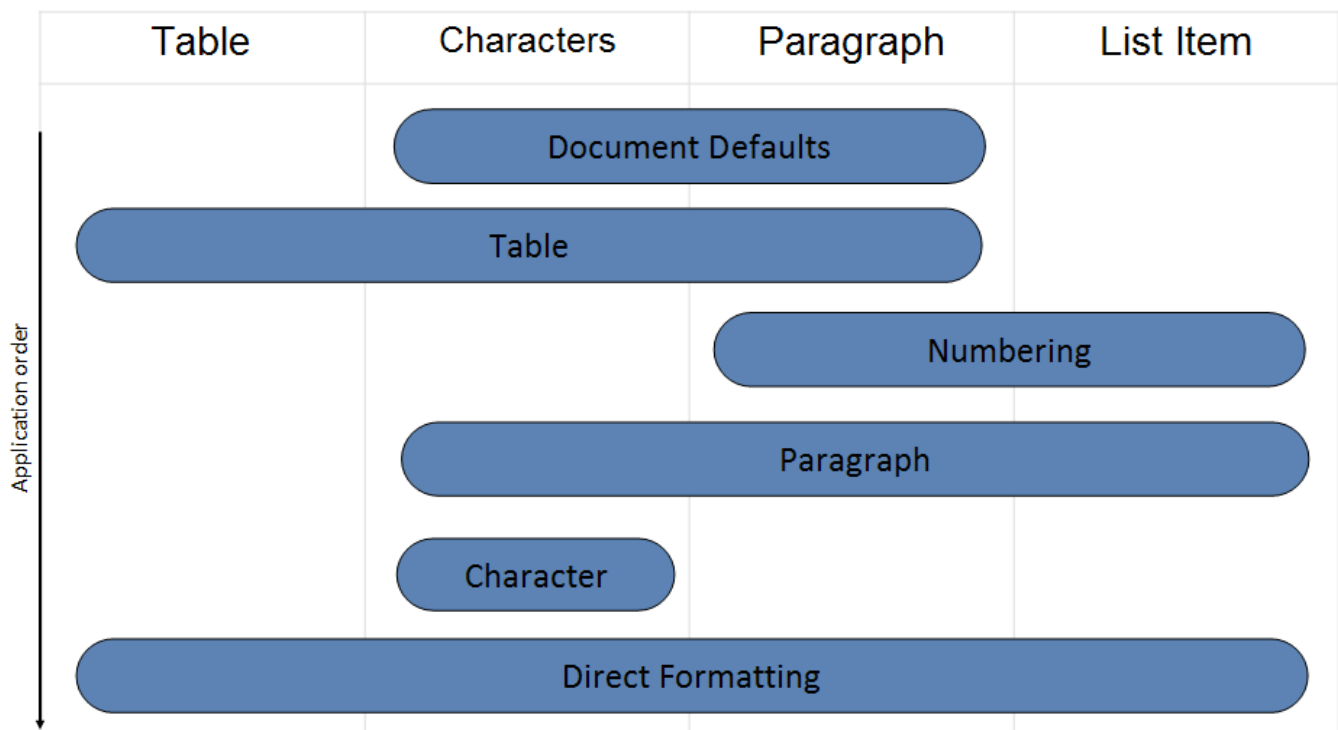
Style inheritance states that styles of any given type may inherit from other styles of that type, and therefore a consumer must 'build up' the style information by following the inheritance tree. This inheritance is defined via the **basedOn** element, which specifies the styleId of the parent style.

To build up the resulting style, a consumer must trace the hierarchy (following each **basedOn** value) back to a style which has no **basedOn** element (is not based on another style). The resulting style is then constructed by following each level in the tree, applying the specified paragraph and/or character properties as appropriate. When properties conflict, they are overridden by each subsequent level (this includes turning OFF a property set at an earlier level). Properties which are not specified simply do not change those specified at earlier levels.

Conversely, a producer should not output any property on a style which has already been set by a previous level of the style hierarchy, as well as those which match the document defaults. This means that if the document defaults or any previous level in a style's hierarchy specify a property which is unchanged at this level, that property should not be part of the style definition in the resulting WordprocessingML.

2.7.2 Style Hierarchy

With the various flavors of styles available (see each of the subclauses below), multiple style types can be applied to the same content within a file, which means that properties must be applied in a specific deterministic order. As with inheritance, the resulting formatting properties set by one type can be unchanged, removed, or altered by following types. The following table illustrates the order of application of these defaults, and which properties are impacted by each:



This process can be described as follows:

- First, the document defaults are applied to all runs and paragraphs in the document.

- Next, the table style properties are applied to each table in the document, following the conditional formatting inclusions and exclusions specified per table.
- Next, numbered item and paragraph properties are applied to each paragraph formatted with a numbering style.
- Next, paragraph and run properties are applied to each paragraph as defined by the paragraph style.
- Next, run properties are applied to each run with a specific character style applied.
- Finally, we apply direct formatting (paragraph or run properties not from styles). If this direct formatting includes numbering, that numbering + the associated paragraph properties are applied.

2.7.3 General Style Properties

General style properties refer to the set of properties which can be used regardless of the type of style.

2.7.3.1 **aliases** (Alternate Style Names)

This element specifies the set of alternative names for the parent style definition. These names may be used in an application's user interface as desired. The alternate names shall be stored in this element's **val** attribute, and each name shall be separated by one or more consecutive comma characters (Unicode character value 002C). All commas present shall be interpreted as separator character and never as part of an alternate style name.

If present, the alternate style names shall be used in the user interface in place of the built-in name specified in the **name** element (§2.7.3.9) when the appropriate value is set in the **stylePaneFormatFilter** element (§2.15.1.87).

If this element is omitted, then the style shall not have any alternate style names.

Attributes	Description
val (String Value)	See 2.3.1.27

2.7.3.2 **autoRedefine** (Automatically Merge User Formatting Into Style Definition)

This element specifies whether an application shall automatically modify this style when the contents of an entire paragraph in the document with this style applied are modified, ensuring that although only a single instance of text with this style was modified, that change is stored on the style and therefore propagated to all locations where the style is in use.

If this element is omitted, then formatting shall not automatically be merged back into the style definition.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.3 **basedOn** (Parent Style ID)

This element specifies the style ID of the parent style from which this style inherits in the style inheritance. The *style inheritance* refers to a set of styles which inherit from one another to produce the resulting set of properties for a single style. The **val** attribute of this element specifies the **styleId** attribute for the parent style in the style inheritance.

If this element is omitted, then this style shall not be based on any other style in the current document (i.e. this element is the root of the style inheritance for a style). If no style in the current document specifies the **styleId** present in the **val** attribute, then this element shall be ignored (i.e. this element is the root of the style inheritance for a style).

If a style with this **styleId** is present, then it shall be subject to the following restrictions:

- If the current style is a table style, then the parent style must also be a table style, or this element shall be ignored.
- If the current style is a paragraph style, then the parent style must also be a paragraph style, or this element shall be ignored.
- If the current style is a character style, then the parent style must also be a character style, or this element shall be ignored.
- If the current style is a numbering style, then this element shall be ignored.

Attributes	Description
val (String Value)	See 2.3.1.27

2.7.3.4 **hidden** (Hide Style From User Interface)

This element specifies whether this style shall be hidden from any and all user interfaces when this document is loaded by an application. If this element is set, then this style may be used to format content (i.e. any content which references this style shall have its properties as normal), but the style shall be hidden from all user interface associated with that application. [*Note*: This setting is typically used to hide styles which are being used internally by an application which should not be used as formatting in a typical case. *end note*]

If this element is omitted, then the style shall not be required to be hidden from the user interface.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.5 **latentStyles** (Latent Style Information)

This element specifies the properties which shall be applied to a set of latent styles for this document. *Latent styles* refer to any set of style definitions known to an application which have not been included in the current document.

When a style definition is embedded in a document, it specifies two distinct groups of properties:

- Behavior properties
- Formatting properties

Obviously, embedding all the styles known to a particular application in each document which it produces would drastically increase the file size. Latent styles provide a way to store pieces of information for the first group (behavior properties) which must be specified for all styles known to an application without requiring the storage of the second group (formatting properties).

Attributes	Description
count (Latent Style Count)	Specifies the number of known styles which shall be initialized to the current latent style defaults when this document is first processed. [<i>Note</i> : This property may be used by an application as needed to ensure that only the number of styles known when this document was created are initialized with the defaults on the parent element, and that all new known styles use their default values. <i>end note</i>] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
defLockedState (Default Style Locking Setting)	Specifies the default setting for the locked element (§2.7.3.7) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§2.7.3.8) exists. If this element is omitted, the default locked state for all latent styles in the current document shall be false . The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
defQFormat (Default Primary Style Setting)	Specifies the default setting for the qFormat element (§2.7.3.14) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§2.7.3.8) exists. If this element is omitted, the default qFormat state for all latent styles in the current document shall be false . The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
defSemiHidden	Specifies the default setting for the semiHidden element (§2.7.3.16) which shall be applied to any

Attributes	Description
(Default Semi-Hidden Setting)	<p>style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§2.7.3.8) exists.</p> <p>If this element is omitted, the default semiHidden state for all latent styles in the current document shall be false.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
defUIPriority (Default User Interface Priority Setting)	<p>Specifies the default setting for the uiPriority element (§2.7.3.19) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§2.7.3.8) exists.</p> <p>If this element is omitted, the default uiPriority state for all latent styles in the current document shall be 99.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
defUnhideWhenUsed (Default Hidden Until Used Setting)	<p>Specifies the default setting for the unhideWhenUsed element (§2.7.3.20) which shall be applied to any style made available by the hosting application which is not explicitly defined in the current document. This setting shall be overridden for every style for which a latent style exception (§2.7.3.8) exists.</p> <p>If this element is omitted, the default unhideWhenUsed state for all latent styles in the current document shall be false.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>

2.7.3.6 [link](#) (Linked Style Reference)

This element specifies the pairing of styles which comprise a linked style. A *linked style* is a grouping of a paragraph style and character style which is used in a user interface to allow the same set of formatting properties to be applied:

- To the contents of one or more entire paragraphs (i.e. as a paragraph style)
- To the contents of one or more runs within a paragraph (i.e. as a character style)

Each style continues to exist independently in the file format as there is both a paragraph and character style present within the **styles** element (§2.7.3.18), however these two styles shall be merged into one and applied appropriately based on whether they are applied to run(s) or paragraph(s), by referencing the styleId attribute of the paired linked style via this element's val attribute.

If this element is omitted, then this style is not part of a linked style pairing. If no style in the current document specifies the styleId present in the val attribute, then this element shall be ignored.

If a style with this styleId is present, then it shall be subject to the following restrictions:

- If the parent style is a table style, then this element shall be ignored.
- If the parent style is a paragraph style, then the parent style must be a character style, or this element shall be ignored.
- If the parent style is a character style, then the parent style must be a paragraph style, or this element shall be ignored.
- If the parent style is a numbering style, then this element shall be ignored.

Attributes	Description
val (String Value)	See 2.3.1.27

2.7.3.7 **locked** (Style Cannot Be Applied)

This element specifies whether an application shall prevent the use of this style when this document is loaded and/or modified. If this element is set, then this style may be used to format existing content (i.e. any content which references this style shall have its properties as normal), but new instances of the style shall be prevented from being applied via all mechanisms associated with that application.

If this element is omitted, then the use of the style shall not be prevented by an application processing this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.8 **IsdException** (Latent Style Exception)

This element specifies the properties which shall be applied a single latent style for this document. *Latent styles* refer to any set of known style definitions which have not been included in the current document.

Attributes	Description
locked (Latent Style Locking Setting)	Specifies the default setting for the locked element (§2.7.3.7) which shall be applied to the latent style with the matching style name value. If this element is omitted, the default locked state for this latent style shall be determined by the defLockedState attribute on the parent latentStyles element. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
name (Primary Style Name)	Specifies the primary name for the style which shall inherit this set of latent style property exceptions. If the current application does not know of an internal primary style with the current name, then this set of latent style exceptions may be ignored. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
qFormat (Latent Style Primary Style Setting)	Specifies the default setting for the qFormat element (§2.7.3.14) which shall be applied to the latent style with the matching style name value. If this element is omitted, the default qFormat state for this latent style shall be determined by the defQFormat attribute on the parent latentStyles element. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
semiHidden (Semi hidden text override)	Specifies the default setting for the semiHidden element (§2.7.3.16) which shall be applied to the latent style with the matching style name value. If this element is omitted, the default semiHidden state for this latent style shall be determined by the defSemiHidden attribute on the parent latentStyles element. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
uiPriority (Override default sorting order)	Specifies the default setting for the uiPriority element (§2.7.3.19) which shall be applied to the latent style with the matching style name value. If this element is omitted, the default uiPriority state for this latent style shall be determined by the defUIPriority attribute on the parent latentStyles element. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
unhideWhenUsed (Unhide when used)	Specifies the default setting for the unhideWhenUsed element (§2.7.3.20) which shall be applied to the latent style with the matching style name value.

Attributes	Description
	If this element is omitted, the default unhideWhenUsed state for this latent style shall be determined by the defUnhideWhenUsed attribute on the parent latentStyles element. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).

2.7.3.9 **name** (Primary Style Name)

This element specifies the primary name for the current style in the document. This name may be used in an application's user interface as desired. The actual primary name for this style is stored in its val attribute.

If present, the alternate style names (§2.7.3.1) shall be used in the user interface in place of the built-in name specified when the appropriate value is set in the **stylePaneFormatFilter** element (§2.15.1.86).

If this element is omitted, then the style shall not have a primary style name.

Attributes	Description
val (String Value)	See 2.3.1.27

2.7.3.10 **next** (Style For Next Paragraph)

This element specifies the style which shall automatically be applied to a new paragraph created following a paragraph with the parent paragraph style applied. [Note: This setting is typically used when the use of the current style is limited to one paragraph at most, and it would typically be undesirable to apply this style to following paragraphs - for example, a title style may specify that its following paragraphs shall return to regular text formatting. *end note*]

If this element is specified on a style of any type other than a paragraph style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a paragraph style, this element shall be ignored.

If this element is omitted, then the following paragraph shall use the same paragraph style as the current paragraph.

Attributes	Description
val (String Value)	See 2.3.1.27

2.7.3.11 **personal** (E-Mail Message Text Style)

This element specifies that the parent style, when in use in the context of an e-mail message, was used by default to format all message text from one or more users. [Note: This setting does not provide any additional semantic about the style, but may be used in the context of e-mail to automatically reformat the contents of the e-mail message while ignoring any content to which styles were deliberately applied (since this style was implicitly applied to message text without user interaction). *end note*]

If this element is specified on a style of any type other than a character style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a character style, this element shall be ignored.

If this element is omitted, then the current style shall not be considered a message text style in the context of e-mail messages.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.12 **personalCompose** (E-Mail Message Composition Style)

This element specifies that the parent style, when in use in the context of an e-mail message, may be used by default to format new message text within the e-mail message. [Note: This setting does not provide any additional semantic about the style, but may be used in the context of e-mail to automatically format the contents of new text in the e-mail message. *end note*]

If this element is specified on a style of any type other than a character style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a character style, this element shall be ignored.

If this element is omitted, then the current style shall not be considered a message composition text style in the context of e-mail messages.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.13 **personalReply** (E-Mail Message Reply Style)

This element specifies that the parent style, when in use in the context of an e-mail message, may be used by default to format existing message text within the e-mail message when a new reply is generated. [*Note: This setting does not provide any additional semantic about the style, but may be used in the context of e-mail to automatically format the contents of existing text in the e-mail message. end note*]

If this element is specified on a style of any type other than a character style, this element shall be ignored. If no style whose styleId matches the val attribute of this element exists or that style is not a character style, this element shall be ignored.

If this element is omitted, then the current style shall not be considered a message reply text style in the context of e-mail messages.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.14 **qFormat** (Primary Style)

This element specifies whether this style shall be treated as a primary style when this document is loaded by an application. If this element is set, then this style has been designated as being particularly important for the current document, and this information may be used by an application in any means desired. [*Note: This setting does not imply any behavior for the style, only that the style is of particular significance for this document. end note*]

If this element is omitted, then the style shall not be considered a primary style for this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.15 **rsid** (Revision Identifier for Style Definition)

This element specifies a unique four digit number which shall be used to determine the editing session in which this style definition was last modified. This value shall follow this following constraint: All document elements which specify the same rsid* values must correspond to changes made during the same editing session. An *editing session* is defined as the period of editing which takes place between any two subsequent save actions. [*Note: This setting does not imply any behavior for the style, only that the style was last modified during one particular editing session. This information may be interpreted by an application in any manner desired. end note*]

If this element is omitted, then no revision identifier shall be associated with the parent style definition.

Attributes	Description
val (Long Hexadecimal Number Value)	Specifies a number value specified as a four digit hexadecimal number), whose contents of this decimal number are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).

2.7.3.16 **semiHidden** (Hide Style From Main User Interface)

This element specifies whether this style shall be hidden from the main user interface when this document is loaded by an application. If this element is set, then this style may be used to format content (i.e. any content which references this style shall have its properties as normal), but the style shall be hidden from the main user interface associated with that application.

[*Note:* The interpretation of a "main" user interface shall not be dictated by this Office Open XML Standard, and may be defined by an application as appropriate.

This setting is intended to define a style property which allows styles to be seen and modified in an advanced user interface, without exposing the style in a less advanced setting, for example, the style which is used to format the contents of a comment should typically not be shown in a simple user interface (as it is uncommon to want to modify it), but would be inappropriate to hide completely using the **hidden** element (§2.7.3.4), as very advanced users may want to change its appearance. *end note*]

If this element is omitted, then the style shall not be required to be hidden from the main user interface.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.3.17 **style** (Style Definition)

This element specifies the definition of a single style within a WordprocessingML document. A *style* is a predefined set of table, numbering, paragraph, and/or character properties which can be applied to regions within a document.

The style definition for any style definition can be divided into three segments:

- General style properties
- Style type
- Type-specific properties

General style properties refers to the set of properties which can be used regardless of the type of style; for example, the style name, additional aliases for the style, a style ID (used by the document content to refer to the style), if style is hidden, if style is locked, etc.

Style types refers to the property on a style which defines the type of style created with this style definition.

WordprocessingML supports six types of style definitions by the values for the style definition's type attribute:

- Paragraph styles
- Character styles
- Linked styles (paragraph + character) [*Note:* Accomplished via the **link** element (§2.7.3.6). *end note*]
- Table styles
- Numbering styles
- Default paragraph + character properties

Type-specific properties refers to the payload of the style: its formatting information as well as any properties which apply only to that type of style.

Attributes	Description
customStyle (User-Defined Style)	Specifies that this style is a user-defined style (i.e. it is not a style which was automatically generated by an application). This setting (specifically a value of true or its equivalents) shall not allow the formatting associated with the style to be changed automatically by an application, but may be used to specify that if the associated style ID is known, certain user interface behaviors may be applied to its definition. If this attribute is omitted, then the style shall be assumed to be a built-in style. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
default (Default Style)	Specifies that this style is the default for this type of style.

Attributes	Description
	<p>This property is used in conjunction with the type attribute to determine the style which is applied to objects that do not explicitly declare a style.</p> <p>If this attribute is not specified for any style, then no properties shall be applied to objects of the specified type. If this attribute is specified by multiple styles, then the last instance of a style with this property shall be used.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
styleId (Style ID)	<p>Specifies a unique identifier for the parent style definition. This identifier shall be used in multiple contexts to uniquely reference this style definition within the document.</p> <p>If multiple style definitions each declare the same value for their styleId, then the first such instance shall keep its current identifier with all other instances being reassigned in any manner desired. This reassignment shall not require references to those style definitions to be 'repaired' in the content (i.e. some content may lose its style definition information, since the document was ill-formed).</p> <p>If this attribute is not specified, then a style ID may be assigned in any manner desired.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
type (Style Type)	<p>Specifies the type of style definition defined by this element. WordprocessingML supports six types of style definitions:</p> <ul style="list-style-type: none"> • Paragraph styles • Character styles • Table styles • Numbering styles • Linked styles (paragraph + character) • Default paragraph + character properties <p>Each of the first four types corresponds to a different value in this attribute, and therefore defines the type of the current style. [Note: The last two types are unique in that they are not simply a style type: a linked style is a pairing of a character and paragraph style via the link element (§2.7.3.6); and the document default properties are defined via the docDefaults element (§2.7.4.1). <i>end note</i>]</p> <p>If this attribute is not specified, then the default value shall be assumed to be paragraph.</p> <p>The possible values for this attribute are defined by the ST_StyleType simple type (§2.18.89).</p>

2.7.3.18 **styles** (Style Definitions)

This element specifies all of the style information stored in the WordprocessingML document: style definitions as well as latent style information.

2.7.3.19 **uiPriority** (Optional User Interface Sorting Order)

This element specifies a number which may be used to sort the set of style definitions in a user interface when this document is loaded by an application and the recommended setting is specified in the **stylePaneSortMethod** element (§2.15.1.87). If this element is set, then this priority shall be used to sort all available styles in ascending value order. If this element is omitted, then the style shall not have an associated priority value and shall be sorted to the end of the list of style definitions (more or less equivalent to a priority value of infinity) when the recommended sort order setting is specified.

Attributes	Description
------------	-------------

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.7.3.20 **unhideWhenUsed** (Remove Semi-Hidden Property When Style Is Used)

This element specifies whether the **semiHidden** property (§2.7.3.16) shall be removed when this style is used by the content of the document. If this element is set, then an application shall ensure that even if the **semiHidden** element is specified on a style, that this property is removed when the document is resaved if the style is referenced by any content in the document.

If this element is omitted, then the style shall not automatically lose the semi-hidden property when it is used in the document contents.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.7.4 Document Defaults

The first formatting information which is applied to all regions of text in a WordprocessingML document when that document is displayed is the document defaults. The document defaults specify the default set of properties which shall be inherited by every paragraph and run of text within all stories of the current WordprocessingML document. If no other formatting information was referenced by that text, these properties would solely define the formatting of the resulting text.

2.7.4.1 **docDefaults** (Document Default Paragraph and Run Properties)

This element specifies the set of default paragraph and run properties which shall be applied to every paragraph and run in the current WordprocessingML document. These properties are applied first in the style hierarchy; therefore they are superseded by any further conflicting formatting, but apply if no further formatting is present.

If this element is omitted, then the document defaults shall be application-defined by the hosting application.

2.7.4.2 **pPr** (Paragraph Properties)

This element specifies the set of paragraph properties which comprise the default paragraph properties for the current WordprocessingML document. [*Rationale: The reason that a **pPr** element is present within the **pPrDefault** element is to allow for easy repurposing of any set of paragraph properties within a WordprocessingML document - since the paragraph properties are always child elements of a single **pPr** element, that element can simply be relocated in its entirety to the desired new location without additional modifications. end rationale*]

If this element is omitted, then the default paragraph properties for the current document are non-existent (i.e. there are no default paragraph properties, and the defaults are therefore application-defined).

2.7.4.3 **pPrDefault** (Default Paragraph Properties)

This element specifies the presence of a set of default paragraph properties for the current document. The actual paragraph properties are stored within the **pPr** child element of the current element.

If this element is omitted, then the default paragraph properties for the current document are non-existent (i.e. there are no default paragraph properties in the document, and the defaults are therefore application-defined).

2.7.4.4 **rPr** (Run Properties)

This element specifies the set of run properties which comprise the default run properties for the current WordprocessingML document. [*Rationale: The reason that an **rPr** element is present within the **rPrDefault** element is to allow for easy repurposing of any set of run properties within a WordprocessingML document - since the run properties are always child elements of a single **rPr** element, that element can simply be relocated in its entirety to the desired new location without additional modifications. end rationale*]

If this element is omitted, then the default run properties for the current document are non-existent (i.e. there are no default run properties, and the defaults are therefore application-defined).

2.7.4.5 **rPrDefault** (Default Run Properties)

This element specifies the presence of a set of default run properties for the current document. The actual run properties are stored within the **rPr** child element of the current element.

If this element is omitted, then the default run properties for the current document are non-existent (i.e. there are no default run properties in the document, and the defaults are therefore application-defined).

2.7.5 **Table Styles**

Table styles are style definitions which apply to the contents of zero or more tables within a document. This definition may imply that the style can only define table properties (properties which apply to the table and its constituent rows and cells), however a table style can also define paragraph properties (properties which apply to the positioning and appearance of paragraphs) as well as character properties (properties which apply to runs) for all of the paragraphs and runs within the specified table in the document.

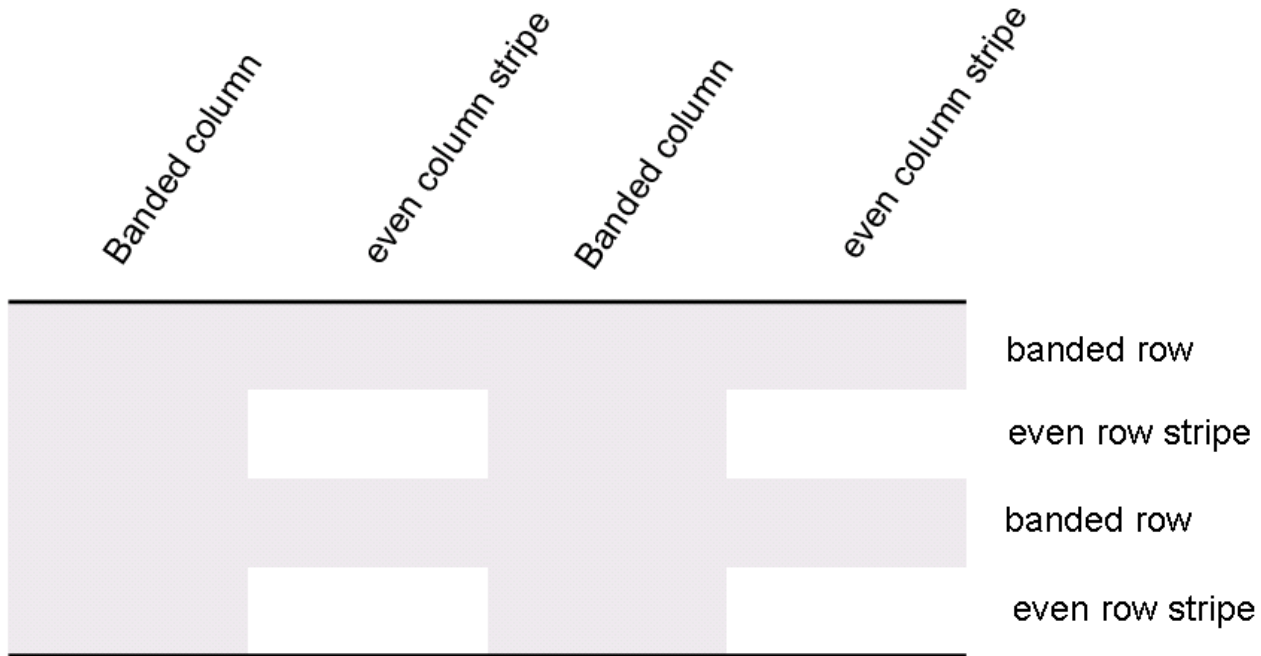
Table styles can only be referenced by tables within a document, and they must be referenced by the **tblStyle** element (§2.4.59) within a table's table properties.

As discussed above, table styles can specify all of the properties that can be applied to a table, as well as paragraph and character properties for the table's contents. However, unlike other style definitions, table styles allow for the definition of conditional formats for different regions of the table.

These table conditional formats are applied to different regions of the table as follows:

Top left cell	Header row	Top right cell
First column	Table body	Last column
Bottom left cell	Footer row	Bottom right cell

All rows in the table can also have conditional formatting on an alternating row/column basis as well as follows:



When specified, these conditional formats shall be applied in the following order (therefore subsequent formats override properties on previous formats):

- Whole table
- Banded columns, even column banding
- Banded rows, even row banding
- First row, last row
- First column, last column
- Top left, top right, bottom left, bottom right

An individual instance of a table defines an association with a table style using the **tblStyle** element in the table's properties (**tblPr**), as discussed above. However, individual tables can choose whether to apply the following aspects of the table's conditional formats individually:

- First row
- Last row
- First column
- Last column
- Row banding
- Column banding

The use or omission conditional formats shall be specified using the **tblLook** element, which contains a bitmask representing which properties are applied and omitted.

2.7.5.1 **pPr** (Table Style Conditional Formatting Paragraph Properties)

This element specifies the set of paragraph properties which shall be applied to all paragraphs within a table which match the conditional formatting type specified on the parent **tblStylePr** element. These properties are applied in the order specified via the style hierarchy.

2.7.5.2 **rPr** (Table Style Conditional Formatting Run Properties)

This element specifies the set of run properties which shall be applied to all runs within a table which match the conditional formatting type specified on the parent **tblStylePr** element. These properties are applied in the order specified via the style hierarchy.

2.7.5.3 **tblPr** (Table Style Conditional Formatting Table Properties)

This element specifies the set of table properties which shall be applied to all regions within a table which match the conditional formatting type specified on the parent **tblStylePr** element. These properties are applied in the order specified via the style hierarchy.

If the current conditional formatting type does not consist of one or more full table rows, then table properties which cannot be applied to a single cell or column may be ignored.

2.7.5.4 **tblPr** (Style Table Properties)

This element specifies the set of table properties which shall be applied to the table. These properties are not conditional and shall always be applied (although they are applied before all conditional formatting properties).

2.7.5.5 **tblStyleColBandSize** (Number of Columns in Column Band)

This element specifies the number of columns which shall comprise each a table style column band for this table style. This element determines how many columns constitute each of the column bands for the current table, allowing column band formatting to be applied to groups of columns (rather than just single alternating columns) when the table is formatted. If this element is omitted, then the default number of columns in a single column band shall be assumed to be 1.

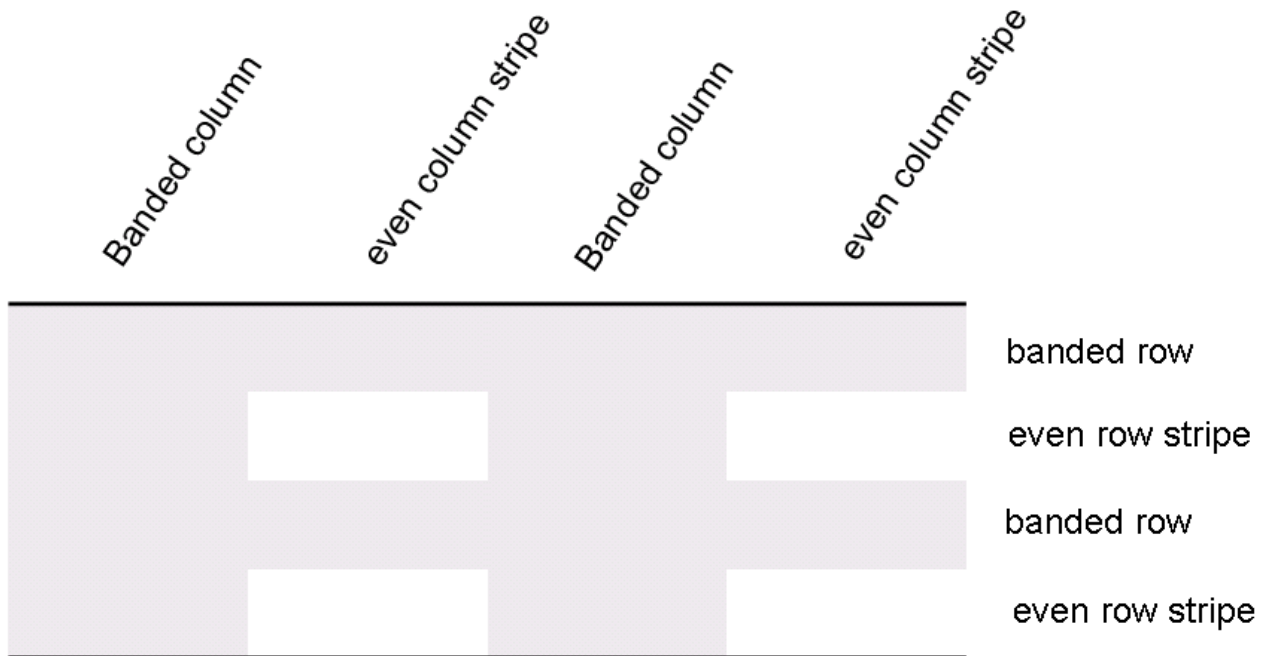
Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.7.5.6 **tblStylePr** (Style Conditional Table Formatting Properties)

This element specifies a set of formatting properties which shall be conditionally applied to the parts of a table which match the requirement specified on the type attribute. These table conditional formats are applied to different regions of the table as follows:

Top left cell	Header row	Top right cell
First column	Table body	Last column
Bottom left cell	Footer row	Bottom right cell

All rows in the table can also have conditional formatting on an alternating row/column basis as well as follows:



When specified, these conditional formats shall be applied in the following order (therefore subsequent formats override properties on previous formats):

- Whole table
- Banded columns, even column banding
- Banded rows, even row banding
- First row, last row
- First column, last column
- Top left, top right, bottom left, bottom right

Attributes	Description
type (Table Style Conditional Formatting Type)	Specifies the section of the table to which the current conditional formatting properties shall be applied. The possible values for this attribute are defined by the ST_TblStyleOverrideType simple type (§2.18.95).

2.7.5.7 **tblStyleRowBandSize** (Number of Rows in Row Band)

This element specifies the number of rows which shall comprise each a table style row band for this table style. This element determines how many rows constitute each of the row bands for the current table, allowing row band formatting to be applied to groups of rows (rather than just single alternating rows) when the table is formatted.

If this element is omitted, then the default number of rows in a single row band shall be assumed to be 1.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.7.5.8 **tcPr** (Style Table Cell Properties)

This element specifies the set of table cell properties which shall be applied to the table. These properties are not conditional and shall always be applied (although they are applied before all conditional formatting properties).

2.7.5.9 **tcPr** (Table Style Conditional Formatting Table Cell Properties)

This element specifies the set of table cell properties which shall be applied to all regions within a table which match the conditional formatting type specified on the parent **tblStylePr** element. These properties are applied in the order specified via the style hierarchy.

2.7.5.10 **trPr** (Table Style Conditional Formatting Table Row Properties)

This element specifies the set of table row properties which shall be applied to all rows within a table which match the conditional formatting type specified on the parent **tblStylePr** element. These properties are applied in the order specified via the style hierarchy.

2.7.5.11 **trPr** (Style Table Row Properties)

This element specifies the set of table row properties which shall be applied to the table. These properties are not conditional and shall always be applied (although they are applied before all conditional formatting properties).

2.7.6 Numbering Styles

Numbering styles are style definitions which specify common style properties for a multi-level numbering format within a document. This means that a numbering style defines only a single paragraph property: a reference to a numbering definition stored in the document's numbering part, using the **numPr** element.

Unlike paragraph and character styles, numbering styles are never directly referenced by content in the document – instead, an abstract numbering definition (covered in the numbering section) specifies that it is actually the underlying numbering information for a numbering style.

2.7.7 Paragraph Styles

Paragraph styles are styles which apply to the contents of an entire paragraph as well as the paragraph mark. This definition implies that the style can define both character properties (properties which apply to text within the document) as well as paragraph properties (properties which apply to the positioning and appearance of the paragraph). Paragraph styles cannot be referenced by runs within a document; they must be referenced by the **pStyle** element (§2.3.1.27) within a paragraph's paragraph properties element.

A paragraph style has three defining type-specific characteristics:

- The type attribute on the style has a value of **paragraph**, which indicates that the following style definition is a paragraph style.
- The **next** element defines an editing behavior which supplies the paragraph style to be automatically applied to the next paragraph when ENTER is pressed at the end of a paragraph of this style.
- The style specifies both paragraph-level and character-level properties using the **pPr** and **rPr** elements, respectively. In this case, the run properties are the set of properties applied to each run in the paragraph.

The paragraph style is then applied to paragraphs by referencing the styleId attribute value for this style in the paragraph properties' **pStyle** element.

2.7.7.1 Numbering in Paragraph Styles

When a paragraph style references a numbering definition and level which shall also be applied, that reference shall be done in a way slightly different from the typical numbering reference as follows:

- When a numbering reference is created as direct formatting, that reference consists of a reference to the numbering definition instance + a numbering level
- When numbering is done as part of a paragraph style, that reference consists of a reference to the numbering definition only. The numbering definition then in turn has a reference to the paragraph style on the level which shall be associated with this style

When numbering is referenced by a paragraph style, its properties shall be applied before the style's properties (the style's paragraph properties shall override the numbering level's paragraph properties).

2.7.7.2 pPr (Style Paragraph Properties)

This element specifies the set of paragraph properties which shall be applied to the paragraph.

2.7.8 Run (Character) Styles

Character styles are styles which apply to the contents of one or more runs of text within a document's contents. This definition implies that the style can only define character properties (properties which apply to text within a paragraph) because it cannot be applied to paragraphs. Character styles can only be referenced by runs within a document, and they must be referenced by the **rStyle** element within a run's run properties element.

A character style has two defining type-specific characteristics:

- The type attribute on the style has a value of **character**, which indicates that the following style definition is a character style.
- The style specifies only character-level properties using the **rPr** element. In this case, the run properties are the set of properties applied to each run which is of this style.

The character style is then applied to runs by referencing the styleId attribute value for this style in the run properties' **rStyle** element.

2.7.8.1 rPr (Run Properties)

This element specifies the set of run properties which shall be applied to the run.

2.8 Fonts

The next component of a WordprocessingML document is storing information about the fonts used in the document.

WordprocessingML stores two pieces of information about fonts:

- (optionally) Information about the font to enable font substitution. *Font substitution* is a process by which an application, when it cannot locate a specific font, attempts to locate the closest possible match to the intended appearance of the font
- (optionally) One or more embedded forms of the font for use on systems which do not have access to the font. When fonts are embedded, they are obfuscated to ensure that they are only used to view the contents of the document in which they are embedded, and for no other purpose.

2.8.1 Font Embedding

Within a WordprocessingML document, *font embedding* refers to a process in which the some or all of the fonts used in the current document are included in that document such that it can be guaranteed that they are available for use when the document is subsequently opened.

Embedded fonts are stored in an Embedded Font part within the package.

When a font is embedded within a WordprocessingML document, it shall be obfuscated to prevent it from being used outside of this document. This obfuscation shall be done using the following algorithm:

- Generate a GUID, which will be used and stored as the obfuscation key
- Reverse the order of the bytes in the GUID (i.e. Big Endian ordering)
- XOR the value with the first 32 bytes of the binary: once against 0-15, once against 16-31
- Store the resulting file in the document, and store the obfuscation key in the fontKey attribute

To retrieve an obfuscated font for viewing the content of this document only, repeat the procedure above to retrieve the original font.

2.8.2 Elements

The following elements comprise the content of the font table:

2.8.2.1 altName (Alternate Names for Font)

This element specifies a set of alternative names which may be used to locate the font specified by the parent element. This set of alternative names is stored in a comma-delimited list, with all adjacent commas ignored (i.e. a value of Name A, Name B is equivalent to Name A,,,,,, Name B).

When an application cannot locate a font using the primary name stored on the font attribute of the **font** element (§2.8.2.10), it should use each alternate name in turn to attempt to locate the font, and use the first font for which it locates a match.

If this element is omitted, then no alternate names are present for the parent font.

Attributes	Description
val (String Value)	See 2.3.1.27

2.8.2.2 charset (Character Set Supported By Font)

This element specifies the character set which is supported by the parent font. This information may be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

The value of this element shall be interpreted as follows:

Value	Description
0x00	Specifies the ANSI character set.
0x01	Specifies the default character set.
0x02	Specifies the Symbol character set.
0x4D	Specifies a Macintosh (Standard Roman) character set.
0x80	Specifies the JIS character set.
0x81	Specifies the Hangul character set.
0x82	Specifies a Johab character set.
0x86	Specifies the GB-2312 character set.
0x88	Specifies the Chinese Big Five character set.
0xA1	Specifies a Greek character set.
0xA2	Specifies a Turkish character set.
0xA3	Specifies a Vietnamese character set.
0xB1	Specifies a Hebrew character set.
0xB2	Specifies an Arabic character set.
0xBA	Specifies a Baltic character set.
0xCC	Specifies a Russian character set.
0xDE	Specifies a Thai character set.
0xEE	Specifies an Eastern European character set.
0xFF	Specifies an OEM character set not defined by this Office Open XML Standard.
Any other value	Application-defined, may be ignored.

If this element is not present, then the character set for this font shall be assumed to be the ANSI character set.

Attributes	Description
val (Value)	Specifies a value specified as single octet (two digit) hexadecimal number whose contents are interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_UcharHexNumber simple type (§2.18.105).

2.8.2.3 **embedBold** (Bold Style Font Style Embedding)

This element specifies information about the embedded font storage for the bold form of a font, when it is embedded. This form is used when bold is applied to a text run.

If this element is omitted, then no bold form of the font is stored in the document. The relationship targeted by the id attribute must be of the embedded font type, or the document shall be considered to be invalid.

Attributes	Description
fontKey (Embedded Font Obfuscation Key)	Specifies the key which was used to obfuscate this embedded font. This key may be used to retrieve the embedded font for the purposes of viewing this WordprocessingML document only, using the algorithm described in §2.8.1. If this attribute is omitted, then no key is provided for this font. The possible values for this attribute are defined by the ST_Guid simple type (§2.18.38).
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17
subsetting (Embedded Font Is Subsetting)	Specifies that the embedded font targeted by the id attribute has been subsetting. <i>Subsetting</i> is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font. If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetting. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).

2.8.2.4 **embedBoldItalic** (Bold Italic Font Style Embedding)

This element specifies information about the embedded font storage for the bold italic form of a font, when it is embedded. This form is used when bold and italics are applied to a text run.

If this element is omitted, then no bold italic form of the font is stored in the document.

Attributes	Description
See 2.8.2.3	

2.8.2.5 **embedItalic** (Italic Font Style Embedding)

This element specifies information about the embedded font storage for the italic form of a font, when it is embedded. This form is used when italics are applied to a text run.

If this element is omitted, then no italic form of the font is stored in the document.

Attributes	Description
See 2.8.2.3	

2.8.2.6 **embedRegular** (Regular Font Style Embedding)

This element specifies information about the embedded font storage for the regular form of a font, when it is embedded.

This form is used when neither bold nor italics is applied to a text run.

If this element is omitted, then no regular form of the font is stored in the document.

Attributes	Description
See 2.8.2.3	

2.8.2.7 **embedSystemFonts** (Embed Common System Fonts)

This element specifies that applications shall embed common system fonts when they are in use and font embedding is enabled for this document using the **embedTrueTypeFonts** element (§2.8.2.8). *Common system fonts* refer to a set of fonts which are typically always present on a machine, and are not defined by this Office Open XML Standard.

If this element is omitted, then the set of fonts defined as common system fonts should not be embedded in the current document when font embedded is turned on. If the **embedTrueTypeFonts** element is omitted or `false`, then this setting has no effect.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.8.2.8 **embedTrueTypeFonts** (Embed TrueType Fonts)

This element specifies that applications shall embed the fonts in use in this document when it is saved. These fonts shall be embedded subject to the algorithm specified in §2.8.1.

If this element is omitted, then fonts in use should not be embedded in the current document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.8.2.9 **family** (Font Family)

This element specifies the font family of the current font. This information may be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then its value shall be assumed to be `auto`.

Attributes	Description
val (Font Family Value)	Specifies the font family for the parent font. The possible values for this attribute are defined by the <code>ST_FontFamily</code> simple type (§2.18.33).

2.8.2.10 **font** (Properties for a Single Font)

This element specifies the properties for one of the fonts used in this document. A **font** element shall be written out for each font face used in the document, and includes:

- The name of the font as used in the document's stories
- (optionally) Font metrics allowing other applications to locate appropriate substitute fonts as needed
- (optionally) Embedded forms of the font

Attributes	Description
name (Primary Font Name)	Specifies the primary name of the current font. This name shall be used to link the information stored in this element with uses of this value in the rFonts element (§2.3.2.24) in document content. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.8.2.11 **fonts** (Font Table Root Element)

This element specifies the root element for a font table part within a WordprocessingML document, and specifies information about the fonts used in this document, each contained within a child **font** element.

2.8.2.12 **notTrueType** (Raster or Vector Font)

This element specifies that this font is not a TrueType or OpenType font, but is rather a raster or vector font. This information may be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then the font shall be assumed to be a TrueType or OpenType font.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.8.2.13 **panose1** (Pansose-1 Typeface Classification Number)

This element specifies the Panose-1 classification number for the current font using the PANOSE Classification Guide, Version 1.2. This information may be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then no Panose-1 information is available.

Attributes	Description
val (Value)	Specifies the Panose-1 classification number for the font, stored as a series of two digit hexadecimal encodings of each digits of the Panose number. The possible values for this attribute are defined by the ST_Panose simple type (§2.18.71).

2.8.2.14 **pitch** (Font Pitch)

This element specifies the font pitch of the current font. This information may be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

If this element is omitted, then its value shall be assumed to be **default**.

Attributes	Description
val (Value)	Specifies the font pitch for the font. The possible values for this attribute are defined by the ST_Pitch simple type (§2.18.72).

2.8.2.15 **saveSubsetFonts** (Subset Fonts When Embedding)

This element specifies that applications shall subset fonts when font embedding is enabled for this document using the **embedTrueTypeFonts** element (§2.8.2.8). *Subsetting* is a mechanism by which only the glyphs used in the contents of this WordprocessingML document are stored in an embedded font, in order to prevent the file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.

If this element is omitted, then the set of fonts should not be subsetted in the current document when font embedded is turned on. If the **embedTrueTypeFonts** element is omitted or **false**, then this setting has no effect.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.8.2.16 sig (Supported Unicode Subranges and Code Pages)

This element specifies information identifying the code pages and Unicode subranges for which the parent font provides glyphs. This information may be used as defined in font substitution logic to locate an appropriate substitute font when this font is not available. This information is determined by querying the font when present and shall not be modified when the font is not available.

When storing Unicode subrange information, the appropriate bit in the bitfield shall only be set if the entire subrange is supported by that font.

If this element is omitted, then no supported code page/Unicode subrange information is available.

Attributes	Description																																				
csb0 (Lower 32 Bits of Code Page Bit Field)	<p>Specifies a four digit hexadecimal encoding of the first 32 bits of the 64-bit code-page bit field that identifies which specific character sets or code pages are supported by the parent font.</p> <p>Each bit in this 32 bits represents the following code page:</p> <table> <tr> <th>Bit</th><th>Description</th></tr> <tr><td>0</td><td>Latin 1</td></tr> <tr><td>1</td><td>Latin 2: Eastern Europe</td></tr> <tr><td>2</td><td>Cyrillic</td></tr> <tr><td>3</td><td>Greek</td></tr> <tr><td>4</td><td>Turkish</td></tr> <tr><td>5</td><td>Hebrew</td></tr> <tr><td>6</td><td>Arabic</td></tr> <tr><td>7</td><td>Windows Baltic</td></tr> <tr><td>8 to 16</td><td>Reserved for Alternate ANSI</td></tr> <tr><td>17</td><td>Thai</td></tr> <tr><td>18</td><td>JIS/Japan</td></tr> <tr><td>19</td><td>Chinese (Simplified)</td></tr> <tr><td>20</td><td>Korean Wansung</td></tr> <tr><td>21</td><td>Chinese (Traditional)</td></tr> <tr><td>22 to 29</td><td>Reserved for Alternate ANSI and OEM</td></tr> <tr><td>30</td><td>Macintosh Character Set (Standard Roman)</td></tr> <tr><td>31</td><td>Symbol Character Set</td></tr> </table> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>	Bit	Description	0	Latin 1	1	Latin 2: Eastern Europe	2	Cyrillic	3	Greek	4	Turkish	5	Hebrew	6	Arabic	7	Windows Baltic	8 to 16	Reserved for Alternate ANSI	17	Thai	18	JIS/Japan	19	Chinese (Simplified)	20	Korean Wansung	21	Chinese (Traditional)	22 to 29	Reserved for Alternate ANSI and OEM	30	Macintosh Character Set (Standard Roman)	31	Symbol Character Set
Bit	Description																																				
0	Latin 1																																				
1	Latin 2: Eastern Europe																																				
2	Cyrillic																																				
3	Greek																																				
4	Turkish																																				
5	Hebrew																																				
6	Arabic																																				
7	Windows Baltic																																				
8 to 16	Reserved for Alternate ANSI																																				
17	Thai																																				
18	JIS/Japan																																				
19	Chinese (Simplified)																																				
20	Korean Wansung																																				
21	Chinese (Traditional)																																				
22 to 29	Reserved for Alternate ANSI and OEM																																				
30	Macintosh Character Set (Standard Roman)																																				
31	Symbol Character Set																																				
csb1 (Upper 32 Bits of Code Page Bit Field)	<p>Specifies a four digit hexadecimal encoding of the upper 32 bits of the 64-bit code-page bit field that identifies which specific character sets or code pages are supported by the parent font.</p> <p>Each bit in this 32 bits represents the following code page:</p> <table> <tr> <th>Bit</th><th>Description</th></tr> </table>	Bit	Description																																		
Bit	Description																																				

Attributes	Description																																		
	<table border="1"> <tr><td>0 to 15</td><td>Reserved for OEM</td></tr> <tr><td>16</td><td>IBM Greek</td></tr> <tr><td>17</td><td>MS-DOS Russian</td></tr> <tr><td>18</td><td>MS-DOS Nordic</td></tr> <tr><td>19</td><td>Arabic</td></tr> <tr><td>20</td><td>MS-DOS Canadian French</td></tr> <tr><td>21</td><td>Hebrew</td></tr> <tr><td>22</td><td>MS-DOS Icelandic</td></tr> <tr><td>23</td><td>MS-DOS Portuguese</td></tr> <tr><td>24</td><td>IBM Turkish</td></tr> <tr><td>25</td><td>IBM Cyrillic</td></tr> <tr><td>26</td><td>Latin 2</td></tr> <tr><td>27</td><td>MS-DOS Baltic</td></tr> <tr><td>28</td><td>Greek (former 437G)</td></tr> <tr><td>29</td><td>Arabic (AMSO 708)</td></tr> <tr><td>30</td><td>WE/Latin 1</td></tr> <tr><td>31</td><td>US</td></tr> </table> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>	0 to 15	Reserved for OEM	16	IBM Greek	17	MS-DOS Russian	18	MS-DOS Nordic	19	Arabic	20	MS-DOS Canadian French	21	Hebrew	22	MS-DOS Icelandic	23	MS-DOS Portuguese	24	IBM Turkish	25	IBM Cyrillic	26	Latin 2	27	MS-DOS Baltic	28	Greek (former 437G)	29	Arabic (AMSO 708)	30	WE/Latin 1	31	US
0 to 15	Reserved for OEM																																		
16	IBM Greek																																		
17	MS-DOS Russian																																		
18	MS-DOS Nordic																																		
19	Arabic																																		
20	MS-DOS Canadian French																																		
21	Hebrew																																		
22	MS-DOS Icelandic																																		
23	MS-DOS Portuguese																																		
24	IBM Turkish																																		
25	IBM Cyrillic																																		
26	Latin 2																																		
27	MS-DOS Baltic																																		
28	Greek (former 437G)																																		
29	Arabic (AMSO 708)																																		
30	WE/Latin 1																																		
31	US																																		
usb0 (First 32 Bits of Unicode Subset Bitfield)	Specifies the first 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).																																		
usb1 (Second 32 Bits of Unicode Subset Bitfield)	Specifies the second 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).																																		
usb2 (Third 32 Bits of Unicode Subset Bitfield)	Specifies the third 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).																																		
usb3 (Fourth 32 Bits of Unicode Subset Bitfield)	Specifies the fourth 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).																																		

2.9 Numbering

Numbering refers to symbols - Arabic numerals, Roman numerals, symbol characters ("bullets"), text strings, etc. - in WordprocessingML that are used to label individual paragraphs of text.

The basis for all numbering in WordprocessingML is specified via two structures:

- abstract numbering definitions

- numbering definition instances

Abstract numbering definitions define the appearance and behavior of a specific set of numbered paragraphs in a document. Because this construct is abstract, they are not be directly referenced by document content, but rather they must be inherited by a *numbering definition instance*, which itself is referenced by document content.

2.9.1 abstractNum (Abstract Numbering Definition)

This element specifies a set of properties which shall dictate the appearance and behavior of a set of numbered paragraphs in a WordprocessingML document. These properties are collectively called an *abstract numbering definition*, and are the basis for all numbering information in a WordprocessingML document.

Although an abstract numbering definition contains a complete set of numbering, it shall not be directly referenced by content (hence the use of *abstract*). Instead, these properties shall be inherited by a numbering definition instance using the **num** element (§2.9.16), which can then itself be referenced by content.

Attributes	Description
abstractNumId (Abstract Numbering Definition ID)	Specifies a unique number which shall be used as the identifier for this abstract numbering definition. This unique number shall be referenced by any numbering definition instance in order to inherit the properties specified by this abstract numbering definition. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.9.2 abstractNumId (Abstract Numbering Definition Reference)

This element specifies the abstract numbering definition information whose properties shall be inherited by the parent numbering definition instance.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.3 ilvl (Numbering Level Reference)

This element specifies the numbering level of the numbering definition instance which shall be applied to the parent paragraph.

This numbering level is specified on either the abstract numbering definition's **lvl** element (§2.9.7), and may be overridden by a numbering definition instance level override's **lvl** element (§2.9.6).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.4 isLgl (Display All Levels Using Arabic Numerals)

This element specifies whether or not all previous levels displayed for a given numbering level's text shall be displayed using the decimal number format, regardless of the actual number format of that level in the list. [Note: This numbering style is often referred to as the legal numbering style. end note]

If this element is present, then all numbering levels present in the **lvlText** element (§2.9.12) shall be converted to their decimal equivalents when they are displayed in this level in the numbering format. If this element is omitted, then each level is displayed using the **numFmt** (§2.9.18) of that level.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.9.5 legacy (Legacy Numbering Level Properties)

This element specifies that a given numbering level is from an earlier word processing application which did not support the full richness of the numbering properties supported by WordprocessingML.

These properties shall be used to render any numbered paragraph which references this numbering level if the legacy attribute is set. [Note: Using this element in generated WordprocessingML documents is not recommended, as updated numbering structures in WordprocessingML should be used in its place. This element is provided solely to save and roundtrip the numbering properties of legacy word processing products in WordprocessingML such that they are recreated if the document is resaved in an older word processor format. *end note*]

Attributes	Description
legacy (Use Legacy Numbering Properties)	<p>Specifies whether the legacy numbering properties present for this numbering level shall be used to format the numbering for any paragraph which references it.</p> <p>A value of <code>on</code>, <code>1</code>, or <code>true</code> for this attribute value specifies that the legacy numbering properties shall be applied. This is the default value for this attribute, and is implied when the attribute is omitted.</p> <p>A value of <code>off</code>, <code>0</code>, or <code>false</code> for this attribute value specifies that the legacy numbering properties shall not be used, and shall be explicitly turned off.</p> <p>The possible values for this attribute are defined by the <code>ST_OnOff</code> simple type (§2.18.66).</p>
legacyIndent (Legacy Indent)	<p>Specifies the indentation which shall be applied to a legacy numbering symbol from the text margin of the document. This value is specified in twentieths of a point.</p> <p>If this attribute is not present, then no indentation shall be applied with respect to the margin. The possible values for this attribute are defined by the <code>ST_SignedTwipsMeasure</code> simple type (§2.18.87).</p>
legacySpace (Legacy Spacing)	<p>Specifies the indentation which shall be applied between a legacy numbering symbol and the accompanying text of the associated paragraph in the document. This value is specified in twentieths of a point.</p> <p>If this attribute is not present, then no indentation shall be applied with respect to the paragraph text.</p> <p>The possible values for this attribute are defined by the <code>ST_TwipsMeasure</code> simple type (§2.18.104).</p>

2.9.6 lvl (Numbering Level Override Definition)

This element specifies the appearance and behavior of a specific numbering level within a given numbering level definition override defined using the **lvlOverride** element (§2.9.9).

A numbering level override definition is identical to a numbering level definition, except for the fact that it is defined as part of a numbering definition instance using the **num** element (§2.9.16) rather than as part of an abstract numbering definition using the **abstractNum** element (§2.9.1).

Attributes	Description
ilvl (Numbering Level)	<p>Specifies the numbering level definition that is to be defined by this set of numbering properties.</p> <p>This override is a zero-based index of the number of list levels in the document.</p> <p>The possible values for this attribute are defined by the <code>ST_DecimalNumber</code> simple type (§2.18.15).</p>
tentative (Tentative)	Specifies that a given numbering level was been saved by a producer but was not used in the

Attributes	Description
Numbering)	<p>parent document. This means that this numbering level may be redefined by a future consumer without changing the actual content of the document.</p> <p>A value of <code>on</code>, <code>1</code>, or <code>true</code> for this attribute value specifies that the numbering level is not used in the current document's contents.</p> <p>A value of <code>off</code>, <code>0</code>, or <code>false</code> for this attribute value specifies that the numbering level is used in the parent document and cannot be redefined without changing its contents. This is the default value for this attribute, and is implied when this attribute is omitted.</p> <p>If this attribute is equal to <code>on</code>, <code>1</code>, or <code>true</code>, the WordprocessingML for a given document will contain the numbering level information associated with this numbering level, but the 'tentative' numbering level(s) shall not be represented in any of the hosting application's user interface pertaining to numbering levels.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
tplc (Template Code)	<p>Specifies a unique hexadecimal value which may be used to specify a location within an application's user interface in which this numbering level shall be displayed. The method by which this value is interpreted shall be application-defined. If this attribute is omitted, then this numbering may be displayed in any location chosen by the consumer.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>

2.9.7 **lvl** (Numbering Level Definition)

This element specifies the appearance and behavior of a numbering level within a given abstract numbering definition. A numbering level contains a set of properties for the display of the numbering for a given numbering level within an abstract numbering definition.

A numbering level definition is identical to a numbering level override definition, except for the fact that it is defined as part of a numbering definition instance using the **abstractNum** element (§2.9.1) rather than as part of an abstract numbering definition using the **num** element (§2.9.16).

Attributes	Description
ilvl (Numbering Level)	<p>Specifies the numbering level definition that is to be defined by this set of numbering properties.</p> <p>This override is a zero-based index of the number of list levels in the document.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
tentative (Tentative Numbering)	<p>Specifies that a given numbering level was been saved by a producer but was not used in the parent document. This means that this numbering level may be redefined by a future consumer without changing the actual content of the document.</p> <p>A value of <code>on</code>, <code>1</code>, or <code>true</code> for this attribute value specifies that the numbering level is not used in the current document's contents.</p> <p>A value of <code>off</code>, <code>0</code>, or <code>false</code> for this attribute value specifies that the numbering level is used in the parent document and cannot be redefined without changing its contents. This is the default value for this attribute, and is implied when this attribute is omitted.</p> <p>If this attribute is equal to <code>on</code>, <code>1</code>, or <code>true</code>, the WordprocessingML for a given document will contain the numbering level information associated with this numbering level, but the 'tentative' numbering level(s) shall not be represented in any of the hosting application's user interface</p>

Attributes	Description
	pertaining to numbering levels. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
tplc (Template Code)	Specifies a unique hexadecimal value which may be used to specify a location within an application's user interface in which this numbering level shall be displayed. The method by which this value is interpreted shall be application-defined. If this attribute is omitted, then this numbering may be displayed in any location chosen by the consumer. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).

2.9.8 **lvljc (Justification)**

This element specifies the type of justification used on a numbering level's text within a given numbering level. This justification is applied relative to the text margin of the parent numbered paragraph in the document.

If omitted, the paragraph shall have left justification relative to the text margin in left-to-right paragraphs, and right justification relative to the text margin in right-to-left paragraphs.

A numbering level's text is the numeral, symbol, character, graphic, etc. used to create a numbered paragraph as defined by the **lvlText** element (§2.9.12).

Attributes	Description
val (Alignment Type)	See 2.3.1.13

2.9.9 **lvlOverride (Numbering Level Definition Override)**

This element specifies an optional override which shall be applied in place of zero or more levels from the abstract numbering definition for a given numbering definition instance. Each instance of this element is used to override the appearance and behavior of a given numbering level definition within the given abstract numbering definition.

Attributes	Description
ilvl (Numbering Level ID)	Specifies the numbering level of a given abstract numbering definition to be overridden. If this number conflicts with the ilvl of the child lvl element, then the latter shall be ignored. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.9.10 **lvlPicBulletId (Picture Numbering Symbol Definition Reference)**

This element specifies a picture which shall be used as a numbering symbol for a given numbering level by referring to a picture numbering symbol definition's **numPictBullet** element (§2.9.21). This reference is made through this element's **val** attribute.

The picture shall be added to the numbering level by replacing each character in the **lvlText** with one instance of this image

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.11 **lvlRestart (Restart Numbering Level Symbol)**

This element specifies a one-based index which determines when a numbering level should restart to its **start** value (§2.9.27). A numbering level restarts when an instance of the specified numbering level, which shall be higher (earlier than the this level) is used in the given document's contents.

If this element is omitted, the numbering level shall restart each time the previous numbering level is used. If the specified level is higher than the current level, then this element shall be ignored. As well, a value of 0 shall specify that this level shall never restart.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.12 **lvlText (Numbering Level Text)**

This element specifies the textual content which shall be displayed when displaying a paragraph with the given numbering level.

All text in this element's **val** attribute shall be taken as literal text to be repeated in each instance of this numbering level, except for any use of the percent symbol (%) followed by a number, which shall be used to indicate the one-based index of the number to be used at this level. Any number of a level higher than this level shall be ignored.

When the % syntax is used, the number shall be incremented for each subsequent paragraph of that level (sequential or not), until the restart level is seen between two subsequent paragraphs of this level.

Attributes	Description
null (Level Text Is Null Character)	Specifies that a null character shall be used as the numbering symbol for a given numbering level. If the val attribute contains any content, then this attribute shall be ignored. If this attribute is omitted, then the null string shall not be used in place of the empty string. [Note: A null character is different from an empty string. <i>end note</i>] The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
val (Level Text)	Specifies the actual text to be used for the numbering level when it is referenced in the document's content. If this attribute is not specified, then the empty string shall be used as the level's text. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.9.13 **multiLevelType (Abstract Numbering Definition Type)**

This element specifies the type of numbering defined by a given abstract numbering type. This information shall only be used by a consumer to determine user interface behaviors for this numbering definition, and shall not be used to limit the behavior of the list (i.e. a list with multiple levels marked as `singleLevel` shall not be prevented from using levels 2 through 9).

If this element is omitted, then the list shall be assumed to be of any type desired by the consumer.

Attributes	Description
val (Abstract Numbering Definition Type)	Specifies the specific type of numbering enabled by a given abstract numbering definition. The possible values for this attribute are defined by the ST_MultiLevelType simple type (§2.18.64).

2.9.14 name (Abstract Numbering Definition Name)

This element specifies the name of a given abstract numbering definition. This name may be surfaced in order to provide a user friendly alias for a given numbering definition, but shall not influence the behavior of the list - two identical definitions with different **name** elements shall behave identically.

If this element is omitted, then this abstract numbering definition shall have no name.

Attributes	Description
val (String Value)	See 2.3.1.27

2.9.15 nsid (Abstract Numbering Definition Identifier)

This element associates a unique hexadecimal ID to the parent abstract numbering definition. This number shall be identical for two abstract numbering definitions that are based from the same initial numbering definition - if a document is repurposed and the underlying numbering definition is changed, it shall maintain its original **nsid**.

If this element is omitted, then the list shall have no **nsid** and one may be added by a producer arbitrarily.

[*Note*: This element may be used to determine the abstract numbering definition to be applied to a numbered paragraph copied from one document and pasted into another. Consider a case in which a given numbered paragraph associated with a abstract numbering definition with **nsid** FFFFFFF23, is pasted among numbered paragraphs associated with a completely different appearance and an abstract numbering definition with an **nsid** of FFFFFFF23. Here, because of the distinction enabled by the identical **nsid** values, the hosting application would not have to arbitrarily keep the pasted numbered paragraph associated with its original abstract numbering definition, as it may use the information provided by the abstract numbering definition's identical **nsid** values to know that those two numbering sets are identical, and merge the paragraphs into the target numbering format. *end note*]

Attributes	Description
val (Long Hexadecimal Number Value)	See 2.7.3.15

2.9.16 num (Numbering Definition Instance)

This element specifies a unique instance of numbering information that can be referenced by zero or more paragraphs within the parent WordprocessingML document.

This instance requires the referencing of a base abstract numbering definition through the abstractNumId child element (§2.9.2). This element also can be used to specify a set of optional overrides applied to zero or more levels from the abstract numbering definition inherited by this instance second through the optional **lvlOverride** child elements (§2.9.9).

Attributes	Description
numId (Numbering Definition Instance ID)	Specifies a unique ID which any numbered paragraph which wishes to inherit these numbering properties shall reference using the numPr element (§2.3.1.19). The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.9.17 numbering (Numbering Definitions)

This element specifies the formatting, display, and functionality of numbering - Arabic numerals, Roman numerals, symbol characters ("bullets"), text strings, etc. - in WordprocessingML documents, which are used to label individual paragraphs of text.

2.9.18 numFmt (Numbering Format)

This element specifies the number format which shall be used to display all numbering at this level in the numbering definition. This information is used to replace the level text string %x, where x is a particular one-based level index, with the

appropriate value. This value shall be calculated by counting the number of paragraphs at this level since the last restart using the numbering system defined in the `val` attribute.

If omitted, the level shall be assumed to be of type `decimal`.

Attributes	Description
val (Numbering Format Type)	Specifies the number format that shall be used for all numbering in the parent object. The possible values for this attribute are defined by the <code>ST_NumberFormat</code> simple type (§2.18.65).

2.9.19 numId (Numbering Definition Instance Reference)

This element specifies the numbering definition instance which shall be used for the given parent numbered paragraph in the WordprocessingML document.

A value of 0 for the `val` attribute shall never be used to point to a numbering definition instance, and shall instead only be used to designate the removal of numbering properties at a particular level in the style hierarchy (typically via direct formatting).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.20 numIdMacAtCleanup (Last Reviewed Abstract Numbering Definition)

This element specifies to a consumer the progress in the last attempt made by the application to remove unused abstract numbering definitions from a given document. If a legacy document is opened by a consumer, it may choose to remove abstract numbering definition which are 'orphaned' (have no associated numbering definition instances). This element is used by those consumers to indicate their progress (if not complete) in reviewing existing abstract numbering definitions. [Note: Removing unused abstract numbering definition from a document will reduce the file size, but is not required. *end note*]

If omitted, then all abstract numbering definitions shall be considered reviewed.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.21 numPicBullet (Picture Numbering Symbol Definition)

This element specifies the appearance and behavior of a specific picture to be used as the numbering symbol within a numbering level definition in a document, and is the basis for all picture numbering symbol information in a WordprocessingML document.

This element is not used directly within abstract numbering definitions but rather is referenced through its `numPicBulletId` attribute by the **lvlPicBulletId** element (§2.9.10) used within numbering level definitions.

Attributes	Description
numPicBulletId (Picture Numbering Symbol ID)	Specifies a unique ID for this picture bullet definition which shall be used to reference this picture bullet from a numbering level definition. The possible values for this attribute are defined by the <code>ST_DecimalNumber</code> simple type (§2.18.15).

2.9.22 numStyleLink (Numbering Style Reference)

This element specifies an abstract numbering does not contain the actual numbering properties for its type, but rather serves as a reference to a numbering style stored in the document, which shall be applied when this abstract numbering definition is referenced, and itself points at the actual underlying abstract numbering definition to be used.

The numbering style that is to be applied when this abstract numbering definition is referenced is identified by the string contained in **numStyleLink**'s **val** attribute.

Attributes	Description
val (String Value)	See 2.3.1.27

2.9.23 pict (Picture Numbering Symbol Properties)

This element specifies the properties for a picture which shall be used as a picture numbering symbol in a given document, using the VML syntax.

2.9.24 pPr (Numbering Level Associated Paragraph Properties)

This element specifies the paragraph properties which shall be applied as part of a given numbering level within the parent numbering definition. These paragraph properties are applied to any numbered paragraph that references the given numbering definition and numbering level.

Paragraph properties specified on the numbered paragraph itself override the paragraph properties specified by **pPr** elements within a numbering **lvl** element (§2.9.6, §2.9.7).

2.9.25 pStyle (Paragraph Style's Associated Numbering Level)

This element specifies the name of a paragraph style which shall automatically this numbering level when applied to the contents of the document. When a paragraph style is defined to include a numbering definition, any numbering level defined by the **numPr** element (§2.3.1.19) shall be ignored, and instead this element shall specify the numbering level associated with that paragraph style.

If this element references a style which does not exist, or is not a paragraph style, then it may be ignored.

Attributes	Description
val (String Value)	See 2.3.1.27

2.9.26 rPr (Numbering Symbol Run Properties)

This element specifies the run properties which shall be applied to the numbering level's text specified in the **lvlText** element (§2.9.12) when it is applied to paragraphs in this document.

These run properties are applied to all numbering level text used by a given abstract numbering definition and numbering level. It should be noted that run properties specified on a numbered paragraph itself, or on text runs within a numbered paragraph, are separate from the run properties specified by **rPr** elements within a numbering level, as the latter affects only the numbering text itself, not the remainder of runs in the numbered paragraph.

2.9.27 start (Starting Value)

This element specifies the starting value for the numbering used by the parent numbering level within a given numbering level definition. This value is used when this level initially starts in a document, as well as whenever it is restarted via the properties set in the **lvlRestart** element (§2.9.11).

If this element is omitted, then the starting value shall be zero (0).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.28 startOverride (Numbering Level Starting Value Override)

This element specifies the number which the specified level override shall begin with. This value is used when this level initially starts in a document, as well as whenever it is restarted via the properties set in the **lvlRestart** element (§2.9.11). If they disagree, this value shall override the starting number of the child **lvl** element (§2.9.6).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.9.29 styleLink (Numbering Style Definition)

This element specifies that the parent abstract numbering definition is the base numbering definition for the specified numbering style referenced in its **val** attribute.

If this element is omitted, or it references a style which does not exist, then this numbering definition shall not be the underlying properties for a numbering style.

[Note: Numbering styles are never directly referenced by paragraphs or runs in the document – instead, an abstract numbering definition specifies that it contains the underlying numbering information for a numbering style, and one or more numbering definition instances reference a numbering definition which inherits from it. The numbering style itself is just a friendly name on an abstract numbering definition. *end note*]

Attributes	Description
val (String Value)	See 2.3.1.27

2.9.30 suff (Content Between Numbering Symbol and Paragraph Text)

This element specifies the content which shall be added between a given numbering level's text and the text of every numbered paragraph which references that numbering level.

If this element is omitted, then its value shall be assumed to be `tab`.

Attributes	Description
val (Character Type Between Numbering and Text)	Specifies the character which shall follow the list number. The possible values for this attribute are defined by the <code>ST_LevelSuffix</code> simple type (§2.18.52).

2.9.31 tmpl (Numbering Template Code)

This element specifies a unique hexadecimal code which may be used to determine a location within application user interface in which this abstract numbering definition shall be displayed.

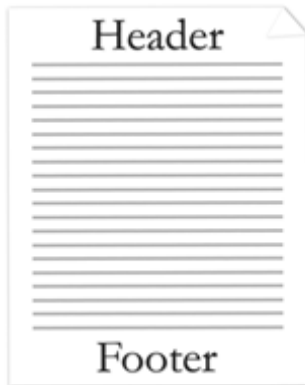
If this element is omitted, then this abstract numbering definition may be displayed in any location chosen by the consumer.

Attributes	Description
val (Long Hexadecimal Number Value)	See 2.7.3.15

2.10 Headers and Footers

Headers and footers refer to text, graphics or data (such as page number, date, document title, and so on) that can appear at the top or bottom of each page in a WordprocessingML document.

A header appears in the top margin (above the main document content on the page), while a footer appears in the bottom margin of a document page (below the main document content on the page).



Since WordprocessingML is a flow-based format, headers and footers are applied by specifying the headers and footers for all pages in a particular section of a document.

Within each section of a document there may be up to three different types of headers and footers:

- First page header/footer
- Odd page header/footer
- Even page header/footer

First page headers and footers specify a unique header or footer which shall appear on the first page of a section. Odd page headers and footers specify a unique header and footer which shall appear on all odd numbered pages for a given section. Even page headers and footers specify a unique header and footer which shall appear on all even numbered pages in a given section.

2.10.1 evenAndOddHeaders (Different Even/Odd Page Headers and Footers)

This element specifies whether sections in this document shall have different headers and footers for even and odd pages (an odd page header/footer and an even page header/footer).

If the `val` attribute is set to `true`, then each section in the document shall use an odd page header for all odd numbered pages in the section, and an even page header for all even numbered pages in the section (counting each page in the section starting from one, regardless of the page numbering settings for the section). If the `val` attribute is set to `false`, then all pages in a section shall use the odd page header.

This setting does not affect the presence of a first page header on each section, which is specified using the **titlePg** element (§2.10.6). If a first page header is specified, then all subsequent pages shall have this setting applied, including the first page in the odd/even page count.

If this element is set to `false` and an even page header is specified, then it shall be ignored and only the odd page header shall be displayed. Conversely, if this element is set to `true` and either header type is omitted for a given section, then a blank header shall be created as needed (another header type shall not be used in its place).

If this element is omitted, then its value shall be assumed to be `false`.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.10.2 footerReference (Footer Reference)

This element specifies a single footer which shall be associated with the current section in the document. This footer shall be referenced via the `id` attribute, which specifies an explicit relationship to the appropriate Footer part in the WordprocessingML package.

If the relationship type of the relationship specified by this element is not

<http://schemas.openxmlformats.org/officeDocument/2006/footer>, is not present, or does not have a `TargetMode` attribute value of `Internal`, then the document shall be considered non-conformant.

Within each section of a document there may be up to three different types of footers:

- First page footer
- Odd page footer

- Even page footer

The footer type specified by the current **footerReference** is specified via the type attribute.

If any type of footer is omitted for a given section, then the following rules shall apply.

- If no **footerReference** for the first page footer is specified and the **titlePg** element is specified, then the first page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created. If the **titlePg** element is not specified, then no first page footer shall be shown, and the odd page footer shall be used in its place.
- If no **footerReference** for the even page footer is specified and the **evenAndOddHeaders** element is specified, then the even page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created. If the **evenAndOddHeaders** element is not specified, then no even page footer shall be shown, and the odd page footer shall be used in its place.
- If no **footerReference** for the odd page footer is specified then the even page footer shall be inherited from the previous section or, if this is the first section in the document, a new blank footer shall be created.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17
type (Header or Footer Type)	Specifies the type of header or footer specified by the target relationship ID. This type determines the page(s) on which the current header or footer shall be displayed. If any section contains more than a single header or footer of each type, then the document shall be considered non-conformant. The possible values for this attribute are defined by the ST_HdrFtr simple type (§2.18.40).

2.10.3 ftr (Footer)

This element specifies the content for a single footer for use within one or more sections of a WordprocessingML document.

Within the **ftr** element, the content of the element is similar to the content of the **body** (§2.2.2) element, and contains what is referred to as *block-level markup* - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

2.10.4 hdr (Header)

This element specifies the content for a single header for use within one or more sections of a WordprocessingML document.

Within the **hdr** element, the content of the element is similar to the content of the **body** (§2.2.2) element, and contains what is referred to as *block-level markup* - markup which can exist as a sibling element to paragraphs in a WordprocessingML document.

2.10.5 headerReference (Header Reference)

This element specifies a single header which shall be associated with the current section in the document. This header shall be referenced via the id attribute, which specifies an explicit relationship to the appropriate Header part in the WordprocessingML package.

If the relationship type of the relationship specified by this element is not

<http://schemas.openxmlformats.org/officeDocument/2006/header>, is not present, or does not have a TargetMode attribute value of Internal, then the document shall be considered non-conformant.

Within each section of a document there may be up to three different types of headers:

- First page header

- Odd page header
- Even page header

The header type specified by the current **headerReference** is specified via the type attribute.

If any type of header is omitted for a given section, then the following rules shall apply.

- If no **headerReference** for the first page header is specified and the **titlePg** element is specified, then the first page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created. If the **titlePg** element is not specified, then no first page header shall be shown, and the odd page header shall be used in its place.
- If no **headerReference** for the even page header is specified and the **evenAndOddHeaders** element is specified, then the even page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created. If the **evenAndOddHeaders** element is not specified, then no even page header shall be shown, and the odd page header shall be used in its place.
- If no **headerReference** for the odd page header is specified then the even page header shall be inherited from the previous section or, if this is the first section in the document, a new blank header shall be created.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17
type (Header or Footer Type)	See 2.10.2

2.10.6 titlePg (Different First Page Headers and Footers)

This element specifies whether the parent section in this document shall have a different header and footer for its first page.

If the val attribute is set to **true**, then the parent section in the document shall use a first page header for the first page in the section. If the val attribute is set to **false**, then the first page in the parent section shall use the odd page header.

This setting does not affect the presence of even and odd page header on all sections, which is specified using the **evenAndOddHeaders** element (§2.10.1).

If this element is set to **false** and a first page header is specified, then it shall be ignored and only the odd page header shall be displayed. Conversely, if this element is set to **true** and the first page header type is omitted for the given section, then a blank header shall be created as needed (another header type shall not be used in its place).

If this element is omitted, then its value shall be assumed to be **false**.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.11 Footnotes and Endnotes

Footnotes and *endnotes* are separate text storied used in documents and books to show the source of borrowed material or to enter explanatory or supplementary information which does not interrupt the normal reading flow of the document.

Footnotes are typically located at the bottom of a page or beneath text being referenced, and *endnotes* are typically placed at the end of a document or at the end of a section. If document has been divided up into one or more sections, each section of a document may contain endnotes.

Both footnotes and endnotes consist of two parts:

- A note reference mark in the body text to indicate that additional information is in a footnote or endnote, with a numbering system used for each to tell readers whether to look for the note at the end of the page or the end of the document or section.

- The actual footnote or endnote story content.

2.11.1 continuationSeparator (Continuation Separator Mark)

This element specifies the presence of a continuation separator mark within the current run. A continuation separator mark is a horizontal line which spans the width of the main story's text extents.

[Note: The continuation separator mark is typically used within the context of continuation separator footnotes or endnotes. These footnote and endnote types define the footnote/endnote used to separate the contents of the main document story from continuation of footnotes or endnotes which began on a previous page. *end note*]

2.11.2 endnote (Endnote Content)

This element specifies the content of a single endnote within a WordprocessingML document. Each endnote shall be represented by a single **endnote** element, which may contain any valid *block-level content*.

Attributes	Description
id (Footnote/Endnote ID)	Specifies a unique ID which shall be used to match the contents of a footnote or endnote to the associated footnote/endnote reference mark in the document using the footnoteRef or endnoteRef element, as appropriate. If this attribute is omitted, then this footnote or endnote shall have no ID. If more than one footnote shares the same ID, then this document shall be considered non-conformant. If more than one endnote shares the same ID, then this document shall be considered non-conformant. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
type (Footnote/Endnote Type)	Specifies the type of footnote or endnote contained within the current footnote or endnote content definition. If this attribute is omitted, then it shall be considered to be of type <code>normal</code> . If a footnote or endnote is not of type <code>normal</code> , then it shall not be referenced by a footnoteReference or endnoteReference element within the main document story. The possible values for this attribute are defined by the ST_FtnEdn simple type (§2.18.36).

2.11.3 endnote (Special Endnote List)

This element specifies the ID for all endnotes which are located in the current document that are not of type `normal`. Each other type of endnote shall be referenced in this list, or it shall not be loaded. If an endnote is not listed beneath this element, and it is required by the document content, then the document shall be considered non-conformant.

Attributes	Description
id (Footnote/Endnote ID)	See 2.11.2

2.11.4 endnotePr (Document-Wide Endnote Properties)

This element specifies the endnote properties for the current document. Each of these properties are stored as a child element within the **endnotePr** element.

These properties may be overridden for a specific section via the section-wide **endnotePr** element (§2.11.5).

2.11.5 endnotePr (Section-Wide Endnote Properties)

This element specifies the endnote properties for the current section. Each of these properties are an override of the document-wide endnote properties (§2.11.4) and are stored as a child element within the **endnotePr** element.

If this element is omitted for a given section, then that section shall use the endnote properties defined at the document-wide level.

2.11.6 endnoteRef (Endnote Reference Mark)

This element specifies the presence of an endnote reference mark. An *endnote reference mark* is a run of automatically numbered text which follows the numbering format set forth via the **numFmt** element (§2.11.18).

If an endnote reference mark is specified within a run which is not part of an endnote, then that endnote reference mark may be ignored.

2.11.7 endnoteReference (Endnote Reference)

This element specifies the presence of an endnote reference. An *endnote reference* is a run of automatically numbered text which references a particular endnote within the parent document, and inherits the endnote reference mark's numbering.

If an endnote reference is specified within a footnote or endnote, then the document shall be considered non-conformant.

Attributes	Description
customMarkFollows (Suppress Footnote/Endnote Reference Mark)	<p>Specifies that the current footnote or endnote shall not have an associated footnote or endnote reference mark, as appropriate.</p> <p>This attribute shall be used to specify that a particular footnote or endnote shall not increment the numbering for its associated footnote/endnote numbering format, so that the use of a footnote with a custom footnote mark does not cause a missing value in the footnote/endnote values. The display of the mark is specified via the footnoteRef/endnoteRef elements, as appropriate.</p> <p>If this attribute is omitted, then the footnote or endnote reference mark shall not be skipped when incrementing over this footnote or endnote.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
id (Footnote/Endnote ID Reference)	<p>Specifies the footnote or endnote which is being referenced by the current footnote or endnote reference in the document.</p> <p>If the resulting footnote or endnote ID is not present in the footnotes or endnote part (as appropriate), then this document shall be considered non-conformant.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>

2.11.8 endnotes (Document Endnotes)

This element specifies the set of all endnotes in the document, including endnote separators and continuation notices. This element is the root node for the Endnotes part.

2.11.9 footnote (Special Footnote List)

This element specifies the ID for all footnotes which are located in the current document that are not of type normal. Each other type of footnote shall be referenced in this list, or it shall not be loaded. This means that if a special footnote is not listed beneath this element, and it is required by the document content, then the document shall be considered non-conformant.

Attributes	Description
id (Footnote/Endnote ID)	See 2.11.2

2.11.10 footnote (Footnote Content)

This element specifies the content of a single footnote within a WordprocessingML document. Each footnote shall be represented by a single **footnote** element, which may contain any valid *block-level content*.

Attributes	Description
id (Footnote/Endnote ID)	See 2.11.2
type (Footnote/Endnote Type)	See 2.11.2

2.11.11 footnotePr (Document-Wide Footnote Properties)

This element specifies the footnote properties for this document. Each property is stored as a unique element within the **footnotePr** element.

These properties may be overridden for a specific section via the section-wide **footnotePr** element (§2.11.12).

2.11.12 footnotePr (Section-Wide Footnote Properties)

This element specifies the footnote properties for the current section. Each of these properties are an override of the document-wide footnote properties (§2.11.11) and are stored as a child element within the **footnotePr** element.

If this element is omitted for a given section, then that section shall use the footnote properties defined at the document-wide level.

2.11.13 footnoteRef (Footnote Reference Mark)

This element specifies the presence of a footnote reference mark. A *footnote reference mark* is a run of automatically numbered text which follows the numbering format set forth via the footnote **numFmt** element (§2.11.17).

If a footnote reference mark is specified within a run which is not part of a footnote, then that footnote reference mark may be ignored.

2.11.14 footnoteReference (Footnote Reference)

This element specifies the presence of a footnote reference. A *footnote reference* is a run of automatically numbered text which references a particular footnote within the parent document, and inherits the footnote reference mark's numbering. If an footnote reference is specified within a footnote or endnote, then the document shall be considered non-conformant.

Attributes	Description
See 2.11.7	

2.11.15 footnotes (Document Footnotes)

This element specifies the set of all footnotes in the document, including footnote separators and continuation notices. This element is the root node for the Footnotes part.

2.11.16 noEndnote (Suppress Endnotes In Document)

This element specifies that all endnotes in this document shall not be displayed or printed. If this element is placed on any section break other than the first section break in the document, it shall be ignored.

If this element is omitted, endnotes shall not be suppressed in the current document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.11.17 numFmt (Footnote Numbering Format)

This element specifies the numbering format which shall be used to determine the footnote or endnote reference mark value for all automatically numbered footnote and endnote reference marks (those without the suppressRef attribute set). If this element is omitted, then the numbering format shall be assume to be decimal.

Attributes	Description
val (Numbering Format Type)	See 2.9.18

2.11.18 numFmt (Endnote Numbering Format)

This element specifies the numbering format which shall be used to determine the footnote or endnote reference mark value for all automatically numbered footnote and endnote reference marks (those without the suppressRef attribute set). If this element is omitted, then the numbering format shall be assume to be decimal.

Attributes	Description
val (Numbering Format Type)	See 2.9.18

2.11.19 numRestart (Footnote and Endnote Numbering Restart Location)

This element specifies when all automatic numbering for the footnote or endnote reference marks shall be restarted. When restarted, the next automatically numbered footnote or endnote in the document (each type is handled independently) shall restart to the specified **numStart** value (§2.11.20).

If this element is omitted, then automatic numbering shall not be restarted between each page or section (a vlaue of continuous).

Attributes	Description
val (Automatic Numbering Restart Value)	Specifies when the automatic numbering shall be restarted for the current set of footnotes or endnotes. The possible values for this attribute are defined by the ST_RestartNumber simple type (§2.18.80).

2.11.20 numStart (Footnote and Endnote Numbering Starting Value)

This element specifies the starting number or character for the first automatically numbered footnotes or endnote in the document, as well as the first automatically numbered footnotes after each restart point specified by the **numRestart** element (§2.11.19). This value shall be specified in decimal number units, then translated accordingly to the appropriate numbering format.

If this element is omitted, then the starting value shall be 1.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.11.21 pos (Footnote Placement)

This element specifies where footnotes shall be placed on the page when they are referenced by text in the current document.

If this element is present at the section level, then it shall be ignored.

If this element is omitted at the document level, then footnotes shall be located at the bottom of the current page.

Attributes	Description
val (Footnote Position Type)	Specifies the position of footnotes in the document. The possible values for this attribute are defined by the ST_FtnPos simple type (§2.18.37).

2.11.22 pos (Endnote Placement)

This element specifies where endnotes shall be placed on the page when they are referenced by text in the current document.

If this element is present at the section level, then it shall be ignored.

If this element is omitted at the document level, then endnotes shall be located at the end of the document.

Attributes	Description
val (Endnote Position Type)	Specifies the position of endnotes on the parent section or the document. The possible values for this attribute are defined by the ST_EdnPos simple type (§2.18.25).

2.11.23 separator (Footnote/Endnote Separator Mark)

This element specifies the presence of a separator mark within the current run. A *separator mark* is a horizontal line which spans part of the width text extents.

[Note: The separator mark is typically used within the context of separator footnotes or endnotes. These footnote and endnote types define the footnote/endnote used to separate the contents of the main document story from the contents of footnotes or endnotes on that page. *end note*]

2.12 Glossary Document

Within a WordprocessingML file, the *glossary document* is a supplemental storage location for additional document content which shall travel with the document, but which shall not be displayed for printed as part of the main document until it is explicitly added to that document by deliberate action.

The glossary document shall also be afforded a separate instance of all of the relationships which are provided on the main document part - this means that the glossary document shall have its own style definitions, numbering definitions, comments, headers, footers, etc. within the WordprocessingML document.

Within the glossary document, each distinct region of document content is referred to as a *glossary document entry*, and is defined via the **docPart** element (§2.12.5). These document parts may contain any block-level WordprocessingML element, and may also have a set of classifications and behaviors applied to them via the glossary document entry's properties.

2.12.1 behavior (Entry Insertion Behavior)

This element specifies a single behavior which shall be applied to the contents of the parent glossary document entry (§2.12.5) when it is added to the main document story of a WordprocessingML document. These behaviors shall be used to format the surrounding WordprocessingML around insertion, and do not require the presence of a user interface (i.e. applications without a user interface shall also utilize these settings).

Attributes	Description
val (Insertion Behavior Value)	Specifies the insertion behavior which shall be associated with the current glossary document entry. The possible values for this attribute are defined by the ST_DocPartBehavior simple type (§2.18.18).

2.12.2 behaviors (Entry Insertion Behaviors)

This element specifies the set of behaviors which shall be applied to the contents of the parent glossary document entry (§2.12.5) when it is added to the main document story of a WordprocessingML document. Since multiple behaviors can be

specified for a single part, the sum total of all behaviors shall be used to insert the parent entry into the contents of the WordprocessingML document.

2.12.3 category (Entry Categorization)

This element specifies the categorization for the parent glossary document entry. This categorization shall not imply any behaviors around the entry, and is only used to organize the set of glossary document entries within an application or user interface (i.e. to disambiguate between two entries with the same entry name (§2.12.13)).

2.12.4 description (Description for Entry)

This element specifies a description for the contents of this glossary document entry. This description may contain any string content, and allows the entry to have additional information contained within the definition for this glossary document entry. [Note: This description may be surfaced in a user interface, for example. *end note*]

Attributes	Description
val (String Value)	See 2.3.1.27

2.12.5 docPart (Glossary Document Entry)

This element specifies the details for a single glossary document entry contained in the document. This glossary document entry may consist of one or both of the following:

- The glossary document entry's properties, which define its name, categorization, and behaviors
- The glossary document entry's contents, which consists of one or more block-level elements of WordprocessingML content

Each of these two components is specified by one of the child elements of this element, as seen in the child elements table below.

2.12.6 docPartBody (Contents of Glossary Document Entry)

This element specifies the contents of the parent glossary document entry (§2.12.5). These contents shall consist of one or more block-level elements, analogous to the **body** element (§2.2.2) of the main document story for the current document.

When the contents of a glossary document entry are added to a document, the styles, numbering definitions, and all other related parts for this entry shall be taken from the relationships from the Glossary Document part and not from the main document part. These references shall be moved to their main document equivalents when the entry is added to the document.

When the part is inserted, it shall be inserted as though its last paragraph mark does not exist (the content of the final paragraph mark shall be merged with the contents of the paragraph into which this entry is being added).

2.12.7 docPartPr (Glossary Document Entry Properties)

This element specifies the set of properties which shall be applied to the parent glossary document entry. These properties define its name, categorization, and behaviors.

2.12.8 docParts (List of Glossary Document Entries)

This element specifies the collection of glossary document entries which are stored in the current Glossary Document part.

2.12.9 gallery (Gallery Associated With Entry)

This element specifies the predefined gallery into which the current glossary document part shall be classified. This classification, although its enumeration values may be interpreted to imply semantics around the contents of the parent glossary document entry, shall only be used to classify and sort this entry (via an application or a user interface).

Attributes	Description
------------	-------------

Attributes	Description
val (Gallery Value)	Specifies the classification of gallery which shall be associated with the parent glossary document entry. The possible values for this attribute are defined by the ST_DocPartGallery simple type (§2.18.19).

2.12.10 glossaryDocument (Glossary Document Root Element)

This element specifies the root element for a glossary document part within a WordprocessingML document. A glossary document is an supplementary document story in a WordprocessingML that shall be afforded all of the relationships of the Main Document part, such as:

- Style definitions
- Numbering definitions
- Comments
- Headers/footers
- Etc.

The entries stored in this part shall have all of its implicit relationships target these parts, rather than their analogues stored off of the main document part.

2.12.11 guid (Entry ID)

This element specifies a unique identifier (specified using a 128-bit GUID stored on the val attribute) that uniquely identifies this document building block. [Note: This unique identifier may be used by an application to uniquely reference a single building block regardless of different naming, for example when the same part has different names for localization purposes. *end note*]

Attributes	Description
val (GUID Value)	Specifies a 128-bit globally unique identifier (GUID) value as defined by the simple type referenced below. The contents of this GUID shall be interpreted based on the context of the parent XML element. If this attribute is omitted, its value shall be assumed to be null (i.e. no GUID shall be associated with the parent XML element). The possible values for this attribute are defined by the ST_Guid simple type (§2.18.38).

2.12.12 name (Category Associated With Entry)

This element specifies the category into which the current glossary document part shall be classified. This classification may consist of any string value as determined by its contents, and shall only be used to classify and sort this entry (via an application or a user interface).

Attributes	Description
val (String Value)	See 2.3.1.27

2.12.13 name (Entry Name)

This element specifies a name for the contents of this glossary document entry. This name may contain any string content, and allows the entry to have a friendly identifier contained within the definition for this glossary document entry. [Note: This name may be surfaced in a user interface, for example. *end note*]

Attributes	Description
decorated (Built-In)	Specifies that the name for the current entry is a built-in entry which should not be displayed in

Attributes	Description
Entry)	the user interface. [Note: This information may be used by an application as needed, for example, to disambiguate an entry from one with the same name, ensuring that the built-in entry can be uniquely identified by the application. <i>end note</i>] If this attribute is omitted, its value shall be assumed to be <code>false</code> . The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
val (Name Value)	Specifies a string value which contains the name of the current glossary document entry. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.12.14 style (Associated Paragraph Style Name)

This element specifies the style ID for a paragraph style which shall be associated with the current glossary document entry. This paragraph style associated shall not imply anything about the formatting or content of the glossary document entry, and shall only be used to filter and/or sort this entry (via an application or a user interface). [Note: One example of the level of classification offered by this element is to only show it as available when the formatting of the paragraph matches the specified style. *end note*]

Attributes	Description
val (String Value)	See 2.3.1.27

2.12.15 type (Entry Type)

This element specifies a single type which shall be applied to the properties of the parent glossary document entry (§2.12.5). Each of these types may, based on their values, influence the visibility and behavior of the parent glossary document entry as defined by the associated simple type information.

Attributes	Description
val (Type Value)	Specifies the value for the current entry type. The possible values for this attribute are defined by the ST_DocPartType simple type (§2.18.20).

2.12.16 types (Entry Types)

This element specifies the set of types which shall be applied to the properties of the parent glossary document entry (§2.12.5). Each of these types may, based on their values, influence the visibility and behavior of the parent glossary document entry.

Attributes	Description
all (Entry Is Of All Types)	Specifies that the current glossary document is all types. This attribute shall override any information specified as child elements of this element and shall ensure that the current entry is associated with all available types. If this attribute is omitted, then its default value shall be assumed to be <code>false</code> . The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).

2.13 Annotations

Within a WordprocessingML document, *annotations* refer to various types of supplementary markup which may be stored inside or around a region of text within the document's contents.

Within a document's contents, annotations are stored in one of three different methods:

- Inline
- "Cross Structure"

- Properties

These three forms are needed in order to maintain compatibility with both the legacy annotations functionality of current word processing applications and the requirements of an XML-based format (i.e. wellformedness of the resulting XML markup). These three forms are referenced within the individual annotation types described in the following sub clauses.

2.13.1 Inline Annotations

Inline annotations describe all annotations which do not require special handling in order to maintain the XML wellformedness requirements of the resulting WordprocessingML output. In these cases, a single XML element shall encapsulate the entire contents of the document content which is being annotated.

2.13.2 "Cross Structure" Annotations

"Cross structure" annotations describe the class of annotations which can span portions of WordprocessingML markup . In these cases, the annotation's region is delimited by two elements: a start element and an end element. These two elements mark the start and end points of the annotated content, but do not contain it. The pairing of the start and end marker are linked via a common value for their id attributes

2.13.3 Property Annotations

Property annotations describe the class of annotations which are stored as a property on an object . In these cases, the annotation's semantics are defined by the property, as they can affect content and/or formatting.

2.13.4 Comments

Comments describe annotations which are anchored to a region of document content, but which contain an arbitrary amount of block-level content stored in their own separate document stories. Within a WordprocessingML document, comments are stored in a separate Comments part within the document package.

A comment in a WordprocessingML document is divided into two components:

- The comment anchor (the text on which the comment applies)
- The comment content (the contents of the comment)

The *comment anchor* is the cross structure annotation which defines the region of text on which the comment is anchored. The *comment content* is the content stored in the comments part which contains the actual content of the comment.

2.13.4.1 **annotationRef** (Comment Information Block)

This element specifies the presence of an annotation reference mark at the current location in the comment. An *annotation reference mark* is an information block that represents the metadata about the current comment within the document. This annotation reference mark should typically consist of the initials and a unique integer associated with its position in the document, but may be displayed in any desired format.

If this element is omitted from a single comment's contents, then an annotation reference mark may be added at the start of the comment in reading order (right in a right-to-left paragraph or left in a left-to-right paragraph). As well, an annotation reference mark may be relocated as desired within a comment's content.

2.13.4.2 **comment** (Comment Content)

This element specifies the content of a single comment stored in the Comments part of a WordprocessingML document. If a comment is not referenced by document content via a matching id attribute on a valid use of the **commentReference** element (§2.13.4.5), then it may be ignored when loading the document. If more than one comment shares the same value for the id attribute, then only one comment shall be loaded and the others may be ignored.

Attributes	Description
author (Annotation Author)	Specifies the author for an annotation within a WordprocessingML document. If this attribute is omitted, then no author shall be associated with the parent annotation type. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

Attributes	Description
date (Annotation Date)	Specifies the date information for an annotation within a WordprocessingML document. The use of this information is outside of the scope of this Office Open XML Standard. If this attribute is omitted, then no date information shall be associated with the parent annotation type. The possible values for this attribute are defined by the ST_DateTime simple type (§Error! Reference source not found.).
id (Annotation Identifier)	See 2.13.5.4
initials (Initials of Comment Author)	Specifies the initials of the author of the current comment. This information may be used to format and present the associated comment information block (§2.13.4.1), or in any user interface supported by an application. If this attribute is omitted, then no author shall be associated with the current comment in the document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.13.4.3 **commentRangeEnd** (§2.13.4.3);

This element specifies the end of the range around which a comment is anchored in the content of the WordprocessingML document. The id attribute on this element shall be used to link the corresponding comment anchor range start element and comment reference.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **commentRangeStart** element (§2.13.4.4) with a matching id attribute value, then it shall be considered the single anchor point for the associated comment reference.
- If this element appears without a corresponding **commentReference** element (§2.13.4.5) in the current document story with a matching id attribute value, then it shall be ignored and the comment has no associated range.
- If this element appears in a comment content story (§2.13.4.2), then it may be ignored.

Attributes	Description
displacedByCustomXml (Annotation Marker Relocated For Custom XML Markup)	Specifies that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document. This element only has an effect when the custom XML element is block-level (i.e. surrounds an entire paragraph), as in this scenario the logical and physical placement of the annotation and custom XML element may differ. Specifically, in this case, the custom XML is presented <i>*around*</i> the block-level object it encloses (the paragraph, table, table row, or table cell), but is physically represented within that same object (i.e. within the paragraph, table, table row or table cell). This requirement stems from the fact that there is no location for the location of the annotation within the document at its logical location (around a table, for example). If this element is omitted, then the annotation shall be anchored inside of all block-level custom XML elements in the paragraph. If this element is present, but no block-level custom XML tag is located at the position it specifies (before or after), then it shall be ignored. The possible values for this attribute are defined by the ST_DisplacedByCustomXml simple type (§2.18.16).
id (Annotation Identifier)	See 2.13.5.4

2.13.4.4 **commentRangeStart (§2.13.4.4);**

This element specifies the start of the range around which a comment is anchored in the content of the WordprocessingML document. The id attribute on this element shall be used to link the corresponding comment anchor range end element and comment reference.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **commentRangeEnd** element (§2.13.4.3) with a matching id attribute value, then it shall be considered the single anchor point for the associated comment reference.
- If this element appears without a corresponding **commentReference** element (§2.13.4.5) in the current document story with a matching id attribute value, then it shall be ignored and the comment content has no associated range.
- If this element appears in a comment content story (§2.13.4.2), then it may be ignored.

Attributes	Description
See 2.13.4.3	

2.13.4.5 **commentReference (Comment Content Reference Mark)**

This element specifies the presence of a comment content reference mark, which links the comment content (§2.13.4.2) with the contents of a document story. This link is established by matching the comment whose id attribute matches the id attribute on this element. The resulting comment is anchored to the range with comment range elements with the same id attribute values (if present) as follows:

- If either or both of the **commentRangeStart** and **commentRangeEnd** elements (§2.13.4.4; §2.13.4.3) are present, then the comment reference shall anchor the comment to the resulting range.
- If neither element is present, then the comment reference shall anchor the comment to its current location.

If this element appears in a comment content story (§2.13.4.2), then it may be ignored. If no comment exists with an id attribute which matches the id attribute on this element, then this document is non-conformant.

Attributes	Description
id (Annotation Identifier)	See 2.13.5.4

2.13.4.6 **comments (Comments Collection)**

This element specifies all of the comments defined in the current document. It is the root element of the Comments part of a WordprocessingML document.

2.13.5 **Revisions**

Revisions in WordprocessingML provide a mechanism for storing information about the evolution of the document (i.e. the set of modifications made to a document by one of more authors). When an application adds revisions to the content of a WordprocessingML document, they are specifying this by storing either (depending on the revision type):

- The current state of the document (a deletion stores the current state of the text as deleted, and implies that its original state was the content used to exist)
- The initial state of the document (a run's initial properties are explicitly stored in a previous run properties block, as the current run properties are always those that are the child of the **rPr** element (§2.7.8.1))

A revision consists of two required pieces of information:

- The revision type (specified via the name of the revision element)
- A unique revision identifier (used to uniquely identify revisions)

As well as optional information:

- The author of the revision
- The date and time of the revision

Within a WordprocessingML document, the following types of revisions may be used to track the changes to a document:

- Insertions
- Deletions
- Moves
- Changes to run/paragraph/table/numbering/section properties
- Changes to custom XML markup

2.13.5.1 **cellDel** (Table Cell Deletion)

This element specifies that the parent table cell shall be treated as though it was deleted from the document while revisions were being recorded. This means that although the table cell element exists in the structure of the table, the table cell technically no longer exists in the document.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.2 **cellIns** (Table Cell Insertion)

This element specifies that the parent table cell shall be treated as though it was inserted into the document while revisions were being recorded.

Attributes	Description
See 2.13.5.1	

2.13.5.3 **cellMerge** (Vertically Merged/Split Table Cells)

This element specifies that the vertical merge state of the parent table cell has been modified while revisions were being tracked for the document. The **vmerge** and **vmergeOrig** attributes on this element specify the original and revised vertical merge states of the table cell.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4
vMerge (Revised Vertical Merge Setting)	Specifies the vertical merge setting which was applied to the parent table cell by this revision. If this attribute is omitted, then no revised vertical merge setting is supplied for this revision (if neither this nor the vmergeOrig attribute is specified, the revision may be ignored). The possible values for this attribute are defined by the ST_AnnotationVMerge simple type (§2.18.3).
vMergeOrig (Vertical Merge Setting)	Specifies the vertical merge setting which was removed from the parent table cell by this revision.

Attributes	Description
Removed by Revision)	If this attribute is omitted, then the original vertical merge setting shall be assumed to be rest (not merged). The possible values for this attribute are defined by the ST_AnnotationVMerge simple type (§2.18.3).

2.13.5.4 **customXmlDelRangeEnd** (Custom XML Markup Deletion End)

This element specifies the end of a region in which custom XML markup has been deleted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup deletion start marker in the document.

Providing a physical representation of custom XML markup results in regions which can be inserted and deleted, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this deletion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlDelRangeStart** element (§2.13.5.5) with a matching id attribute value, then it shall be ignored and no deletions shall be present in the document.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If multiple end elements exist with the same id attribute value, then the first instance in the document shall be used and subsequent elements should be treated as unmatched (no corresponding start).

Attributes	Description
id (Annotation Identifier)	See 2.13.5.4

2.13.5.5 **customXmlDelRangeStart** (Custom XML Markup Deletion Start)

This element specifies the beginning of a region in which all custom XML markup has been deleted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup deletion end marker in the document.

Providing a physical representation of custom XML markup results in regions which can be inserted and deleted, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this deletion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlDelRangeEnd** element (§2.13.5.4) with a matching id attribute value, then this revision is ill-formed, and the revision may be ignored or all custom XML from this point forward may be treated as deleted.
- If this element and its paired end encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If multiple start elements exist with the same id attribute value, then the each instance in the document shall be matched with an end in document order, and unmatched starts (no corresponding end) shall be handled as described above.

Attributes	Description
See 2.13.5.1	

2.13.5.6 **customXmlInsRangeEnd** (Custom XML Markup Insertion End)

This element specifies the end of a region within which all custom XML markup has been inserted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup insertion start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this insertion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlInsRangeStart** element (§2.13.5.7) with a matching id attribute value, then it shall be ignored and no insertions shall be present in the document.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If multiple end elements exist with the same id attribute value, then the first instance in the document shall be used and subsequent elements should be treated as unmatched (no corresponding start).

Attributes	Description
id (Annotation Identifier)	See 2.13.5.4

2.13.5.7 **customXmlInsRangeStart** (§2.13.5.7);

This element specifies the beginning of a region in which all custom XML markup has been inserted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup insertion end marker in the document.

Providing a physical representation of custom XML markup start and end tags results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this deletion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlInsRangeEnd** element (§2.13.5.6) with a matching id attribute value, then this revision is ill-formed, and the revision may be ignored or all custom XML from this point forward may be treated as inserted.
- If this element and its paired end encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If multiple start elements exist with the same id attribute value, then the each instance in the document shall be matched with an end in document order, and unmatched starts (no corresponding end) shall be handled as described above.

This element specifies the end of a region within which all custom XML markup has been inserted and tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML markup insertion start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this insertion applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlInsRangeStart** element (§2.13.5.7) with a matching id attribute value, then it shall be ignored and no insertions shall be present in the document.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If multiple end elements exist with the same id attribute value, then the first instance in the document shall be used and subsequent elements should be treated as unmatched (no corresponding start).

Attributes	Description
See 2.13.5.1	

2.13.5.8 **customXmlMoveFromRangeEnd** (Custom XML Markup Move Source End)

This element specifies the end of a region within which all custom XML markup was moved to another location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move source start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move source applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlMoveFromRangeStart** element (§2.13.5.9) with a matching id attribute value, then it shall be ignored and no move source information shall be applied to the custom XML elements by this element.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If this element and its paired start occur outside of a valid move source container (§2.13.5.24; §2.13.5.23) with a matching move destination container (§2.13.5.28; §2.13.5.27), then custom XML markup in this region shall be treated as if it was deleted
- If multiple end elements exist with the same id attribute value, then the first instance in the document shall be used and subsequent elements should be treated as unmatched (no corresponding start).

Attributes	Description
id (Annotation Identifier)	See 2.13.5.4

2.13.5.9 **customXmlMoveFromRangeStart** (Custom XML Markup Move Source Start)

This element specifies the start of a region within which all custom XML markup was moved to another location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move source end marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move source applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlMoveFromRangeStart** element (§2.13.5.9) with a matching id attribute value, then it shall be ignored and no move source information shall be applied to the custom XML elements by this element.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If this element and its paired end occur outside of a valid move source container (§2.13.5.24; §2.13.5.23) with a matching move destination container (§2.13.5.28; §2.13.5.27), then custom XML markup in this region shall be treated as if it was deleted
- If multiple start elements exist with the same id attribute value, then the each instance in the document shall be matched with an end in document order, and unmatched starts (no corresponding end) shall be handled as described above.

Attributes	Description
------------	-------------

Attributes	Description
See 2.13.5.1	

2.13.5.10 **customXmlMoveToRangeEnd** (Custom XML Markup Move Destination Location End)

This element specifies the end of a region within which all custom XML markup was moved to this location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move destination start marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move destination applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlMoveToRangeStart** element (§2.13.5.11) with a matching id attribute value, then it shall be ignored and no move destination information shall be applied to the custom XML elements by this element.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If this element and its paired start occur outside of a valid move source container (§2.13.5.24; §2.13.5.23) with a matching move destination container (§2.13.5.28; §2.13.5.27), then custom XML markup in this region shall be treated as if it was inserted
- If multiple end elements exist with the same id attribute value, then the first instance in the document shall be used and subsequent elements should be treated as unmatched (no corresponding start).

Attributes	Description
id (Annotation Identifier)	See 2.13.5.4

2.13.5.11 **customXmlMoveToRangeStart** (Custom XML Markup Move Destination Location Start)

This element specifies the start of a region within which all custom XML markup was moved to this location in the document and this move was tracked as a revision. The id attribute on this element shall be used to link this element with the corresponding custom XML move destination end marker in the document.

Providing a physical representation of the start and end tags of custom XML markup results in regions which can be inserted and deleted independently, but cannot be encapsulated by a single revision element, since their representation in WordprocessingML is the start or end XML tag for the custom XML markup which it represents. Therefore, the start/end "cross structure" annotation format surrounds the WordprocessingML region to which this move destination applies.

The following restrictions shall be applied to this element:

- If this element occurs without a corresponding **customXmlMoveFromRangeEnd** element (§2.13.5.8) with a matching id attribute value, then it shall be ignored and no move source information shall be applied to the custom XML elements by this element.
- If this element and its paired start encapsulate a range with no custom XML markup, then they shall be ignored and may be omitted when the document is subsequently saved.
- If this element and its paired end occur outside of a valid move source container (§2.13.5.24; §2.13.5.23) with a matching move destination container (§2.13.5.28; §2.13.5.27), then custom XML markup in this region shall be treated as if it was inserted
- If multiple start elements exist with the same id attribute value, then the each instance in the document shall be matched with an end in document order, and unmatched starts (no corresponding end) shall be handled as described above.

Attributes	Description
See 2.13.5.1	

2.13.5.12 **del** (Deleted Run Content)

This element specifies that the inline-level content contained within it shall be treated as deleted content which has been tracked as a revision.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.13 **del** (Deleted Paragraph)

This element specifies that the paragraph mark delimiting the end of a paragraph within a WordprocessingML document shall be treated as deleted (i.e. the contents of this paragraph are no longer delimited by this paragraph mark, and are combined with the following paragraph - but those contents shall not automatically be marked as deleted) as part of a tracked revision.

Attributes	Description
See 2.13.5.1	

2.13.5.14 **del** (Deleted Table Row)

This element specifies that the parent table row shall be treated as a deleted row whose deletion has been tracked as a revision. This setting shall not imply any revision state about the table cells in this row or their contents (which must be revision marked independently), and shall only affect the table row itself.

Attributes	Description
See 2.13.15.1	

2.13.5.15 **del** (Deleted Math Control Character)

This element specifies that the Office Open XML Math control character which contains this element was deleted and tracked as a revision.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.16 **ins** (Inserted Table Row)

This element specifies that the parent table row shall be treated as an inserted row whose insertion has been tracked as a revision. This setting shall not imply any revision state about the table cells in this row or their contents (which must be revision marked independently), and shall only affect the table row itself.

Attributes	Description
See 2.13.5.1	

2.13.5.17 **ins** (Inserted Math Control Character)

This element specifies that the Office Open XML Math control character which contains this element was inserted and tracked as a revision.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.18 **ins** (Inserted Paragraph)

This element specifies that the paragraph mark delimiting the end of a paragraph within a WordprocessingML document shall be treated as deleted (i.e. the contents of this paragraph are no longer delimited by this paragraph mark, and are combined with the following paragraph) as part of a tracked revision.

Attributes	Description
See 2.13.5.1	

2.13.5.19 **ins** (Inserted Numbering Properties)

This element specifies that the numbering information defined by the parent element shall be treated as numbering information which was recorded as an insertion using revisions.

Attributes	Description
See 2.13.5.1	

2.13.5.20 **ins** (Inserted Run Content)

This element specifies that the inline-level content contained within it shall be treated as inserted content which has been tracked as a revision.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.21 **moveFrom** (Move Source Run Content)

This element specifies that the inline-level content contained within it shall be treated as content which has been moved away from this location and tracked as a revision.

The following restrictions shall be applied to this content:

- If this element occurs outside of a valid move source container (§2.13.5.24; §2.13.5.23) for which a matching move destination container (§2.13.5.28; §2.13.5.27) exists in the document, then content in this region shall be treated as deleted, rather than moved.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.22 **moveFrom** (Move Source Paragraph)

This element specifies that the parent paragraph has been moved away from this location and tracked as a revision. This does not imply anything about the revision state of the contents of the paragraph, and applies only to the existence of the paragraph as its own unique paragraph.

The following restrictions shall be applied to this content:

- If this element occurs outside of a valid move source container (§2.13.5.24; §2.13.5.23) for which a matching move destination container (§2.13.5.28; §2.13.5.27) exists in the document, then content in this region shall be treated as deleted, rather than moved.

Attributes	Description
See 2.13.5.1	

2.13.5.23 **moveFromRangeEnd** (Move Source Location Container - End)

This element specifies the end of a region whose move source contents are part of a single named move. When a move source is stored as a revision in a WordprocessingML document, two pieces of information must be stored about that move source:

- A set of pieces of content which were moved - both inline-level content (§2.13.5.21) and paragraphs (§2.13.5.22)
- A move source container (or "bookmark") which specifies that all content within it which marked as a move source is part of a single named move. The name attribute on the move container links a group of move source content with the corresponding group of move destination content.

This element defines the end of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding start of a move source container in the document.

The following restrictions are applied to the use of this element:

- If this element occurs without a corresponding **moveFromRangeStart** element (§2.13.5.24) with a matching id attribute value, then it shall be ignored and no move source container exists
- If this element and its paired end occur without a matching move destination container (§2.13.5.28; §2.13.5.27), then moved content in this region shall be treated as if it was deleted
- If multiple move source containers surround the same text, the last valid container (determined by the location of the container start elements, in document order) should be the container associated with that text.

Attributes	Description
See 2.13.4.3	

2.13.5.24 **moveFromRangeStart** (Move Source Location Container - Start)

This element specifies the start of a region whose move source contents are part of a single named move. When a move source is stored as a revision in a WordprocessingML document, two pieces of information must be stored about that move source:

- A set of pieces of content which were moved - both inline-level content (§2.13.5.21) and paragraphs (§2.13.5.22)
- A move source container (or "bookmark") which specifies that all content within it which marked as a move source is part of a single named move. The name attribute on the move container links a group of move source content with the corresponding group of move destination content.

This element defines the start of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding end of a move source container in the document.

The following restrictions are applied to the use of this element

- If this element occurs without a corresponding **moveFromRangeEnd** element (§2.13.5.23) with a matching id attribute value, then it shall be ignored and no move source container exists
- If this element and its paired end occur without a matching move destination container (§2.13.5.28; §2.13.5.27), then moved content in this region shall be treated as if it was deleted
- If multiple start elements exist with the same id attribute value, then the each instance in the document shall be matched with an end in document order, and unmatched starts (no corresponding end) shall be handled as described above.
- If multiple move source containers surround the same text, the last valid container (determined by the location of the container start elements, in document order) should be the container associated with that text.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
colFirst (First Table Column Covered By Bookmark)	<p>Specifies the zero-based index of the first column in this row which shall be part of this bookmark.</p> <p>When a bookmark is contained within a table, it is possible for that bookmark to only cover cells within a certain column and row range within that table, by specifying:</p> <ul style="list-style-type: none"> • The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row. • The first column included in the bookmark for each of the specified row(s) via this attribute. • The last column included in the bookmark for each of the specified row(s) via the colLast attribute. • The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row. <p>If this attribute appears, then the colLast attribute must also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value exceeds the value of colLast or the number of columns in the table, then both values should be ignored.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
colLast (Last Table Column Covered By Bookmark)	<p>Specifies the zero-based index of the last column in this row which shall be part of this bookmark.</p> <p>When a bookmark is contained within a table, it is possible for that bookmark to only cover cells within a certain column and row range within that table, by specifying:</p> <ul style="list-style-type: none"> • The first row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkStart element in the first table cell in that row.

Attributes	Description
	<ul style="list-style-type: none"> The first column included in the bookmark for each of the specified row(s) via the colFirst attribute. The last column included in the bookmark for each of the specified row(s) via this attribute. The last row for which the specified columns are part of the table bookmark. This is accomplished by placing the bookmarkEnd element at the end of that table row. <p>If this attribute appears, then the colFirst attribute must also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value does not equal or exceed the value of colFirst or the number of columns in the table, then both values should be ignored.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
date (Annotation Date)	See 2.13.4.2
displacedByCustomXML (Annotation Marker Relocated For Custom XML Markup)	See 2.13.4.3
id (Annotation Identifier)	See 2.13.5.4
name (Bookmark Name)	<p>Specifies the bookmark name.</p> <p>If multiple bookmarks in a document share the same name, then the first bookmark (defined by the location of the bookmarkStart element in document order) shall be maintained, and all subsequent bookmarks should be ignored.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>

2.13.5.25 **moveTo** (Move Destination Paragraph)

This element specifies that the parent paragraph has been moved to this location and tracked as a revision. This does not imply anything about the revision state of the contents of the paragraph, and applies only to the existence of the paragraph as its own unique paragraph.

The following restrictions shall be applied to this content:

- If this element occurs outside of a valid move destination container (§2.13.5.28; §2.13.5.27) for which a matching move source container (§2.13.5.24; §2.13.5.23) exists in the document, then content in this region shall be treated as inserted, rather than moved.

Attributes	Description
See 2.13.5.1	

2.13.5.26 **moveTo** (Move Destination Run Content)

This element specifies that the inline-level content contained within it shall be treated as content which has been moved to this location and tracked as a revision.

The following restrictions shall be applied to this content:

- If this element occurs outside of a valid move destination container (§2.13.5.28; §2.13.5.27) for which a matching move source container (§2.13.5.24; §2.13.5.23) exists in the document, then content in this region shall be treated as inserted, rather than moved.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.27 **moveToRangeEnd** (Move Destination Location Container - End)

This element specifies the end of a region whose move destination contents are part of a single named move. When a move source is stored as a revision in a WordprocessingML document, two pieces of information must be stored about that move destination:

- A set of pieces of content which were moved - both inline-level content (§2.13.5.26) and paragraphs (§2.13.5.25)
- A move destination container (or "bookmark") which specifies that all content within it which marked as a move destination is part of a single named move. The name attribute on the move container links a group of move destination content with the corresponding group of move source content.

This element defines the end of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding start of a move destination container in the document.

The following restrictions are applied to the use of this element:

- If this element occurs without a corresponding **moveToRangeStart** element (§2.13.5.28) with a matching id attribute value, then it shall be ignored and no move source container exists
- If this element and its paired end occur without a matching move source container (§2.13.5.24; §2.13.5.23), then moved content in this region shall be treated as if it was inserted
- If multiple move destination containers surround the same text, the last valid container (determined by the location of the container start elements, in document order) should be the container associated with that text.

Attributes	Description
See 2.13.4.3	

2.13.5.28 **moveToRangeStart** (Move Destination Location Container - Start)

This element specifies the start of the region whose move destination contents are part of a single named move. When a move destination is stored as a revision in a WordprocessingML document, two pieces of information must be stored about that move destination:

- A set of pieces of content which were moved - both inline-level content (§2.13.5.26) and paragraphs (§2.13.5.25)
- A move destination container (or "bookmark") which specifies that all content within it which marked as a move destination is part of a single named move. The name attribute on the move container links a group of move destination content with the corresponding group of move source content.

This element defines the start of the latter piece of the move revision data - the container. The id attribute on this element shall be used to link this element with the corresponding end of a move destination container in the document.

The following restrictions are applied to the use of this element

- If this element occurs without a corresponding **moveToRangeEnd** element (§2.13.5.27) with a matching id attribute value, then it shall be ignored and no move source container exists
- If this element and its paired end occur without a matching move source container (§2.13.5.24; §2.13.5.23), then moved content in this region shall be treated as if it was inserted
- If multiple start elements exist with the same id attribute value, then the each instance in the document shall be matched with an end in document order, and unmatched starts (no corresponding end) shall be handled as described above.

- If multiple move destination containers surround the same text, the last valid container (determined by the location of the container start elements, in document order) should be the container associated with that text.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
colFirst (First Table Column Covered By Bookmark)	See 2.13.5.24
colLast (Last Table Column Covered By Bookmark)	See 2.13.5.24
date (Annotation Date)	See 2.13.4.2
displacedByCustomXML (Annotation Marker Relocated For Custom XML Markup)	See 2.13.4.3
id (Annotation Identifier)	See 2.13.5.4
name (Bookmark Name)	See 2.13.5.24

2.13.5.29 **numberingChange** (Previous Numbering Field Properties)

This element specifies the previous state of the numbering displayed by a LISTNUM field (§2.16.5.40) within a WordprocessingML document when additional LISTNUM fields are added and revisions are being tracked.

[*Rationale:* The legacy numbering mechanism provided by the LISTNUM field relies on the presence of fields in the run content of the document, rather than being a paragraph property (as numbering typically is represented). For this reason, these fields must store their previous state as a unique revision type on the field character of the numbering field. *end rationale*]

If this element is supplied for a field which is not of type LISTNUM as defined by its field codes (§2.16.5), then this property shall be ignored.

For numbering fields, the original attribute shall specify the previous numbering displayed by the parent LISTNUM field within a WordprocessingML document. This information is a performance-enhancing cache of the state of the numbering before the revision to allow applications to show the previous state without having to recalculate all of the LISTNUM fields in the document.

If this attribute is omitted, then no previous numbering value is implied and applications may choose to calculate this value, or display no previous numbering value.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

Attributes	Description
original (Previous Numbering Value)	<p>Specifies the previous numbering displayed by the parent numbering change revision. Its format is specified by the parent element.</p> <p>If this attribute is omitted, then no previous numbering value is implied and applications may choose to calculate this value, or display no previous numbering value.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>

2.13.5.30 numberingChange (Previous Paragraph Numbering Properties)

This element specifies the previous state of the numbering on a paragraph when revisions are being tracked.

[*Rationale*: This mechanism is simply used to provide storage for revisions to numbering produced by legacy word processing applications, and applications are encouraged to use the **pPrChange** element to store these changes as changes to the paragraph properties instead. *end rationale*]

For paragraph numbering, the original attribute shall specify the previous numbering definition for an individual paragraph of text within a WordprocessingML document while revisions are being tracked.

The value of original is represented as separate numbering level definitions defined as follows:

<[%[numbering level]:[nfc value]:[numbering format]:[separator]>[repeat if more than one level]

where

- **numbering level** – The level for which the numbering definition is defined
- **nfc value** – The value of the numbering style at the specific numbering level
- **numbering format** – The nfc value of the numbering format, as referenced in the table below.
- **separator** – The separator used to separate the numbering level definitions

The numbering format values are mapped as follows:

nfc Value	ST_NumberFormat enumeration equivalent
0	decimal
1	upperRoman
2	lowerRoman
3	upperLetter
4	lowerLetter
5	ordinal
6	cardinalText
7	ordinalText
8	hex
9	chicago
10	ideographDigital
11	japaneseCounting
12	Aiueo
13	Iroha
14	decimalFullWidth
15	decimalHalfWidth
16	japaneseLegal
17	japaneseDigitalTenThousand

nfc Value	ST_NumberFormat enumeration equivalent
18	decimalEnclosedCircle
19	decimalFullWidth2
20	aiueoFullWidth
21	irohaFullWidth
22	decimalZero
23	bullet
24	ganada
25	chosung
26	decimalEnclosedFullstop
27	decimalEnclosedParen
28	decimalEnclosedCircleChinese
29	ideographEnclosedCircle
30	ideographTraditional
31	ideographZodiac
32	ideographZodiacTraditional
33	taiwaneseCounting
34	ideographLegalTraditional
35	taiwaneseCountingThousand
36	taiwaneseDigital
37	chineseCounting
38	chineseLegalSimplified
39	chineseCountingThousand
40	Application-defined. May be ignored.
41	koreanDigital
42	koreanCounting
43	koreanLegal
44	koreanDigital2
45	hebrew1
46	arabicAlpha
47	hebrew2
48	arabicAbjad
49	hindiVowels
50	hindiConsonants
51	hindiNumbers
52	hindiCounting
53	thaiLetters
54	thaiNumbers
55	thaiCounting

nfc Value	ST_NumberFormat enumeration equivalent
56	vietnameseCounting
57	numberInDash
58	russianLower
59	russianUpper
60 or above	Application-defined. May be ignored.

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4
original (Previous Numbering Value)	See 2.13.5.29

2.13.5.31 **pPrChange** (Revision Information for Paragraph Properties)

This element specifies the details about a single revision to a set of paragraph properties in a WordprocessingML document. This element stores this revision as follows:

- The child element of this element contains the complete set of paragraph properties which were applied to this paragraph before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of paragraph properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.32 **rPrChange** (Revision Information for Run Properties)

This element specifies the details about a single revision to a set of run properties in a WordprocessingML document. This element stores this revision as follows:

- The child element of this element contains the complete set of run properties which were applied to this run before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of run properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2

Attributes	Description
Date)	
id (Annotation Identifier)	See 2.13.5.4

2.13.5.33 **rPrChange** (Revision Information for Run Properties on the Paragraph Mark)

This element specifies the details about a single revision to a set of run properties applied to a paragraph mark within a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of run properties which were applied to this paragraph mark before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of run properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.34 **sectPrChange** (Revision Information for Section Properties)

This element specifies the details about a single revision to a set of section properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of section properties which were applied to the parent section before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of section properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.35 **tblGridChange** (Revision Information for Table Grid Column Definitions)

This element specifies the details about a single revision to a table's grid column definitions within a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the definition of the table grid which was applied to the parent table before this revision

Attributes	Description
id (Annotation Identifier)	See 2.13.5.4

Attributes	Description
Identifier)	

2.13.5.36 **tblPrChange** (Revision Information for Table Properties)

This element specifies the details about a single revision to a set of table properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of table properties which were applied to the parent table before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.37 **tblPrExChange** (Revision Information for Table-Level Property Exceptions)

This element specifies the details about a single revision to a set of table-level property exceptions in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of table-level property exceptions which were applied to the parent table row before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table-level exception properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.38 **tcPrChange** (Revision Information for Table Cell Properties)

This element specifies the details about a single revision to a set of table cell properties in a WordprocessingML document.

This element stores this revision as follows:

- The child element of this element contains the complete set of table cell properties which were applied to the parent table before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table cell properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2

Attributes	Description
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.5.39 **trPrChange** (Revision Information for Table Row Properties)

This element specifies the details about a single revision to a set of table row properties in a WordprocessingML document. This element stores this revision as follows:

- The child element of this element contains the complete set of table row properties which were applied to the parent table row before this revision
- The attributes of this element contain information about when this revision took place (i.e. when these properties became a 'former' set of table row properties).

Attributes	Description
author (Annotation Author)	See 2.13.4.2
date (Annotation Date)	See 2.13.4.2
id (Annotation Identifier)	See 2.13.5.4

2.13.6 Bookmarks

Within a WordprocessingML document, *bookmarks* refer to arbitrary regions of content which are bounded and have a unique name associated with them.

Because bookmarks are a legacy word processing function which predates the concepts of XML and well-formedness, they can start and end at any location within a document's contents and therefore must use the "cross-structure" annotation format described in §2.13.2.

2.13.6.1 **bookmarkEnd** (Bookmark End)

This element specifies the end of a bookmark within a WordprocessingML document. This end marker is matched with the appropriately paired start marker by matching the value of the id attribute from the associated **bookmarkStart** element. If no **bookmarkStart** element exists prior to this element in document order with a matching id attribute value, then this element is ignored and no bookmark is present in the document with this name.

Attributes	Description
See 2.13.4.3	

2.13.6.2 **bookmarkStart** (Bookmark Start)

This element specifies the start of a bookmark within a WordprocessingML document. This start marker is matched with the appropriately paired end marker by matching the value of the id attribute from the associated **bookmarkEnd** element. If no **bookmarkEnd** element exists subsequent to this element in document order with a matching id attribute value, then this element is ignored and no bookmark is present in the document with this name.

If a bookmark begins and ends within a single table, it is possible for that bookmark to cover discontinuous parts of that table which are logically related (e.g. a single column in a table). This type of placement for a bookmark is accomplished (and described in detail) on the colFirst and colLast attributes on this element.

Attributes	Description
------------	-------------

Attributes	Description
colFirst (First Table Column Covered By Bookmark)	See 2.13.5.24
colLast (Last Table Column Covered By Bookmark)	See 2.13.5.24
displacedByCustomXML (Annotation Marker Relocated For Custom XML Markup)	See 2.13.4.3
id (Annotation Identifier)	See 2.13.5.4
name (Bookmark Name)	See 2.13.5.24

2.13.7 Range Permissions

Range permissions in a WordprocessingML document refer to a special type of bookmark used to control which subset(s) of users may edit a particular region of a document. Range permissions specify the user or set of users which are allowed to edit all content between them whenever the document protection specified by the **documentProtection** element (§2.15.1.28) is enabled and set to **readOnly** or **comments**.

Like bookmarks, range permissions are a legacy word processing function which predates the concepts of XML and well-formedness, so they can start and end at any location within a document's contents and therefore must use the "cross-structure" annotation format described in §2.13.2.

2.13.7.1 **permEnd** (Range Permission End)

This element specifies the end of a single range permission within a WordprocessingML document. This end marker is matched with the appropriately paired start marker by matching the value of the **id** attribute from the associated **permStart** element.

If no **permStart** element exists prior to this element in document order with a matching **id** attribute value, then this element is ignored and no range permission is present in the document.

Attributes	Description
displacedByCustomXML (Annotation Displaced By Custom XML Markup)	See 2.13.4.3
id (Annotation ID)	See 2.13.5.4

2.13.7.2 **permStart** (Range Permission Start)

This element specifies the start of a range permission within a WordprocessingML document. This start marker is matched with the appropriately paired end marker by matching the value of the **id** attribute from the associated **permEnd** element. If no **permEnd** element exists subsequent to this element in document order with a matching **id** attribute value, then this element is ignored and no range permission is present in the document.

If a range permission begins and ends within a single table, it is possible for that permission to cover discontinuous parts of that table which are logically related (e.g. a single column in a table). This type of placement for a range permission is accomplished (and described in detail) on the **colFirst** and **colLast** attributes on this element.

Attributes	Description
colFirst (First Table Column Covered By Range Permission)	<p>Specifies the zero-based index of the first column in this row which shall be part of this range permission.</p> <p>When a range permission is contained within a table, it is possible for that range permission to only cover cells within a certain column and row range within that table, by specifying:</p> <ul style="list-style-type: none"> • The first row for which the specified columns are part of the table range permission. This is accomplished by placing the permStart element in the first table cell in that row. • The first column included in the range permission for each of the specified row(s) via this attribute. • The last column included in the range permission for each of the specified row(s) via the colLast attribute. • The last row for which the specified columns are part of the table range permission. This is accomplished by placing the permEnd element at the end of that table row. <p>If this attribute appears, then the colLast attribute must also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a range permission which is not contained in a table, then their values should be ignored. If this value exceeds the value of colLast or the number of columns in the table, then both values should be ignored.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
colLast (Last Table Column Covered By Range Permission)	<p>Specifies the zero-based index of the last column in this row which shall be part of this range permission.</p> <p>When a range permission is contained within a table, it is possible for that range permission to only cover cells within a certain column and row range within that table, by specifying:</p> <ul style="list-style-type: none"> • The first row for which the specified columns are part of the table range permission. This is accomplished by placing the permStart element in the first table cell in that row. • The first column included in the range permission for each of the specified row(s) via the colFirst attribute. • The last column included in the range permission for each of the specified row(s) via this attribute. • The last row for which the specified columns are part of the table range permission. This is accomplished by placing the permEnd element at the end of that table row. <p>If this attribute appears, then the colFirst attribute must also appear (regardless of where this bookmark is located) or the document shall be considered non-conformant. If this attribute and its pair occur on a bookmark which is not contained in a table, then their values should be ignored. If this value does not equal or exceed the value of colFirst or the number of columns in the table, then both values should be ignored.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
displacedByCustomXML (Annotation Displaced By Custom XML Markup)	See 2.13.4.3
ed (Single User For Range Permission)	Specifies a single user for which this range permission shall be enabled (i.e. a user which shall be able to edit this range when document protection is enabled).

Attributes	Description
	<p>This editor can be stored in one of the following forms:</p> <ul style="list-style-type: none"> DOMAIN\username - for users whose access shall be authenticated using the current user's domain credentials user@domain.com - for users whose access shall be authenticated using the user's e-mail address as credentials <p>user - for users whose access shall be authenticated using the current user's machine credentials</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>
edGrp (Editor Group For Range Permission)	<p>Specifies an alias (or editing group) which shall be used to determine if the current user shall be allowed to edit this range of the document. This mechanism simply provides a set of predefined editing groups which may be associated with user accounts by applications in any desired manner. The possible values for this attribute are defined by the ST_EdGrp simple type (§2.18.24).</p>
id (Annotation ID)	See 2.13.4.2

2.13.8 Spelling & Grammar

The final type of annotation stored in a WordprocessingML document, *spelling and grammar errors* are annotations used to specify the locations of existing spelling and grammatical errors within the contents of a document.

[*Rationale*: When a WordprocessingML document is saved, applications may choose to save currently flagged spelling and grammar errors, for two reasons:

- In order to increase the performance subsequent loads of the document (as those load operations can rely on the persisted proofing state of the document)
- In order to store words which shall not be marked as proofing errors regardless of how they would normally be flagged by the proofing tools engine (i.e. to store spelling and grammar exceptions).

end rationale]

2.13.8.1 proofErr (Proofing Error Anchor)

This element specifies the presence of a start or end anchor for a single proofing error within a WordprocessingML document.

When proofing errors are stored in a document, their semantics shall be interpreted as follows:

- Each proofing error with a type attribute value of **spellStart** shall be linked with the next error with a type attribute of **spellEnd**. If one does not exist, then this error should be ignored.
- Each proofing error with a type attribute value of **spellEnd** which was not preceded by an error with a type attribute value of **spellStart** (that was not previously matched to an end) should be ignored.
- Each proofing error with a type attribute value of **gramStart** shall be linked with the next error with a type attribute of **gramEnd**. If one does not exist, then this error should be ignored.
- Each proofing error with a type attribute value of **gramEnd** which was not preceded by an error with a type attribute value of **gramStart** (that was not previously matched to an end) should be ignored.

Attributes	Description
type (Proofing Error Anchor Type)	<p>Specifies the type of proofing error anchor at this location in the document. This type implies the necessary semantics for this element as defined by the parent element.</p> <p>The possible values for this attribute are defined by the ST_ProofErr simple type (§2.18.76).</p>

2.14 Mail Merge

Mail merge refers to an operation by which WordprocessingML documents may work in conjunction with data from an external data source, importing this data into a document according to a set of codes contained in WordprocessingML known as fields.

A WordprocessingML document that contains the **mailMerge** element (§2.14.20) and is therefore connected to an external data source, is known as a *source document*. In addition to being connected to an external data source and containing fields, a source document may contain any regular WordprocessingML constructs such as:

- Text runs
- Paragraphs
- Images
- Tables
- Numbering
- Etc.

There are two key parts of the mail merge data stored in a WordprocessingML document

10. Information connecting a document to an external data source
11. Information populating fields within that document with external data.

Once the fields in a merged document have been populated with external data, mail merge has been completed and the resulting files are known as *mail merged documents* or simply *merged documents*.

The mail merge settings for a WordprocessingML document are stored in two locations:

- The standard mail merge settings are stored as the child elements of the **mailMerge** element (§2.14.20)
- A set of additional mail merge settings stored in the **odso** element (§2.14.25), and collectively referred to as the Office Data Source Object settings. The *Office Data Source Object* is an extension to the standard settings stored with a mail merge which performs two functions: First, it provides additional information about the mail merge data source, specifically: information about how to map the columns in the data source to MERGEFIELD fields and information about records which shall be included and excluded when creating merged documents. Second, it provides an alternate set of connection information which should be used when the **dataType** element (§2.14.10) specifies a value of **native**. This alternate connection string provides additional connection information for applications which choose to support the ODSO connection string syntax.

2.14.1 active (Record Is Included in Mail Merge)

This element specifies whether a specific record from the specified external data source shall be imported into a merged WordprocessingML document when the mail merge defined for a source document is performed. If this element's **val** attribute is **false**, then the record specified by the parent element shall not used to create a merged document. If this element is omitted for a given record, the data record associated with it shall be imported into a merged WordprocessingML document when the mail merge is performed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.14.2 activeRecord (Record Currently Displayed In Merged Document)

This element specifies that the hosting application shall display the given record from the specified external data source in place of the MERGEFIELD fields (§2.16.5.42) its data is mapped to via the **fieldMapData** element (§2.14.15) in a merged document. When this element is present, the **val** attribute shall specify the one-based index of the record from that data source which shall be used to populate this document.

If the **activeRecord** element is omitted with the **viewMergedData** element's **val** attribute equal to **true**, the hosting application shall behave as if the **activeRecord** element's **val** attribute was equal to 1. If the **viewMergedData** element (§2.14.36) is omitted or present with a **val** attribute equal to **Off**, **0**, or **false**, then this element shall be ignored. If the **activeRecord** record is given a **val** attribute that is less than one or greater than the number of records in the specified external data source, the hosting application shall treat this **val** attribute as if it were equal to 1.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.14.3 addressFieldName (Column Containing E-mail Address)

This element specifies the column within a given external data source that contains e-mail addresses. This element is specified independently of the field mappings specified for a given merged document via the **fieldMapData** element (§2.14.15).

If this element is omitted, or no column exists in the data source with this column name, then the source document specifies that no e-mail address data shall be associated with this mail merge.

[*Note:* This element is generally used to allow the e-mailing of merged documents resulting from populating the fields within a merged document with external data.

This element is independent of the field mapping specified for a given merged document via the **fieldMapData** element (§2.14.15). This separation enables applications to email the documents resulting from the population of WordprocessingML fields with external data regardless of the presence or absence of a field mapped to external data specifying email addresses. *end note*]

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.4 checkErrors (Mail Merge Error Reporting Setting)

This element specifies the type of error reporting which shall be conducted by an application when performing a mail merge against the specified source data.

The type of error reporting implied by this element shall be defined as follows:

- Simulate the population of fields with mapped external data and report errors in a new document if the **val** attribute is equal to 1.
- While populating fields with mapped external data, pausing to report each error as it occurs if the **val** attribute is equal to 2.
- Populate fields with mapped external data and report errors in a new document if the **val** attribute is equal to 3.
- Application-defined behaviors may be used if the **val** attribute is equal to any other value.

If this element is omitted, or its value is set to a value outside of those specified below that is not understood by the hosting application, then its value shall be assumed to be 2.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.14.5 colDelim (Column Delimiter for Data Source)

This element specifies the character which shall be interpreted as the column delimiter used to separate columns within external data sources. The character representing the specific delimiter used for the external data source referenced by a source or merged WordprocessingML document is specified via a decimal number representing the decimal number for the Unicode character representation within this element's **val** attribute.

If this element is omitted, then no column delimiter shall be specified for the data source in this mail merge.

[*Example:* Consider the following WordprocessingML fragment:

```
<w:colDelim w:val="44" />
```

Here, the **colDelim** element's **val** attribute specifies that the given external data source is using the comma character (,) to delimit column data, as 44 is the decimal value for the Unicode character representation of a comma. *end Example*]

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.14.6 column (Index of Column Containing Unique Values for Record)

This element specifies the column within the specified external data source that contains unique data for the current record within that data source. This element shall be used in conjunction with the **uniqueTag** element (§2.14.35) to maintain a relationship between a specific record within an external data source and a given source or merged document. The **val** attribute on this element shall be interpreted as a zero-based index into the columns specified by the data source, specifying the resulting column as the column in which the **uniqueTag** element shall be looked up.

If this element specifies a column number which exceeds the number of columns in the specified external data source, then its value shall be ignored.

[*Note:* This information is necessary as part of a mail merge as records may be added or deleted from external data sources, and a means must be provided to maintain record-specific inclusion or exclusion data using the **active** element (§2.14.1) and the affected external data record when the WordprocessingML document is reconnected to the external data source irrespective of the ordering of the records within the external data source. *end note*]

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.14.7 column (Index of Column Being Mapped)

This element specifies the zero-based index of the column within a given external data source which shall be mapped to the local name of a specific MERGEFIELD field (§2.16.5.42) specified by the parent field mapping data. The **val** attribute specifies this index value, which is used to look up the appropriate column in the data source.

If this element is omitted, or its value exceeds the number of columns in the associated data source, then the index of the referenced column shall be assumed to be 0.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.14.8 connectionString (Data Source Connection String)

This element specifies the connection string used to reconnect to an external data source. The string within this element's **val** attribute shall contain the connection string that the hosting application shall pass to a external data source access application to enable the WordprocessingML document to be reconnected to the specified external data source.

[*Note:* This string is generally comprised of a series of name/value pairs, delimited by semicolons, determined by the data source access application and the external data source that is accessed. *end note*]

If this string is omitted, then no legacy connection string shall be associated with this mail merge.

This connection string should be ignored under the following conditions:

- The **udl** element (§2.14.34) is present within the mail merge data
- The **dataType** element (§2.14.10) is set to **native**
- The current application is able to use the information contained in the **odso** element (§2.14.25) to access the data source

[*Guidance:* In this case, using the connection string in the **udl** element will provide an equal or greater amount of connection information for the mail merge data source for clients which support it. *end guidance*]

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.9 dataSource (Data Source File Path)

This element specifies the relationship whose target is the location of the external data source to be connected to a given WordprocessingML document to perform the mail merge (for a source document) or to find the associated field data (for a merged document).

If this element is omitted, then no file location is specified for the data source for the current mail merge. If no relationship exists with the given relationship ID, or this relationship is not of type

<http://schemas.openxmlformat.org/officeDocument/2006/relationships/mailMergeSource> then this document shall be considered non-conformant.

The data source location may also be ignored under the following conditions:

- The **src** element (§2.14.30) is present within the mail merge data
- The **dataType** element (§2.14.10) is set to **native**
- The current application is able to use the information contained in the **odso** element (§2.14.25) to access the data source

[Guidance: In this case, using the data source file path in the **src** element will provide an equal or greater amount of information for the mail merge data source for clients which can consume it. *end guidance*]

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.14.10 dataType (Data Source Type)

This element specifies the type of external data source to be connected to via the Dynamic Data Exchange (DDE) system (such as a spreadsheet or database), or the alternative method of data access if the Dynamic Data Exchange system is not used. This setting is purely a suggestion of the data source access mechanism which shall be used, and may be ignored in favor of an alternative mechanism if one is present.

Attributes	Description
val (Value)	Specifies the exact type of external data source to which a given merged WordprocessingML document is to be connected. The possible values for this attribute are defined by the ST_MailMergeDataType simple type (§2.18.58).

2.14.11 destination (Merged Document Destination)

This element specifies what the result which shall be generated when a mail merge is carried out on a given WordprocessingML source document. In other words, this element is used to specify what is to be done with the merged documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.

If this element is omitted, then the default destination of merged documents shall be assumed to be of type **newDocument**.

[Note: The aspects of the mail merge outside of connecting to an external data source and populating the fields within a given merged document with external data from the specified external data source are not specified by this Office Open XML Standard.

For example, if a given merged WordprocessingML document contains a **destination** element with its **val** attribute equal to **email**, the hosting application may surface a user interface specific to creating emails with the data resulting from populating fields within a given merged WordprocessingML document with external data from the specified external data

source. WordprocessingML only provides a flag (via the **destination** element) to tell the hosting application to surface this user interface. *end note*]

Attributes	Description
val (Mail Merge Merged Document Type)	Specifies the type of merged documents which shall be the result of carrying out a mail merge on a given source WordprocessingML document. The possible values for this attribute are defined by the ST_MailMergeDest simple type (§2.18.59).

2.14.12 doNotSuppressBlankLines (Remove Blank Lines from Merged Documents)

This element specifies how an application performing the mail merge shall handle blank lines in the merged documents resulting from the mail merge. Typically, when a mail merge is performed, any blank lines which result from lines whose sole contents are merge fields with no content are removed from the merged document in order to prevent extraneous blank lines from appearing in the merged documents. When this element is present, the merged documents which are generated from the mail merge shall not have any blank lines removed before they are sent to their destination format. If this element is omitted, the merged documents generated from this mail merge shall have all blank lines suppressed if they consist of only merge fields with values consisting of empty strings.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.14.13 dynamicAddress (Use Country-Based Address Field Ordering)

This element specifies that the contents of the AddressBlock MERGEFIELD field shall be dynamically ordered based on the country associated with the current record or if the country-invariant version of the address field shall be used in its place. [*Rationale*: When a source document is combined with the contents of a data source in order to produce multiple merged documents, it is often necessary to use an address form specific to the destination country for each particular record in the data source, rather than one static address form for all records. *end rationale*] If this element is set to true, then the mail merge shall use an address form suited to the country associated with the current record in the external data source.

If this element is omitted, then the form of the address shall be dynamically determined based on the country specified in the current record.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.14.14 fHdr (First Row of Data Source Contains Column Names)

This element specifies that a hosting application shall treat the first row of data in the specified external data source as a header row containing the names of each column in the data source, rather than data to populate mapped fields in a merged document. When present, this information shall not change the indices specified in the **recipientData** elements (§2.14.28), but shall indicate that the first row is not part of the mail merge when it is performed.

If this element is omitted, then the first row of the data source shall not be considered a header row when a mail merge is performed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.14.15 fieldMapData (External Data Source to Merge Field Mapping)

This element specifies how a column specified in the external data source that has been connected to a WordprocessingML document shall be mapped to the pre-defined MERGEFIELD fields (§2.16.5.42) within the given merged document's contents. Each instance of a **fieldMapData** element contains the information needed to map one column in the external data source to a single type of pre-defined MERGEFIELD field for the purposes of the mail merge in the current document.

2.14.16 headerSource (Header Definition File Path)

This element specifies the location of a file that contains the column header information which shall be used when connecting to an external data source that does not have column header data specified. Specifically, this element specifies a file that corresponds with the aforementioned external data source. [Note: Column headers are needed to enable a hosting application to associate external data source's columns to fields via the **fieldMapData** element (§2.14.15).

If this element is omitted, then the column header definition data is not specified in an external file and shall be retrieved from the primary data source associated with the mail merge.

Attributes	Description
id (Relationship to Part)	See 2.3.3.17
Namespace: .../officeDocument/2006/relationships	

2.14.17 lid (Merge Field Name Language ID)

This element specifies the language ID for the language which was used to generate the merge field name which was associated with a given column in the data source, as specified by the **fieldMapData** element (§2.14.15). This element specifies that when this field mapping is processed by an application, it shall interpret the merge field name as the name for the specified language.

If this element is omitted, then the mapped field names specified in the current document may be interpreted using any method desired by the consuming application (i.e. no language data is included with the field mapping information).

Attributes	Description
val (Language Code)	See 2.3.3.14

2.14.18 linkToQuery (Query Contains Link to External Query File)

This element specifies that the current WordprocessingML document's query string, stored in the **query** element (§2.14.26) and used to specify the data to be imported from the external data source, actually contains a reference to an external query file which contains the actual query data to be used against the specified external data source for the mail merge. This query shall mimic a SQL query and be of the following form: `SELECT * FROM <query file path>`.

If this element is omitted, then the query specified for the data source attached to the current document shall be assumed to not be a query containing a link to an external file.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.14.19 mailAsAttachment (Merged Document To E-Mail Attachment)

This element specifies that, after importing external data into fields to generate a series of destination WordprocessingML documents as e-mails, the resulting documents should be emailed as an attachment rather than the body of the actual e-mail.

If this element is omitted, then its value shall be assumed to be `false` (i.e. the destination source is not an e-mail attachment). If the **destination** element (§2.14.11) specifies that the merged document destination is not `email`, then this element shall be ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.14.20 mailMerge (Mail Merge Settings)

This element specifies all of the mail merge information for a document that has been connected to an external data source as part of a mail merge operation.

The document which contains this mail merge data may be of one of two types:

- A *source document*, the document which contains all of the information for the mail merge, and is used in conjunction with an application to connect to an external data source and create one document for each record in that data source.
- A *merged document*, a document which contains all of the information for the mail merge as well as a reference to a single specific record which shall be used to populate the values of all of the merge fields in that document.

The information in this element shall contain all data needed to connect to a data source and populate any merge fields in the document with data from that data source.

2.14.21 mailSubject (Merged E-mail or Fax Subject Line)

This element specifies the text which shall appear in the subject line of the e-mails or faxes that result after the actions of a mail merge have imported external data into fields within a merged WordprocessingML document whose destination, as specified in the **destination** element (§2.14.21), is `email` or `fax`.

If this element is omitted, then no subject line text shall be associated with each merged document produced via a mail merge using the specified mail merge data. If the **destination** element (§2.14.11) specifies that the merged document destination is not `email` or `fax`, this element shall be ignored.

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.22 mainDocumentType (Source Document Type)

This element specifies the type of a given WordprocessingML source document.

If this element is omitted, then its value shall be assumed to be `formLetters`.

[Note: This element is generally used in conjunction with the behavior of an application to customize aspects of the mail merge user interface and experience independent of the WordprocessingML file format. For example, if a given WordprocessingML merged document contains a **mainDocumentType** element with its `val` attribute equal to `envelopes`, the hosting application may surface a piece of user interface specific to creating envelopes when the given document is opened.

In addition, what a hosting application does with the documents that result from importing external data into specified fields can be determined based on the **mainDocumentType** element, but other than this, is independent of a given merged document's WordprocessingML. For example, if a given merged WordprocessingML document contains a **mainDocumentType** element with its `val` attribute equal to `email`, the hosting application may call a email service after importing external data into specified fields, in order to generate emails containing the resulting documents.

WordprocessingML simply provides the **mainDocumentType** that can serve as a trigger for an application to surface user interface specific to a type of mail merge. *end note*]

Attributes	Description
val (Mail Merge Source Document)	Specifies the type of source document which is specified by the given WordprocessingML document.

Attributes	Description
Type)	The possible values for this attribute are defined by the ST_MailMergeDocType simple type (§2.18.60).

2.14.23 mappedName (Predefined Merge Field Name)

This element specifies the predefined WordprocessingML MERGEFIELD field name which shall be mapped to the column number specified by the **column** element (§2.14.7) within this field mapping. *[Guidance: This element allows the current column from the specified data source to be mapped to a predefined field name, allowing applications to have one standard set of field names to use regardless of the data source column names, for example, to create the address formats to place into an ADDRESSBLOCK field. end guidance]*

If this element is omitted, then the current data source column mapping shall not have a predefined merge field name mapped to its contents, and shall only be referenced via the data source column name specified by the **name** element (§2.14.24) when referenced by one or more MERGEFIELD fields. If the application does not have a predefined merge field whose name matches the name specified using the val attribute, then this element may be ignored.

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.24 name (Data Source Name for Column)

This element specifies the column name within a given external data source for the column whose index is specified via the **column** element (§2.14.7). This data source name provides a column name which shall be used to map a specific MERGEFIELD field in the document, as specified by the parent field mapping data. The val attribute specifies the name of this column in the data source when the connection is initially established, which is then used permanently to link columns in the database to MERGEFIELD fields in the document.

If this element is omitted, no data source name is provided for the current column.

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.25 odso (Office Data Source Object Settings)

This element specifies a group of additional settings for the mail merge information which comprise an extension to the standard settings stored with a mail merge which performs two functions:

- First, it provides additional information about the mail merge data source, specifically: information about how to map the columns in the data source to MERGEFIELD fields and information about records which shall be included and excluded when creating merged documents, and column delimiters used in text data sources. This information may be used regardless of the value of the **dataType** element (§2.14.10) when it is present.
- Second, it provides an alternate set of connection information which should be used when the **dataType** element (§2.14.10) specifies a value of **native**. This alternate connection string provides additional connection information for applications which choose to support the ODSO connection string syntax. If the **dataType** element (§2.14.10) specifies that the data type of the current mail merge is not **native**, then the second group of settings specified within this element shall be ignored in favor of their non-ODSO equivalents.

2.14.26 query (Query For Data Source Records To Merge)

This element contains the Structured Query Language string that shall be run against the specified external data source to return the set of records from the external data which shall be imported into merged WordprocessingML documents when the mail merge operation is performed.

If this element is omitted, then no query shall be associated with the current data source.

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.27 recipientData (Reference to Inclusion/Exclusion Data for Data Source)

This element shall specify a reference to the part which contains data about whether the set of records in the associated data source have been explicitly included or excluded from the specified mail merge. Only those records which shall not be used to generate merged WordprocessingML documents must be stored within the referenced part, as all records shall be merged by default as part of the mail merge operation. [*Guidance:* Applications may choose to store only those records which are excluded for efficiency, or a list of all records in order to determine which set of records were added/removed between mail merge operations. *end guidance*]

[*Rationale:* When defining a mail merge, it is possible that a user wishes to connect to a specified data source, but specify only a subset of the records returned by the query specified by the **query** element (§2.14.26) which shall be merged as part of the mail merge operation. This element allows applications to utilize a separate part to store this information, either the shared part defined by this Office Open XML Standard, or an application-specific part as needed. *end rationale*]

If the relationship type of the relationship specified by this element is not

<http://schemas.openxmlformats.org/officeDocument/2006/mailMergeRecipientData>, is not present, or does not have a TargetMode attribute value of Internal, then the document shall be considered non-conformant. If an application cannot process external content of the content type specified by the targeted part, then it may be ignored.

This Office Open XML Standard defines one shared mechanism for storing this data: using the Mail Merge Recipient Data part. This mechanism shall be used if the associated data source has a column which may be used as the unique key. However, when using data sources which do not have a unique key, applications may store their own part (of an application-defined content type) using this relationship.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.14.28 recipientData (Data About Single Data Source Record)

This element specifies information about a single record within an external data source. If a record shall be merged into a merged document, then no information is needed about that record within this part. However, if a given record shall not be merged into a merged document, then the value of the unique key for that record shall be stored within the **uniqueTag** element as a child of this element (along with the **active** element) to indicate this exclusion.

[*Note:* This mapping is necessary in place of simply using the element order to correspond to the record indices in the external data source, as records may be added or deleted from external data sources, and a means must be provided to maintain WordprocessingML external record specific data like that specified in the **active** element (§2.14.1) and the corresponding external data record when the WordprocessingML document is reconnected to the external data source; irrespective of the ordering of the records within the external data source. In other words, this element, and its child elements enable merged WordprocessingML documents to maintain the relationship between the records within an external data and record specific WordprocessingML parameters. *end note*]

2.14.29 recipients (Inclusion/Exclusion Data for Data Source)

This element specifies all of the inclusion/exclusion data for the contents of the specified mail merge data source. It is the root element for the Mail Merge Recipient Data part.

2.14.30 src (ODSO Data Source File Path)

This element specifies the relationship whose target is the location of the external data source to be connected to a given WordprocessingML document to perform the mail merge (for a source document) or to find the associated field data (for a merged document) when the merge type, specified using the **dataType** element (§2.14.10), is set to **native**.

If this element is omitted, then no file location is specified for the data source for the current mail merge. If no relationship exists with the given relationship ID, or this relationship is not of the Mail Merge Data Source relationship type, then this document shall be considered non-conformant.

The data source location is only used under the following conditions:

- The **dataType** element (§2.14.10) is set to **native**
- The current application is able to use the information contained in the **odso** element (§2.14.25) to access the data source

Attributes	Description
id (Relationship to Part)	See 2.3.3.17
Namespace: .../officeDocument/2006/relationships	

2.14.31 table (Data Source Table Name)

This element specifies the particular set of data that a source or merged WordprocessingML document shall be connected to within an external data source containing multiple data sets. In other words, when connecting to a WordprocessingML document to an external data source that may have more than one repository of data within it, such as a database with multiple tables or a spreadsheet with multiple worksheets, this element is used to distinguish the specific table or spreadsheet from which data will be imported from within the external data source.

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.32 type (Merge Field Mapping)

This element specifies if a given mail merge field has been mapped to a column in the given external data source or not. If this element is omitted, then the field mapping shall be considered to be of type **null** (i.e. not mapped).

Attributes	Description
val (Merge Field Mapping Type)	Specifies if the given mail merge field has been mapped to a column in the given external data source (i.e. if the merge field mapping is valid or not). The possible values for this attribute are defined by the ST_MailMergeOdsoFMDFieldType simple type (§2.18.61).

2.14.33 type (ODSO Data Source Type)

This element specifies the type of external data source to be connected to via as part of the ODSO connection information for this mail merge. This setting is purely a suggestion of the data source type which is being used for this mail merge, and may be ignored in favor of an alternative mechanism if one is present.

Attributes	Description
val (Data Source Type Value)	Specifies the type of an external data source used for a mail merge operation. The possible values for this attribute are defined by the ST_MailMergeSourceType simple type

Attributes	Description
	(§2.18.62).

2.14.34 udl (UDL Connection String)

This element specifies the Universal Data Link (UDL) connection string used to reconnect to an external data source. The string within this element's **val** attribute shall contain the connection string that the hosting application shall pass to a external data source access application to enable the WordprocessingML document to be reconnected to the specified external data source.

If this string is omitted, then no UDL connection string shall be associated with the ODSO data for this mail merge.

This connection string is only used under the following conditions:

- The **dataType** element (§2.14.10) is set to **native**
- The current application is able to use the information contained in the **odso** element (§2.14.25) to access the data source

Attributes	Description
val (String Value)	See 2.3.1.27

2.14.35 uniqueTag (Unique Value for Record)

This element specifies the contents of a given record within the specified external data source, in the column containing unique data for every record within the external data source. given external data source. This element is used in conjunction with the **column** (§2.14.7) element to maintain a relationship between the records within an external data source and a given merged WordprocessingML document.

The contents of this attribute shall be the base64-encoded value of the unique tag value as specified by the data source.

[Note: This information is necessary as part of a mail merge as records may be added or deleted from external data sources, and a means must be provided to maintain record-specific inclusion or exclusion data using the **active** element (§2.14.1) and the affected external data record when the WordprocessingML document is reconnected to the external data source irrespective of the ordering of the records within the external data source. *end note*]

The possible values for this element are defined by the XML Schema base64Binary datatype.

2.14.36 viewMergedData (View Merged Data Within Document)

This element specifies that a specific merged document shall display the data from the specified external data source where merge fields have been inserted. The **activeRecord** element (§2.14.2) is used to specify which record within the external data source is to have its applicable data displayed where applicable within the WordprocessingML merged document.

If the **activeRecord** element is not present in the WordprocessingML for the document with the **viewMergedData**'s **val** attribute equal to on, the hosting application may behave as if the **activeRecord** element's **val** attribute was equal to 1. This element is ignored if the **viewMergedData** (§2.14.36) element is not present or present with a **val** attribute equal to Off, 0, or false.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15 Settings

Within a WordprocessingML document, *settings* specify stored preferences which shall be used when processing the contents of the document. These settings are typically divided into three categories:

- *Document Settings* - settings which influence the appearance and behavior of the current document, as well as store document-level state.

- *Compatibility Settings* - settings which tell applications to perform behaviors which are designed to maintain visual output of previous word processing applications. These settings are for backward compatibility and are all ignorable.
- *Web Settings* - settings which affect how this document shall be handled when it is saved as HTML. These settings exist primarily for backward compatibility reasons and are all ignorable.

The first two groups are stored in the Document Settings part, and the last group is stored in the Web Settings part.

2.15.1 Document Settings

The first group of settings stored in WordprocessingML is document settings. These settings specify all document-level properties which affect the handling of the current document.

2.15.1.1 **activeWritingStyle** (Grammar Checking Settings)

This element specifies information about the parameters of the grammar checking which was performed on the contents of the current WordprocessingML document. [Note: This information may be used as desired by applications; for example, to determine if the current grammar checking state, specified by the **proofState** element (§2.15.1.65) is sufficient. *end note*]

Attributes	Description
appName (Application Name)	Specifies the name of the application which specified the grammatical settings contained on the attributes for this element. If an application reads these settings and does not understand the value of this attribute, then its settings may be ignored and the application's default settings used instead. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
checkStyle (Check Stylistic Rules With Grammar)	Specifies if the grammar content checking performed on this document included stylistic rules for the document content. If specified, applications which support this functionality shall check stylistic rules as well as grammatical ones when checking the grammatical content of this document. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
dllVersion (Grammatical Check Engine Version)	Specifies the version of the engine that was used to check the grammatical content of the WordprocessingML document. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
lang (Writing Style Language)	Specifies the language of the engine used to perform the grammatical content checking. The possible values for this attribute are defined by the ST_Lang simple type (§2.18.50).
nlCheck (Natural Language Grammar Check)	Specifies whether the engine that was used to check the grammatical content of the WordprocessingML document performed natural language-based analysis. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
vendorID (Grammatical Engine ID)	Specifies a value indicating a unique ID for the writing style engine that was used to check the grammatical content of the WordprocessingML document. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).

2.15.1.2 **alignBordersAndEdges** (Align Paragraph and Table Borders with Page Border)

This element specifies that paragraph borders specified using the **pBdr** element (§2.3.1.24) and table borders using the **tblBorders** element (§2.4.37) shall be adjusted to align with extents of the page border defined using the **pgBorders** element (§2.6.10) if the spacing between these borders is less than or equal to 10.5 points (one character width) or less

from the page border. The presence of this setting shall ensure there are no gaps of one character width or less between adjoining page and paragraph/table borders, as borders which are perfectly aligning shall not be displayed in favor of the intervening page border.

If this element is omitted, then borders shall not be automatically adjusted to prevent gaps of less than one character width. If the page border is not measured from the text extents using a value of text in the **offsetFrom** attribute on the **pgBorders** element, then it may be ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.3 **alwaysMergeEmptyNamespace** (Do Not Mark Custom XML Elements With No Namespace As Invalid)

This element specifies whether custom XML markup specified via the **customXml** element which has no associated namespace shall be treated as an error and moved into a special error namespace (for the purposes of validation) when the document is opened. If this element is turned on, when an application determines that the current XML markup is in the empty namespace, those elements shall not automatically be moved into an error namespace.

If this element is not present in a WordprocessingML document than custom XML markup which has no associated namespace shall be treated as an error and moved into a special error namespace when the document is opened.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.4 **alwaysShowPlaceholderText** (Use Custom XML Element Names as Default Placeholder Text)

This element specifies that each custom XML element specified using the **customXml** element within this document shall always show some form of in-document placeholder text presentation when it contains no run content. If the **placeholder** element (§2.5.2.24) is present in the custom XML element's properties, then this is the placeholder text displayed and this effect has no effect. If the **placeholder** element is omitted, then the application shall use the name of the element to generate default placeholder text in its place.

If this element is omitted, then custom XML markup which does not contain a **placeholder** element within its properties shall not display any placeholder text.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.5 **attachedSchema** (Attached Custom XML Schema)

This element specifies that the custom XML schema whose target namespace matches the value specified in the **val** attribute should be associated with this document when it is loaded, if such a schema is available to the hosting application. Applications may also load and utilize any additional schemas as well as those explicitly mentioned here. *[Note: These custom XML schemas may then be used to validate the structure of the custom XML markup in the document, etc. end note]*

If no elements of this type are present, then no custom XML schemas have been explicitly associated with the contents of this document.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.1.6 **attachedTemplate** (Attached Document Template)

This element specifies the location of a document template which shall be attached to the current WordprocessingML document if it is accessible and of a format supported by an application. Specifically, this element's **val** attribute shall contain the file path of the associated document template.

If this element is omitted, then the document shall not have an attached document template, and applications should use their default template in its place.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.15.1.7 **autoCaption** (Single Automatic Captioning Setting)

This element specifies what type(s) of objects shall automatically labeled with captions (§2.15.1.17), and with which captions the specified objects shall be labeled as defined in the **caption** element (§2.15.1.16).

Attributes	Description
caption (Caption Used for Automatic Captioning)	Specifies the caption defined in using the caption element (§2.15.1.16) which shall be used to automatically label a given type of object inserted in a WordprocessingML document. The caption settings are linked by matching the value of this attribute with the name attribute of the corresponding caption element. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
name (Identifier of Object to be Automatically Captioned)	Specifies a unique identifier which may be used to associate objects inserted into the document which are to be automatically labeled with a caption when inserted into the WordprocessingML document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.15.1.8 **autoCaptions** (Automatic Captioning Settings)

This element specifies that one or more types of objects, when inserted into a WordprocessingML document, will automatically be labeled with a specific caption defined using the **caption** element (§2.15.1.16).

2.15.1.9 **autoFormatOverride** (Allow Automatic Formatting to Override Formatting Protection Settings)

This element specifies whether formatting automatically applied by an application (i.e. not explicitly applied by a user or an application) shall be allowed to override formatting protection enabled via the formatting attribute on the **documentProtection** element (§2.15.1.9) when those formatting operations would add formatting which has been explicitly disabled.

If this element is omitted, then no automatic formatting rule(s) shall be allowed to override the formatting restrictions enabled for the document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.10 **autoHyphenation** (Automatically Hyphenate Document Contents When Displayed)

This element specifies whether the content of a given WordprocessingML document should automatically be hyphenated by the hosting application before it is displayed, if the application supports this functionality.

If this element is omitted, then hyphenation shall not automatically be performed by application displaying this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.11 **bookFoldPrinting** (Book Fold Printing)

This element specifies if the contents of a given WordprocessingML document should be printed as signatures. *Signatures* are printed *sheets*, which depict several pages of a document that are folded and bound with other signatures to form a booklet, a set of which can be bound together to form a book like publication. Specifically, this element specifies that each page in a given WordprocessingML document should be oriented in a landscape fashion, divided in half vertically with two left margins emanating from the bisector of the page, and two right margins instantiated at the left and right side of each page.

This element is used in conjunction with the **bookFoldPrintingSheets** element (§2.15.1.12) to enable a WordprocessingML document to be printed such that the series of signatures printed may be folded and bound to create a booklet.

This element has no impact on the settings of printer leveraged by the hosting application. In other words, if the printer leveraged by the hosting application has been configured to print on one side of a page, including the WordprocessingML for this element has no effect.

If this element is omitted, then pages shall not be printed as signatures. If the **bookFoldRevPrinting** element (§2.15.1.13) is also specified, then this element shall be ignored.

[*Note*: This element could also be leveraged by the hosting application to notify the application to display two pages per sheets within its user interface to allow for a WYSIWYG user experience. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.12 **bookFoldPrintingSheets** (Number of Pages Per Booklet)

This element shall be used in conjunction with the **bookFoldPrinting** (§2.15.1.11) and **bookFoldRevPrinting** (§2.15.1.13) elements to specify the number of pages to be included in each booklet when printing a series of signatures. Signatures are printed *sheets*, which depict several pages of a document that are to be folded and bound with other signatures to form a booklet. Booklets can be bound together to form a book like publication.

If this element is omitted, then its default behavior shall be to print the contents of the content on a single sheet. A *sheet* is a single piece of paper which is folded and cut to produce a book.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.15.1.13 **bookFoldRevPrinting** (Reverse Book Fold Printing)

This element specifies if pages of a given WordprocessingML document are to be printed as signatures in reverse order. *Signatures* are printed *sheets*, which depict several pages of a document that are folded and bound with other signatures to form a booklet, a set of which can be bound together to form a book like publication. Specifically, this element specifies that each page in a given WordprocessingML document should be oriented in a landscape fashion and divided in half vertically, with two left margins emanating from the bisector of the page, and right margins instantiated at the left and right side of each page.

In addition, this element is used in conjunction with the **bookFoldPrintingSheets** element (§2.15.1.12) to enable given WordprocessingML document to be printed such that the series of signatures printed may be folded and bound to create a booklet.

This element has no impact on the settings of printer leveraged by the hosting application. In other words, if the printer leveraged by the hosting application has been configured to print on one side of a page, including the WordprocessingML for this element has no effect.

If this element is omitted, then pages shall not be printed as reverse book fold signatures. If the **bookFoldPrinting** element (§2.15.1.11) is also specified, then that element shall be ignored, and this element shall be used instead.

[*Note*: This element could also be leveraged by the hosting application to notify the application to display two pages per sheets within its user interface to allow for a WYSIWYG user experience. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.14 **bordersDoNotSurroundFooter** (Page Border Excludes Footer)

This element specifies that a given WordprocessingML document's page border specified using the **pgBorders** element (§2.6.10) should not surround contents of the footer.

If this element is omitted, then the page border shall not exclude the footer on the page. As well, this element shall be ignored if the **pgBorders** element has an **offsetFrom** attribute which is not equal to **text**.

[*Note*: If the **pgBorders** element has a **offsetFrom** attribute equal to **page**, the **bordersDontSurroundFooter** element shall be ignored as specifying the **pgBorders** element with a **offsetFrom** attribute equal to **page** is to specify that the positioning of borders within the document shall be calculated relative to the edge of the page and therefore irrespective of document content in the footer. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.15 **bordersDoNotSurroundHeader** (Page Border Excludes Header)

This element specifies that a given WordprocessingML document's page border specified using the **pgBorders** element (§2.6.10) should not surround contents of the header.

If this element is omitted, then the page border shall not exclude the header on the page. As well, this element shall be ignored if the **pgBorders** element has a **offsetFrom** attribute which is not equal to **text**.

[*Note*: If the **pgBorders** element has a **offsetFrom** attribute equal to **page**, the **bordersDontSurroundHeader** element shall be ignored as specifying the **pgBorders** element with a **offsetFrom** attribute equal to **page** is to specify that the positioning of borders within the document shall be calculated relative to the edge of the page and therefore irrespective of document content in the header. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.16 **caption** (Single Caption Type Definition)

This element specifies the contents and positioning for captions which may be used to automatically label objects in a WordprocessingML document. A *caption* is a string that labels an object included in a WordprocessingML document, and typically consists of a string plus a field which numbers this item within a collection of similar objects.

Attributes	Description
chapNum (Include Chapter Number in Field for Caption)	Specifies whether or not to display numbering associated with the most recent chapter heading in the WordprocessingML document within the caption field. A <i>chapter heading</i> is a paragraph of text within a WordprocessingML document that is formatted with a style that has been specified by the heading attribute to demarcate chapters in documents.

Attributes	Description																						
	<p>Only a style with its styleID attribute equal to Heading1, Heading2, Heading3, Heading4, Heading5, Heading6, Heading7, Heading8, or Heading9 may be specified as the style used to demarcate chapters in a document. The choice of which of these heading levels shall be used to determine the current chapter number is defined by the value of the corresponding heading attribute.</p> <p>If this attribute is omitted, then chapter numbers shall not be included in the resulting caption. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>																						
heading (Style for Chapter Headings)	<p>Specifies the given style that is used to demarcate chapter headings in a document.</p> <p>This value is used to link the chapter headings with paragraphs with a styleID attribute as follows:</p> <table data-bbox="418 611 1346 1104"> <tr> <th>Value</th><th>Description</th></tr> <tr> <td>1</td><td>Style with styleID of Heading1</td></tr> <tr> <td>2</td><td>Style with styleID of Heading2</td></tr> <tr> <td>3</td><td>Style with styleID of Heading3</td></tr> <tr> <td>4</td><td>Style with styleID of Heading4</td></tr> <tr> <td>5</td><td>Style with styleID of Heading5</td></tr> <tr> <td>6</td><td>Style with styleID of Heading6</td></tr> <tr> <td>7</td><td>Style with styleID of Heading7</td></tr> <tr> <td>8</td><td>Style with styleID of Heading8</td></tr> <tr> <td>9</td><td>Style with styleID of Heading9</td></tr> <tr> <td>Any other value</td><td>Application-defined. May be ignored.</td></tr> </table> <p>If this attribute is omitted, then its value shall be assumed to be 1. The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>	Value	Description	1	Style with styleID of Heading1	2	Style with styleID of Heading2	3	Style with styleID of Heading3	4	Style with styleID of Heading4	5	Style with styleID of Heading5	6	Style with styleID of Heading6	7	Style with styleID of Heading7	8	Style with styleID of Heading8	9	Style with styleID of Heading9	Any other value	Application-defined. May be ignored.
Value	Description																						
1	Style with styleID of Heading1																						
2	Style with styleID of Heading2																						
3	Style with styleID of Heading3																						
4	Style with styleID of Heading4																						
5	Style with styleID of Heading5																						
6	Style with styleID of Heading6																						
7	Style with styleID of Heading7																						
8	Style with styleID of Heading8																						
9	Style with styleID of Heading9																						
Any other value	Application-defined. May be ignored.																						
name (Caption Type Name)	<p>Specifies the literal string component of this caption.</p> <p>This value is used as follows:</p> <ul style="list-style-type: none"> It is added to the field containing the chapter number and item number of this object when a caption is automatically added to the document. It is used to uniquely label this caption type, allowing it to be linked with classes of objects via the autoCaption element (§2.15.1.7) <p>It may be used to label this caption type in a user interface. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>																						
noLabel (Do Not Include Name In Caption)	<p>Specifies if the string specified in the name attribute shall be included in the resulting caption when it is automatically added to the document. If set to true, then the label text in the name attribute is omitted when adding the caption.</p> <p>If this attribute is omitted, then the name shall be added to the caption. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>																						
numFmt (Caption Numbering Format)	<p>Specifies the format of the numbering which shall be included in an automatically generated caption to specify the index of this item in that collection (within the current chapter if chapNum is specified, or within the current document story).</p>																						

Attributes	Description
	<p>If this attribute is omitted, then its default value shall be assumed to be <code>decimal</code>.</p> <p>The possible values for this attribute are defined by the <code>ST_NumberFormat</code> simple type (§2.18.65).</p>
pos (Automatic Caption Placement)	<p>Specifies how an automatically inserted caption shall be positioned relative to the object that it is captioning.</p> <p>If this attribute is omitted, then the default value shall be <code>below</code>.</p> <p>The possible values for this attribute are defined by the <code>ST_CaptionPos</code> simple type (§2.18.8).</p>
sep (Chapter Number/Item Index Separator)	<p>Specifies the character which shall be used to separate the chapter number used in this caption from the caption item numbering. A caption format consists of three components:</p> <ul style="list-style-type: none"> • The (optional) literal string • The (optional) chapter number • The index of this caption within the chapter/document <p>When the latter two items are both present, they are delimited using the chapter separator specified by this attribute.</p> <p>If this attribute is omitted, then its default value shall be <code>hyphen</code>. If the chapter number is not part of the caption format, then this parameter shall be ignored.</p> <p>The possible values for this attribute are defined by the <code>ST_ChapterSep</code> simple type (§2.18.9).</p>

2.15.1.17 captions (Caption Settings)

This element specifies the presence of information about captions in a given WordprocessingML document. This information is divided into two components:

- The child element **caption** defines the format for a single type of caption to be automatically added to the document.
- The child element **autoCaptions** defines the types of objects to which a caption format shall automatically be applied.

This information should be used to determine the captions which are automatically added to objects when they are inserted into a WordprocessingML document. [Note: This setting is typically ignored unless it is specified in an application's default template. *end note*]

Captioning leverages fields (§2.16.5) to label objects with reference to either:

- Other captioned objects within a given document
- Other captioned objects within the same chapter in a given document (when chapter numbers are added by specifying the `chapNum` attribute on the caption type).

A *chapter* is a section of text within a WordprocessingML document that is preceded by content with a style that has been specified by to demarcate chapters in documents. Only one style may be specified as the style used for a single caption type to demarcate chapters in a document. A chapter ends immediately above the next instance of content with the style used to demarcate chapters.

[Note: WordprocessingML is designed such that the **caption** element may be used in conjunction with applications to provide a dynamic captioning experience. In other words, an application may use the WordprocessingML in the example above to automatically insert a caption consisting of the string `Table` followed by an incrementing decimal number field below tables when tables are inserted into a WordprocessingML document as defined by the **autoCaption** element (§2.15.1.7). *End note*]

2.15.1.18 **characterSpacingControl** (Character-Level Whitespace Compression)

This element specifies how full-width characters in the current WordprocessingML document should be compressed to remove additional whitespace when the contents of this document are displayed, specifically by specifying the set(s) of characters which may be compressed to remove additional whitespace. [Note: The behavior of this element is functionally identical to the CSS text-justify-trim property. *end note*]

If this element is omitted, then the default value shall be `dontCompress`.

Attributes	Description
val (Value)	Specifies the set(s) of characters which should be compressed when the contents of this document are displayed. The possible values for this attribute are defined by the ST_CharacterSpacing simple type (§2.18.10).

2.15.1.19 **clickAndTypeStyle** (Paragraph Style Applied to Automatically Generated Paragraphs)

This element specifies the paragraph style, specified using the **style** element, which shall be applied to paragraphs which are automatically created when text is inserted into a WordprocessingML document in an area of the document that has no other style associated with it. This style is referenced via the `val` attribute, which stores the style ID of the style (stored in the **styleId** attribute on the style definition).

[Guidance: Consider a WordprocessingML document opened in an application that allows users to place their cursor anywhere within the document editing canvas and enter text. The **clickAndTypeStyle** element should be used to specify the paragraph style to be associated with the paragraph of text entered after a user places their cursor somewhere in the blank document that results in the generation of new paragraphs. *end guidance*]

If this element is omitted, then the default paragraph style (the paragraph style whose default attribute is set to true), shall be used for automatically generated paragraphs. If the style whose **styleId** is specified using the `val` attribute is not a paragraph style or does not exist in the document, then the default paragraph style shall be used instead.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.1.20 **clrSchemeMapping** (Theme Color Mappings)

This element specifies the theme color, stored in the document's Theme part to which the value of this theme color shall be mapped. This mapping enables multiple theme colors to be chained together.

Attributes	Description
accent1 (Accent 1 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the <code>accent1</code> theme color shall be used. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§2.18.12).
accent2 (Accent 2 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the <code>accent2</code> theme color shall be used. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§2.18.12).
accent3 (Accent3)	Specifies the theme color in the document's theme part which shall be used in place of this color

Attributes	Description
Theme Color Mapping)	<p>when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>accent3</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
accent4 (Accent4 Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>accent4</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
accent5 (Accent5 Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>accent5</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
accent6 (Accent6 Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>accent6</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
bg1 (Background 1 Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>light1</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
bg2 (Background 2 Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>light2</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
followedHyperlink (Followed Hyperlink Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>followedHyperlink</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
hyperlink (Hyperlink Theme Color Mapping)	<p>Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content.</p> <p>If this attribute is omitted, then the <code>hyperlink</code> theme color shall be used.</p> <p>The possible values for this attribute are defined by the <code>ST_ColorSchemeIndex</code> simple type (§2.18.12).</p>
t1 (Text 1 Theme	Specifies the theme color in the document's theme part which shall be used in place of this color

Attributes	Description
Color Mapping)	when it is referenced by document content. If this attribute is omitted, then the t1 theme color shall be used. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§2.18.12).
t2 (Text 2 Theme Color Mapping)	Specifies the theme color in the document's theme part which shall be used in place of this color when it is referenced by document content. If this attribute is omitted, then the t2 theme color shall be used. The possible values for this attribute are defined by the ST_ColorSchemeIndex simple type (§2.18.12).

2.15.1.21 **consecutiveHyphenLimit** (Maximum Number of Consecutively Hyphenated Lines)

This element specifies the maximum number of consecutive lines of text that can end with a hyphen when the contents of this document are displayed. Once this limit has been reached, the following line shall not be hyphenated regardless of whether or not it meets the criteria needed for hyphenation.

If this element is omitted or has its val attribute equal to 0, the given WordprocessingML document shall have no limit on the number of consecutive lines of text that may end with a hyphen

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.15.1.22 **decimalSymbol** (Radix Point for Field Code Evaluation)

This element specifies the character that shall be interpreted as the radix point when evaluating the contents of all fields in the current document.

[*Rationale*: When evaluating field instructions based on the contents of the current document, it is necessary to know the character which shall be treated as the radix point in order to prevent changes to the calculation of the same field instructions based on the current user's locale. This element stores the radix point which shall be used to evaluate fields in the contents of this document, irrespective of the locale of the application loading the file. *end rationale*]

If this element is omitted, the application shall use the default radix point of its current locale setting to evaluate field instructions. If this element's attribute value is more than a single character, then the document is non-conformant.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.1.23 **defaultTableStyle** (Default Table Style for Newly Inserted Tables)

This element specifies the table style which shall automatically be applied to the table properties of tables added to this document by an application. Note that it does not change the table style applied to tables which do not reference a style, instead, it automatically applies the style to that table via the **tblStyle** element (§2.4.59). This link is made by referencing the styleId attribute value of the table style which shall be used to format newly inserted tables.

If this element is omitted, then no table style shall automatically be applied to inserted tables (therefore inheriting the default table style). If the referenced style is not present or not a table style, then no table style shall automatically be applied to inserted tables.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.1.24 **defaultTabStop** (Distance Between Automatic Tab Stops)

This element specifies the value which shall be used as the multiplier to generate automatic tab stops in this document. *Automatic tab stops* refer to the tab stop locations which occur after all custom tab stops in the current paragraph have been surpassed.

If this element is omitted, then automatic tab stops should be generated at 720 twentieths of a point (0.5") intervals across the displayed page.

Attributes	Description
val (Measurement in Twentieths of a Point)	Specifies a positive measurement value, specified in twentieths of a point. This value is interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_TwipsMeasure simple type (§2.18.104).

2.15.1.25 **displayBackgroundShape** (Display Background Objects When Displaying Document)

This element specifies whether the images and colors defined in the document's background using the **background** element (§2.2.1) shall be displayed when the document is displayed in print layout view as specified in the **view** element (§2.15.1.93).

If this element is omitted, then background shapes shall not be displayed when the document is displayed in print layout view.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.26 **displayHorizontalDrawingGridEvery** (Distance between Horizontal Gridlines)

This element specifies the number of horizontal grid units defined using the **drawingGridHorizontalSpacing** element (§2.15.1.44) which shall be allowed between subsequent visible horizontal drawing grid lines in this document, if gridlines are being shown. [Note: The display of gridlines is an application-level setting not specified in this Office Open XML Standard. *end note*] The *drawing grid* is a grid which may be used by applications to help position floating objects in the document.

If this element is omitted, then gridlines shall be displayed for each horizontal grid unit.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.15.1.27 **displayVerticalDrawingGridEvery** (Distance between Vertical Gridlines)

This element specifies the number of vertical grid units defined using the **drawingGridVerticalSpacing** element (§2.15.1.46) which shall be allowed between subsequent vertical gridlines in this document, if gridlines are being shown. [Note: The display of gridlines is an application-level setting not specified in this Office Open XML Standard. *end note*] The *drawing grid* is a grid which may be used by applications to help position floating objects in the document.

If this element is omitted, then vertical gridlines shall not be displayed.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.15.1.28 documentProtection (Document Editing Restrictions)

This element specifies the set of document protection restrictions which have been applied to the contents of a WordprocessingML document. These restrictions shall be enforced by applications editing this document when the enforcement attribute is turned on, and should be ignored (but persisted) otherwise. *Document protection* is a set of restrictions used to prevent unintentional changes to all or part of a WordprocessingML document - since this protection does not encrypt the document, malicious applications may circumvent its use. This protection is not intended as a security feature and may be ignored.

If this element is omitted, then no protection shall be applied to this document.

When a password is supplied via an application which shall be hashed and stored in this element, that process shall be done in two stages:

First, the password shall be hashed using the following algorithm:

- Truncate the password to 15 characters.
- Construct a new NULL-terminated string consisting of single-byte characters:
 - Get the single-byte values by iterating through the Unicode characters of the truncated password. For each character, if the low byte is not equal to 0, take it. Otherwise, take the high byte.
- From now on, the single-byte character string is used.
- If the password is empty, return 0.
- Compute the high-order word of the new key:
 - Initialize from the initial code array (see below), depending on the password's length. For each character in the password:
 - For every bit in the character, starting with the least significant and progressing to (but excluding) the most significant, if the bit is set, XOR the key's high-order word with the corresponding word from the encryption matrix
- Compute the low-order word of the new key:
 - Initialize with 0
 - For each character in the password, going backwards, low-order word = (((low-order word SHR 14) AND 0x0001) OR (low-order word SHL 1) AND 0x7FFF) XOR character
 - Lastly, low-order word = (((low-order word SHR 14) AND 0x0001) OR (low-order word SHL 1) AND 0x7FFF) XOR password length XOR 0xCE4B.

Initial code array

The initial code array contains the initial values for the key's high-order word. The initial value depends on the length of the password, as follows:

Password length	Initial value for the key's high-order word
1	0xE1F0
2	0x1D0F
3	0xCC9C
4	0x84C0
5	0x110C
6	0x0E10
7	0xF1CE
8	0x313E
9	0x1872
10	0xE139
11	0xD40F
12	0x84F9
13	0x280C

Password length	Initial value for the key's high-order word
14	0xA96A
15	0x4EC3

Encryption matrix

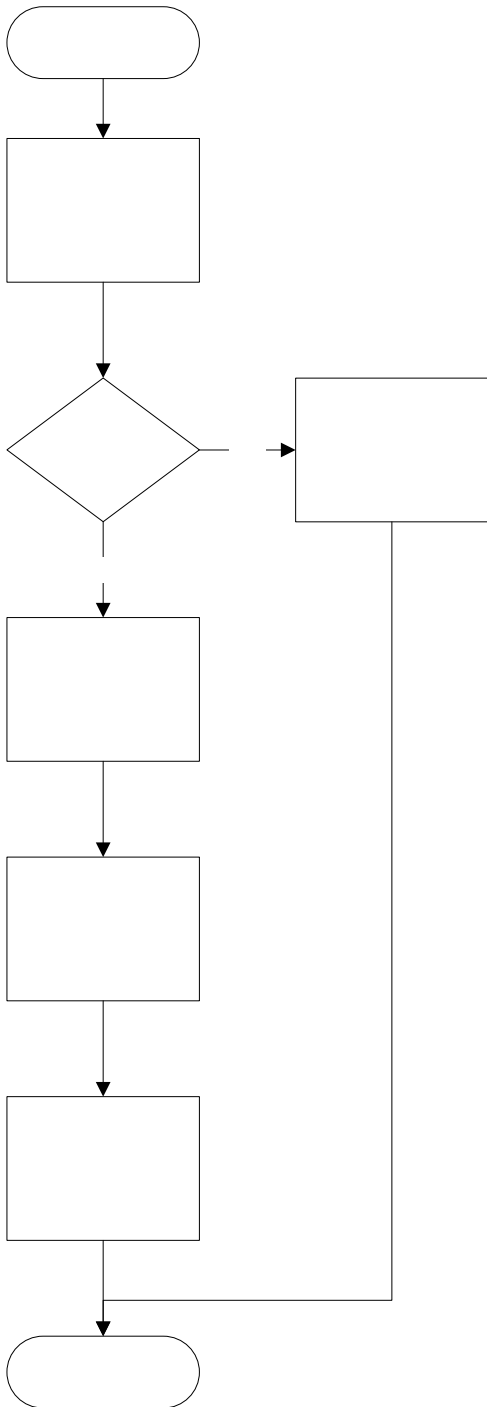
The encryption matrix contains codes used during the calculation of the key's high-order word. As described in the algorithm above, for every bit of the password's characters, if the bit is set, a corresponding value is taken from this encryption matrix and is used to XOR the key's high-order word with it. Each row in the encryption matrix corresponds to a single character from the password, and each of the seven columns corresponds to a particular bit (0-6) in this character. The values are taken in such a way so that the last character of the password uses the last row in the encryption matrix. The next-to-last character uses the next-to-last row in the matrix, and so on. This means that the beginning of the matrix may be unused, depending on the length of the password.

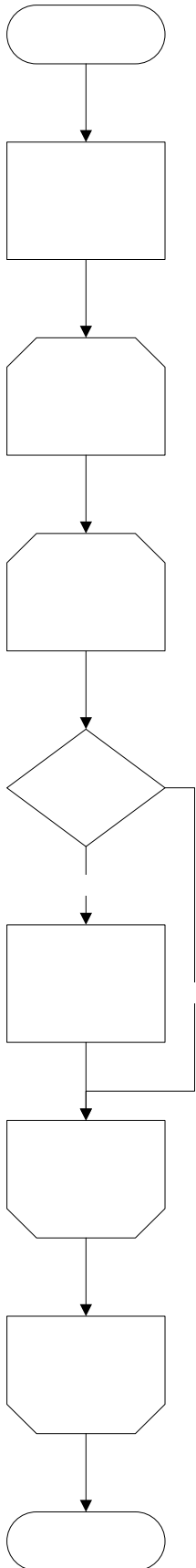
	Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6
Last-14	0xAEFC	0x4DD9	0x9BB2	0x2745	0x4E8A	0x9D14	0x2A09
Last-13	0x7B61	0xF6C2	0xFDA5	0xEB6B	0xC6F7	0x9DCF	0x2BBF
Last-12	0x4563	0x8AC6	0x05AD	0x0B5A	0x16B4	0x2D68	0x5AD0
Last-11	0x0375	0x06EA	0x0DD4	0x1BA8	0x3750	0x6EA0	0xDD40
Last-10	0xD849	0xA0B3	0x5147	0xA28E	0x553D	0xAA7A	0x44D5
Last-9	0x6F45	0xDE8A	0xAD35	0x4A4B	0x9496	0x390D	0x721A
Last-8	0xEB23	0xC667	0x9CEF	0x29FF	0x53FE	0xA7FC	0x5FD9
Last-7	0x47D3	0x8FA6	0x0F6D	0x1EDA	0x3DB4	0x7B68	0xF6D0
Last-6	0xB861	0x60E3	0xC1C6	0x93AD	0x377B	0x6EF6	0xDDEC
Last-5	0x45A0	0x8B40	0x06A1	0x0D42	0x1A84	0x3508	0x6A10
Last-4	0xAA51	0x4483	0x8906	0x022D	0x045A	0x08B4	0x1168
Last-3	0x76B4	0xED68	0xCAF1	0x85C3	0x1BA7	0x374E	0x6E9C
Last-2	0x3730	0x6E60	0xDCC0	0xA9A1	0x4363	0x86C6	0x1DAD
Last-1	0x3331	0x6662	0xCCC4	0x89A9	0x0373	0x06E6	0x0DCC
Last	0x1021	0x2042	0x4084	0x8108	0x1231	0x2462	0x48C4

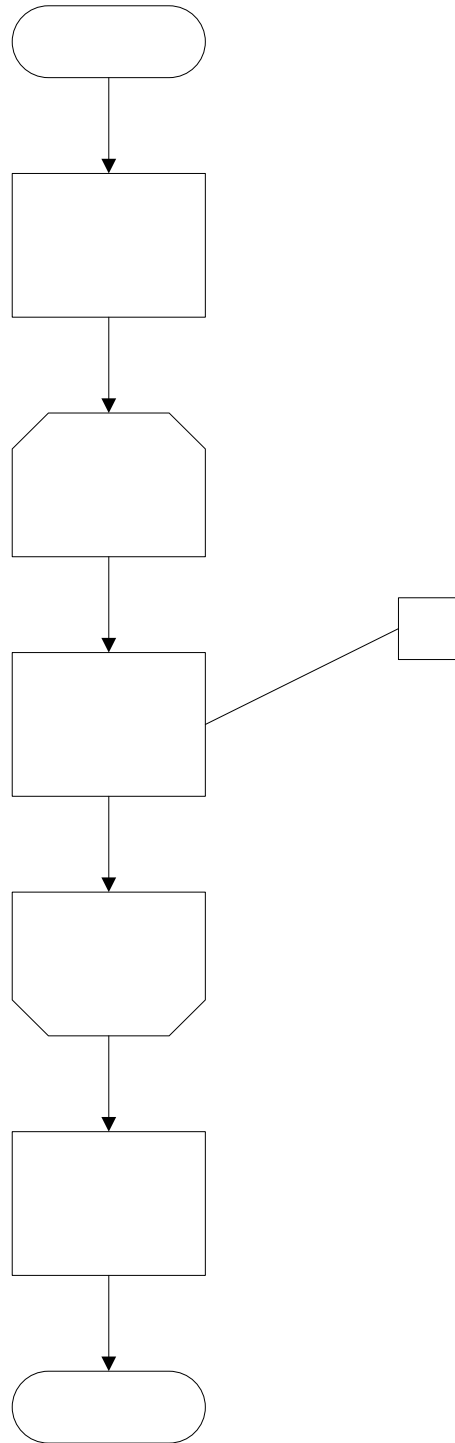
[*Rationale*: This pre-processing step is necessary for compatibility with legacy word processing applications which hashed their password solely using this mechanism. *end rationale*]

Second, the byte order of the result shall be reversed , and that value shall be hashed as defined by the attribute values.

[*Note*: The algorithm above can be stated as follows using diagrams:







end note]

Attributes	Description																																
algIdExt (Cryptographic Algorithm Extensibility)	<p>Specifies that a cryptographic algorithm which was not defined by this Office Open XML Standard has been used to generate the hash value stored with this document.</p> <p>This value, when present, shall be interpreted based on the value of the algIdExtSource attribute in order to determine the algorithm used, which shall be application-defined. [<i>Rationale</i>: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future will likely need to utilize as yet undefined hashing algorithms in order to remain secure. <i>end rationale</i>]</p> <p>If this value is present, the cryptAlgorithmClass, cryptAlgorithmType, and cryptAlgorithmSid attribute values shall be ignored in favor of the algorithm defined by this attribute. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).</p>																																
algIdExtSource (Algorithm Extensibility Source)	<p>Specifies the application which defined the algorithm value specified by the algIdExt attribute. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>																																
cryptAlgorithmClass (Cryptographic Algorithm Class)	<p>Specifies the class of cryptographic algorithm used by this protection. [<i>Note</i>: The initial version of this Office Open XML Standard only supports a single version - hash - but future versions may expand this as necessary. <i>end note</i>]</p> <p>The possible values for this attribute are defined by the ST_Algorithm simple type (§2.18.1).</p>																																
cryptAlgorithmSid (Cryptographic Hashing Algorithm)	<p>Specifies the specific cryptographic hashing algorithm which shall be used along with the salt attribute and user-supplied password in order to compute a hash value for comparison.</p> <p>The possible values for this attribute shall be interpreted as follows:</p> <table border="1" data-bbox="418 1087 1360 1843"> <thead> <tr> <th>Value</th><th>Algorithm</th></tr> </thead> <tbody> <tr><td>1</td><td>MD2</td></tr> <tr><td>2</td><td>MD4</td></tr> <tr><td>3</td><td>MD5</td></tr> <tr><td>4</td><td>SHA-1</td></tr> <tr><td>5</td><td>MAC</td></tr> <tr><td>6</td><td>RIPEMD</td></tr> <tr><td>7</td><td>RIPEMD-160</td></tr> <tr><td>8</td><td>Undefined. Shall not be used.</td></tr> <tr><td>9</td><td>HMAC</td></tr> <tr><td>10</td><td>Undefined. Shall not be used.</td></tr> <tr><td>11</td><td>Undefined. Shall not be used.</td></tr> <tr><td>12</td><td>SHA-256</td></tr> <tr><td>13</td><td>SHA-384</td></tr> <tr><td>14</td><td>SHA-512</td></tr> <tr><td>Any other value</td><td>Undefined. Shall not be used.</td></tr> </tbody> </table> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type</p>	Value	Algorithm	1	MD2	2	MD4	3	MD5	4	SHA-1	5	MAC	6	RIPEMD	7	RIPEMD-160	8	Undefined. Shall not be used.	9	HMAC	10	Undefined. Shall not be used.	11	Undefined. Shall not be used.	12	SHA-256	13	SHA-384	14	SHA-512	Any other value	Undefined. Shall not be used.
Value	Algorithm																																
1	MD2																																
2	MD4																																
3	MD5																																
4	SHA-1																																
5	MAC																																
6	RIPEMD																																
7	RIPEMD-160																																
8	Undefined. Shall not be used.																																
9	HMAC																																
10	Undefined. Shall not be used.																																
11	Undefined. Shall not be used.																																
12	SHA-256																																
13	SHA-384																																
14	SHA-512																																
Any other value	Undefined. Shall not be used.																																

Attributes	Description
	(§2.18.15).
cryptAlgorithmType (Cryptographic Algorithm Type)	Specifies the type of cryptographic algorithm used by this protection. [Note: The initial version of this Office Open XML Standard only supports a single type - typeAny - but future versions may expand this as necessary. end note] The possible values for this attribute are defined by the ST_AlgType simple type (§2.18.2).
cryptProvider (Cryptographic Provider)	Specifies the cryptographic provider which was used to generate the hash value stored in this document. If the user provided a cryptographic provider which was not the system's built-in provider, then that provider shall be stored here so it can subsequently be used if available. If this attribute is omitted, then the built-in cryptographic provider on the system shall be used. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
cryptProviderType (Cryptographic Provider Type)	Specifies the type of cryptographic provider to be used. The possible values for this attribute are defined by the ST_CryptProv simple type (§2.18.14).
cryptProviderTypeExt (Cryptographic Provider Type Extensibility)	Specifies that a cryptographic provider type which was not defined by this Office Open XML Standard has been used to generate the hash value stored with this document. This value, when present, shall be interpreted based on the value of the cryptProviderTypeExtSource attribute in order to determine the provider type used, which shall be application-defined. [Rationale: This extensibility affords the fact that with exponentially increasing computing power, documents created in the future will likely need to utilize as yet undefined cryptographic provider types in order to remain secure. end rationale] If this value is present, the cryptProviderType attribute value shall be ignored in favor of the provider type defined by this attribute. The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.56).
cryptProviderTypeExt Source (Provider Type Extensibility Source)	Specifies the application which defined the provider type value specified by the cryptProviderTypeExt attribute. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
cryptSpinCount (Iterations to Run Hashing Algorithm)	Specifies the number of times the hashing function shall be iteratively run (using each iteration's result as the input for the next iteration) when attempting to compare a user-supplied password with the value stored in the hash attribute. [Rationale: Running the algorithm many times increases the cost of exhaustive search attacks correspondingly. Storing this value allows for the number of iterations to be increased over time to accommodate faster hardware (and hence the ability to run more iterations in less time). end rationale] The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
edit (Document Editing Restrictions)	Specifies the set of editing restrictions which shall be enforced on a given WordprocessingML document, as defined by the simple type referenced below If this attribute is omitted, the consumer shall behave as though there are no editing restrictions applied to this document; equivalent to an attribute value of none. The possible values for this attribute are defined by the ST_DocProtect simple type (§2.18.21).
enforcement (Enforce Document Protection Settings)	Specifies if the document protection settings shall be enforced for a given WordprocessingML document. If the value of this element is off, 0, or false, all the WordprocessingML pertaining to document protection is still preserved in the document, but is not enforced. If the value of this element is on, 1, or true, the document protection is enforced.

Attributes	Description
	<p>If this attribute is omitted, then document protection settings shall not be enforced by applications.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
formatting (Only Allow Formatting With Unlocked Styles)	<p>Specifies if formatting restrictions are in effect for a given WordprocessingML document. This enables the document to restrict the types of styles that may exist in a given WordprocessingML document. Specifically, by setting this attribute's value equal to true, every style whose locked element (§2.7.3.7) has a value of true (or latent styles (§2.7.3.5) whose locked attribute is true) shall not be available for use in the application, nor should any direct formatting. Only styles with a locked value of false may be used.</p> <p>If this attribute is omitted, then no formatting restrictions shall be applied, even when document protection is enforced.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
hash (Password Hash)	<p>Specifies the hash value for the password stored with this document. This value shall be compared with the resulting hash value after hashing the user-supplied password using the algorithm specified by the preceding attributes and parent XML element, and if the two values match, the protection shall no longer be enforced.</p> <p>If this value is omitted, then no password shall be associated with the protection, and it may be turned off without supplying any password.</p> <p>The possible values for this attribute are defined by the XML Schema base64Binary datatype.</p>
salt (Salt for Password Verifier)	<p>Specifies the salt which was prepended to the user-supplied password before it was hashed using the hashing algorithm defined by the preceding attribute values to generate the hash attribute, and which shall also be prepended to the user-supplied password before attempting to generate a hash value for comparison. A <i>salt</i> is a random string which is added to a user-supplied password before it is hashed in order to prevent a malicious party from pre-calculating all possible password/hash combinations and simply using those precalculated values (often referred to as a "dictionary attack").</p> <p>If this attribute is omitted, then no salt shall be prepended to the user-supplied password before it is hashed for comparison with the stored hash value.</p> <p>The possible values for this attribute are defined by the XML Schema base64Binary datatype.</p>

2.15.1.29 **documentType** (Document Classification)

This element specifies the classification of a given WordprocessingML document as a letter, email, or general document.

[*Note:* This element may be used by hosting applications to facilitate customized user interface and/or automatic formatting behaviors based on the 'type' of a given WordprocessingML document. *end note*]

If this element is omitted, then the document shall be classified as a general document.

Attributes	Description
val (Document Classification Value)	<p>Specifies the classification of the document based on the types defined in the referenced simple type definition.</p> <p>The possible values for this attribute are defined by the ST_DocType simple type (§2.18.22).</p>

2.15.1.30 **docVar** (Single Document Variable)

This element specifies the parameters of a single document variable. A *document variable* is a storage location for arbitrary customer data in name/value pairs that will be persisted in a given WordprocessingML document. Specifically, this element specifies through its name and val attributes the name and value pair for a given document variable.

[Note: This mechanism is maintained for legacy compatibility only, and should be avoided in favor of the custom XML data support defined in this Office Open XML Standard. *end note*]

Attributes	Description
name (Document Variable Name)	Specifies the name of the parent document variable. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
val (Document Variable Value)	Specifies the value of the parent document variable. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.15.1.31 **docVars** (Document Variables)

This element specifies the presence of document variables in a WordprocessingML. A *document variable* is a storage location for arbitrary customer data in name/value pairs that will be persisted in a given WordprocessingML document.

[Note: This mechanism is maintained for legacy compatibility only, and should be avoided in favor of the custom XML data support defined in this Office Open XML Standard. *end note*]

2.15.1.32 **doNotAutoCompressPictures** (Do Not Automatically Compress Images)

This element specifies that pictures in this document shall not automatically be compressed when saving the document in order to reduce the overall size of the resulting WordprocessingML document.

If this element is omitted, applications may perform basic compression on images before saving the contents of the document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.33 **doNotDemarcateInvalidXml** (Do Not Show Visual Indicator For Invalid Custom XML Markup)

This element specifies whether a visual cue should be displayed around content contained in a WordprocessingML document which is contained with custom XML markup specified via the **customXml** element when an application determines that the current XML markup (or its contents) violate the constraints of the attached XML schema(s).

If this element is not present in a WordprocessingML document visual cues shall be displayed on content contained in custom XML markup in a WordprocessingML document which is considered to be invalid based on the associated XML schema(s).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.34 **doNotDisplayPageBoundaries** (Do Not Display Visual Boundary For Header/Footer or Between Pages)

This element specifies whether applications displaying this document should display the contents of the header and footer when displaying the document in print layout view (§2.15.1.93) or should collapse those areas as well as the whitespace on all displayed pages so that the text extents are directly following one another. [*Rationale*: Collapsing the ends of pages makes it easier to read the contents of the document, since the text flows between pages without whitespace, while maintaining the WYSIWYG functionality of print layout view for the document's main content. *end rationale*]

If this element is omitted, then all pages should be shown at their full size (including whitespace and headers/footers) when they are displayed in print layout view.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.35 **doNotEmbedSmartTags** (Remove Smart Tags When Saving)

This element specifies if any smart tags specified using the **smartTag** element shall be removed from the contents of this document before it is resaved. This setting shall also prevent the addition of new smart tags to the content of the document.

If this element is omitted, then smart tags shall not be removed from the file when it is saved.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.36 **doNotHyphenateCaps** (Do Not Hyphenate Words in ALL CAPITAL LETTERS)

This element specifies whether or not words comprised of all capital letters shall be hyphenated within a given document when automatic hyphenation is specified via the **autoHyphenation** element (§2.15.1.10).

If this element is omitted, then words in ALL CAPITAL LETTERS shall be hyphenated when the document is hyphenated.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.37 **doNotIncludeSubdocsInStats** (Do Not Include Content in Text Boxes, Footnotes, and Endnotes in Document Statistics)

This element specifies if document content contained in text boxes, footnotes, and endnotes shall be excluded when an application calculates a given document's statistics when these values are calculated and/or displayed by an application. [Note: Some examples of document statistics that an application may chose to calculate are: number of words, number of characters, number of paragraphs, number of pages, number of lines, and so on. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.38 **doNotShadeFormData** (Do Not Show Visual Indicator For Form Fields)

This element specifies whether a visual cue should be displayed around form fields contained in a WordprocessingML document specified via the FORMTEXT, FORMCHECKBOX, or FORMDROPDOWN fields.

If this element is not present in a WordprocessingML document visual cues should be displayed on form fields contained in the document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.39 **doNotTrackFormatting** (Do Not Track Formatting Revisions When Tracking Revisions)

This element specifies that applications shall not track revisions made to the formatting of this WordprocessingML document when the **trackRevisions** element (§2.15.1.90) is turned on.

If this element is omitted, then revisions to formatting shall be generated by changes to the contents of this document when the **trackRevisions** element is turned on.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.40 **doNotTrackMoves** (Do Not Use Move Syntax When Tracking Revisions)

This element specifies that applications shall not track revisions made to this WordprocessingML document as moves when the **trackRevisions** element (§2.15.1.90) is turned on, even when that syntax is appropriate. Instead, applications should use a standard insertion and deletion annotation syntax. Existing moves shall not be modified. [*Rationale*: This element is provided to enable interoperability with earlier word processing applications which do not understand moves. *end rationale*]

If this element is omitted, then move annotations may be generated by changes to the contents of this document when the **trackRevisions** element is turned on as appropriate.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.41 **doNotUseMarginsForDrawingGridOrigin** (Do Not Use Margins for Drawing Grid Origin)

This element specifies that the top-left corner of the page shall not be used as the origin for the drawing grid. The *drawing grid* is a virtual grid which may be used by applications to specify where drawing objects shall be positioned on a page when inserted (i.e. to ensure objects are aligned, etc.). If this element is present the grid shall start at the top-left edge of the page and not the text extents.

If this element is omitted, then the gridlines shall start at the topmost edge of the text extents.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.42 **doNotValidateAgainstSchema** (Do Not Validate Custom XML Markup Against Schemas)

This element specifies that applications shall not validate the custom XML markup in this document against the applicable custom XML schema(s), even when those schemas are available. The application should silently behave as if it was unable to provide this functionality.

If this element is omitted, then applications which support this functionality should attempt to validate the custom XML contents against any available related custom XML schema(s).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.43 **drawingGridHorizontalOrigin** (Drawing Grid Horizontal Origin Point)

This element specifies the distance from of the left edge of the page which shall be used as the origin for the horizontal gridlines used by the drawing grid. The *drawing grid* is a virtual grid which may be used by applications to specify where drawing objects shall be positioned on a page when inserted (i.e. to ensure objects are aligned, etc.). Since the grid always covers the entire page when the **doNotUseMarginsForDrawingGridOrigin** element (§2.15.1.41) is specified, this element shall only affect the starting edge of the first horizontal gridline displayed (i.e. it only adjusts the grid by the modulus of the value against the width of one grid unit).

If this element is omitted, then the gridlines shall start at the leftmost edge of the page. If the **doNotUseMarginsForDrawingGridOrigin** element is not specified, then this element is ignored.

Attributes	Description
val (Measurement in Twentieths of a Point)	See 2.15.1.24

2.15.1.44 **drawingGridHorizontalSpacing** (Drawing Grid Horizontal Grid Unit Size)

This element specifies the width of horizontal grid units in this document. The *drawing grid* is a grid which may be used by applications to help position floating objects in the document.

If this element is omitted, then each horizontal grid unit shall be 180 twentieths of a point (0.125") in width.

Attributes	Description
val (Measurement in Twentieths of a Point)	See 2.15.1.24

2.15.1.45 **drawingGridVerticalOrigin** (Drawing Grid Vertical Origin Point)

This element specifies the distance from of the top edge of the page which shall be used as the origin for the vertical gridlines used by the drawing grid. The *drawing grid* is a virtual grid which may be used by applications to specify where drawing objects shall be positioned on a page when inserted (i.e. to ensure objects are aligned, etc.). Since the grid always covers the entire page when the **doNotUseMarginsForDrawingGridOrigin** element (§2.15.1.41) is specified, this element shall only affect the starting edge of the first vertical gridline displayed (i.e. it only adjusts the grid by the modulus of the value against the width of one grid unit).

If this element is omitted, then the gridlines shall start at the topmost edge of the page. If the **doNotUseMarginsForDrawingGridOrigin** element is not specified, then this element is ignored.

Attributes	Description
val (Measurement in Twentieths of a Point)	See 2.15.1.24

2.15.1.46 **drawingGridVerticalSpacing** (Drawing Grid Vertical Grid Unit Size)

This element specifies the width of vertical grid units in this document. The *drawing grid* is a grid which may be used by applications to help position floating objects in the document.

If this element is omitted, then each vertical grid unit shall be 180 twentieths of a point (0.125") in width.

Attributes	Description
val (Measurement in Twentieths of a Point)	See 2.15.1.24

2.15.1.47 **forceUpgrade** (Upgrade Document on Open)

This element specifies that the contents of this document may be upgraded and that the resulting document shall not have its functionality limited to only those functions compatible with earlier word processing applications. The only actions required as part of upgrading the document are:

- The removal of this and the **uiCompat97to2003** element (§2.15.3.54). This is needed in order to prevent future applications from disabling functionality on the resulting 'upgraded' document. If an application does not know how to upgrade a document, this element and the **uiCompat97to2003** element should be ignored and persisted.
- The removal of all compatibility options (§2.15.3.9) on the document which maintain compatibility with previous word processing applications. The compatibility settings which simply affect a given behavior shall not be turned off.

[Note: The remaining operations which shall be performed as part of upgrading the document are application-defined and outside the scope of this Office Open XML Standard. *end note*]

2.15.1.48 **formsDesign** (Structured Document Tag Placeholder Text Should be Resaved)

This element specifies that the document was last saved while the placeholder text of all structured document tags in this document were being edited. This means that the placeholder text currently displayed in all structured document tags which are displaying the **showingPlcHdr** element (§2.5.2.38) shall be committed to the corresponding glossary document entry as specified using the **docPart** element (§2.12.5) when this document is opened, in order to ensure that the most recent placeholder text is stored in the glossary document entry. If the current placeholder text cannot be saved as a glossary document entry, then it should be modified as needed before saving.

If this element is omitted, then the placeholder text in this document should not automatically be resaved when the document is opened.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.49 **gutterAtTop** (Position Gutter At Top of Page)

This element specifies that a given WordprocessingML document's gutter shall be positioned at the top of the document's pages when the document is displayed. A *gutter* is the white space formed by the inner margins of two pages facing one another; such as the white space between the text on pages of a book when the book is opened.

If this element is omitted, then the gutter shall not be positioned at the top of the page. If the **mirrorMargins** (§2.15.1.57), **bookFoldPrinting** (§2.15.1.11), **bookFoldRevPrinting** (§2.15.1.13), or **printTwoOnOne** (§2.15.1.64) elements are used within a given document, the **gutterAtTop** element shall not be used. Rather, the gutter shall be positioned automatically as necessary to enable the printing and page layout capabilities of these settings.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.50 **hdrShapeDefaults** (Default Properties for VML Objects in Header and Footer)

This element specifies the default parameters for object using the ' (§**Error! Reference source not found.**) inserted in the header and footer of a WordprocessingML document. The definition and semantics of these parameters is described in the VML - Office Drawing subclause (§**Error! Reference source not found.**) of this Office Open XML Standard.

If this element is omitted, then no default properties are applied to VML objects in the header and footer of this document.

2.15.1.51 **hideGrammaticalErrors** (Do Not Display Visual Indication of Grammatical Errors)

This element specifies whether a visual cue should be displayed around run content contained in a WordprocessingML document which has been flagged as a possible grammatical error using the **proofErr** element (§2.13.8.1) or via the application's own grammar engine.

If this element is not present in a WordprocessingML document, visual cues shall be displayed on content contained in a WordprocessingML document which is considered to contain grammatical errors.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.52 **hideSpellingErrors** (Do Not Display Visual Indication of Spelling Errors)

This element specifies whether a visual cue should be displayed around run content contained in a WordprocessingML document which has been flagged as a possible spelling error using the **proofErr** element (§2.13.8.1) or via the application's own spelling engine.

If this element is not present in a WordprocessingML document, visual cues shall be displayed on content contained in a WordprocessingML document which is considered to contain spelling errors.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.53 **hyphenationZone** (Hyphenation Zone)

This element specifies the hyphenation zone which shall be used when automatically or manually hyphenating the contents of this document. The *hyphenation zone* is the amount of whitespace which may be left at the end of a line (or added to justified lines) before hyphenation should be attempted on the next word in the document (in order to reduce the amount of whitespace on the line). A smaller hyphenation zone should reduce the raggedness of the right edge of a given document's body text, as more words will be hyphenated. Conversely, a larger hyphenation zone should increase the raggedness of the right edge of a given document's text, as fewer words will be hyphenated.

If this element is omitted, then a default hyphenation zone of 360 twentieths of a point (0.25") shall be applied when performing hyphenation on this document.

Attributes	Description
val (Measurement in Twentieths of a Point)	See 2.15.1.24

2.15.1.54 **ignoreMixedContent** (Ignore Mixed Content When Validating Custom XML Markup)

This element specifies that applications should ignore all text content which is not contained within a leaf custom XML markup element when validating the contents of the custom XML markup in this document against one or more attached custom XML schema(s). A *leaf element* is a custom XML element which has no child custom XML elements (it is a leaf in the custom XML tree).

If this element is omitted, then text content in leaf elements shall not be ignored when validating the custom XML markup against one or more custom XML schema(s).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.55 **linkStyles** (Automatically Update Styles From Document Template)

This element specifies that styles in the given document shall be updated to match the styles in the attached template specified using the **attachedTemplate** element (§2.15.1.6) when the document is opened by a hosting application. This setting enables the styles contained in documents with attached templates to stay synchronized with the styles used in the attached template.

If this element is omitted, then styles shall not be updated based on the document template regardless of its availability. If the attached template cannot be located or is not a valid file, then this setting should be silently ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.56 **listSeparator** (List Separator for Field Code Evaluation)

This element specifies the character that shall be interpreted as a list item separator when evaluating the contents of all fields in the current document.

[*Rationale*: When evaluating field instructions based on the contents of the current document, it is necessary to know the character which shall be treated as the list separator in order to prevent changes to the calculation of the same field

instructions based on the current user's locale. This element stores the list separator which shall be used to evaluate fields in the contents of this document, irrespective of the locale of the application loading the file. *end rationale*

If this element is omitted, the application shall use the default list separator of its current locale setting to evaluate field instructions. If this element's attribute value is more than a single character, then the document is non-conformant.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.1.57 **mirrorMargins** (Mirror Page Margins)

This element specifies that the left and right margins defined in the section properties shall be swapped on facing pages.

[*Guidance*: This setting is generally used when printing on both sides of pages and binding them like a book. *end guidance*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.58 **noLineBreaksAfter** (Custom Set of Characters Which Cannot End a Line)

This element specifies the set of characters which shall be restricted from ending a line for runs of text which shall be subject to custom line breaking logic using the **kinsoku** element (§2.3.1.16) when the contents of the document are displayed. This constraint shall only apply to text which has been flagged in the language of this rule via the **lang** element (§2.3.2.18) or automatic detection methods outside the scope of this Office Open XML Standard.

If this element is omitted, then no custom set of characters shall be used to restrict the characters which may end a line when using the **kinsoku** element.

Attributes	Description
lang (Language For Which Custom Line Breaking Rule Applies)	Specifies the language of text for which the parent custom line breaking rule shall be applied. Applications supporting this functionality shall support custom line breaking for the following four languages: <ul style="list-style-type: none"> Chinese (Traditional) Chinese (Simplified) Japanese Korean Applications may also support custom line breaking rules for other languages, but this is not required. The possible values for this attribute are defined by the ST_Lang simple type (§2.18.50).
val (Characters For Custom Line Breaking Rule)	Specifies the set of characters which shall be included in the custom line breaking rule. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.15.1.59 **noLineBreaksBefore** (Custom Set Of Characters Which Cannot Begin A Line)

This element specifies the set of characters which shall be restricted from beginning a new line for runs of text which shall be subject to custom line breaking logic using the **kinsoku** element (§2.3.1.16) when the contents of the document are displayed. This constraint shall only apply to text which has been flagged in the language of this rule via the **lang** element (§2.3.2.18) or automatic detection methods outside the scope of this Office Open XML Standard.

If this element is omitted, then no custom set of characters shall be used to restrict the characters which may end a line when using the **kinsoku** element.

Attributes	Description
------------	-------------

Attributes	Description
See 2.15.1.58	

2.15.1.60 **noPunctuationKerning** (Never Kern Punctuation Characters)

This element specifies that punctuation characters shall not be kerned in the current document when kerning is enabled on a run using the **kern** element (§2.3.2.17). *Kerning* refers to a process by which a hosting application shall reduce the spacing of adjacent characters and/or punctuation to improve the visual appearance of text. Well kerned text has a similar amount of blank space between each pair of characters and/or each set of a character and punctuation symbol. When kerning is enabled, Latin text shall always be kerned, and this option shall control whether punctuation characters are also kerned. If this element is omitted, then punctuation characters shall be kerned when kerning is enabled on a given run.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.61 **printFormsData** (Only Print Form Field Content)

This element specifies that printing the contents of this document shall only print the contents of WordprocessingML form fields defined using the FORMTEXT, FORMCHECKBOX, and FORMDROPDOWN field codes in their current locations on the page - all other document contents shall be suppressed.
If this element is omitted, then the contents of the entire document (not just form fields) should be printed according to the normal print settings.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.62 **printFractionalCharacterWidth** (Print Fractional Character Widths)

This element specifies the contents of this document shall be printed with fractional character widths. *Fractional character widths* exist when the spacing between characters is not constant (i.e. a proportional font face is used).
[Note: Fractional character widths are generally used in conjunction with large font sizes to prevent characters from running together or having too much space between one another. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.63 **printPostScriptOverText** (Print PostScript Codes With Document Text)

This element specifies that the PostScript codes specified in WordprocessingML documents containing PRINT fields shall be included in foreground (on the same Z-order as text) with the data printed in the contents of a given WordprocessingML document.
[Note: This setting is maintained to ensure compatibility of legacy word processing documents. The PRINT field should not be used in lieu of newer technologies in this Office Open XML Standard. *end note*]
If this element is omitted, then the contents of PRINT fields shall be printed behind text (i.e. in the background).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.64 **printTwoOnOne** (Print Two Pages Per Sheet)

This element specifies whether two pages should be printed on one sheet of paper when this document is printed. Specifically, this element specifies that each page displayed for the contents in a given WordprocessingML document

should be the page size specified in the section settings divided in half with two top margins originating from the bisector of the page, and bottom margins instantiated at the top and bottom of each page.

If this element is omitted, then pages should be displayed and printed as one per sheet.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.65 **proofState** (Spelling and Grammatical Checking State)

This element specifies if the grammar and spell checking engines of the last application to process this document completed checking the grammar and spelling of a the document before the document was last saved. Applications which modify the document contents without checking spelling or grammar should reset these states as needed.

[*Note*: If this element specifies that an application's grammar and spell checking engines completed checking the grammar and spelling of the document when the document was last saved, then subsequent applications may elect to not run their grammar and spell checking engines when the given WordprocessingML document is loaded.

This may increase the speed with which the hosting application loads the file, and does not compromise the state of the grammar or spell checking of the document, as all errors have already been found and flagged with the **proofErr** element (§2.13.8.1) as the document has not been edited, only loaded, since it was last saved. *end note*]

Attributes	Description
grammar (Grammatical Checking State)	Specifies if an application's grammar checking engine completed checking the grammatical content of the document when it was last saved. If this attribute is omitted, then its value is assumed to be dirty (not complete). The possible values for this attribute are defined by the ST_Proof simple type (§2.18.75).
spelling (Spell Checking State)	Specifies if an application's spell checking engine completed checking the spelling of the document when it was last saved. If this attribute is omitted, then its value is assumed to be dirty (not complete). The possible values for this attribute are defined by the ST_Proof simple type (§2.18.75).

2.15.1.66 **readModeInkLockDown** (Freeze Document Layout)

This element specifies the exact set of page and text sizing parameters which shall be used to display the contents of a WordprocessingML document. This element shall only affect the display of WordprocessingML documents as follows:

- When the **actualPage** attribute is specified with a value of **true**, the given WordprocessingML document's pages shall be rendered as they would normally be displayed. The resulting pages may have their magnification setting changed as desired. All other attributes shall be ignored.
- When the **actualPage** attribute is specified with a value of **false**, the given WordprocessingML document's pages shall be rendered as *virtual pages* when loaded by a conforming hosting application irrespective of the given WordprocessingML document's view (§2.15.1.93). *Virtual pages* are pages with no correlation with the printed layout of a given WordprocessingML document that have been scaled by a conforming hosting application to improve the readability of a given WordprocessingML document when it is displayed. Specifically, the **w** and **h** attributes specify the width and height of the virtual pages, and the **fontSz** attribute specifies the scaling to be applied to text within the given WordprocessingML document.

Attributes	Description
actualPg (Use Actual Pages, Not Virtual Pages)	Specifies if applications shall render this WordprocessingML document with actual pages, not virtual pages. <i>Actual pages</i> are pages rendered as they will be printed. A value of true specifies that the given WordprocessingML document's pages will be rendered as

Attributes	Description
	they are printed, and the w, h, and fontSz attributes shall be ignored. A value of <code>false</code> specifies that the given WordprocessingML document's pages shall be rendered as virtual pages using the other attributes on this element. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
fontSz (Font Size Scaling)	Specifies the percentage that text in a given WordprocessingML document shall be scaled by before it is displayed on a virtual page. The attribute's value stores the percentage specified as an integer who units correspond to the percentage that text runs shall be scaled to . This attribute shall only be used if the actualPage attribute equals <code>off</code> , <code>0</code> , or <code>false</code> . The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).
h (Virtual Page Height)	Specifies the height of the virtual pages which shall be used in this document. This value is specified in pixels. This attribute shall only be used if the actualPage attribute equals <code>off</code> , <code>0</code> , or <code>false</code> . The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§2.18.73).
w (Virtual Page Width)	Specifies the width of the virtual pages which shall be used in this document. This value is specified in pixels. This attribute shall only be used if the actualPage attribute equals <code>off</code> , <code>0</code> , or <code>false</code> . The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§2.18.73).

2.15.1.67 **removeDateAndTime** (Remove Date and Time from Annotations)

This element specifies that the date and time information shall be removed from all annotations which are present in the current document when it is saved. Annotations store this information in the date attribute on the annotation's XML element.

If this element is omitted, then date information shall not be removed when the document is saved. If the **removePersonalInformation** element is not turned on, then this setting shall be ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.68 **removePersonalInformation** (Remove Personal Information from Document Properties)

This element specifies that hosting applications shall remove all personal information of document authors upon saving a given WordprocessingML document. The definition and extent of personal information is not defined by this Office Open XML Standard.

If this element is omitted, then personal information shall not be removed when the document is saved.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.69 **revisionView** (Visibility of Annotation Types)

This element specifies which forms of annotations shall be visible for a WordprocessingML document when it is displayed. This setting shall not affect whether annotations are added or persisted, it shall only affect the display of the annotations which exist in the document's contents (persisted or in memory).

If this element is omitted, then all forms of annotations shall be visible.

Attributes	Description
comments (Display Comments)	Specifies if comments should be included when the contents of this document are displayed. If this attribute is omitted, then comments shall be displayed when annotations are visible based on application-level settings. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
formatting (Display Formatting Revisions)	Specifies if revisions to properties (i.e. formatting revisions) should be included when the contents of this document are displayed. If this attribute is omitted, then formatting revisions shall be displayed when annotations are visible based on application-level settings. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
inkAnnotations (Display Ink Annotations)	Specifies if ink annotations, specified in VML syntax (Error! Reference source not found.), should be included when the contents of this document are displayed. If this attribute is omitted, then ink annotations shall be displayed when annotations are visible based on application-level settings. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
insDel (Display Content Revisions)	Specifies if revisions to content (i.e. insertions, deletions, and moves) should be included when the contents of this document are displayed. If this attribute is omitted, then insertions, deletions, and moves shall be displayed when annotations are visible based on application-level settings. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).
markup (Display Visual Indicator Of Markup Area)	Specifies if the application shall visually indicate any additional non-printing area used to display annotations when the annotations in this document are displayed. If this attribute is omitted, then any additional non-printing area shall be indicated when they are visible based on application-level settings. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).

2.15.1.70 **rsid** (Single Session Revision Save ID)

This element specifies the revision save ID that was associated with a single editing session for a document. An editing session is a span of time that begins and ends with any event that produces an editable file, such as a save or an e-mail send, and contains no such event. When revision save IDs are added to a document, they shall follow these rules:

- Every editing session shall be assigned a revision save ID that is larger than all earlier ones in the same file
- Revision save IDs should be randomly generated based on the current time (to minimize the chance that two disparate editing sessions starting with the same immediate predecessor are assigned the same revision save ID)
- Changes to document content in an editing session shall be stamped with the current revision save ID using the appropriate rsid* attributes
- An identical **rsid** value between two documents with the same **rsidRoot** (§2.15.1.71) shall indicate the same editing sessions

[Note: A revision save ID should be treated as unique within the context of all documents with the same **rsidRoot** value. Although in practice it is possible for two independent sessions to result in the same value, this outcome is extremely rare as the values are based on the current time. However, the meaning of two revision save IDs is not defined for documents with a different **rsidRoot**. Applications may use this information as desired. *end note*]

Attributes	Description
val (Long Hexadecimal Number Value)	See 2.7.3.15

2.15.1.71 **rsidRoot** (Original Document Revision Save ID)

This element specifies the revision save ID which was associated with the first editing session for this document. *[Note: This information shall be identical between any number of copies of the same document, as they all originate from the same original editing session. Applications may use this information as desired. end note]*

If this element is omitted, then the original document revision save ID is unknown.

Attributes	Description
val (Long Hexadecimal Number Value)	See 2.7.3.15

2.15.1.72 **rsids** (Listing of All Revision Save ID Values)

This element specifies the set of revision save ID values for the current document. *Revision save ID values* refer to four digit hexadecimal values which uniquely identify an editing session in the life of the current document. An *editing session* is the period of time between two subsequent save operations by an application.

[Guidance: The set of revision save IDs stored with a document only supplies information about the editing session in which document components were last saved, which may be used by applications in any manner desired. end guidance]

If this element is omitted, then no information is available about the set of revision save ID values for this document.

Child Elements	Subclause
rsid §2.15.1.70; rsidRoot §2.15.1.71	

2.15.1.73 **saveFormsData** (Only Save Form Field Content)

This element specifies that saving the contents of this document shall only save the contents of WordprocessingML form fields defined using the FORMTEXT, FORMCHECKBOX, and FORMDROPDOWN field codes in a comma-delimited text format which does not conform to this Office Open XML Standard (i.e. it is a one-way export from a WordprocessingML document).

[Rationale: This setting is typically used to allow duplication of paper forms in electronic WordprocessingML document form, allowing the resulting content to be extracted as a comma-delimited text file. end rationale]

If this element is omitted, then the contents of the entire document (not just form fields) should be saved according to the definition of WordprocessingML in this Office Open XML Standard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.74 **saveInvalidXml** (Allow Saving Document As XML File When Custom XML Markup Is Invalid)

This element specifies that this document should be capable of being saved into a format consisting of a single XML file (not defined by this Office Open XML Standard) when its contents are invalid based on the custom XML markup contained in the document. This setting has no effect on documents that do not contain custom XML markup, or that do contain custom XML markup but do not have a schema attached. *[Guidance: Because this setting specifies behavior when saving to an alternative file format not defined by this Office Open XML Standard, this behavior is optional. end guidance]*

If this element is omitted, then applications should not allow this document to be saved into a single XML file when its contents are invalid based on the custom XML markup contained in the document. If the **doNotValidateAgainstSchema** element (§2.15.1.42) is set, then the XML is never "invalid" and this property is ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.75 **savePreviewPicture** (Generate Thumbnail For Document On Save)

This element specifies if a document's Thumbnail part should be generated for the contents of the first page of this document when saved by application which support document thumbnail generation.

If this element is omitted, then applications may choose to save a thumbnail, however, that behavior is not required. If this element is specified, a thumbnail must be produced if that functionality is supported.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.76 **saveThroughXslt** (Custom XSL Transform To Use When Saving As XML File)

This element specifies the location of a custom XSL transform which shall be used when this document is saved as a single XML file (in a format not defined by this Office Open XML Standard). If this element is omitted, then no custom XSL transform shall be used when saving this file as a single XML file. If the **useXSLTWhenSaving** element (§2.15.1.92) is omitted or set to **false**, then this transform shall not be applied when the document is saved as a single XML file.

Attributes	Description
id (XSL Transformation Location) Namespace: .../officeDocument/2006/relationships	Specifies an explicit relationship to the location of the XSL Transformation which shall be applied. The relationship targeted by this element shall be of type http://schemas.openxmlformats.org/officeDocument/2006/relationships/transform , or this document shall be declared invalid. The possible values for this attribute are defined by the ST_RelationshipId simple type (\$Error! Reference source not found.).
solutionID (Local Identifier for XSL Transform)	Specifies a string identifier which may be used to locate the XSL transform to be applied. The semantics of this attribute are not defined by this Office Open XML Standard - applications may use this information in any application-defined manner to resolve the location of the XSL transform to apply. If this attribute is omitted, then no local identifier is specified for the XSL transform. If both this and the xslt attributes are present, then this data shall be used first, and the latter shall only be used if this information cannot be used successfully. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.15.1.77 **saveXmlDataOnly** (Only Save Custom XML Markup)

This element specifies that the contents of this document shall be saved as an XML file containing only the custom XML markup in this document in its regular form. The resulting document will not conform to this Office Open XML Standard (i.e. this is an export-only save option for a WordprocessingML document).

[*Rationale*: This setting is typically used to extract custom XML markup from a WordprocessingML document for further processing by XML-enabled applications. *end rationale*]

If this element is omitted, then the contents of the entire document (not just custom XML markup) should be saved according to the definition of WordprocessingML in this Office Open XML Standard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.78 settings (Document Settings)

This element specifies the settings that are applied to a WordprocessingML document. This element is the root element of the Document Settings part in a WordprocessingML document.

2.15.1.79 shapeDefaults (Default Properties for VML Objects in Main Document)

This element specifies the default parameters for object using the VML syntax (**Error! Reference source not found.**) inserted in the body (the main document story, comments, footnotes, and endnotes) of the WordprocessingML document. The definition and semantics of these parameters is described in the VML - Office Drawing subclause (**Error! Reference source not found.**) of this Office Open XML Standard.

If this element is omitted, then no default properties are applied to VML objects in the body of this document.

2.15.1.80 showEnvelope (Show E-Mail Message Header)

This element specifies that an e-mail message header shall be displayed when this document is opened, if an e-mail header is supported by the application opening the file.

If this element is omitted, then applications shall not display the e-mail message header automatically when this file is opened, even if one is available in the application opening the file.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.81 showXMLTags (Show Visual Indicators for Custom XML Markup Start/End Locations)

This element specifies that some visual indicator shall be provided for the start and end locations of custom XML markup present in this document, if any.

If this element is omitted, then applications should not provide any visual indicator of the locations of custom XML markup start/end tags.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.82 smartTagType (Supplementary Smart Tag Information)

This element specifies optional supplementary information about one or more smart tags (§2.5.1.9) used in the current WordprocessingML document. This supplementary data is linked to the smart tag to which it applies via its name and namespaceuri attributes.

Attributes	Description
name (Smart Tag Name)	Specifies the name of the smart tag within the document for which supplementary data is provided. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
namespaceuri (Smart Tag Namespace)	Specifies the namespace URI of the smart tag for which supplementary data is provided. If this attribute is omitted, the URI shall be assumed to be null (no associated URI). The possible values for this attribute are defined by the ST_String simple type (§2.18.88).
url (Smart Tag Supplementary URL)	Specifies a URL provided for a particular smart tag type in this document. [<i>Note</i> : This URL is typically used to provide access to a URL for additional updates to this smart tag type as requested by the smart tag provider. <i>end note</i>] If this attribute is omitted, then no supplementary URL is provided for this type.

Attributes	Description
	The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.15.1.83 **strictFirstAndLastChars** (Use Strict Kinsoku Rules for Japanese Text)

This element specifies that the strict set of Kinsoku rules shall be applied to Japanese text in this document when the **kinsoku** element (§2.3.1.16) is applied to that text. The resulting line breaking rules are provided on the **kinsoku** element. If this element is omitted, then standard rules shall apply to Japanese text when the **kinsoku** element is applied to that text.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.84 **styleLockQFSet** (Prevent Replacement of Styles Part)

This element specifies whether applications shall prevent the replacement of the complete set of styles stored in the Styles part when editing this document. This setting should not preclude the editing or removal of individual styles, instead, it should only prevent the removal and replacement of the entire styles part in a single operation (either through a user interface or a programmatic operation).

If this element is omitted, then applications may allow the replacement of the entire styles part in this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.85 **styleLockTheme** (Prevent Modification of Themes Part)

This element specifies whether applications shall prevent the modification of the document's theme information stored in the Theme part when editing this document. This setting should not preclude the use of the theme information, instead, it should only prevent the modification of the theme part in a single operation (either through a user interface or a programmatic operation).

If this element is omitted, then applications may allow the replacement or modification of the theme part in this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.86 **stylePaneFormatFilter** (Suggested Filtering for List of Document Styles)

This element specifies a set of suggested filters which should be applied to the list of document styles in this application if the styles are displayed in a user interface.

The val attribute of this element contains a bitmask of the following filtering options:

Value	Description
0x0001	Specifies that all styles present in the styles part should be displayed in the list of document styles.
0x0002	Specifies that only styles with the customStyle attribute should be displayed in the list of document styles.
0x0004	Specifies that all latent styles should be displayed in the list of document styles.
0x0008	Specifies that only styles used in the document should be displayed in the list of document styles.
0x0010	Undefined. Shall not be used.
0x0020	Specifies that heading styles (styles with a styleId of Heading1 to Heading9) should be displayed in the list of document styles when the previous style is used in the document and/or is present in the styles part.
0x0040	Specifies that numbering styles should be displayed in the list of document styles.

Value	Description
0x0080	Specifies that table styles should be displayed in the list of document styles.
0x0100	Specifies that all unique forms of run-level direct formatting should be displayed in the list of document styles as though they were each a unique style.
0x0200	Specifies that all unique forms of paragraph-level direct formatting should be displayed in the list of document styles as though they were each a unique style.
0x0400	Specifies that all unique forms of direct formatting of numbering data should be displayed in the list of document styles as though they were each a unique style.
0x0800	Specifies that all unique forms of direct formatting of tables should be displayed in the list of document styles as though they were each a unique style.
0x1000	Specifies that a style should be present which removes all formatting and styles from text.
0x2000	Specifies that heading styles with a styleId of Heading1 to Heading3 should always be displayed in the list of document styles.
0x4000	Specifies that styles should only be shown the semiHidden element (§2.7.3.16) is false and the hidden element (§2.7.3.4) is false .
0x8000	Specifies that primary names for styles should not be shown if an alternate name using the name element (§2.7.3.9) exists.
Any other value	Undefined. Shall not be used.

If this element is omitted, then all settings defined by this element are turned off.

Attributes	Description
val (Two Digit Hexadecimal Value)	See 2.4.51

2.15.1.87 stylePaneSortMethod (Suggested Sorting for List of Document Styles)

This element specifies a suggested sorting which should be applied to the list of document styles in this application if the styles are displayed in a user interface.

The val attribute of this element specifies one of the following sorting options:

Value	Description
0x0000	Specifies that styles which are visible should be sorted by their names.
0x0001	Specifies that styles which are visible should be sorted by their UI priority using the uiPriority element (§2.7.3.19).
0x0002	Specifies that styles which are visible should be sorted by the default sorting of the host application.
0x0003	Specifies that styles which are visible should be sorted by the font which they apply.
0x0004	Specifies that styles which are visible should be sorted by the style on which they are based using the basedOn element (§2.7.3.3).
0x0005	Specifies that styles which are visible should be sorted by their style types (i.e. character, linked, paragraph).
Any other value	Undefined. Shall not be used.

If this element is omitted, then styles which are visible should be sorted by the default sorting of the host application.

Attributes	Description
val (Two Digit Hexadecimal Value)	See 2.4.51

2.15.1.88 **summaryLength** (Percentage of Document to Use When Generating Summary)

This element specifies the size for automatic document summaries performed on the content of a WordprocessingML document. An *automatic document summary* is a subset of text contained in a document deemed by the hosting application to summarize the content of the WordprocessingML document. The **val** attribute of this element specifies the size of an automatic document summary to be performed on a given WordprocessingML document as a percentage of the total size of the given WordprocessingML document. Performing an automatic document summary is a runtime operation outside the scope of this Office Open XML Standard.

If this element is omitted, then applications may summarize this document to any desired size.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.15.1.89 **themeFontLang** (Theme Font Languages)

This element specifies the language which shall be used to determine the appropriate theme fonts in the document's Theme part which map to the major/minor theme fonts.

These mappings are performed as follows:

- For **majorAscii/majorHAnsi**, locate the **font** element (**Error! Reference source not found.**) in the **majorFont** element (**Error! Reference source not found.**) in the theme part for the language specified by the **val** attribute
- For **majorBidi**, locate the **font** element in the **majorFont** element in the theme part for the language specified by the **bidi** attribute
- For **majorEastAsia**, locate the **font** element in the **majorFont** element in the theme part for the language specified by the **eastAsia** attribute
- For **minorAscii/minorHAnsi**, locate the **font** element in the **minorFont** element (**Error! Reference source not found.**) in the theme part for the language specified by the **val** attribute
- For **minorBidi**, locate the **font** element in the **minorFont** element in the theme part for the language specified by the **bidi** attribute
- For **minorEastAsia**, locate the **font** element in the **minorFont** element in the theme part for the language specified by the **eastAsia** attribute

If this element is omitted, then the default fonts for each region as specified by the **latin**, **ea**, and **cs** elements (**Error! Reference source not found.**; **Error! Reference source not found.**; **Error! Reference source not found.**) should be used.

Attributes	Description
See 2.3.2.18	

2.15.1.90 **trackRevisions** (Track Revisions to Document)

This element specifies that applications shall track revisions made to the WordprocessingML document. *Revisions* are changes to a WordprocessingML document which are recorded such that they can be viewed independently, accepted or removed, and reverted if needed. When revisions are tracked, the resulting WordprocessingML markup in the Revisions subclause of this document describes the necessary syntax.

If this element is omitted, then revisions shall not be generated by changes to the contents of this document.

Attributes	Description
------------	-------------

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.91 **updateFields** (Automatically Recalculate Fields on Open)

This element specifies whether the fields contained in this document should automatically have their field result recalculated from the field codes when this document is opened by an application which supports field calculations. [*Note*: Some fields are always recalculated (e.g. the page numbering), therefore this element only affects fields which are typically not automatically recalculated on opening the document. Also note that this setting shall not supersede any document protection (§2.15.1.28) or write protection (§2.15.1.94) settings. *end note*]

If this element is omitted, then fields should not automatically be recalculated on opening this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.92 **useXSLTWhenSaving** (Save Document as XML File through Custom XSL Transform)

This element specifies that this document should be saved through the custom XSLT transform defined by the **saveThroughXslt** element (§2.15.1.76) in this document when it is saved as a single XML file (not defined by this Office Open XML Standard). [*Guidance*: Because this setting specifies behavior when saving to an alternative file format not defined by this Office Open XML Standard, this behavior is optional. *end guidance*]

If the **saveXmlDataOnly** element (§2.15.1.77) is specified, then the single XML file to be transformed is the custom XML markup of the document, otherwise, it a format outside the scope of this Office Open XML Standard. If the XSL transform specified by the **saveThroughXslt** element is not present, then this setting should be ignored.

If this element is omitted, then this document should not be saved through a custom XSL transform when it is saved as a single XML file.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.1.93 **view** (Document View Setting)

This element specifies the manner in which the contents of this document should be displayed when opened by an application.

If this element is omitted, then an application may view the document in any desired default state.

Attributes	Description
val (Document View Setting Value)	<p>Specifies the view which shall be used to render the contents of a WordprocessingML document.</p> <p>Applications may omit support for one or more of the views defined by the ST_View simple type (referenced below). If a WordprocessingML document containing an unsupported view is loaded by an application, it shall fall back to its default view (equivalent to use of the enumeration value none).</p> <p>The possible values for this attribute are defined by the ST_View simple type (§2.18.111).</p>

2.15.1.94 **writeProtection** (Write Protection)

This element specifies the write protection settings which have been applied to a WordprocessingML document. *Write protection* refers to a mode in which the document's contents cannot be edited, and the document cannot be resaved using the same file name. This setting is independent of the **documentProtection** (§2.15.1.28) element, but like document protection, this setting is not intended as a security feature and may be ignored.

When present, the write protection shall result in one of two write protection behaviors:

- If the password attribute is present, or both attributes are omitted, then the application shall prompt for a password to exit write protection. If the supplied password does not match the hash value in this attribute, then write protection shall be enabled.
- If only the recommended attribute is present, the application should provide user interface recommending that the user open this document in write protected state. If the user chooses to do so, the document shall be write protected, otherwise, it shall be opened fully editable.

If this element is omitted, then no write protection shall be applied to the current document.

Attributes	Description
algIdExt (Cryptographic Algorithm Extensibility)	See 2.15.1.28
algIdExtSource (Algorithm Extensibility Source)	See 2.15.1.28
cryptAlgorithmClass (Cryptographic Algorithm Class)	See 2.15.1.28
cryptAlgorithmSid (Cryptographic Hashing Algorithm)	See 2.15.1.28
cryptAlgorithmType (Cryptographic Algorithm Type)	See 2.15.1.28
cryptProvider (Cryptographic Provider)	See 2.15.1.28
cryptProviderType (Cryptographic Provider Type)	See 2.15.1.28
cryptProviderTypeExt (Cryptographic Provider Type Extensibility)	See 2.15.1.28
cryptProviderTypeExt Source (Provider Type Extensibility Source)	See 2.15.1.28
cryptSpinCount (Iterations to Run Hashing Algorithm)	See 2.15.1.28
hash (Password Hash)	See 2.15.1.28
recommended (Recommend Write Protection in User Interface)	Specifies that applications should provide user interface recommending that the user open this document in write protected state. If the user chooses to do so, the document shall be write protected, otherwise, it shall be opened fully editable. If this attribute is omitted, then user interface recommending that the user open this document in

Attributes	Description
	<p>write protected state should not be provided. If the password attribute is also specified, then this setting shall be ignored.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
salt (Salt for Password Verifier)	See 2.15.1.28

2.15.1.95 zoom (Magnification Setting)

This element specifies the magnification level which should be applied to a document when it is displayed by an application. The zoom level is specified with the use of two attributes stored on this element:

- **val**, which stores the type of zoom applied to the document
- **percent**, which stores the zoom percentage to be used when rendering the document

If both attributes are present, then the percent attribute shall be treated as a 'cached' value and only used when the value none is specified for the **val** attribute.

If this element is omitted, then applications may display the document in any desired magnification setting.

Attributes	Description
percent (Zoom Percentage)	<p>Specifies the zoom percentage that should be applied when a given WordprocessingML document is rendered by conforming hosting applications. This value is the zoom percentage specified as an integer whose units correspond to the zoom percentage.</p> <p>If this attribute is omitted, then applications may use any desired default percentage for the magnification.</p> <p>If the val attribute instantiated in addition to the percent attribute, then the percent attribute shall be treated as a cached value and only used when the value none is specified for the val attribute. If the value specified exceeds the maximum zoom level available in a conforming hosting application, the conforming hosting application shall display the document using its maximum zoom level. Correspondingly, if the value specified is less than the minimum zoom level available in the conforming hosting application, the conforming hosting application shall display the document using its minimum zoom level.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>
val (Zoom Type)	<p>Specifies the type of zoom which shall be applied to a given document on open.</p> <p>If this attribute is not present, then the document shall be displayed as though the value had been set to none, and should rely on the value of the percent attribute for the actual zoom percentage.</p> <p>The possible values for this attribute are defined by the ST_Zoom simple type (§2.18.115).</p>

2.15.2 Web Page Settings

The next group of settings stored in WordprocessingML is web page settings. These settings specify two categories of settings:

- Settings which are related to HTML documents (i.e. frameset definitions) that may be used in WordprocessingML documents as well
- All settings which affect how this document shall be handled when it is saved as HTML. Actually saving a document as HTML is outside of the scope of this Office Open XML Standard, but in order to ensure the maximum

interoperability between a WordprocessingML document and an HTML document, settings not explicitly stored elsewhere are stored in these settings.

2.15.2.1 **allowPNG** (Allow PNG as Graphic Format)

This element specifies that applications shall allow use of the PNG file format when the contents of this WordprocessingML document are saved as a web page. This includes all supporting images used as part of this HTML web page.

If this element is omitted from the document, then the PNG file format shall not be allowed when this document is saved as a web page, and that another suitable file format (such as the JPEG file format) should be utilized in its place.

[*Note*: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support the reading of PNG images. However, although PNG utilizes a lossless compression algorithm, JPEG uses 'lossy' compression and may in some cases result in lower fidelity images. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.2 **blockquote** (Data for HTML blockquote Element)

This element specifies that the current **div** element does not represent an HTML **div** element, but rather represents an HTML **blockquote** element. This element shall specify that this container shall be written out using the **blockquote** element if this document is subsequently saved as HTML.

If this element is omitted, then the current **div** element does not represent an HTML **blockquote** element. If both this element and the **bodyDiv** element (§2.15.2.3) are specified, then this element shall take precedence in all cases.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.3 **bodyDiv** (Data for HTML body Element)

This element specifies that the current **div** element does not represent an HTML **div** element, but rather represents formatting properties on the HTML **body** element. This element shall specify that the properties specified by this container shall be written out onto the **body** element if this document is subsequently saved as HTML.

If this element is omitted, then the current **div** element does not represent an HTML **body** element. If both this element and the **blockquote** element (§2.15.2.2) are specified, then this element shall be ignored. If this element is specified on any **div** which is not the main **div** element for the document, then this element shall be ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.4 **bottom** (Bottom Border for HTML div)

This element specifies the border which shall be displayed at the bottom of the boundaries of the current HTML **div** object. If this element is omitted, then this HTML **div** object shall not have a bottom border.

Attributes	Description
See 2.3.1.4	

2.15.2.5 **color** (Frameset Splitter Color)

This element specifies the color of the splitters within the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and may be ignored for all nested framesets in this document.

If this element is omitted, then the default color of the splitter may be automatically determined by the application displaying this WordprocessingML document (equivalent to a **val** attribute value of **auto**).

Attributes	Description
See 2.3.2.5	

2.15.2.6 **div** (Information About Single HTML div Element)

This element specifies information about a single HTML **div**, **body**, or **blockquote** element which was included in this document, so that that information (which is stored on a logical structure with no direct analog in WordprocessingML) may be maintained when an HTML document is stored in the WordprocessingML format.

The **div** element stores the following information about these structures:

- The child HTML **div**, and **blockquote** elements
- The borders for the element
- The margins for the element

When the resulting WordprocessingML document is displayed by an application, the settings specified by this information shall be reflected in the formatting of the resulting paragraphs (i.e. this information shall not only be used when the document is resaved in the HTML format).

Attributes	Description
id (div Data ID)	<p>Specifies a unique decimal number which shall be used to associate one or more structures in the WordprocessingML content with this HTML div information.</p> <p>When a WordprocessingML structure (a paragraph or a table row) is associated with div information, it shall be associated with the set of information which most immediately contains the current object.</p> <p>The ID specified by this attribute is then referenced by the divId element for all structures which are immediately contained within the specified HTML div.</p> <p>The possible values for this attribute are defined by the ST_DecimalNumber simple type (§2.18.15).</p>

2.15.2.7 **divBdr** (Set of Borders for HTML div)

This element specifies the set of borders for the boundaries of the current HTML **div**, **body**, or **blockquote** element, using the four border types defined by its child elements.

If this element is omitted, then there shall be no borders associated with the current HTML **div**, **body**, or **blockquote** element.

2.15.2.8 **divs** (Information about HTML div Elements)

This element specifies all information about the set of HTML **div** elements (as well as the **body** and **blockquote** elements) which were included in this document, so that that information (which is stored on a logical structure with no direct analog in WordprocessingML) may be maintained when an HTML document is stored in the WordprocessingML format.

The **divs** element stores the following information about these structures:

- The parent/child structure of HTML **div**, **blockquote**, and **body** elements
- The borders for each of these elements
- The margins for each of these elements

When the resulting WordprocessingML document is displayed by an application, the settings specified by this information shall be reflected in the formatting of the resulting paragraphs (i.e. this information shall not only be used when the document is resaved in the HTML format).

2.15.2.9 **divsChild** (Child div Elements Contained within Current div)

This element specifies the set of HTML **div** or **blockquote** elements which are contained within the current HTML **div**, **body**, or **blockquote** element, establishing the parent/child hierarchy of the original set of these elements.

When an HTML document containing these objects is saved in the WordprocessingML format, WordprocessingML objects store a reference to their most immediate parent **div**, **body**, or **blockquote** element using the **divId** element. However, since only a single reference is stored, this information is often insufficient to determine the appropriate parent/child hierarchy for the original HTML **div** data, so it can be applied appropriately. This element allows that hierarchy to be stored, as child HTML **div** elements are stored within the **childDivs** element.

2.15.2.10 **doNotOrganizeInFolder** (Do Not Place Supporting Files in Subdirectory)

This element specifies that applications shall not automatically place all supporting files (images which are part of this HTML web page, etc.) in a subdirectory when the contents of this WordprocessingML document are saved as a web page. Typically, applications which save a document as a web page consisting of multiple files save all supporting files in a subdirectory next to the main HTML file (in order to keep those files organized). This element specifies the files shall be placed in the same directory as the actual web page.

If this element is omitted from the document, then all supporting files should be saved into a subdirectory beneath the main web page file when this document is saved as a web page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.11 **doNotRelyOnCSS** (Do Not Rely on CSS for Font Face Formatting)

This element specifies whether applications may rely on the CSS properties for font face (the **font-family** property) when saving this WordprocessingML document as a web page. If this element is utilized, then the HTML **font** element should be used either in place of or in concert with these CSS properties in order to specify the font face formatting for the resulting web page.

If this element is omitted, then applications may choose to rely on the CSS properties for font face as desired.

[*Note*: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support the reading of these CSS properties when attempting to read and display the resulting web page, in order to maximize the fidelity of the resulting output. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.12 **doNotSaveAsSingleFile** (Recommend Web Page Format over Single File Web Page Format)

This element specifies that applications should recommend that new web page files generated using this WordprocessingML document use a multi-file web page format (HTML), rather than a single-file web page format (MHTML) when this document is saved as an HTML web page. This setting shall not prevent the use of the MHTML format; it shall only cause applications to recommend (via a default) a non single-file format when saving as a web page.

[*Note*: This setting is primarily intended for applications which explicitly support a "Save as Web Page..." action, in order to determine the default setting for the resulting web page. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.13 **doNotUseLongFileNames** (Do Not Use File Names Longer than 8.3 Characters)

This element specifies that applications shall ensure that the file names for all files generated when saving this document as a web page do not exceed eight characters with a three character extension. This includes all supporting files (images which are part of this HTML web page, etc.).

[*Note*: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support the reading of long file names when attempting to read and display the resulting web page. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.14 **encoding** (Output Encoding When Saving as Web Page)

This element specifies the encoding which shall be used for the contents of this WordprocessingML document when it is saved as an HTML web page. The set of encodings supported by this element shall be derived from the standard set of character set definitions provided at <http://www.iana.org/assignments/character-sets>.

If this element is omitted, then the default encoding for the current system shall be used when this document is saved as a web page. If the value of the **val** attribute is unknown or supported by an application, then the default encoding for the current system shall be used when this document is saved as a web page.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.2.15 **flatBorders** (Frameset Splitter Border Style)

This element specifies the 3D style of the splitters within the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and may be ignored for all nested framesets in this document. When this property is turned on, the borders for this frameset shall be flat (not 3D), otherwise they may be presented as 3D splitter when they are displayed.

If this element is omitted, then the default style of the splitter should be a 3D splitter.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.16 **frame** (Single Frame Properties)

This element specifies the properties for a single frame within a frameset document. When a document defines a frameset using the **frameset** element; that frameset is composed of a set of frames, each of which is specified by a single **frame** element.

2.15.2.17 **frameLayout** (Frameset Layout)

This element specifies the order in which the frames (and nested framesets) in a frameset shall be displayed. When a frameset is created, it can only contain frames which are stacked in one direction:

- Vertically (one on top of another)
- Horizontally (one next to another)

This element specifies how the frames in this frameset are stacked, which shall also be used to interpret the sizes defined by the **sz** element (§2.15.2.39) for each frame. In order to determine the ordering of the constituent frames within this frameset, the ordering of the child **frame** and **frameset** elements shall be used.

If this element is omitted, then the frames in this frameset shall be stacked vertically on top of one another (a row frameset).

Attributes	Description
val (Frameset Layout Value)	Specifies the type of layout which shall be used to display the contents of the frames and nested framesets within this frameset, as defined by the simple type referenced below. The possible values for this attribute are defined by the ST_FrameLayout simple type (§2.18.34).

2.15.2.18 **frameset** (Root Frameset Definition)

This element specifies that this document is the container for a frameset. This WordprocessingML element is analogous to the **frameset** element in HTML.

When the **frameset** element is present within a document, that document shall serve as a frameset definition only; all of its normal document content shall therefore not be displayed as long as it contains at least one child **frame** or **frameset** element.

If this element is omitted, then the currently document shall not be treated as a frameset definition; its regular document content shall be displayed.

2.15.2.19 **frameset** (Nested Frameset Definition)

This element specifies a frameset which has been nested within another frameset within a WordprocessingML document. This WordprocessingML element is analogous to the **frameset** element in HTML (when that frameset is the child of another **frameset** element).

2.15.2.20 **framesetSplitbar** (Frameset Splitter Properties)

This element specifies the properties for the splitters associated with this frameset. A *splitter* is a horizontal or vertical line which visually separates the contents of one frame from another within a frameset.

If this element is omitted, then the default parameters for each of the child frameset properties shall be used for all splitters in this frameset.

2.15.2.21 **left** (Left Border for HTML div)

This element specifies the border which shall be displayed at the left of the boundaries of the current HTML **div** object.

If this element is omitted, then this HTML **div** object shall not have a left border.

Attributes	Description
See 2.3.1.4	

2.15.2.22 **linkedToFile** (Maintain Link to Existing File)

This element specifies that the file referenced by the **sourceFileName** element (§2.15.2.38) as the basis for the current frame shall not be changed, even when the file defined by the parent frameset is moved - i.e. the link shall remain exactly as specified.

[*Guidance:* Typically, when a document is incorporated into a frameset, a copy of that document is made such that all files encompassing the frameset are stored in a single subdirectory (so they can be moved as a single unit). However, if the link to the current file is absolute and shall not be changed even when the location of the main frameset document, then this element shall be set to indicate that setting. *end guidance*]

If this element is omitted, then a new file may be created as necessary when the parent frameset document is resaved to another location.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.23 **marBottom** (Bottom Margin for HTML div)

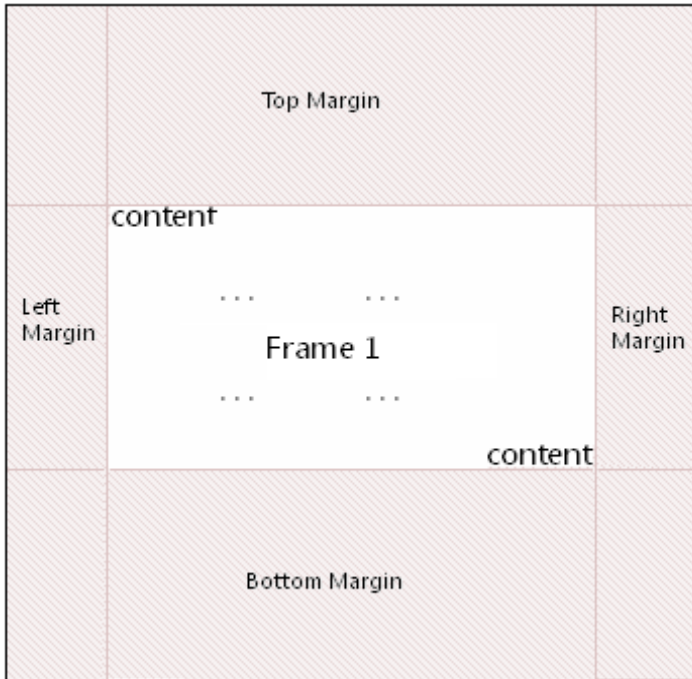
This element specifies the margin which shall be displayed at the bottom of the boundaries of the current HTML **div** object.

If this element is omitted, then this HTML **div** object shall not have a bottom margin.

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	See 2.3.2.33

2.15.2.24 **marH** (Top and Bottom Margin for Frame)

This element specifies the top and bottom margin height for a single frame in a frameset document, as follows:



This height is expressed in pixels.

If this element is omitted, then no top or bottom margin shall be used for this frame.

Attributes	Description
val (Measurement in Pixels)	Specifies a value whose contents shall contain a positive whole number, whose contents consist of a positive measurement in pixels. The contents of this measurement shall be interpreted based on the context of the parent XML element. The possible values for this attribute are defined by the ST_PixelsMeasure simple type (§2.18.73).

2.15.2.25 **marLeft** (Left Margin for HTML div)

This element specifies the margin which shall be displayed at the left of the boundaries of the current HTML div object.

If this element is omitted, then this HTML div object shall not have a left margin.

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	See 2.3.2.33

2.15.2.26 **marRight** (Right Margin for HTML div)

This element specifies the margin which shall be displayed at the right of the boundaries of the current HTML div object.

If this element is omitted, then this HTML div object shall not have a right margin.

Attributes	Description
val (Positive or	See 2.3.2.33

Attributes	Description
Negative Value in Twentieths of a Point)	

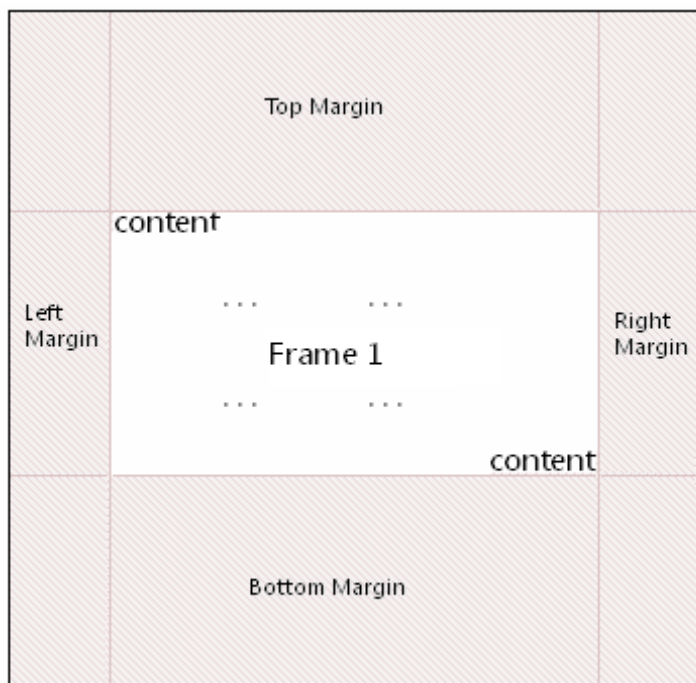
2.15.2.27 **marTop** (Top Margin for HTML div)

This element specifies the margin which shall be displayed at the top of the boundaries of the current HTML div object. If this element is omitted, then this HTML div object shall not have a top margin.

Attributes	Description
val (Positive or Negative Value in Twentieths of a Point)	See 2.3.2.33

2.15.2.28 **marW** (Left and Right Margin for Frame)

This element specifies the left and right margin height for a single frame in a frameset document, as follows:



This height is expressed in pixels.

If this element is omitted, then no left or right margin shall be used for this frame.

Attributes	Description
val (Measurement in Pixels)	See 2.15.2.24

2.15.2.29 **name** (Frame Name)

This element specifies the name of a single frame within a frameset document. This property is analogous to the name attribute on the frame element in HTML.

[Note: The name of a frame may be used in web pages that reference a frame via targeted links, etc. *end note*]

If this element is omitted, then the current frame shall have no name associated with it.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.2.30 **noBorder** (Do Not Display Frameset Splitters)

This element specifies whether the splitters shall be displayed for the contents of the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and may be ignored for all nested framesets in this document. If this element is present, then no splitters shall be displayed, and all other frameset splitter properties may be ignored.

If this element is omitted, then the splitters in this document shall be displayed as defined by the **w** and **color** elements.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.31 **noResizeAllowed** (Frame Cannot Be Resized)

This element specifies whether or not the size of the current frame shall be modifiable (i.e. whether the frame can be resized) when the contents of this document are saved as HTML and displayed in a web browser. When this element is set, the size of the frame shall be set to its current values. This property is analogous to the **noresize** attribute on the **frame** element in HTML.

If this element is omitted, the size of the frame shall be modifiable (the frame may be resized when it is displayed).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.32 **optimizeForBrowser** (Disable Features Not Supported by Target Web Browser)

This element specifies whether applications should attempt to detect the target web browser for any web page produced from this document, and subsequently disable all user interface and output which is not supported by that target web browser.

The target web browser can be determined by the state of the following elements:

- **allowPNG**
- **doNotRelyOnCSS**
- **relyOnVML**
- **doNotSaveWebPagesAsSingleFile**

The following table determines how this determination is made:

Settings	Target Browser
allowPNG is off doNotRelyOnCSS is on relyOnVML is off doNotSaveWebPagesAsSingleFile is on	Microsoft Internet Explorer 3.0 or later Netscape Navigator 3.0 or later
allowPNG is off doNotRelyOnCSS is off relyOnVML is off doNotSaveWebPagesAsSingleFile is on	Netscape Navigator 4.0 or later
allowPNG is off doNotRelyOnCSS is off relyOnVML is off doNotSaveWebPagesAsSingleFile is off	Microsoft Internet Explorer 4.0 or later
allowPNG is off doNotRelyOnCSS is off	Microsoft Internet Explorer 5.0 or later

Settings	Target Browser
relyOnVML is on doNotSaveWebPagesAsSingleFile is off	
allowPNG is on doNotRelyOnCSS is off relyOnVML is on doNotSaveWebPagesAsSingleFile is off	Microsoft Internet Explorer 6.0 or later

If this element is omitted, then no user interface or output which is not supported by that target web browser shall be disabled.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.33 **pixelsPerInch** (Pixels per Inch for Graphics/Images)

This element specifies the number of pixels per inch (or density) that will be used for the display of pictures or table cells when a WordprocessingML document is saved as a web page. The size that is specified by this element affects the size of the pictures or table cells relative to the size of text in the document. The pixels per inch (ppi) measurement is relative to the screen resolution, and the resulting physical dimensions of the resulting image or cell in pixels (which are used in web pages, but not for printed documents) are the result of the original dimensions (in inches) multiplied by the number of pixels per inch.

The range of values for this element is typically from 19 to 480 pixels per inch. The common settings for popular screen sizes are 72, 96, and 120 pixels per inch.

If this element is omitted, then a default size of 96 pixels per inch shall be used when determining the number of pixels for images and/or table cells within this document.

[Note: This setting is typically only specified if the target screen resolution for the web page is known, as defined by the **targetScreenSz** element (§2.15.2.41) to set the optimum screen size for the web page. *end note*]

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.15.2.34 **relyOnVML** (Utilize VML When Saving as Web Page)

This element specifies whether applications may utilize the Vector Markup Language format when saving the content of this WordprocessingML document as a web page, when graphical elements which can leverage this format are present in the document.

If this element is omitted, then a graphic image format should be used either in place of or in concert with the Vector Markup Language output in order to specify the formatting and positioning for objects which are part of the resulting web page.

[Note: This setting is intended for applications to save web pages which can be supported by legacy web browsers which do not support Vector Markup Language when attempting to read and display the resulting web page. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.35 **right** (Right Border for HTML div)

This element specifies the border which shall be displayed at the right of the boundaries of the current HTML div object.

If this element is omitted, then this HTML div object shall not have a right border.

Attributes	Description
See 2.3.1.4	

2.15.2.36 **saveSmartTagsAsXml** (Save Smart Tag Data in XML Property Bag)

This element specifies that the information pertaining to all smart tags () in the current document shall be saved into a separate XML-based property bag at the head of the web page when this WordprocessingML document is saved as a web page.

If this element is omitted, then the smart tag data of this document shall not be saved into a separate XML-compliant property bag within the HTML output when this document is saved as a web page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.2.37 **scrollbar** (Scrollbar Display Option)

This element specifies when a scrollbar shall be visible for the contents of the current frame. When this element is set, the val attribute determines exactly when the scrollbar shall be visible. This property is analogous to the scrolling attribute on the frame element in HTML.

If this element is omitted, the scrollbar shall only be displayed when the contents of the frame exceed the visible space for the frame (i.e. when the scrollbar is needed to display all of the content).

Attributes	Description
val (Scrollbar Display Option Value)	Specifies the criteria under which a scrollbar shall be displayed along with the contents of this frameset, as defined by the simple type referenced below. The possible values for this attribute are defined by the ST_FrameScrollbar simple type (§2.18.35).

2.15.2.38 **sourceFileName** (Source File for Frame)

This element specifies the ID for the relationship which specifies the source file for a single frame within a frameset document.

The relationship referenced by this element's id attribute shall exist in the relationship part item for the Web Settings part, or this document shall be considered non-conformant. Also, the type of the relationship referenced by this element's id attribute shall be <http://schemas.openxmlformats.org/officeDocument/2006/relationships/frame>, or this document shall be considered non-conformant.

If this element is omitted, then no source file is present for the current frame, and one may be created dynamically as needed to display content within the frame.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.15.2.39 **sz** (Frame Size)

This element specifies the size for a single frame within a frameset.

This size shall be interpreted based on the contents of the **frameLayout** element (§2.15.2.17) for the parent frameset, as follows:

- If the val attribute on that element is **cols**, then this element specifies the width of the frame
- If the val attribute on that element is **rows**, then this element specifies the height of the frame

Once the axis of this measurement has been established using the criteria above, the actual value of the measurement shall be determined by the following:

- If the val attribute ends in an asterisk (*), then this measurement is a relative measurement (relative to all other frames in this frameset).
- If the val attribute ends in a percentage symbol (%), then this measurement is a percentage of the height and/or width of the parent window, respectively.
- Otherwise, the value of the val attribute specifies the size of the frame in pixels. This measurement shall be interpreted in the context of the **pixelsPerInch** element (§2.15.2.33) to determine the width of the resulting measurement in inches.

If this element is omitted, then no information shall be implied about the size of the current frame.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.2.40 **sz** (Nested Frameset Size)

This element specifies the size for a frameset that has been nested within another frameset. If this size appears on a root frameset, then it may be ignored and the main frameset shall encompass the entire window.

This size shall be interpreted based on the contents of the **frameLayout** element (§2.15.2.17) for the parent frameset (not the current nested frameset), as follows:

- If the val attribute on that element is **cols**, then this element specifies the width of the frameset
- If the val attribute on that element is **rows**, then this element specifies the height of the frameset

Once the axis of this measurement has been established using the criteria above, the actual value of the measurement shall be determined by the following:

- If the val attribute ends in an asterisk (*), then this measurement is a relative measurement (relative to all other frames in this frameset).
- If the val attribute ends in a percentage symbol (%), then this measurement is a percentage of the height and/or width of the parent frameset, respectively.
- Otherwise, the value of the val attribute specifies the size of the frameset in pixels. This measurement shall be interpreted in the context of the **pixelsPerInch** element (§2.15.2.33) to determine the width of the resulting measurement in inches.

If this element is omitted, then no information shall be implied about the size of the current frameset.

Attributes	Description
val (String Value)	See 2.3.1.27

2.15.2.41 **targetScreenSz** (Target Screen Size for Web Page)

This element specifies the ideal minimum target screen size (width by height, specified in pixels) on which web pages generated when saving this document will be displayed. This setting may be used to optimize the output of web pages produced from this document.

If this element is omitted, then the target screen size for web pages produced from this document shall be assumed to be 800x600.

Attributes	Description
val (Target Screen Size Value)	Specifies the target screen size for web pages produced by this document, as defined by the simple type referenced below. The possible values for this attribute are defined by the ST_TargetScreenSz simple type (§2.18.92).

2.15.2.42 **top** (Top Border for HTML div)

This element specifies the border which shall be displayed at the top of the boundaries of the current HTML div object. If this element is omitted, then this HTML div object shall not have a top border.

Attributes	Description
See 2.3.1.4	

2.15.2.43 **w** (Frameset Splitter Width)

This element specifies the width of the splitters within the frameset in this WordprocessingML document. This element shall only be honored on the root frameset for this document, and may be ignored for all nested framesets in this document.

If this element is omitted, then the default width of the splitters in this document shall be 4.5 points (90 twentieths of a point) wide. If the **noBorder** element (§2.15.2.30) is also specified, then this element shall be ignored and no splitters shall be displayed.

Attributes	Description
val (Measurement in Twentieths of a Point)	See 2.15.1.24

2.15.2.44 **webSettings** (Web Page Settings)

This element specifies the set of web page settings that have been specified for a single WordprocessingML document. This element is the root element for the Web Settings part within a WordprocessingML document.

2.15.3 Compatibility Settings

The last group of settings stored in WordprocessingML is compatibility settings. *Compatibility Settings* are optional settings used to preserve visual fidelity of documents created in earlier word processing applications. Some of these settings provide affordance for specific behaviors, described in detail below; and others simply instruct applications to mimic the behavior of an existing word processing application.

If compatibility settings are needed, they are stored in the Document Settings part.

It is important to note that all compatibility settings are optional in nature - applications may freely ignore all behaviors described within this section and these settings should not be added unless compatibility is specifically needed in one or more cases. The compatibility settings are provided for backward compatibility with documents created in legacy applications. As such, a number of the settings reference specific applications and specific versions of those applications. This is solely for backward compatibility reasons, and any of those settings are ignorable.

2.15.3.1 **adjustLineHeightInTable** (Add Document Grid Line Pitch To Lines in Table Cells)

This element specifies whether a document grid defined using the **docGrid** element (§2.6.5) that specifies a line grid (manually adding additional pitch to each line in the section) shall also be applied to lines within table cells in this section. Typically, when additional line pitch is added to all lines in a section via the document grid, it is not applied to text in tables. This element, when present with a val attribute value of true (or equivalent), specifies that additional line pitch shall be added to lines in table cells.

Attributes	Description
------------	-------------

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.2 **alignTablesRowByRow** (Align Table Rows Independently)

This element specifies whether applications shall align each row within a table independently based on the alignment setting of the **jc** element (§2.4.22) when displaying the contents of a table in a WordprocessingML document. When the justification of a table using the **jc** element is typically applied, that alignment is applied to the contents of the table (the table is centered, left justified, or right-aligned), and then individual rows are laid out based on the resulting table's position. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that each table row shall be independently aligned based on the table alignment setting, ignoring the placement of all other rows.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.3 **allowSpaceOfSameStyleInTable** (Allow Contextual Spacing of Paragraphs in Tables)

This element specifies whether the suppression of additional space (contextual spacing) defined using the **contextualSpacing** element (§2.3.1.9) shall be applied to paragraphs contained within tables. Typically, the rules for the removal of additional paragraph spacing via the **contextualSpacing** element are applied to all paragraphs in a WordprocessingML document. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that this setting shall always be ignored for paragraphs in table cells (and additional spacing shall be allowed).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.4 **applyBreakingRules** (Use Legacy Ethiopic and Amharic Line Breaking Rules)

This element specifies whether applications shall use a legacy set of line breaking rules when determining line breaks for text consisting of Ethiopic and/or Amharic characters. Typically, when line breaking this text, applications should allow line breaks to occur after a character between the UTF-16 (hexadecimal) values 0x1361 and 0x1368 when those characters appear in the document's content. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that when a line break would occur after a character between the UTF-16 hexadecimal values 0x1361 and 0x1368, the line break shall occur before all instances of these characters (i.e. no break opportunity shall be afforded after a character in this range).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.5 **autofitToFirstFixedWidthCell** (Allow Table Columns To Exceed Preferred Widths of Constituent Cells)

This element specifies that when performing an AutoFit on a table in a WordprocessingML document in order to display it, applications shall alter that logic slightly in order to mimic the behavior of a previous word processing application. Normally, the AutoFit behavior of a table is as is described in the associated simple type. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that this logic shall be changed as follows:

- If the width of a grid column in a table has been set by a preferred table cell width, then that column's width may be enlarged by the content of cells which themselves do not have a preferred width (in contrast, the normal logic never allows the content of cells to override a preferred width on a grid column).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.6 **autoSpaceLikeWord95** (Emulate Word 95 Full-Width Character Spacing)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 95) when determining the spacing between full-width East Asian characters in a document's content.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.7 **balanceSingleByteDoubleByteWidth** (Balance Single Byte and Double Byte Characters)

This element specifies whether applications shall balance the width of Single Byte Character Set characters and Double Byte Character Set characters when rendering WordprocessingML documents. Specifically, this element specifies to adjust the fixed pitch fonts' half-width space character and full-width space character to attain a 1 to 2 ratio.

[*Note:* This element is used with East Asian content. Layout and line breaking for East Asian text is dependent on the character width. Half width characters (or Hankaku characters) are one half of an em wide, and full width characters (or Zenkaku characters) are one em wide. Legacy encoding often used a single byte to encode half-width characters and two bytes to encode full width characters. *end note*]

Typically, no adjustment is done on any character when it is displayed as part of a WordprocessingML document. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that character sizes shall be adjusted as needed to meet the 1:2 ratio described above.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.8 **cachedColBalance** (Use Cached Paragraph Information for Column Balancing)

This element specifies whether applications shall incorrectly calculate the height of a paragraph for the purposes of column balancing when rendering WordprocessingML documents. Specifically, this element specifies that when a paragraph's lines have differing heights, an application shall treat this paragraph as though it had only one line equaling the full paragraph height, regardless of the actual number of lines in the paragraph.

[*Guidance:* It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from a legacy application. *end guidance*]

Typically, lines are correctly measured for their height when balancing columns as part of a WordprocessingML document. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall perform the incorrect calculation in the conditions described above.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.9 **compat** (Compatibility Settings)

This element specifies a set of optional compatibility options for the current document.

All settings in this section are optional, but some are very commonly used by different languages, and those which are typically used are as follows:

For Thai, Lao, Khmer, Tibetan, and Armenian:

- The **applyBreakingRules** setting (§2.15.3.4)

For East Asian languages:

- The **adjustLineHeightInTable** setting (§2.15.3.1)
- The **balanceSingleByteDoubleByteWidth** setting (§2.15.3.7)
- The **doNotExpandShiftReturn** setting (§2.15.3.15)
- The **doNotLeaveBackslashAlone** setting (§2.15.3.16)
- The **spaceForUL** setting (§2.15.3.43)
- The **ulTrailSpace** setting (§2.15.3.55)
-

2.15.3.10 **convMailMergeEsc** (Treat Backslash Quotation Delimiter as Two Quotation Marks)

This element specifies whether applications should perform a conversion of the contents of a mail merge data source when reading those contents in order to perform a mail merge operation with their contents.

Typically, the contents of a mail merge data source are read in exactly as specified when performing a mail merge with the contents of a data source. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall interpret delimiters composed of a backslash and quotation mark (\"") as two quotation marks (""), within external data sources to be connected to via a mail merge.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.11 **displayHangulFixedWidth** (Always Use Fixed Width for Hangul Characters)

This element specifies whether applications should assume that all characters in the Hangul Syllables Unicode sub range (character values between 0xAC00 and 0xD7FF) are of a single fixed width or shall use the characters widths defined by the font in use (typical for a proportional width font).

Typically, applications shall retrieve the character width for any character in a document from the associated font, allowing each character to be of its own width (a proportional width character). This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall instead assume a single fixed width for all characters in the Hangul Syllables sub range, by reading the width of Unicode character 0x4E00 from the associated font and using that width for all Hangul characters (or, if that character is not present, the next available character in the font).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.12 **doNotAutofitConstrainedTables** (Do Not AutoFit Tables To Fit Next To Wrapped Objects)

This element specifies whether applications shall allow tables to be resized to the remaining available line width when they are using the AutoFit algorithm and part of that line is filled by a shape with a wrapping type with a value of **square** or **tight**.

Typically, a table which is AutoFit and has a preferred width shall have its width reduced in order to allow a floating shape to wrap around its contents within the document, as that shape simply reduces the width of the line and the AutoFit algorithm applies to the remaining line width. This element, when present with a **val** attribute value of **true** (or

equivalent), specifies that tables shall never have any preferred width overridden to allow them to wrap around that floating object, and shall instead be pushed to the next full width line in the document to be displayed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.13 **doNotBreakConstrainedForcedTable** (Don't Break Table Rows Around Floating Tables)

This element specifies whether applications shall allow a table row to be split in two when its contents are displayed under the following circumstances:

- The table row exceeds one page in height (it must be split into two pages)
- The table row would need to be split in order to accommodate a floating table also on the page (tables which have been set to floating using the **tblpPr** element (§2.4.54))

Typically, assuming the **cantSplit** property (§2.4.6) is not set, a table row which cannot fit on one single page shall be split as needed around any floating table on a page, in order to allow its contents to be fully displayed across two or more pages. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that table rows which exceed one page in height shall never be split around floating tables in the document, and shall instead be displayed on the first page below the floating table, even if that means that part of the table row is clipped by the edge of the page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.14 **doNotBreakWrappedTables** (Do Not Allow Floating Tables To Break Across Pages)

This element specifies whether applications shall allow tables which have been set to floating using the **tblpPr** element (§2.4.54) shall be allowed to break across multiple pages when needed.

Typically, a table whose contents cannot all be displayed on one page is broken as needed across multiple pages in order to preserve the location of the table (just as a paragraph of multiple lines is broken across pages as needed). This element, when present with a **val** attribute value of **true** (or equivalent), specifies that floating tables shall never be broken across pages, and shall instead be put on the first page by adjusting the starting position of the table as needed to fit on that single page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.15 **doNotExpandShiftReturn** (Don't Justify Lines Ending in Soft Line Break)

This element specifies whether applications should fully justify the contents of incomplete lines which end in a soft line break when the parent paragraph is fully justified using the **jc** element (§2.3.1.13).

Typically, applications shall fully justify all lines in a paragraph when that setting is specified using the **jc** element except for the last line in the paragraph (the line ending with the paragraph mark). This element, when present with a **val** attribute value of **true** (or equivalent), specifies that any line which ends in a soft line break shall also not be fully justified when the paragraph specifies that setting.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.16 **doNotLeaveBackslashAlone** (Convert Backslash To Yen Sign When Entered)

This element specifies whether applications should automatically convert the backslash character into the yen character when it is added through user keyboard input.

Typically, no automatic conversion of one character to another is done when characters are entered by the user. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that all entries of the backslash (\) character shall automatically be converted to a yen symbol (¥) when the former is entered.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.17 **doNotSnapToGridInCell** (Do Not Snap to Document Grid in Table Cells with Objects)

This element specifies whether a document grid defined using the **docGrid** element (§2.6.5) shall be applied to the contents of table cells in that section which also contain floating objects defined using the Vector Markup Language syntax. Note that the floating object must be part of the cell, and simply not displayed over the cell due to its anchoring relative to another part of the document.

Typically, if a floating object is present in a table cell, then that setting shall have no impact on whether East Asian text in that cell is snapped to the document grid (as text is always snapped to the grid). This element, when present with a `val` attribute value of `true` (or equivalent), specifies that whenever a floating object defined using VML is present in a table cell, that the cell's contents shall not be snapped to the document grid.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.18 **doNotSuppressIndentation** (Do Not Ignore Floating Objects When Calculating Paragraph Indentation)

This element specifies whether applications should ignore the presence of floating objects when calculating the starting position of paragraphs which are wrapped around floating objects defined using the Vector Markup Language (VML) syntax. Typically, the presence of a floating object on the same line or lines as a paragraph shall only affect the text when the floating object occurs where that text would normally be presented. .

This element, when present with a `val` attribute value of `true` (or equivalent), specifies that floating objects shall always impact paragraphs on the same line in two ways:

- If the paragraph is not numbered, then it shall tightly wrap any floating object which precedes it on the same line, ignoring its own indentation settings.
- If the paragraph is numbered using the **numPr** element (§2.3.1.19), then it shall calculate and use its full indent relative to the edge of the floating object, not relative to the edge of the page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.19 **doNotSuppressParagraphBorders** (Do Not Suppress Paragraph Borders Next To Frames)

This element specifies whether applications should suppress paragraph borders defined using the **pBdr** element (§2.3.1.24) when those borders would be displayed next to the contents of paragraphs which have been defined as frames using the **framePr** element (§2.3.1.11).

Typically, when a paragraph's borders appear next to a frame, those borders are suppressed to avoid having two borders in close proximity. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that those borders shall not be suppressed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.20 **doNotUseEastAsianBreakRules** (Do Not Compress Compressible Characters When Using Document Grid)

This element specifies whether applications should compress characters with identical compression rules when the document grid has been defined using the **docGrid** element (§2.6.5). *Compression rules* refer to the additional bearing on the left and/or right side of a typical character, which can be compressed as needed without modifying the actual width of the character (its breadth).

Typically, punctuation characters with an identical set of compression rules are compressed when the contents of a document are displayed. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that if a document grid is defined for the current section, compression shall never be performed on any character - all compressible characters shall be individually snapped to the document grid.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.21 **doNotUseHTMLParagraphAutoSpacing** (Use Fixed Paragraph Spacing for HTML Auto Setting)

This element specifies whether applications should use a fixed definition when interpreting automatic paragraph spacing defined by a value of **true** (or equivalent) on the **beforeAutospacing** and/or **afterAutospacing** attributes on the **spacing** element (§2.3.1.33).

Typically, applications shall interpret these settings to match the behavior of most HTML user agents, mimicking the default spacing above and below an HTML **p** element without additional spacing information. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that those two attributes shall result in the following settings for each value:

- **beforeAutospacing** = 5 points of spacing before
- **afterAutospacing** = 10 points of spacing after

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.22 **doNotUseIndentAsNumberingTabStop** (Ignore Hanging Indent When Creating Tab Stop After Numbering)

This element specifies whether applications shall use the custom tab stop generated by the hanging indent (if any) when advancing the text after the numbering for a numbered paragraph.

Typically, a hanging indent on a paragraph creates a virtual custom tab stop at that location, and therefore a tab added after the numbering on a numbered paragraph by the **suff** element (§2.9.30) shall advance to that tab stop, so that the text of the numbered paragraph begins at that location. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that a tab stop added as the suffix to the numbering of a numbered paragraph shall ignore that virtual custom tab stop and shall instead advance to the next real tab stop (custom or automatic) on the current line.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.23 **doNotVertAlignCellWithSp** (Don't Vertically Align Cells Containing Floating Objects)

This element specifies whether applications shall vertically align the contents of a table cell, even when the contents of that table cell include one or more floating objects defined using the Vector Markup Language syntax. Note that the floating object must be part of the cell, and simply not displayed over the cell due to its anchoring relative to another part of the document.

Typically, if the alignment of a table cell in a WordprocessingML document is specified, then the entire contents of that cell are aligned as specified. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that whenever a floating object defined using VML is present in a table cell, that no vertical alignment shall be applied to the contents of that cell, and the contents of the cell shall instead always be top aligned to the cell's contents.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.24 **doNotVertAlignInTxbx** (Ignore Vertical Alignment in Textboxes)

This element specifies whether applications shall allow text within text boxes to be vertically aligned when the `v-text-anchor` property is set within the parent VML shape.

Typically, if when the `v-text-anchor` property is set within the parent VML shape, then based on the value of that property, the text is top, center, or bottom aligned appropriately. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that the property shall be ignored, and instead the contents of the table shall always be top-aligned.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.25 **doNotWrapTextWithPunct** (Do Not Allow Hanging Punctuation With Character Grid)

This element specifies whether applications shall allowing hanging punctuation when:

- The **overflowPunct** element (§2.3.1.21) is turned on for a paragraph
- A document grid is defined using the **docGrid** element (§2.6.5) which defines the number of characters per line

Typically, paragraphs which allow hanging punctuation shall allow the number of characters on a line as specified by the document grid to be exceeded by one in order to allow for hanging punctuation. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that the document grid shall never be exceeded for hanging punctuation.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.26 **footnoteLayoutLikeWW8** (Emulate Word 6.x/95/97 Footnote Placement)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 6.x/95/97) when determining the placement of the contents of footnotes relative to the page on which the footnote reference occurs. This emulation typically involves some and/or all of the footnote being inappropriately placed on the page following the footnote reference.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.27 **forgetLastTabAlignment** (Ignore Width of Last Tab Stop When Aligning Paragraph If It Is Not Left Aligned)

This element specifies how applications should handle the final tab stop on a line when aligning the contents of a paragraph as specified by the **jc** element (§2.3.1.13) in the paragraph's properties.

Typically, aligning the contents of a paragraph involves the following:

- Determining the layout of that line before the alignment (including all tab stops)
- Aligning the resulting contents of the line

This is done to ensure that tab stops on a line do not change when the contents of the paragraph are aligned (i.e. the tab stops should not have to take into account the paragraph alignment).

This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall ignore the additional line width generated by the last tab stop (and only the last tab stop) when the alignment of the tab stop as defined by the **val** attribute on the **tab** element (§2.3.1.37) is not **left** (or **bar**, which as defined by this Office Open XML Standard, is not a tab stop per se) when determining the width of the line. The resulting full line shall then be aligned at the position where the line would have been aligned without that tab stop.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.28 **growAutofit** (Allow Tables to AutoFit Into Page Margins)

This element specifies whether applications shall allow a table which is using the AutoFit table layout algorithm to extend beyond the margins of the page if the minimum width of each table cell would result in an overall table width which is wider than those page margins.

Typically, if a table is using the AutoFit layout algorithm, then based on the definition of that logic, each column in the table shall be increased to the minimum width of its contents (e.g. the longest non-breaking run of text contained within it and/or the width of an inline image contained in one of its cells) until the overall width of the table reaches that of the text extents on the page, at which point text shall be broken and images shall be clipped as needed to maintain the width of the table at the page width (i.e. the page width is an immutable maximum width for the table). This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the minimum width of the cells shall not be constrained by the page width, and instead the table shall be allowed to extend into the page margins as needed in order to meet the minimum widths of each of its cells.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.29 **layoutRawTableWidth** (Ignore Space Before Table When Deciding If Table Should Wrap Floating Object)

This element specifies how tables which have been indented from the margin using the **tblInd** element (§2.4.48) shall be wrapped around floating objects defined using the Vector Markup Language (VML) syntax.

Typically, when a table is positioned next to a floating object, the table shall only remain next to the object if it can fit in the remaining space on the line when considering the full width needed for the table: the space before the table, plus the width of the table. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the calculation determining whether the table shall fit next to the object shall not include the space before the table, even if that means that the table is actually clipped by the object.

Attributes	Description
------------	-------------

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.30 **layoutTableRowsApart** (Allow Table Rows to Wrap Inline Objects Independently)

This element specifies whether tables which are wrapping around floating objects defined using the Vector Markup Language (VML) syntax shall wrap around the object as a whole, or if each table row shall individually wrap the object as needed (causing a more stuttered, yet tighter, wrapping of the object).

Typically, when a table wraps around a floating object, the table must wrap the object as a unit (i.e. the whole table square wraps the object). This element, when present with a **val** attribute value of **true** (or equivalent), specifies that wrapping is applied to each row in the table one by one, even if it means that each row has a different resulting position with respect to the table.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.31 **lineWrapLikeWord6** (Emulate Word 6.0 Line Wrapping for East Asian Text)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 6.0) when determining the whitespace compression of the final character on each line in the document. This emulation typically results in characters ending a line that may be compressed on the right being compressed on the right irrespective to whether the compression will allow another character to be included on the given line or not.

Typically, applications shall not perform this compatibility. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.32 **mwSmallCaps** (Emulate Word 5.x for the Macintosh Small Caps Formatting)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 5.x for the Macintosh) when determining the resulting formatting when the **smallCaps** element (§2.3.2.31) is applied to runs of text within this WordprocessingML document. This emulation typically results in small caps which are smaller than typical small caps at most font sizes.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.33 **noColumnBalance** (Do Not Balance Text Columns within a Section)

This element specifies whether the contents of sections with multiple columns defined using the **cols** element (§2.6.4) should automatically be balanced. In terms of column layout, *balancing* is the act of attempting to ensure that the number of lines in each column is equivalent (rather than completely filling one column before populating the next).

Typically, column balancing is automatically performed on the contents of sections with multiple columns. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that column balancing shall not occur, and each

column shall be filled individually until the end of the current page, until all text has been displayed, even if this means one or more columns are unused.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.34 **noExtraLineSpacing** (Do Not Center Content on Lines With Exact Line Height)

This element specifies whether an exact line height using the **spacing** element (§2.3.1.33) in the paragraph's properties, each line shall not be automatically centered within the given amount of line spacing.

Typically, if the exact amount of spacing allotted to a line via the paragraph properties exceeds the amount of space required by that line, then the line of text shall be automatically centered when the text of the document is displayed. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that all additional spacing shall instead be placed below the normal layout of the line of text.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.35 **noLeading** (Do Not Add Leading Between Lines of Text)

This element specifies whether the additional leading specified by the current font face shall be added between each line of text when that text is displayed. *Leading* refers to the additional spacing requested by a particular font in order to ensure that letters on subsequent lines do not display in a fashion where they are positioned too closely together.

Typically, leading should be added as specified by the associated font. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the additional leading specified by the font shall never be output when the text is displayed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.36 **noSpaceRaiseLower** (Do Not Increase Line Height for Raised/Lowered Text)

This element specifies whether the height which is allotted to any given line of text when the contents of this document are displayed shall include additional spacing in order to ensure that all raised and/or lowered text can be fully displayed.

Typically, any extra space needed is added to the line to prevent raised and lowered text from being truncated or hidden. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the height of the line shall be determined solely by the spacing settings on the parent paragraph, and any raised/lowered text shall just be clipped if it exceeds that space.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.37 **noTabHangInd** (Do Not Create Custom Tab Stop for Hanging Indent)

This element specifies whether applications should always create a hanging indent as a custom tab stop when handling tabs within the contents of a WordprocessingML paragraph. The **dontUseIndentAsNumberingTabStop** element (§2.15.3.22) specifies if this tab stop shall be used in the case of a tab added as the suffix to numbering in a numbered paragraph, while this element handles the same functionality in the generic case (i.e. this element, when set, renders that setting irrelevant as the tab stop is never used).

Typically, the hanging indent on a paragraph shall be treated as a custom tab stop location within that paragraph, allowing the first tab on the first line in the paragraph to advance to the location of the hanging indent. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that no custom tab stop shall be created for a hanging indent on a line under any circumstances.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.38 **printBodyTextBeforeHeader** (Print Body Text before Header/Footer Contents)

This element specifies the order in which the contents of the main document story and any headers and/or footers shall be sent to the printer.

Typically, the contents of a document are sent to the printer as follows:

- First, the contents of headers/footers are sent to the printer
- Finally, the contents of the main document story are sent to the printer

This element, when present with a **val** attribute value of **true** (or equivalent), specifies that this order shall be reversed, and that the body text shall be sent to the printer before any header/footer text. This reversal allows for the processing of PostScript codes in the text layer in the same order as afforded by some legacy word processing applications.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.39 **printColBlack** (Print Colors as Black And White without Dithering)

This element specifies the way in which colored text and/or objects shall be handled when printed to a printer whose printer settings indicate that it can only handle black and white text.

Typically, the contents of a colored document are sent to a black and white printer using grayscale (different shades of gray) to represent each of the possible colors. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that colors will not be printed as mapped shades of grey, but rather exclusively in solid black and white. This setting prevents the fuzzy look that may occur when gray or blue content is dithered. *Dithering* is the process by which colors are simulated using various patterns of black dots on a white background

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.40 **selectFldWithFirstOrLastChar** (Select Field When First or Last Character Is Selected)

This element specifies whether applications should automatically select the entire contents of a field in a WordprocessingML document when the first or last character is selected.

Typically, users can select any character individually within the result of a field in the document. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that selecting the first or last character of that field result shall automatically result in the selection of the entire field.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.41 **shapeLayoutLikeWW8** (Emulate Word 97 Text Wrapping Around Floating Objects)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 97) when determining how to wrap text around floating objects using **topAndBottom** wrapping defined using the Vector Markup Language (VML) syntax. This emulation typically results in the wrapping above and below the object allowing the text to wrap more tightly around the object (text wraps more tightly around the object than it normally would).

[Guidance: To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications

wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*

Typically, applications shall not perform this compatibility. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.42 **showBreaksInFrames** (Display Page/Column Breaks Present in Frames)

This element specifies whether applications should honor the presence of page and/or column breaks which are present within the contents of paragraphs which have been defined as frames using the **framePr** element (§2.3.1.11).

Typically, breaks within frames shall be ignored and shall have no effect on the display of the paragraph in which they are contained. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that rather than completely ignoring these breaks, applications should display the break and move the remaining frame content, and all subsequent text, to the next page and/or column, as needed.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.43 **spaceForUL** (Add Additional Space Below Baseline For Underlined East Asian Text)

This element specifies whether East Asian content in a WordprocessingML document which has been underlined using the **u** element shall have additional descent added to the properties of the font in order to ensure that there is adequate spacing between the characters in the font and the underlining applied to the text.

Typically, no adjustments are made to the contents of text runs containing East Asian text which have been underlined. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that whenever the following conditions are met:

- The text run contains East Asian characters
- The text run is not using baseline font alignment as defined by the **textAlignment** property

That the larger of the following two values will be added to the descent property of that font in order to provide additional padding between the text characters and the underline:

- 3 percent of the font size
- 40 twentieths of a point (31 twentieths of a point for Japanese text)

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.44 **spacingInWholePoints** (Only Expand/Condense Text By Whole Points)

This element specifies how applications should apply text expansion/compression defined using the **spacing** element (§2.3.2.33) within a set of run properties.

Typically, as defined in the **spacing** element, text within runs in a WordprocessingML document may be expanded or compressed in increments of twentieths of a point. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that the expansion and compression of text shall only be performed in increments of points. Any value which is not equal to an expansion or compression of a whole point shall be rounded down to the nearest whole point when the text is expanded/compressed within the WordprocessingML document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.45 **splitPgBreakAndParaMark** (Always Move Paragraph Mark to Page after a Page Break)

This element specifies whether a page break shall automatically complete the line on which it appears, moving the end of the paragraph to a new line on the next page, or if it shall behave as true run-level content within its current paragraph. Typically, a page break defined using the **br** element (§2.3.3.1) is treated as run-level content, which means that although it delimits the end of the page, if there is no content after it within the current paragraph, that the paragraph shall also end on that page. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that a page break shall always immediately end the current page, moving the paragraph mark which delimits the end of its parent paragraph to a new line on the next page.

Note that this setting only affects the case where there is no run-level content after the page break within the paragraph - if any further run content appears in the paragraph it shall appear on subsequent lines on the next page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.46 **subFontBySize** (Increase Priority Of Font Size During Font Substitution)

This element specifies whether applications shall increase the priority of font size when performing font substitution in a WordprocessingML document. *Font substitution* is the process by which an application determines which font to use in place of a font that is referenced by a document, but is not available to the application trying to display the document. Typically, applications may perform font substitution using any mechanism available. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that finding a font with a similar font size shall have increased precedence when doing font substitution for this document.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.47 **suppressBottomSpacing** (Ignore Exact Line Height for Last Line on Page)

This element specifies whether an exact line height specified using the **spacing** element (§2.3.1.33) with a **lineRule** attribute value of **exact** shall be ignored for the last line on each page.

Typically, if an exact line height has been specified using the **spacing** element, then all lines within that paragraph have the necessary line spacing added to them in order to meet this constraint. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that no additional spacing shall be added below the last line on each page as a result of these line spacing requirements - a line shall be placed on the bottom of the page if its characters fit on that page ignoring the necessary space after.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.48 **suppressSpacingAtTopOfPage** (Ignore Minimum Line Height for First Line on Page)

This element specifies whether the minimum line height specified using the **spacing** element (§2.3.1.33) with a **lineRule** attribute value of **atLeast** shall be ignored for the first line on each page.

Typically, if a minimum line height has been specified using the **spacing** element, then all lines within that paragraph have the necessary line spacing added to them in order to meet this constraint. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that no additional spacing shall be added above the first line on each page as a result of this line spacing requirements - the top of the text characters on the first line shall be at the top edge of the page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.49 **suppressSpBfAfterPgBrk** (Do Not Use Space Before On First Line After a Page Break)

This element specifies that applications should not postpone any before paragraph spacing to the first line containing content after a page break.

Typically, a page break defined using the **br** element (§2.3.3.1) is treated as run-level content, which means that although it delimits the end of the page, if there is no content after it within the current paragraph, that the paragraph shall also end on that page. However, in the case where there is additional run-level content within the same paragraph, that content, although part of the same paragraph as the page break, is displayed on the following page.

This leads to a situation where the only run content on the page with the page break is the break itself, with all subsequent content on the following page. In this case, applications shall apply the value specified by the **spacing** element's before attribute to the first line on the new page (since it is ostensibly the only page with content in that paragraph).

This element, when present with a val attribute value of true (or equivalent), specifies the paragraph before spacing shall not be 'postponed' in this way - if the line with the page break has no content, then the **spacing** element's before attribute is simply ignored.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.50 **suppressTopSpacing** (Ignore Minimum and Exact Line Height for First Line on Page)

This element specifies whether the minimum line height specified using the **spacing** element (§2.3.1.33) with a lineRule attribute value of atLeast or exact shall be ignored for the first line on each page.

Typically, if a minimum or exact line height has been specified using the **spacing** element, then all lines within that paragraph have the necessary line spacing added to them in order to meet this constraint. This element, when present with a val attribute value of true (or equivalent), specifies that no additional spacing shall be added above the first line on each page as a result of these line spacing requirements - the top of the text characters on the first line shall be at the top edge of the page.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.51 **suppressTopSpacingWP** (Emulate WordPerfect 5.x Line Spacing)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (WordPerfect 5.x) when determining the resulting spacing between lines in a paragraph using the **spacing** element (§2.3.1.33). This emulation typically results in line spacing which is reduced from its normal size.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a val attribute value of true (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.52 **swapBordersFacingPages** (Swap Paragraph Borders on Odd Numbered Pages)

This element specifies whether left and right paragraph borders defined under the **pBdr** element (§2.3.1.24) shall be swapped under conditions where it is possible that the those pages are intended to be used to create a book-like publication.

Typically, no changes shall be made to the positions of paragraph borders defined under the **pBdr** element - a right border is always on the right, and a left border is always on the left. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that under the two following conditions:

- The margins in this document are mirrored using the **mirrorMargins** element (§2.15.1.57)
- The header/footers in this document are different on even and odd numbered pages using the **evenAndOddHeaders** element (§2.10.1)

That paragraph borders on odd-numbered pages will be swapped - that is, left borders shall be displayed on the right and right borders shall be displayed on the left.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.53 **truncateFontHeightsLikeWP6** (Emulate WordPerfect 6.x Font Height Calculation)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (WordPerfect 6.x) when determining the character height for characters in a font. This emulation typically results slightly truncated character heights.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.54 **uiCompat97To2003** (Disable Features Incompatible With Earlier Word Processing Formats)

Disable UI functionality that is not compatible with Word97-2003

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.55 **ulTrailSpace** (Underline All Trailing Spaces)

This element specifies whether applications shall display underlining beneath all trailing spaces in the contents of a line when those contents are underlined. *Trailing spaces* are all space characters which are not followed by non-space characters on the same line.

Typically, applications do not display underlining on all trailing spaces which have the underline property applied to them. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that all characters with underline applied, including trailing spaces, shall display underlining if it is applied to that content.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.56 **underlineTabInNumList** (Underline Following Character Following Numbering)

This element specifies whether applications shall underline the character following the numbering defined using the **suff** element (§2.9.30) when both the numbering itself and the first letter of the corresponding numbered paragraph is underlined.

Typically, the tab or space character generated between numbering and the corresponding paragraph of text is never formatted, since it is automatically generated by the **suff** element. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the tab or space shall tab or space shall be underlined the same way as the numbering symbol itself in the following conditions:

- The numbering is underlined
- The first character of the paragraph is underlined

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.57 **useAltKinsokuLineBreakRules** (Use Alternate Set of East Asian Line Breaking Rules)

This element specifies an alternate set of characters which may be used to determine which characters can begin and/or end a line when kinsoku line breaking rules are enabled using the **kinsoku** element (§2.3.1.16).

Typically, the characters used to determine which characters shall not end a line are those listed by the **kinsoku** element in the paragraph properties subclause of this document. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the following settings shall be used instead (for brevity, only those settings which are different are listed below):

Chinese (Simplified)

- Cannot start a line:
!),.,;?]}‘’~—||”....、。 ” 々>》」』】〕! ” ’) , . : ; ?] ` | } ~ ¢
- Cannot end a line:
([{““ < 《 「 『 【 〔 〔 (. [{ £ ¥

Chinese (Traditional)

- Cannot start a line:
!),.,;?]}¢——”•.....、。 >》」』】〕 ” : || { ~~~~~~?!) !) , . : ; ? | }、

Korean

- Cannot end a line:
([{£¥““ < 《 「 『 【 〔 \$ ([{ ₩

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.58 **useAnsiKerningPairs** (Use ANSI Kerning Pairs from Fonts)

This element specifies whether applications shall use the ANSI or Unicode kerning pair information from fonts stored in the document when displaying those characters within the document's contents.

Typically, applications shall use the Unicode kerning pair information in order to determine all possible kerning pairs in the fonts in use. This element, when present with a **val** attribute value of **true** (or equivalent), specifies that the ANSI kerning information shall be used instead.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.59 useFELayout (Do Not Bypass East Asian/Complex Script Layout Code)

This element specifies that applications shall not bypass code relating to the layout of East Asian and/or Complex Script characters when presenting this document.

[*Guidance:* Previous word processing applications relied on this flag to determine whether to perform functions which allow for the correct layout of East Asian and Complex Script text. Although current applications no longer rely on this flag (as they should correctly use the Unicode subranges and code pages of the text in use), this flag should be output in order to ensure that files with this content can be viewed correctly in previous word processors. *end guidance*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.60 useNormalStyleForList (Do Not Automatically Apply List Paragraph Style To Bulleted/Numbered Text)

This element specifies whether applications shall automatically apply the paragraph style with the styleId attribute ListParagraph when numbering is applied to a paragraph currently formatted using the default paragraph style. Typically, when a paragraph is formatted using the default paragraph style, and numbering is subsequently applied, the paragraph style with the styleId attribute ListParagraph when numbering is applied to ensure that paragraph properties are appropriate for a numbered paragraph. This element, when present with a val attribute value of true (or equivalent), specifies that no alternate paragraph style shall ever be applied

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.61 usePrinterMetrics (Use Printer Metrics To Display Documents)

This element specifies whether applications shall use the printer metrics of the currently active printer when determining how to display the contents of a WordprocessingML document. *Printer metrics* are printer-specific settings which can be queried to tell an application how and where text shall be displayed on a printed page.

Typically, applications display the content of a document in a device independent manner - the application is therefore not changing the layout of a document based on the currently attached printer, and instead shall dictate to the printer where characters shall be presented on the page when printed. This element, when present with a val attribute value of true (or equivalent), specifies that the metrics of the current printer shall be used to display the document instead.

Specifically, when this setting is enabled, the printer metrics are used to determine the number of pixels per logical inch along the screen width and height. This should then be used to compute the pixel height of the fonts requested when displaying the document, as well as to scale between any logical units within the document (e.g. drawing object sizes) to the appropriate device units. Those units would then need to be scaled back into screen units for final display to a screen, but not scaled again when displayed to a printer.

[*Note:* On the Windows platform, you can use the GetDeviceCaps function to retrieve device-specific information for the specified printer. For this specific setting, you can use GetDeviceCaps(hdc, LOGPIXELSX) and GetDeviceCaps(hdc, LOGPIXELSY) with a printer DC to retrieve the number of pixels per logical inch along the screen width and height. With this, you can then use those DPI metrics to compute a pixel value for the font request in the LOGFONT structure (the LOGFONT structure defines the attributes of a font). A common formula to do this is $S_{px} = S_{pts} * \frac{LOGPIXELSY}{72}$. *end note*]

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.62 useSingleBorderforContiguousCells (Use Simplified Rules For Table Border Conflicts)

This element specifies whether applications should use an alternate simplified algorithm when handling conflicts between adjacent table borders within a table.

Typically, the conflicts between two adjacent table borders are handled using the conflict resolution algorithm defined in §2.4.38 of this Office Open XML Standard. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that rather than using that algorithm to determine the outcome of the conflict to two adjacent borders, that the following logic shall be used instead:

- Cell borders shall supersede table borders
- Cell borders to the right shall supersede cell borders to the left (i.e. the rightmost border wins in conflicts between vertical borders)
- Cell borders below shall supersede cell borders above (i.e. the bottommost border wins in conflicts between horizontal borders)

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.63 **useWord2002TableStyleRules** (Emulate Word 2002 Table Style Rules)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 2002) when determining the formatting resulting from table styles applied to tables within a WordprocessingML document.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.64 **useWord97LineBreakRules** (Emulate Word 97 East Asian Line Breaking)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (Microsoft Word 97) when determining the line breaking rules for East Asian text within a WordprocessingML document.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.65 **wpJustification** (Emulate WordPerfect 6.x Paragraph Justification)

This element specifies that applications shall emulate the behavior of a previously existing word processing application (WordPerfect 6.x) when performing full paragraph justification using a `val` attribute value of `both` on the `jc` element (§2.3.1.13). This alternate justification method involves biasing towards compressing rather than expanding spaces when needed to justify a line.

[*Guidance:* To faithfully replicate this behavior, applications must imitate the behavior of that application, which involves many possible behaviors and cannot be faithfully placed into narrative for this Office Open XML Standard. If applications wish to match this behavior, they must utilize and duplicate the output of those applications. It is recommended that

applications not intentionally replicate this behavior as it was deprecated due to issues with its output, and is maintained only for compatibility with existing documents from that application. *end guidance*]

Typically, applications shall not perform this compatibility. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that applications shall attempt to mimic that existing word processing application in this regard.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.66 **wpSpaceWidth** (Space width)

Set the width of a space like WordPerfect 5.x.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.15.3.67 **wrapTrailSpaces** (Line Wrap Trailing Spaces)

This element specifies whether applications shall perform line wrapping on trailing spaces in the contents of a line when displaying in it a paragraph. *Trailing spaces* are all space characters which are not followed by non-space characters on the same line.

Typically, applications do not line wrap trailing spaces, instead allowing an unbounded number of trailing spaces on a line, with the next non-space character starting at the first character position on the next line. This element, when present with a `val` attribute value of `true` (or equivalent), specifies that all characters, including trailing spaces, shall be line wrapped normally.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.16 Fields & Hyperlinks

Most text in a word processing document is static; that is, unless it is directly changed as the result of editing, its contents remain the same, no matter how the rest of the document might change. However, certain useful pieces of information can change value over the life of a document. Consider the case of a reference to a page number, as in "For more information on this topic, see page 56." Clearly, hard coding the page number as 56 means that that number will need to be manually replaced as the document's size or layout is changed. Even a simple change to any margin, line spacing, or font size can invalid such references.

Fields provide a mechanism for placeholders, such as page reference numbers, that can be added to a document such that those placeholders are replaced by their corresponding values when the document is rendered for display or print. Other applications for fields include, but are not limited to, automatic numbering of tables and figures, document creation and current date and time, document author information, and the computation of totals for a table column.

A *field* is a set of codes that instructs a WordprocessingML consumer to insert text, graphics, page numbers, and other material into a document automatically. The text or graphics inserted into a document when a consumer carries out a field's codes is referred to as the *field result* for that field. The act of carrying out a field's codes is referred to as a *field update*. As to how or when any field is updated is outside the scope of this Office Open XML Standard.

2.16.1 Syntax

The general syntax of a field is as follows:

field:
field-type [*instruction*]

field-type:

date-and-time
document-automation
document-information
equations-and-formulas
index-and-tables
links-and-references
mail-merge
numbering
user-information
form-field

date-and-time:

CREATEDATE | DATE | EDITTIME | PRINTDATE | SAVEDATE | TIME

document-automation:

COMPARE | DOCVARIABLE | GOTOBUTTON | IF | MACROBUTTON | PRINT

document-information:

AUTHOR | COMMENTS | DOCPROPERTY | FILENAME | FILESIZE | INFO
| KEYWORDS | LASTSAVEDBY | NUMCHARS | NUMPAGES | NUMWORDS | SUBJECT
| TEMPLATE | TITLE

equations-and-formulas:

= formula | ADVANCE | EQ | SYMBOL

index-and-tables:

INDEX | RD | TA | TC | TOA | TOC | XE

links-and-references:

AUTOTEXT | AUTOTEXTLIST | BIBLIOGRAPHY | CITATION | HYPERLINK | INCLUDEPICTURE
| INCLUDETTEXT
| LINK | NOTEREF | PAGEREF | QUOTE | REF | STYLEREF

mail-merge:

ADDRESSBLOCK | ASK | COMPARE | DATABASE | FILLIN | GREETINGLINE | IF
| MERGEFIELD | MERGEREC | MERGESEQ | NEXT | NEXTIF | SET | SKIPIF

numbering:

AUTONUM | AUTONUMGL | AUTONUMOUT | BARCODE | LISTNUM | PAGE | REVNUM
| SECTION | SECTIONPAGES | SEQ

user-information:

USERADDRESS | USERINITIALS | USERNAME

form-field:

FORMCHECKBOX | FORMDROPDOWN | FORMTEXT

instruction:

field
field-argument
switches
field-argument switches
switches field-argument

field-argument:

["] text ["]

switches:

switch

switch switches

switch:

formatting-switch

field-specific-switch

formatting-switch:

date-and-time-formatting-switch

numeric-formatting-switch

general-formatting-switch

field-specific-switch:

\field-switch-character [field-argument]

field-switch-character:

!

one or two Latin letters

formula is discussed in §2.16.3, and *formatting-switches* are discussed in §2.16.4.

If the *text* in a *field-argument* contains white space, the delimiting double-quote characters shall be present; otherwise, they are optional. To include a double-quote character in *text*, it shall be preceded with a backslash (\). To include a backslash character in *text*, it shall be preceded with another backslash (\).

Arbitrary amount of white space can occur before the first token, after the last token, and between successive tokens, including no white space at all.

Except for = *formula*, the terminals of *field-type* are alphabetic tokens. These tokens are called *field-type names*. Field-type names are case-insensitive.

field-switch-characters are case-insensitive.

There is no ordering of *switch* entries in *switches*.

2.16.2 XML representation

Fields shall be implemented in XML using either of two approaches:

- As a *simple field implementation*, using the **fldSimple** element, or
- As a *complex field implementation*, using a set of runs involving the **fldChar** and **instrText** elements.

For a simple field implementation, only one element, **fldSimple**, shall be used, in which case, its *instr* attribute shall contain a *field*, and the body of the element shall contain the most recently updated field result.

For a complex field implementation, a set of runs shall be used with each run containing, in sequence, the following elements:

- **fldChar** with attribute *fldCharType* value *begin*,
- One or more **instrText** elements, which, collectively, contain a complete *field*,
- Optionally,
 - **fldChar** with attribute *fldCharType* value *separate*, which separates the field from its field result,
 - Any number of runs and paragraphs that contains the most recently updated field result, and
- **fldChar** with attribute *fldCharType* value *end*.

[Note: Fields that are for display purposes only have no need to, and do not, store a field result. *end note*]

[Note: Every simple field implementation for a given field has a corresponding complex field implementation. However, not every complex field implementation has a corresponding simple field implementation. If some characters in a *field* have different run properties than others, that field must be implemented using multiple runs, and that requires that complex field implementation be used. For an example, see §2.16.4.3, where the first letter of a DATE field is made bold, underlined, and red, while the other letters have none of these properties. *end note*]

As shown in §2.16.1, the *instruction* of one *field* can be another *field*, allowing fields to nest. In such cases, the XML run sequence for the inner field is defined at the point of reference for that inner field, inside the outer field's XML run sequence.

2.16.3 Formulas and expressions

A field instruction can involve a calculation via a *formula*:

formula:

expression

where *expression* can be an arbitrary complex arithmetic expression involving constants (§2.16.3.1), bookmarks that refer to *expressions* (§2.16.3.2), arithmetic and logical operators (§2.16.3.3), functions (§2.16.3.4), values of cells in a table (§2.16.3.5), and *fields* that result in a single value. *expression* can contain grouping parentheses to document the default precedence or to override it.

All arithmetic terms in an *expression* are real numbers. Infinities and NaN (Not-a-Number) are not supported.

2.16.3.1 Constants

A constant has the following form:

constant:

number

number:

whole-number-part [.]

. *fractional-part*

whole-number-part . *fractional-part*

whole-number-part:

series of one or more decimal digits

fractional-part:

series of one or more decimal digits

2.16.3.2 Bookmarks

Any arbitrary piece of text and/or graphics in a WordprocessingML document can be assigned a name, called a *bookmark*. If a bookmark references text that represents an *expression*, that bookmark's name can be used as an operand in another *expression*. If a whole field is bookmarked, its bookmark name can also be used as an operand in an *expression*.

2.16.3.3 Operators

The *operators* permitted in *expression* are:

Operators		
Operator	Description	Precedence
-	Unary minus	highest
^	Powers and roots	
*	Multiplication	
/	Division	
%	Percentage	
+	Addition	
-	Subtraction	
=	Equal to	
<>	Not equal to	

Operators		
<	Less than	lowest
<=	Less than or equal to	
>	Greater than	
>=	Greater than or equal to	

Operators in *expression* having the same precedence associate left-to-right.

The equality, inequality, and relational operators yield 1 for true and 0 for false. An expression with value 0 tests logically false while one with any non-zero value tests true.

2.16.3.4 Functions

A *function* is a predefined procedure that computes and returns a result. Functions defined below with a parameter list of *list* accept two or more arguments separated by commas (,) or semicolons (;). As to which separator is permitted, is defined by the document's **listSeparator** (§2.15.1.56) element. Arguments to functions can be constants, formulas, or bookmark names that refer to constants or formulas. The functions AVERAGE, COUNT, MAX, MIN, PRODUCT, and SUM can also accept references to table cells as arguments. In the context of a table cell, functions taking a *list* also accept a single argument that designates a named-list of contiguous cells (§2.16.3.5). Function names are not case-sensitive, and white space can occur between a function's name and its argument list, if any.

The functions supported are as follows:

Functions	
Function	Description
ABS(<i>x</i>)	Returns the absolute value of <i>x</i> .
AND(<i>x</i> , <i>y</i>)	Returns 1 if the logical expressions <i>x</i> and <i>y</i> are both true; otherwise, it returns 0.
AVERAGE(<i>list</i>)	Returns the average value of the items in <i>list</i> .
COUNT(<i>list</i>)	Returns the number of items in <i>list</i> .
DEFINED(<i>x</i>)	Returns 1 if the expression <i>x</i> is well formed; otherwise, it returns 0.
FALSE	Returns 0.
INT(<i>x</i>)	Returns the value of the integer part of <i>x</i> .
MAX(<i>list</i>)	Returns the largest value in <i>list</i> .
MIN(<i>list</i>)	Returns the smallest value in <i>list</i> .
MOD(<i>x</i> , <i>y</i>)	Returns the value $x - ny$, for some integer n such that, if y is nonzero, the result has the same sign as x and magnitude less than the magnitude of y . If y is zero, a diagnostic shall be issued. (y need not be a whole number.)
NOT(<i>x</i>)	Returns 0 if the logical expression <i>x</i> is true, or 1 if the expression is false.
OR(<i>x</i> , <i>y</i>)	Returns 1 if either or both logical expressions <i>x</i> and <i>y</i> are true; otherwise, it returns 0.
PRODUCT(<i>list</i>)	Returns the result of multiplying together all members in <i>list</i> .
ROUND(<i>x</i> , <i>y</i>)	Returns the value of <i>x</i> rounded to the specified number of decimal places indicated by floor(<i>y</i>), where floor has the mathematical meaning. If <i>y</i> is negative, any fractional part is discarded and the integer part of the value is rounded to the corresponding power of 10.
SIGN(<i>x</i>)	Returns 1 if <i>x</i> is positive; returns 0 if <i>x</i> is zero; and returns -1 if <i>x</i> is negative.

Functions	
SUM(<i>list</i>)	Returns the sum of the items in <i>list</i> .
TRUE	Returns 1.

2.16.3.5 Table cell references

Items in a WordprocessingML table are organized into rows and columns with the box formed by the intersection of a row and column being called a *cell*. Cells have names such as A1, A2, B1, B2, and so on, with the letter representing a column and the number representing a row. The cell at the top-left corner of each table is named A1. Column letters are not case-sensitive.

A *cell reference* shall be one of the following:

- The name of a cell.
- A comma-separated set of cell names.
- A cell range where a colon (:) is used to separate the first and last cells in a designated range of cells that has a contiguous rectangular shape. Specifying a row or column's name only as the first and last cell in a range, selects that whole row or column, regardless of the number of rows and columns the table has now or might have in the future.

An *expression* inside a table's cell can have operands that are references to other cells in that table.

When used in a table cell, the functions taking a *list* argument can have a single argument of ABOVE, BELOW, LEFT, or RIGHT, spelled in any case combination. Such lists designate, respectively, all the cells above, below, to the left of, or to the right of that cell. However, the designated range terminates if a cell with blank or non-numeric contents is reached, except that if the first cell is blank, it is treated as containing 0.

An *expression* used outside a table or in a cell of one table can refer to cells in a second table by making a bookmark to that second table and qualifying cell names in that table by their table name using the form

(*tableBookmarkName cellReference*)

2.16.4 Field formatting

The result of a field has a *format*, either by default or because that field contains a *formatting-switch*. There are three kinds of field formatting: date and time (§2.16.4.1), numeric (§2.16.4.2), and general (§2.16.4.3).

2.16.4.1 Date and time formatting

date-and-time-formatting-switch:

\@ ["] *switch-argument* ["]

A *date-and-time-formatting-switch* specifies the format of a date or time result. [Note: This switch is sometimes called a *picture* switch because it allows the use of symbols to represent the format of the field result. *end note*] If the result of a field is not a date or time, this switch has no effect.

Quotation marks are required around *switch-argument* if it contains white space; otherwise, they are optional.

If no *date-and-time-formatting-switch* is present, a date or time result is formatted in an implementation-defined manner.

A date and time *switch-argument* is made up of a series of *picture items*.

Date and Time Formatting Picture Items	
Picture Item	Description
d	Formats the day of(the week or day of the month as a number without a leading 0 for single-digit days.
dd	Formats the day of the week or day of the month as a number with a leading 0 for single-digit days.
ddd	Formats the day of the week or month in its abbreviated form according to the language specified by the lang element (§2.3.2.18) on the run containing the field instructions.

Date and Time Formatting Picture Items	
dddd	Formats the day of the week as its full name according to the language specified by the lang element (§2.3.2.18) on the run containing the field instructions.
M	Formats the month as a number without a leading 0 for single-digit months.
MM	Formats the month as a number with a leading 0 for single-digit months.
MMM	Formats the month in its abbreviated form according to the language specified by the lang element (§2.3.2.18) on the run containing the field instructions.
MMMM	Formats the month as its full name according to the language specified by the lang element (§2.3.2.18) on the run containing the field instructions.
yy	Formats the year as two digits with a leading 0 for years 0–9.
yyyy	Formats the year as four digits.

In the following time formats, a lowercase h indicates that time is based on a 12-hour clock, while uppercase H indicates time is based on a 24-hour clock.

Time Formatting Picture Items	
Picture Item	Description
h or H	Formats the hour without a leading 0 for single-digit hours.
hh or HH	Formats the hour with a leading 0 for single-digit hours.
m	Formats the minutes without a leading 0 for single-digit minutes.
mm	Formats the minutes with a leading 0 for single-digit minutes.
am/pm or AM/PM	Formats using an am/AM or pm/PM suffix.

Miscellaneous Formatting Picture Items	
Picture Item	Description
Other character	Includes the specified character in the result at that position. [Note: Commonly used characters are colon (:), hyphen (-), asterisk (*), slash (/), and space. <i>end note</i>]
'text'	Includes <i>text</i> in the result.
`numbered-item`	Includes, in Arabic numerals, the number of the preceding item numbered as a caption or resulting from a SEQ field (§2.16.5.63). <i>numbered-item</i> shall be the same name as <i>identifier</i> in that SEQ field.

2.16.4.2 Numeric formatting

numeric-formatting-switch:

\# ["] *switch-argument* ["]

A *numeric-formatting-switch* specifies the format of a numeric result. If the result of a field is not a number, this switch has no effect.

Quotation marks are required around *switch-argument* if it contains white space; otherwise, they are optional.

If no *numeric-formatting-switch* is present, a numeric result is formatted without leading spaces or trailing fractional zeros. If the result is negative, a leading minus sign is present. If the result is a whole number, no radix point is present.

A numeric *switch-argument* is made up of a series of *picture items*.

Numeric Formatting Picture Items	
Picture Item	Description
0	Specifies the requisite numeric positions to display in the result. If the result does not include a digit in that position, 0 is displayed.
#	Specifies the requisite numeric positions to display in the result. If the result does not include a digit in that position, a space is displayed. Extra fractional digits are rounded off.
x	Drops digits to the left of the x placeholder. If the placeholder is to the right of the decimal point, the result is rounded to that place.
.	Indicates the radix-point position. The radix-point character displayed is locale-specific.
,	Separates groups of three digits. The separator character displayed is locale-specific.
-	Prepends a minus sign to a negative result, or prepends a space if the result is positive or 0.
+	Prepends a plus sign to a positive result, a minus sign to a negative result, or a space if the result is 0.
Other character	Includes the specified character in the result at that position.
'text'	Includes text in the result.
`numbered-item`	Includes, in Arabic numerals, the number of the preceding item numbered as a caption or resulting from a SEQ field (§2.16.5.63). <i>numbered-item</i> shall be the same name as <i>identifier</i> in that SEQ field.
<i>positive-result ; negative-result</i>	Specifies different sets of picture items for positive and negative results. A zero value uses the positive picture.
<i>positive-result ; negative-result ; zero-result</i>	Specifies different sets of picture items for positive, negative, and zero results.

2.16.4.3 General formatting

general-formatting-switch:

* ["] *switch-argument* ["]

A *general-formatting-switch* specifies a variety of formats for a numeric or text result. If the result type of a field does not correspond to the format specified, this switch has no effect.

Quotation marks are required around *switch-argument* if it contains white space; otherwise, they are optional.

A *switch-argument* is made up of a series of *picture items*.

General Formatting Switch Arguments	
Switch Argument	Description
AIUEO	Formats a numeric result using hiragana characters in the traditional a-i-u-e-o order. Corresponds to an ST_NumberFormat enumeration value of <code>aiueo</code> .
ALPHABETIC	Formats a numeric result as one or more occurrences of an uppercase alphabetic Latin character. Value 1 results in the letter A, value 2 results in the letter B, and so on up to value 26, which results in the letter Z. For values greater than 26, 26 is repeatedly subtracted from the value until the result is 26 or less. The result value determines which letter to use, and the same letter is repeated for each time 26 was subtracted from the original value.
alphabetic	Formats a numeric result as one or more occurrences of an lowercase alphabetic Latin

General Formatting Switch Arguments	
	character. Value 1 results in the letter a, value 2 results in the letter b, and so on up to value 26, which results in the letter z. For values greater than 26, 26 is repeatedly subtracted from the value until the result is 26 or less. The result value determines which letter to use, and the same letter is repeated for each time 26 was subtracted from the original value.
Arabic	Formats a numeric result using Arabic cardinal numerals. Corresponds to an ST_NumberFormat enumeration value of decimal.
ARABICABJAD	Formats a numeric result using ascending Abjad numerals. Corresponds to an ST_NumberFormat enumeration value of arabicAbjad.
ARABICALPHA	Formats a numeric result using characters in the Arabic alphabet. Corresponds to an ST_NumberFormat enumeration value of arabicAlpha.
ArabicDash	Formats a numeric result using Arabic cardinal numerals, with a prefix of "- " and a suffix of "-". Corresponds to an ST_NumberFormat enumeration value of numberInDash.
BAHTTEXT	Formats a numeric result using the given Thai style.
Caps	Capitalizes the first letter of each word.
CardText	Formats a numeric result as lowercase cardinal text. Corresponds to an ST_NumberFormat enumeration value of cardinalText.
CHARFORMAT	See the discussion following this table.
CHINESENUM1	Formats a numeric result using ascending numbers from the Chinese counting system. Corresponds to an ST_NumberFormat enumeration value of chineseCounting.
CHINESENUM2	Formats a numeric result using sequential numbers from the Chinese simplified legal format. Corresponds to an ST_NumberFormat enumeration value of chineseLegalSimplified.
CHINESENUM3	Formats a numeric result using sequential numbers from the Chinese counting thousand system. Corresponds to an ST_NumberFormat enumeration value of chineseCountingThousand.
CHOSUNG	Formats a numeric result using sequential numbers from the Korean Chosung format. Corresponds to an ST_NumberFormat enumeration value of chosung.
CIRCLENUM	Formats a numeric result using decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character for numbers in the range 1–20. For non-negative numbers outside this range, formats them as with ARABIC. Corresponds to an ST_NumberFormat enumeration value of decimalEnclosedCircle.
DBCHAR	Formats a numeric result using double-byte Arabic numbering. Corresponds to an ST_NumberFormat enumeration value of decimalFullWidth.
DBNUM1	Formats a numeric result using sequential digital ideographs, using the appropriate character. Corresponds to an ST_NumberFormat enumeration value of ideographDigital.
DBNUM2	Formats a numeric result using sequential numbers from the Korean counting system. Corresponds to an ST_NumberFormat enumeration value of koreanCounting.
DBNUM3	Formats a numeric result using sequential numbers from the Japanese legal counting

General Formatting Switch Arguments	
	system. Corresponds to an ST_NumberFormat enumeration value of <code>japaneseLegal1</code> .
DBNUM4	Formats a numeric result using sequential numbers from the Japanese digital ten thousand counting system. Corresponds to an ST_NumberFormat enumeration value of <code>japaneseDigitalTenThousand</code> .
DollarText	Formats a numeric result in the following form: <i>integer-part-as-cardinal-text</i> and <i>nn/100</i> The fractional part is rounded to two decimal places, <i>nn</i> , and is formatted using Arabic cardinal numerals.
FirstCap	Capitalizes the first letter of the first word.
GANADA	Formats a numeric result using sequential numbers from the Korean Ganada format. Corresponds to an ST_NumberFormat enumeration value of <code>ganada</code> .
GB1	Formats a numeric result using decimal numbering followed by a period, using the enclosed alphanumeric glyph character. Corresponds to an ST_NumberFormat enumeration value of <code>decimalEnclosedFullstop</code> .
GB2	Formats a numeric result using decimal numbering enclosed in parenthesis, using the enclosed alphanumeric glyph character. Corresponds to an ST_NumberFormat enumeration value of <code>decimalEnclosedParen</code> .
GB3	Formats a numeric result using decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character. Once the specified sequence reaches 11, the numbers may be replaced with non-enclosed equivalents. Corresponds to an ST_NumberFormat enumeration value of <code>decimalEnclosedCircleChinese</code> .
GB4	Formats a numeric result using decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character. Once the specified sequence reaches 11, the numbers may be replaced with non-enclosed equivalents. Corresponds to an ST_NumberFormat enumeration value of <code>ideographEnclosedCircle</code> .
HEBREW1	Formats a numeric result using Hebrew numerals. Corresponds to an ST_NumberFormat enumeration value of <code>hebrew1</code> .
HEBREW2	Formats a numeric result using the Hebrew alphabet. Corresponds to an ST_NumberFormat enumeration value of <code>hebrew2</code> .
Hex	Formats the numeric result using uppercase hexadecimal digits. Corresponds to an ST_NumberFormat enumeration value of <code>hex</code> .
HINDIARABIC	Formats a numeric result using Hindi numbers. Corresponds to an ST_NumberFormat enumeration value of <code>hindiNumbers</code> .
HINDICARDTEXT	Formats a numeric result using sequential numbers from the Hindi counting system. Corresponds to an ST_NumberFormat enumeration value of <code>hindiCounting</code> .
HINDILETTER1	Formats a numeric result using Hindi vowels. Corresponds to an ST_NumberFormat enumeration value of <code>hindiVowels</code> .
HINDILETTER2	Formats a numeric result using Hindi consonants. Corresponds to an ST_NumberFormat enumeration value of <code>hindiConsonants</code> .

General Formatting Switch Arguments	
IROHA	Formats a numeric result using the Japanese iroha. Corresponds to an ST_NumberFormat enumeration value of <code>iroha</code> .
KANJINUM1	Formats a numeric result using a Japanese style using sequential digital ideographs, using the appropriate character.
KANJINUM2	Formats a numeric result using the Japanese counting system. Corresponds to an ST_NumberFormat enumeration value of <code>japaneseCounting</code> .
KANJINUM3	Formats a numeric result using the Japanese legal counting system.
Lower	All letters are lowercase.
MERGEFORMAT	See the discussion following this table.
Ordinal	Formats a numeric result using lowercase ordinal Arabic numerals. Corresponds to an ST_NumberFormat enumeration value of <code>ordinal</code> .
OrdText	Formats a numeric result as lowercase ordinal text. Apart from being used to round off the whole number part, the fractional part is not used. Corresponds to an ST_NumberFormat enumeration value of <code>ordinalText</code> .
Roman	Formats a numeric result using uppercase Roman numerals. Corresponds to an ST_NumberFormat enumeration value of <code>upperRoman</code> .
roman	Formats a numeric result using lowercase Roman numerals. Corresponds to an ST_NumberFormat enumeration value of <code>lowerRoman</code> .
SBCHAR	Formats a numeric result using single-byte Arabic numbering. Corresponds to an ST_NumberFormat enumeration value of <code>decimalHalfWidth</code> .
THAIARABIC	Formats a numeric result using Thai numbers. Corresponds to an ST_NumberFormat enumeration value of <code>thaiNumbers</code> .
THAICARDTEXT	Formats a numeric result using sequential numbers from the Thai counting system. Corresponds to an ST_NumberFormat enumeration value of <code>thaiCounting</code> .
THAILETTER	Formats a numeric result using Thai letters. Corresponds to an ST_NumberFormat enumeration value of <code>thaiLetters</code> .
Upper	All letters are uppercase.
VIETCARDTEXT	Formats a numeric result using Vietnamese numerals. Corresponds to an ST_NumberFormat enumeration value of <code>vietnameseCounting</code> .
ZODIAC1	Formats a numeric result using sequential numerical traditional ideographs. Corresponds to an ST_NumberFormat enumeration value of <code>ideographTraditional</code> .
ZODIAC2	Formats a numeric result using sequential zodiac ideographs. Corresponds to an ST_NumberFormat enumeration value of <code>ideographZodiac</code> .
ZODIAC3	Formats a numeric result using sequential traditional zodiac ideographs. Corresponds to an ST_NumberFormat enumeration value of <code>ideographZodiacTraditional</code> .

The general formatting switch argument CHARFORMAT is used to set the visual appearance of a field's value by setting the first run in that field's *field-type* name to the desired state using any of the elements that can be directly nested inside the run property element, **rPr**.

If a format specified directly in the first run of a field's *field-type* name conflicts with a general formatting switch, the general formatting switch is ignored.

The general formatting switch argument MERGEFORMAT is used to apply formatting directly to part of a result such that when that result is updated, the formatting is preserved. The formatting is expressed in XML using an **rPr** element on the run that contains the most recently updated field result.

2.16.5 Field definitions

The set of fields is divided into the following functional categories:

Category	Description	Fields
Date and Time	Inserts the current date and/or time, or date and/or time of some kind of event.	CREATEDATE (§2.16.5.16), DATE (§2.16.5.18), EDITTIME (§2.16.5.21), PRINTDATE (§2.16.5.54), SAVEDATE (§2.16.5.60), TIME (§2.16.5.72)
Document Automation	Compares values and takes action based on outcome, run macros, and sends a code to a printer.	COMPARE (§2.16.5.15), DOCVARIABLE (§2.16.5.20), GOTOBUTTON (§2.16.5.29), IF (§2.16.5.32), MACROBUTTON (§2.16.5.41), PRINT (§2.16.5.53)
Document Information	Inserts or stores information about the document.	AUTHOR (§2.16.5.4), COMMENTS (§2.16.5.14), DOCPROPERTY (§2.16.5.19), FILENAME (§2.16.5.23), FILESIZE (§2.16.5.24), INFO (§2.16.5.36), KEYWORDS (§2.16.5.37), LASTSAVEDBY (§2.16.5.38), NUMCHARS (§2.16.5.48), NUMPAGES (§2.16.5.49), NUMWORDS (§2.16.5.50), SUBJECT (§2.16.5.67), TEMPLATE (§2.16.5.71), TITLE (§2.16.5.73)
Equations and Formulas	Defines formulas and calculates results; inserts symbols.	= <i>formula</i> (§2.16.3), ADVANCE (§2.16.5.2), EQ (§2.16.5.22), SYMBOL (§2.16.5.68)
Form Fields	Allows the insertion of form fields.	FORMCHECKBOX (§2.16.5.26), FORMDROPDOWN (§2.16.5.27), FORMTEXT (§2.16.5.28)
Index and Tables	Defines entries for, and builds, a table of contents, table of figures, and table of authorities.	INDEX (§2.16.5.35), RD (§2.16.5.57), TA (§2.16.5.69), TC (§2.16.5.70), TOA (§2.16.5.74), TOC (§2.16.5.75), XE (§2.16.5.79)
Links and References	Inserts information from another place in the same document, from a different document or file, or from an AutoText entry.	AUTOTEXT (§2.16.5.8), AUTOTEXTLIST (§2.16.5.9), BIBLIOGRAPHY (§2.16.5.11), CITATION (§2.16.5.13), HYPERLINK (§2.16.5.31), INCLUDEPICTURE (§2.16.5.33), INCLUDETTEXT (§2.16.5.34), LINK (§2.16.5.39), NOTEREF (§2.16.5.47), PAGEREF (§2.16.5.52), QUOTE (§2.16.5.56), REF (§2.16.5.58), STYLEREF (§2.16.5.66)
Mail Merge	Defines information that is to be used in a mail merge.	ADDRESSBLOCK (§2.16.5.1), ASK (§2.16.5.3), COMPARE (§2.16.5.15), DATABASE (§2.16.5.17), FILLIN (§2.16.5.25), GREETINGLINE (§2.16.5.30), IF (§2.16.5.32), MERGEFIELD (§2.16.5.42), MERGEREC (§2.16.5.43), MERGESEQ (§2.16.5.44), NEXT (§2.16.5.45), NEXTIF (§2.16.5.46), SET (§2.16.5.64), SKIPIF (§2.16.5.65)

Category	Description	Fields
Numbering	Specifies numbering for document items such as sections and pages; also bar codes.	AUTONUM (§2.16.5.5), AUTONUMLGL (§2.16.5.6), AUTONUMOUT (§2.16.5.7), BARCODE (§2.16.5.10), LISTNUM (§2.16.5.40), PAGE (§2.16.5.51), REVNUM (§2.16.5.59), SECTION (§2.16.5.61), SECTIONPAGES (§2.16.5.62), SEQ (§2.16.5.63)
User Information	Stores or inserts the name, initials, or address of the document user.	USERADDRESS (§2.16.5.76), USERINITIALS (§2.16.5.77), USERNAME (§2.16.5.78)

2.16.5.1 ADDRESSBLOCK

Syntax:

ADDRESSBLOCK [*switches*]

Description: Inserts a mail merge address block.

Field Value: The address block.

Switches: Zero or more of the following *field-specific-switches*.

\c <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies whether to include the name of the country/region: a value of 0 causes the country/region to be omitted; a value of 1 causes it to be included, and a value of 2 causes country/region to be included, but only if it is different from the value for \e.
\d	Specifies that the address is to be formatted according to the country/region of the recipient. If this switch is not used, then addresses are formatted according to some implementation-specific preference.
\e <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies which country/region to exclude from the address block. [Note: This is useful when your mailing contains a mix of domestic and international recipients. <i>end note</i>] To exclude the names of more than one country or region, use a \e switch for each one.
\f <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the name and address format by providing a template of merge-field placeholders.
\l <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the language ID used to format the address. The default is to use the language ID of the first character of the document.

2.16.5.2 ADVANCE

Syntax:

ADVANCE [*switches*]

Description: Moves the starting point of text that follows the field to the right or left, up or down, or to a specific horizontal or vertical position. The switches used by this field can cause text to overlap. Text will not display if it is moved to the previous or next page, or beyond the print margins of the current page.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

\d <i>field-argument</i>	Moves the text that follows the field down by the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> .
\l <i>field-argument</i>	Moves the text that follows the field left by the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> .
\r <i>field-argument</i>	Moves the text that follows the field right by the integral number of points specified

	by <i>text</i> in this switch's <i>field-argument</i> .
<code>\u <i>field-argument</i></code>	Moves the text that follows the field up by the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> .
<code>\x <i>field-argument</i></code>	Moves the text that follows the field the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> from the left edge of the column, frame, or text box.
<code>\y <i>field-argument</i></code>	Moves the text that follows the field the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> . This shift is the vertical position relative to the page. The entire line of text that contains the field is moved. This switch is ignored if it specifies a location outside the page margins or if the switch is used inside any of the following: table, text box, footnote, endnote, annotation, header, or footer.

2.16.5.3 ASK

Syntax:

ASK *field-argument-1* *field-argument-2* [*switches*]

field-argument-1:

field-argument

field-argument-2:

field-argument

Description: Prompts the user to enter information and assigns the bookmark designated by *field-argument-1* to represent the user's response. *text* in *field-argument-2* specifies the prompt text, which is displayed in a dialog box. The prompt is displayed each time the ASK field is updated. A response remains assigned to the bookmark until a new response is entered.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

<code>\d <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> specifies a default response if one is not entered. If no default response is specified, the most recent response is used. To specify a blank entry as the default, <i>field-argument</i> shall be "".
<code>\o</code>	When used in a mail merge main document, this causes the display of the prompt once instead of each time a new data record is merged. The same response is inserted in each resulting merged document.

2.16.5.4 AUTHOR

Syntax:

AUTHOR [*field-argument*] [*switch*]

Description: Retrieves, and optionally sets, the document author's name, as recorded in the **Creator** element of the Core File Properties part or, if *field-argument* is present, the name specified by *text* in *field-argument*. Specifying a *field-argument* shall change **Creator** to *text*.

Field Value: The document author's name.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.5 AUTONUM

Syntax:

AUTONUM [*switches*]

Description: In paragraphs formatted with one of the nine built-in heading styles, paragraph numbering restarts at 1 in each successive heading level. If headings that contain AUTONUM fields are followed by body text paragraphs that also contain AUTONUM fields, the paragraph numbering of the body text is restarted at 1 after each heading. If the headings don't contain AUTONUM fields, body text paragraphs that contain AUTONUM fields are numbered in a continuous, sequential series throughout the document. [Note: This field is supported for legacy reasons, It is recommended that LISTNUM (§2.16.5.40) be used instead. *end note*]

The XML generated for a complex field implementation shall not have the optional field value stored.

Field Value: A new paragraph number in ascending sequential order.

Switches: Zero or one of the *general-formatting-switches*, or zero or more of the following *field-specific-switches*.

<code>\s <i>field-argument</i></code>	<i>text</i> in this switch's <i>field-argument</i> specifies the separator character to be used. If \s is omitted, a period (.) is used.
---------------------------------------	--

2.16.5.6 AUTONUMLGL

Syntax:

AUTONUMLGL [*switches*]

Description: For legal and technical publications, use the nine built-in heading styles to format headings in the document, and then insert an AUTONUMLGL field at the beginning of each heading paragraph. The numbers reflect the heading levels that correspond to the heading styles. If an AUTONUMLGL field is inserted in paragraphs of body text paragraphs not formatted with built-in heading styles, the number of the preceding heading is included in the paragraph number. [Note: This field is supported for legacy reasons, It is recommended that LISTNUM (§2.16.5.40) be used instead. *end note*]

This field only makes sense in terms of multi-level headings. Given the following headings:

Heading 1
 Heading 2
 Heading 2
 Heading 1
 this field allows
 1. Heading 1
 1.1. Heading 2
 1.2. Heading 2
 2. Heading 1

At each level, the numbering sequence does two things—it increments specific to that level, and it includes the value from the previous level.

The XML generated for a complex field implementation shall not have the optional field value stored.

Field Value: A new paragraph number in ascending sequential order.

Switches: Zero or one of the *general-formatting-switches*, or zero or more of the following *field-specific-switches*.

<code>\e</code>	Removes the trailing separator (period).
<code>\s <i>field-argument</i></code>	<i>text</i> in this switch's <i>field-argument</i> specifies the separator character to be used. If \s is omitted, a period (.) is used.

2.16.5.7 AUTONUMOUT

Syntax:

AUTONUMOUT

Description: Use the nine built-in heading styles to format headings in the document, and then insert an AUTONUMOUT field at the beginning of each heading paragraph. The numbers reflect the heading levels that correspond to the heading styles. [Note: This field is supported for legacy reasons, It is recommended that LISTNUM (§2.16.5.40) be used instead. *end note*]

The XML generated for a complex field implementation shall not have the optional field value stored.

This field allows the numbering to be incremented based on the heading level. Given the following:

{AutoNumOut} Heading 1

{AutoNumOut} Heading 2

{AutoNumOut} Heading 2

{AutoNumOut} Heading 1

results in

I. Heading 1

A. Heading 2

B. Heading 2

II. Heading 1

Field Value: A paragraph number.

Switches: None.

2.16.5.8 AUTOTEXT

Syntax:

AUTOTEXT *field-argument*

Description: Inserts the AutoText entry whose name is specified by *text* in *field-argument*.

Regarding XML generation, the field result is the value of the autotext. [*Note:* This can be arbitrarily complex and involve VML *end note*]

Field Value: The specified AutoText entry.

Switches: None.

2.16.5.9 AUTOTEXTLIST

Syntax:

AUTOTEXTLIST *field-argument* [*switches*]

Description: Creates a shortcut menu based on AutoText entries in the active template. The list can vary based on the styles applied to the AutoText entries. *text* in *field-argument* is inserted into the document.

A complex field XML implementation shall be used, and the *field-argument* text shall be placed in one or more runs between the separate and end parts of the **fldChar** element.

Field Value: *text* in *field-argument*.

Switches: Zero or more of the following *field-specific-switches*.

\s <i>field-argument</i>	Specifies that the list is to contain entries based on the style specified by <i>text</i> in this switch's <i>field-argument</i> . Without this switch, entries of the current paragraph style appear. If there are no entries for the current style, all entries appear. The style can be a paragraph style or a character style.
\t <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the text to show in the ScreenTip.

2.16.5.10 BARCODE

Syntax:

BARCODE *field-argument* [*switch*]

Description: Produces a postal bar code is a machine-readable form of address used by the U.S. Postal Service. The barcode is in the form of either a POSTNET delivery-point bar code or a Facing Identification Mark (FIM). *text* in *field-argument* can be either a postal address or a bookmark name. In the case of a postal address, all that is needed is a 5-digit or 9-digit ZIP code; the rest of the address is superfluous.

Field Value: A postal bar code.

Switches: Zero or more of the following *field-specific-switches*.

\b	Indicates that <i>text</i> in <i>field-argument</i> is the name of a bookmark.
\f <i>field-argument</i>	Inserts a Facing Identification Mark (FIM). <i>text</i> in this switch's <i>field-argument</i> shall

	be either "A" (courtesy reply mark) or "C" (business reply mark).
\u	Indicates that <i>text</i> in <i>field-argument</i> is a U.S. postal address.

2.16.5.11 BIBLIOGRAPHY

Syntax:

BIBLIOGRAPHY [*switch*]

Description: Retrieves and displays the contents of the document's Bibliography part in the bibliographic style specified within the SelectedStyle attribute of the **Sources** (§Error! Reference source not found.) element of the Bibliography part.

Field Value: The formatted bibliographic data for all sources in the current document.

Switches: The following *field-specific-switch*.

\l <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the locale that shall be used in to format the bibliographic sources in the document that do not specify a locale using the LCID (§Error! Reference source not found.) element.
\f <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the locale that shall be used to filter the bibliographic data to only the sources in the document that have a value matching <i>field-argument</i> in the LCID (§Error! Reference source not found.) element.
\m <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies that only the source with a Tag (§Error! Reference source not found.) element value matching <i>field-argument</i> shall be displayed in the bibliography.

2.16.5.12 BIDIOUTLINE

Syntax:

BIDIOUTLINE

Description: Sets the output as being right-to-left. This field is like AUTONUMOUT (§2.16.5.7), except for differences in Arabic/Hebrew numbering

Field Value: A paragraph number.

Switches: None.

2.16.5.13 CITATION

Syntax:

CITATION *field-argument* [*switch*]

Description: Displays the contents of the **Source** (§Error! Reference source not found.) element with a **Tag** (§Error! Reference source not found.) element value matching *field-argument* using the bibliographic style specified within the SelectedStyle attribute of the **Sources** (§Error! Reference source not found.) element of the Bibliography part.

Field Value: The comments relating to the current document.

Switches: The following *field-specific-switch*.

\l <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the locale which shall be used in conjunction with the specified bibliographic style to format the citation in the document.
\f <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the prefix which shall be prepended to the citation.
\s <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the suffix which shall be appended to the citation.
\p <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the page number associated with the

	citation.
<code>\v <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> specifies the volume number associated with the citation.
<code>\n</code>	Specifies that the author information shall be suppressed from the citation.
<code>\t</code>	Specifies that the title information shall be suppressed from the citation.
<code>\y</code>	Specifies that the year information shall be suppressed from the citation.
<code>\m <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> specifies the Tag (\$Error! Reference source not found.) element value for another source to be included in this citation's field result.

2.16.5.14 COMMENTS

Syntax:

COMMENTS [*field-argument*] [*switch*]

Description: Retrieves, and optionally sets, the comments relating to the current document, as recorded in the **Description** element of the Core File Properties part or, if *field-argument* is present, the comments specified by *text* in *field-argument*. Specifying a field-argument shall change **Description** to *text*.

Field Value: The comments relating to the current document.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.15 COMPARE

Syntax:

COMPARE Expression-1 Operator Expression-2

Expression-1:

expression

Expression-2:

expression

Description: Compares the values designated by *Expression-1* and *Expression-2* using the operator designated by *Operator*. [Note: This field can be used to create compound logical comparisons with AND and OR functions in a formula, and then by using the result of the formula in an IF field. *end note*]

Operator can be any one of the six relational and equality operators specified for *operator* (§2.16.3.3).

If *Operator* is = or <>, *Expression-2* can contain a question mark (?) to represent any single character, or an asterisk (*) to represent any string of characters. The expression shall be enclosed in quotation marks so that it is compared as a character string. If an asterisk is used in *Expression-2*, the portion of *Expression-1* that corresponds to the asterisk, plus any remaining characters in *Expression-2*, shall NOT exceed 128 characters.

Field Value: 1 if the comparison is true, or 0 if the comparison is false.

Switches: None.

2.16.5.16 CREATEDATE

Syntax:

CREATEDATE[switches]

Description: Retrieves the date and time at which the document was created, as recorded in the **DateCreated** element of the Core File Properties part. By default, the Gregorian calendar is used and the *date-and-time-formatting-switch* used is implementation-defined.

Field Value: The date and time at which the document was created.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

<code>\h</code>	Use the Hijri/Lunar calendar.
<code>\s</code>	Use the Saka Era calendar.

2.16.5.17 DATABASE

Syntax:

DATABASE[switches]

Description: Inserts the results of a database query into a WordprocessingML table. If the number of columns is 62 or more, the field inserts the results of a query in columns separated by tabs. The DATABASE field contains all the information needed to connect to a database and perform an SQL query. Each time the field is updated, the database is queried again.

Field Value: The results of a database query as a WordprocessingML table.

Switches: Zero or more of the following *field-specific-switches*.

\b <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies which attributes of the format set by the \l switch are to be applied to the table. If the \l switch is blank, the \b switch value shall be 16 (AutoFit). <i>text</i> can have a value that is the bitwise-or of any combination of the following: 0, None 1, Borders 2, Shading 4, Font 8, Color 16, AutoFit 32, Heading Rows 64, Last Row 128, First Column 256, Last Column
\c <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a connection to the data.
\d <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the complete path and file name of the database. Used for all database queries except a query to an SQL database table using ODBC.
\f <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the integral record number of the first data record to insert
\h	Inserts the field names from the database as column headings in the resulting table.
\l <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the format that is to be applied to the result of the database query. If this switch is used and the \b switch doesn't specify the table attributes, an unformatted table is inserted.
\o <i>field-argument</i>	Inserts data at the beginning of a merge. By adding the \o switch to the database field, it will only get the data for the database field at the beginning of a merge instead of once for each record merged. This is a performance optimization and should only be used when the database field doesn't rely on record specific information to gather.
\s <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a set of SQL instructions. Each quotation mark in the instructions shall be preceded by a backslash (\).
\t <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the integral record number of the last data record to insert.

2.16.5.18 DATE

Syntax:

DATE [switches]

Description: Retrieves the current date and time. By default, the Gregorian calendar is used, and the *date-and-time-formatting-switch* used is implementation-defined.

Field Value: The current date and time.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri/Lunar calendar.
\l	If no <i>date-and-time-formatting-switch</i> is used, the date shall use the date format last used by the hosting application when inserting a new DATE field.
\s	Use the Saka Era calendar.

[Note: For some *date-and-time-formatting-switches*, the DATE and TIME (§2.16.5.72) fields can produce the same result. end note]

2.16.5.19 DOCPROPERTY

Syntax:

DOCPROPERTY docprop-category [field-argument] [switches]

docprop-category:

AUTHOR | *BYTES* | *CATEGORY* | *CHARACTERS* | *CHARACTERSWITHSPACES*
 | *COMMENTS* | *COMPANY* | *CREATETIME* | *HYPERLINKBASE* | *KEYWORDS*
 | *LASTPRINTED* / *LASTSAVEDBY* | *LASTSAVEDTIME* | *LINES* | *MANAGER*
 | *NAMEOFAPPLICATION* | *ODMADOCID* | *PAGES* | *PARAGRAPHS* | *REVISIONNUMBER*
 | *SECURITY* | *SUBJECT* | *TEMPLATE* | *TITLE* | *TOTALEDITINGTIME* | *WORDS*

Description: Retrieves the indicated document information. For some combinations of DOCPROPERTY and *docprop-category*, there is an equivalent field, in which case, the description for the combination can be obtained from that field. For those combinations not having an equivalent field, the description is shown directly. When used directly, some of the equivalent fields allow the value of the designated property to be changed. However, when the corresponding DOCPROPERTY field is used, such values shall not be changed. This is indicated in the following table by "Read-only operation."

docprop-category	Corresponding Field	Description
AUTHOR	AUTHOR (§2.16.5.4)	Read-only operation.
BYTES	FILESIZE (§2.16.5.24)	
CATEGORY	No equivalent	The contents of the Category element of the Core File Properties part.
CHARACTERS	NUMCHARS (§2.16.5.48)	
CHARACTERSWITHSPACES	No equivalent	Like NUMCHARS, but includes all white space characters as well.
COMMENTS	COMMENTS (§2.16.5.11)	Read-only operation.
COMPANY	No equivalent	The contents of the Company element of the Application-Defined File Properties part.
CREATETIME	CREATEDATE (§2.16.5.16)	

docprop-category	Corresponding Field	Description
HYPERLINKBASE	No equivalent	The contents of the HyperlinkBase element of the Application-Defined File Properties part.
KEYWORDS	No equivalent	The contents of the Keywords element of the Core File Properties part.
LASTPRINTED	PRINTDATE (§2.16.5.54)	
LASTSAVEDBY	LASTSAVEDBY (§2.16.5.38)	
LASTSAVEDTIME	SAVEDATE (§2.16.5.60)	
LINES	No equivalent	The contents of the Lines element of the Application-Defined File Properties part.
MANAGER	No equivalent	The contents of the Manager element of the Application-Defined File Properties part.
NAMEOFAPPLICATION	No equivalent	The contents of the Application element of the Application-Defined File Properties part.
ODMADOCID		
PAGES	NUMPAGES (§2.16.5.49)	
PARAGRAPHS	No equivalent	The contents of the Paragraphs element of the Application-Defined File Properties part.
REVISIONNUMBER	REVNUM (§2.16.5.59)	
SECURITY	No equivalent	The contents of the DocSecurity element of the Application-Defined File Properties part.
SUBJECT	SUBJECT (§2.16.5.67)	Read-only operation.
TEMPLATE	TEMPLATE (§2.16.5.71)	
TITLE	TITLE (§2.16.5.73)	Read-only operation.
TOTALEDITINGTIME	EDITTIME (§2.16.5.21)	
WORDS	No equivalent	The contents of the Words element of the Application-Defined File Properties part.

Field Value: The indicated document information.

2.16.5.20 DOCVARIABLE

Syntax:

DOCVARIABLE*field-argument*

Description: Inserts the string assigned to the document variable designated by *text* in *field-argument*. Each WordprocessingML document has a collection of variables. This field is used to access and display the contents of **docVar** (§2.15.1.30) elements in the Document Settings part.

Field Value: The value of the specified document variable.

Switches: None.

2.16.5.21 EDITTIME

Syntax:

EDITTIME[*switch*]

Description: Retrieves the total editing time, in minutes, since the document was created, as recorded in the **TotalTime** element of the Application-Defined File Properties part. By default, the *numeric-formatting-switch* or *general-formatting-switch* used is implementation-defined.

Field Value: The total editing time, in minutes.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

2.16.5.22 EQ

Syntax:

EQ*eq-primary-switch* [*switches*] ([*eq-argument-list*])

eq-argument-list is a list of arguments separated using a separator character. For implementations using a period (.) as the radix point, the separator character is a comma (,). For implementations using a comma (,) as the radix point, the separator character is a semicolon (;).

Description: Computes the specified mathematical equation.

Field Value: The result of the specified mathematical equation. [Note: The result of an EQ field can be used as an argument in another EQ field's *eq-argument-list*. end note]

Switches: One of the following *eq-primary-switches*: \a, \b, \d, \f, \i, \l, \o, \r, \s, and \x. Each of these switches has one or more subswitches, as shown below.

\a produces an array using the argument values in *eq-argument-list* (which are in row-major order) and the *field-specific-switches* below:

\ac	Alignment is centered in each array column.
\al	Alignment is left in each array column.
\ar	Alignment is right in each array column.
\co <i>field-argument</i>	The number of columns in the array is specified by <i>text</i> in this switch's <i>field-argument</i> . In the absence of this switch, the number is 1.
\hs <i>field-argument</i>	Adds the integral number of points of horizontal spacing specified by <i>text</i> in this switch's <i>field-argument</i> between columns.
\vs <i>field-argument</i>	Adds the integral number of points of vertical spacing specified by <i>text</i> in this switch's <i>field-argument</i> between lines.

\b brackets the single element in *eq-argument-list* in a size appropriate for that element. The default form of brackets is parentheses. The *field-specific-switches* below may be used:

\bc \char	Uses the character designated by <i>char</i> as both the left and right bracket character. However, if <i>char</i> is {, [, (, or <, that character is used for the left bracket, and },],), or >, respectively, is used for right bracket.
\lc \char	Uses the character designated by <i>char</i> as the left bracket character.

<code>\rc</code> <i>char</i>	Uses the character designated by <i>char</i> as the right bracket character.
------------------------------	--

`\d` Controls where the next character following the EQ field is drawn (that is, the displacement). *eq-argument-list* shall have no arguments. The *field-specific-switches* below may be used:

<code>\ba</code> <i>field-argument</i>	Draws to the left (backward) the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> .
<code>\fo</code> <i>field-argument</i>	Draws to the right (forward) the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> .
<code>\li</code>	Underlines the space up to the next character.

`\f` Creates a fraction with the first argument as numerator and the second argument as denominator, centered above and below the division line, respectively. *eq-argument-list* shall have exactly two arguments. There are no *field-specific-switches* for this switch.

`\i` Creates an integral using the specified or default symbol and three elements. The first argument is the lower limit, the second is the upper limit, and the third is the integrand. *eq-argument-list* shall have exactly three arguments. The *field-specific-switches* below may be used:

<code>\fc</code> <i>char</i>	Uses the character designated by <i>char</i> as the fixed-height character for the symbol.
<code>\in</code>	Uses an inline format with the limits displayed to the right of the symbol instead of above and below it.
<code>\pr</code>	Uses the symbol Capital pi and creates a product.
<code>\su</code>	Uses the symbol Capital sigma and creates a summation.
<code>\vc</code> <i>char</i>	Uses the character designated by <i>char</i> as the variable-height character for the symbol. The symbol matches the height of the third argument.

`\lf` Creates a list from an arbitrary number of arguments. There are no *field-specific-switches* for this switch.

`\o` Using an arbitrary number of arguments, displays each successive argument on top of the previous one. Each character is displayed within an invisible character box, with the switches being available to align the boxes on top of one another.

The *field-specific-switches* below may be used:

<code>\ac</code>	Alignment character box center (the default).
<code>\al</code>	Alignment character box left.
<code>\ar</code>	Alignment character box right.

`\r` Creates a radical. *eq-argument-list* shall have either one or two arguments. If it has one argument, the result is the square root of that argument. If it has two arguments, the result is the *n*th root of the second argument, where *n* is the first argument. There are no *field-specific-switches* for this switch.

`\s` Creates a subscript or superscript. One or more arguments are permitted. If more than one element is specified, the elements are stacked and left-aligned. The *field-specific-switches* below may be used:

<code>\ai</code> <i>field-argument</i>	Adds space above a line in a paragraph by the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> . The default is 2 points.
<code>\di</code> <i>field-argument</i>	Adds space below a line in a paragraph by the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> .
<code>\do</code> <i>field-argument</i>	Moves a single argument below the adjacent text by the integral number of points specified by <i>text</i> in this switch's <i>field-argument</i> . The default is 2 points.
<code>\up</code> <i>field-argument</i>	Moves a single argument above the adjacent text by the integral number of

	points specified by <i>text</i> in this switch's <i>field-argument</i> .
--	--

\x Creates one or more border segments around a single argument. By default, all four borders are added. *eq-argument-list* shall have no arguments. The *field-specific-switches* below may be used:

\bo	Draws a horizontal border below the argument.
\le	Draws a vertical border to the left of the argument.
\ri	Draws a vertical border to the right of the argument.
\to	Draws a horizontal border above the argument.

2.16.5.23 FILENAME

Syntax:

FILENAME [*switch*]

Description: Retrieves the name of the current document as stored on disk.

Field Value: The name of the current document.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper, and zero or one of the following *field-specific-switches*.

\p	Include the full file path name.
----	----------------------------------

2.16.5.24 FILESIZE

Syntax:

FILESIZE [*switches*]

Description: Retrieves the size of the current document in bytes. [*Note:* This information is not stored inside the document's XML. It needs to be obtained from the file system. *end note*]

Field Value: The size of the current document in bytes.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch* and zero or one of the following *field-specific-switches*.

\k	Round to the nearest kilobyte.
\m	Round to the nearest megabyte.

2.16.5.25 FILLIN

Syntax:

FILLIN [*field-argument*] [*switch*]

Description: Prompts the user to enter text. *text* in *field-argument* contains the prompt. The prompt is displayed each time the field is updated. When a new document is created based on a template containing FILLIN fields, those fields are updated automatically.

Field Value: The user's response.

Switches: Zero or more of the following *field-specific-switches*.

\d <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a default response if one is not entered. If no default response is specified, the most recent response is used. To specify a blank entry as the default, <i>field-argument</i> shall be "".
\o	When used in a mail merge main document, this causes the display of the prompt once instead of each time a new data record is merged. The same

	response is inserted in each resulting merged document.
--	---

2.16.5.26 FORMCHECKBOX

Syntax:

FORMCHECKBOX

Description: Inserts a check box style form field which, when the editing of form fields is enabled using the **documentProtection** element (§2.15.1.28), can be checked and unchecked. An instance of this field shall be accompanied by a use of the **ffData** element (§2.16.17) which contains the form field's properties.

Field Value: A check box based on the properties of the **ffData** element (§2.16.17).

Switches: None.

2.16.5.27 FORMDROPDOWN

Syntax:

FORMDROPDOWN

Description: Inserts a drop-down list style form field which, when the editing of form fields is enabled using the **documentProtection** element (§2.15.1.28), can be used to select an entry in the list. An instance of this field shall be accompanied by a use of the **ffData** element (§2.16.17) which contains the form field's properties.

Field Value: A drop-down list based on the properties of the **ffData** element (§2.16.17).

Switches: None.

2.16.5.28 FORMTEXT

Syntax:

FORMTEXT

Description: Inserts a text box style form field which, when the editing of form fields is enabled using the **documentProtection** element (§2.15.1.28), can be typed into. An instance of this field shall be accompanied by a use of the **ffData** element (§2.16.17) which contains the form field's properties.

Field Value: A text box based on the properties of the **ffData** element (§2.16.17).

Switches: None.

2.16.5.29 GOTOBUTTON

Syntax:

GOTOBUTTON field-argument-1 field-argument-2

field-argument-1:
expression

field-argument-2:
expression

Description: Inserts a jump command, such that when it is activated, the insertion point of the document is moved to the location specified by *text* in *field-argument-1*. *text* can be a bookmark, a page number, or some other item (as described below). The page number can be a reference resulting from a REF field. The other items than can be locations are:

a n	annotation
f n	footnote
l n	line
p n	page
s n	section

where n is an integer that designates the n th occurrence of the corresponding item (which is not necessarily the item numbered n).

text in *field-argument-2* is the text or graphic "button" that appears in the document, such that it can be selected to activate the jump. [Note: The BOOKMARK and INCLUDEPICTURE fields make for some interesting button possibilities. *end note*] The text or graphic shall appear on one line in the field result; otherwise, an error occurs.

Field Value: None.

Switches: None.

2.16.5.30 GREETINGLINE

Syntax:

GREETINGLINE [*switches*]

Description: Inserts a mail merge greeting line.

Field Value: The greeting line.

Switches: Zero or more of the following *field-specific-switches*.

\c <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the text to include in the merge field if the name field in the data source is blank.
\c <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the format of the name included in the field.
\l <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the language ID used to format the name. it defaults to the language ID of the first character of the document.

2.16.5.31 HYPERLINK

Syntax:

HYPERLINK *field-argument* [*switches*]

Description: When selected, causes control to jump to the location specified by *text* in *field-argument*. That location can be a bookmark or a URL.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

\l <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies a location in the file, such as a bookmark, where this hyperlink will jump.
\m	Appends coordinates to a hyperlink for a server-side image map.
\n	Causes the destination site to be opened in a new window.
\o <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the ScreenTip text for the hyperlink.
\t <i>field-argument</i>	<p><i>text</i> in this switch's <i>field-argument</i> specifies the target to which the link should be redirected. Use this switch to link from a frames page to a page that you want to appear outside of the frames page. The permitted values for <i>text</i> are:</p> <ul style="list-style-type: none"> • <code>_top</code>, whole page (the default) • <code>_self</code>, same frame • <code>_blank</code>, new window • <code>_parent</code>, parent frame

2.16.5.32 IF

Syntax:

I F *Expression-1* *Operator* *Expression-2* *field-argument-1* *field-argument-2*

Expression-1:
expression

Expression-2:
expression

field-argument-1:
expression

field-argument-2:
expression

Description: Compares the values designated by *Expression-1* and *Expression-2* using the operator designated by *Operator*.

Operator can be any one of the six relational and equality operators specified for *operator* (§2.16.3.3).

If *Operator* is = or <>, *Expression-2* can contain a question mark (?) to represent any single character, or an asterisk (*) to represent any string of characters. The expression shall be enclosed in quotation marks so that it is compared as a character string. If an asterisk is used in *Expression-2*, the portion of *Expression-1* that corresponds to the asterisk, plus any remaining characters in *Expression-2*, shall NOT exceed 128 characters.

Field Value: *field-argument-1* if the comparison is true; otherwise, *field-argument-2*.

Switches: None.

2.16.5.33 INCLUDEPICTURE

Syntax:

I NCLUDEPI CTURE *field-argument* [*switches*]

Description: Retrieves the picture contained in the document named by *field-argument*. If *field-argument* contains white space, it shall be enclosed in double quotes. If *field-argument* contains any backslash characters, each one shall be preceded directly by another backslash character.

Field Value: The specified picture.

Switches: Zero or more of the following *field-specific-switches*.

\c <i>field-argument</i>	If <i>text</i> in this switch's <i>field-argument</i> identifies the graphics filter to be used.
\d	Reduce the file size by not storing graphics data with the document.

2.16.5.34 INCLUDETTEXT

Syntax:

I NCLUDETTEXT *field-argument-1* [*field-argument-2*] [*switches*]

field-argument-1:
field-argument

field-argument-2:
field-argument

Description: Inserts all or part of the text and graphics contained in the document named by *field-argument-1*. If the document is a WordprocessingML document, the portion marked by the optional bookmark *field-argument-2* is inserted. If no such bookmark is specified here, the whole document is inserted. If the document is an XML file, the fragment referred to by an XPath expression in the \x switch is inserted. If no such switch is specified, the whole XML file is inserted.

If *field-argument-1* contains white space, it shall be enclosed in double quotes. If *field-argument-1* contains any backslash characters, each one shall be preceded directly by another backslash character.

Field Value: The specified text and graphics.

Switches: Zero or more of the following *field-specific-switches*.

\!	Prevents this field from being updated unless all fields in the inserted text are first updated in their original document.
\c <i>field-argument</i>	Specifies that the file specified by <i>field-argument-2</i> shall be processed by a document filter whose name matches the corresponding <i>field-argument</i> value. Possible <i>field-argument</i> values are implementation-defined.
\n <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a namespace mapping for XPath queries. This switch is required if the \x switch refers to an element by name in an XML file that declares a namespace.
\t <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies an XSLT for formatting XML data.
\x <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the XPath for returning a fragment of data in an XML file.

2.16.5.35 INDEX

Syntax:

I NDEX [*switches*]

Description: Builds an index using the index entries specified by XE fields (§2.16.5.79), and inserts that index at this place in the document. Each index entry and subentry is a separate paragraph unless the \r switch is used, in which case, an index entry and all its subentries together make up a paragraph.

Field Value: The index.

Switches: Zero or more of the following *field-specific-switches*.

\b <i>field-argument</i>	Builds an index for the portion of the document marked by the bookmark indicated by <i>text</i> in this switch's <i>field-argument</i> .
\c <i>field-argument</i>	Builds an index having the number of columns per page specified by <i>text</i> in this switch's <i>field-argument</i> . That number can be 1–4. Without this switch, the number of columns is 1.
\d <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a sequence of characters that is used to separate sequence numbers and page numbers when the \s switch is used. By default, a hyphen (-) is used.
\e <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a sequence of characters that is used to separate an index entry and its first page number. By default, a comma (,) and space sequence is used. If <i>text</i> contains a horizontal tab character, the page number list is right justified in the column.
\f <i>field-argument</i>	Builds an index using only those entries having the entry type (§2.16.5.79) specified by <i>text</i> in this switch's <i>field-argument</i> . Without this switch, all entries included.
\g <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies a sequence of characters that is used to separate the start and end of a page range. By default, an en dash is used.
\h <i>field-argument</i>	Builds an index such that the <i>text</i> in this switch's <i>field-argument</i> occurs as a heading—formatted with the Index Heading style—at the start of each set of entries for any given letter. If the first letter in <i>text</i> is A or a, that letter is replaced with the corresponding letter for each letter set. To replace the default heading with a blank line, use a space as <i>text</i> .

<code>\k <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> specifies a sequence of characters that is used to separate an index entry and its cross reference (as produced by an XE entry (§2.16.5.79) having a <code>\t</code> switch). By default, a period (.) and space sequence is used.
<code>\l <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> specifies a sequence of characters that is used to separate two page numbers in a page number list. By default, a comma (,) and space sequence is used.
<code>\p <i>field-argument</i></code>	Builds an index using only those entries whose first letter is in the range of letters specified by <i>text</i> in this switch's <i>field-argument</i> . The letter range has the form <i>startLetter</i> - <i>endLetter</i> . If <i>startLetter</i> is !, entries whose first character is not a letter, are also included, as are the letters starting from A. The letters in the range can be either upper- or lowercase.
<code>\r</code>	Runs subentries into the same line as the main entry. Colons (:) separate main entries from subentries; semicolons (;) separate subentries.
<code>\s <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> is used as a sequence name, and the sequence number is included along with the page number, these numbers being separated by a hyphen (-), by default. Use the <code>\d</code> switch to specify a separator character other than the default.
<code>\y</code>	Enables the use of yomi text for index entries.
<code>\z <i>field-argument</i></code>	The <i>text</i> in this switch's <i>field-argument</i> specifies the language ID used to generate the index as defined in the ST_LangCode (§2.18.51) simple type.

2.16.5.36 INFO

Syntax:

INFO **info-category** [*field-argument*] [*switches*]

info-category:

AUTHOR | **COMMENTS** | **CREATEDATE** | **EDITTIME** | **FILENAME** | **FILESIZE** |
KEYWORDS | **LASTSAVEDBY** | **NUMCHARS** | **NUMPAGES** | **NUMWORDS** | **PRINTDATE** |
REVNUM | **SAVEDATE** | **SUBJECT** | **TEMPLATE** | **TITLE**

A field of this kind is treated as if **INFO** was omitted and *info-category* was a *field-type name*.

2.16.5.37 KEYWORDS

Syntax:

KEYWORDS [*field-argument*] [*switch*]

Description: Retrieves, and optionally sets, the document's keywords, as recorded in the **Keywords** element of the Core File Properties part or, if *field-argument* is present, the subject specified by *text* in *field-argument*. Specifying a *field-argument* shall change **Keywords** to *text*. The **Keywords** element contains a string of text whose format and semantics is unspecified by this Office Open XML Standard.

Field Value: The document's keywords

Switches: One of the following *general-formatting-switches*: `* Caps`, `* FirstCap`, `* Lower`, or `* Upper`.

2.16.5.38 LASTSAVEDBY

Syntax:

LASTSAVEDBY [*switch*]

Description: Retrieves the name of the user who last modified and saved the current document, as recorded in the **LastModifiedBy** element of the Core File Properties part.

Field Value: The name of the user who last modified and saved the current document.

Switches: One of the following *general-formatting-switches*: `* Caps`, `* FirstCap`, `* Lower`, or `* Upper`.

2.16.5.39 LINK

Syntax:

LINK *field-argument-1 field-argument-2 [field-argument-3] [switches]*

field-argument-1:
field-argument

field-argument-2:
field-argument

field-argument-3:
field-argument

Description: For information copied from another application, this field links that information to its original source file using OLE. The application type of the link information is specified by *field-argument-1*. The name and location of the source file is specified by *field-argument-2*. *field-argument-3* specifies the portion of the source file that's being linked.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

\a	Causes this field to be updated automatically.
\b	Inserts the linked object as a bitmap.
\d	Don't store the graphic data with the document, thus reducing the file size.
\f <i>field-argument</i>	Causes the linked object to update its formatting in a particular way, according to the integral value of <i>text</i> in this switch's <i>field-argument</i> . The possible values are: <ul style="list-style-type: none"> • 0 Maintain the formatting of the source file • 1 Not supported • 2 Match the formatting of the destination document • 3 Not supported • 4 Maintain the formatting of the source file, if the source file is a SpreadsheetML workbook • 5 Match the formatting of the destination document, if the source file is a SpreadsheetML workbook
\h	Inserts the linked object as HTML format text.
\p	Inserts the linked object as a picture.
\r	Inserts the linked object in rich-text format (RTF).
\t	Inserts the linked object in text-only format.
\u	Inserts the linked object as Unicode text.

2.16.5.40 LISTNUM

Syntax:

LISTNUM[*field-argument*] [*switches*]

Description: Computes the next integral number from the current or a specific series, or a specific number from the next or specific series. This field can be used anywhere in a paragraph, not just at its start. A LISTNUM field can be incorporated into numbering from a simple or outline-numbered list. *text* in *field-argument* is used to associate a LISTNUM field with a specific list. To emulate the behavior of the AUTONUM (§2.16.5.5), AUTONUMLGL (§2.16.5.6), and AUTONUMOUT (§2.16.5.7) fields, use the list names NumberDefault, LegalDefault, and OutlineDefault names, respectively. By default, the NumberFormat list is used.

The XML generated for a complex field implementation shall not have the optional field value stored. There are nine levels of list, and, assuming \s 1 for each, the result style used for each is as follows:

1	1)	4	(1)	7	1.
2	a)	5	(a)	8	a.
3	iii)	6	(iii)	9	iii.

Field Value: The next integral number from the current or a specific series, or a specific number from the next or specific series.

Switches: Zero or more of the following *field-specific-switches*.

\l <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> is an integer that specifies the level in the list, overriding the default behavior of the field. If \l is omitted, for a new series, the default value is 3; otherwise, the current level is continued.
\s <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> is an integer that specifies the starting value for this field. If \s is omitted, for a new series, the default value is 1; otherwise, the current series numbering is continued.

2.16.5.41 MACROBUTTON

Syntax:

MACROBUTTON *field-argument-1 field-argument-2*

field-argument-1:

field-argument

field-argument-2:

field-argument

Description: Allows the macro or command designated by *text* in *field-argument-1* to be run. *text* in *field-argument-2* designates the text or graphic to appear as the "button" that is selected to run the macro or command.

Field Value: *field-argument-2*.

Switches: None.

2.16.5.42 MERGEFIELD

Syntax:

MERGEFIELD *field-argument* [*switch*]

Description: Retrieves the name of a data field designated by *text* in *field-argument* within the merge characters in a mail merge main document. When the main document is merged with the selected data source, information from the specified data field is inserted in place of the merge field.

The name designated by *text* shall match exactly the field name in the header record.

Field Value: The name of a data field designated by *text* in *field-argument*.

Switches: Zero or more of the following *field-specific-switches*.

\b <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the text to be inserted before the MERGEFIELD field if the field is not blank.
\f <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> specifies the text to be inserted after the MERGEFIELD field if the field is not blank.
\m	Specifies that the MERGEFIELD field is a mapped field.
\v	Enables character conversion for vertical formatting.

2.16.5.43 MERGEREC

Syntax:
MERGEREC

Description: Results in «MERGEREC». Use this in a mail merge to print the number of the corresponding merged data record in each resulting merged document. The number reflects the sequential order of the data records that were selected and possibly sorted for merging with the active main document. It does not indicate the actual order of the records as they occur in the physical data source. [Note: A personnel database might contain thousands of records. However, to send a form letter to employees who've reached their five-year anniversary with your company, you'd select as your data source only the records of those five-year employees, a much smaller set of records. To print a physical record number, you must include a record number field in the data source and insert the corresponding merge field in the main document. *end note*]

Field Value: «MERGEREC».

Switches: None.

2.16.5.44 MERGESEQ

Syntax:
MERGESEQ

Description: Counts the number of data records that were successfully merged with the main document. Merged records are numbered starting from 1 each time documents are merged. [Note: The number might be different from the value inserted by the MERGEREC field. *end note*]

Field Value: The number of data records that were successfully merged with the main document.

Switches: None.

2.16.5.45 NEXT

Syntax:
NEXT

Description: Merges the next data record into the current resulting merged document, rather than starting a new merged document. [Note: This field is used when setting up a mailing label and envelope main document during a mail merge. *end note*]

Field Value: None.

Switches: None.

2.16.5.46 NEXTIF

Syntax:

NEXTIF Expression-1 Operator Expression-2

Expression-1:
expression

Expression-2:
expression

Description: Compares the values designated by *Expression-1* and *Expression-2* using the operator designated by *Operator*. If the comparison is true, the next data record is merged into the current merge document. (Merge fields that follow the NEXTIF in the main document are replaced by values from the next data record rather than the current data record.) If the comparison is false, the next data record is merged into a new merge document.

Operator can be any one of the six relational and equality operators specified for *operator* (§2.16.3.3).

A NEXTIF fields shall NOT be used in a footnote, an endnote, annotation, a header, a footer, or a data source. A NEXTIF field shall NOT be nested within any field.

Field Value: None.

Switches: None.

2.16.5.47 NOTEREF

Syntax:

NOTEREF *field-argument* [*switches*]

Description: Inserts the mark of the footnote or endnote that is marked by the bookmark specified by *text* in *field-argument*.

Field Value: The mark of the footnote or endnote.

Switches: Zero or more of the following *field-specific-switches*.

\f	For a footnote, inserts the reference mark with the same character formatting as the Footnote Reference style. For an endnote, inserts the reference mark with the same character formatting as the Endnote Reference style.
\h	Inserts a hyperlink to the bookmarked endnote or footnote.
\p	Inserts the relative position of the footnote or endnote. If the NOTEREF field occurs before the bookmark, the result is "below". If the NOTEREF field occurs after the bookmark, the result is "above".

2.16.5.48 NUMCHARS

Syntax:

NUMCHARS [*switch*]

Description: Retrieves the number of characters in the current document, as recorded in the **Characters** element of the Application-Defined File Properties part.

Field Value: The number of characters in the current document.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

2.16.5.49 NUMPAGES

Syntax:

NUMPAGES [*switch*]

Description: Retrieves the number of pages in the current document, as recorded in the **Pages** element of the Application-Defined File Properties part.

Field Value: The number of pages in the current document.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

2.16.5.50 NUMWORDS

Syntax:

NUMWORDS [*switch*]

Description: Retrieves the number of words in the current document, as recorded in the **Words** element of the Application-Defined File Properties part.

Field Value: The number of words in the current document.

Switches: Zero or one *numeric-formatting-switch* or *general-formatting-switch*.

2.16.5.51 PAGE

Syntax:

PAGE [*switches*]

Description: Retrieves the number of the current page.

Field Value: The number of the current page.

Switches: Zero or more *general-formatting-switches*.

2.16.5.52 PAGeref

Syntax:

PAGeref *field-argument* [*switches*]

Description: Inserts the number of the page containing the bookmark specified by *text* in *field-argument* for a cross-reference.

Field Value: The number of the page containing the bookmark.

Switches: Zero or one of the *general-formatting-switches*, zero or one of the *numeric-formatting-switches*, and zero or more of the following *field-specific-switches*.

\h	Creates a hyperlink to the bookmarked paragraph.
\p	Causes the field to display its position relative to the source bookmark. If the PAGeref field is on the same page as the bookmark, it omits "on page #" and returns "above" or "below" only. If the PAGeref field is not on the same page as the bookmark, the string "on page #" is used.

2.16.5.53 PRINT

Syntax:

PRINT *field-argument* [*switches*]

Description: Sends the printer-specific control code characters specified by *text* in *field-argument* to the selected printer.

Field Value: None.

Switches: Zero or more of the following *field-specific-switches*.

\p <i>field-argument-1</i> <i>field-argument-2</i>	<p>Allows PostScript strings to be sent to the printer as native PostScript codes. PostScript commands embedded in the document are carried out in the order in which they are inserted.</p> <p>The y-coordinate space used for PostScript commands is as follows: The graphics origin (0,0) is in the lower-left corner of the page, and the positive directions are up and to the right. PostScript drawing instructions take place within a drawing rectangle. The graphics origin is translated to the lower-left corner of the drawing rectangle.</p> <p><i>text</i> in this switch's <i>field-argument-1</i> defines the drawing rectangle on which the subsequent PostScript instructions operate.</p> <p><i>text</i> in this switch's <i>field-argument-2</i> contains the PostScript instructions.</p>
---	---

2.16.5.54 PRINTDATE

Syntax:

PRINTDATE [*switches*]

Description: Retrieves the date and time on which the document was last printed, as recorded in the **LastPrinted** element of the Core File Properties part. By default, the Gregorian calendar is used and the *date-and-time-formatting-switch* used is implementation-defined. For a document that has never been printed, the date and time corresponds to 0000-00-00T00:00:00 local time and each text component is XXX.

Field Value: The date and time on which the document was last printed.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri/Lunar calendar.
\s	Use the Saka Era calendar.

2.16.5.55 PRIVATE

Syntax:

PRIVATE

Description: Provides a private storage area. This field is used to store data for documents converted from other file formats. The field contains data needed for converting a document back to its original file format.

A PRIVATE field is formatted as hidden text.

Field Value: None.

Switches: None.

2.16.5.56 QUOTE

Syntax:

QUOTE *field-argument*

Description: Retrieves the text specified by *text* in *field-argument*. This text can include any other fields except AUTONUM, AUTONUMGL, AUTONUMOUT, and SYMBOL.

Field Value: The specified text.

Switches: One or more of the *date-and-time-formatting-switch*, *general-formatting-switch*, or *date-and-time-formatting-switches*, depending on the type of *field-argument*.

2.16.5.57 RD

Syntax:

RD *field-argument* [*switch*]

Description: *field-argument* identifies a file to include when creating a table of contents, a table of authorities, or an index using a TOC (§2.16.5.75), TOA (§2.16.5.74), or INDEX field (§2.16.5.35). RD fields that reference a series of files must be in the same order as the files in the final document. If the location includes a long file name containing spaces, *field-argument* shall contain delimiting quotes. A single backslash in the file path shall be preceded directly by a backslash.

For a complex field implementation in XML the optional field-value storage is not needed.

Field Value: None.

Switches: One of the following *field-specific-switches*:

\p	Indicates that the path is relative to the current document.
----	--

2.16.5.58 REF

Syntax:

[REF] *field-argument* [*switches*]

Description: Inserts the text or graphics represented by the bookmark specified by *text* in *field-argument*. The bookmark shall be defined in the current document. Provided the bookmark name is not exactly the same as a field name, the REF prefix can be omitted. If the text marked by the bookmark contains a paragraph mark, the text preceding the REF field assumes the formatting of the paragraph in the bookmark.

Field Value: The specified text or graphics.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper, and zero or one of the following *field-specific-switches*.

\d <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the character sequence that is used to separate sequence numbers and page numbers.
\f	Increments footnote, endnote, and annotation numbers that are marked by the bookmark, and inserts the corresponding footnote, endnote, and comment text.
\h	Creates a hyperlink to the bookmarked paragraph.

\n	For a referenced paragraph, causes the field result to have the entire paragraph number without trailing periods. No information about prior levels is displayed unless it is included as part of the current level.
\p	Causes the field result to contain the position relative to the source bookmark using the word "above" or "below." If the REF field appears in the document before the bookmark, it evaluates to "below". If the REF field appears after the bookmark, it evaluates to "above". If the REF field appears within the bookmark, an error is returned. This switch can also be used in conjunction with the \n, \r, and \w switches, in which case, "above" or "below" is appended to the end of the field result.
\r	Inserts the entire paragraph number of the bookmarked paragraph in relative context—or relative to its position in the numbering scheme — without trailing periods.
\t	Causes the REF field to suppress non-delimiter or non-numerical text when used in conjunction with the \n, \r, or \w switch.
\w	Inserts the paragraph number of the bookmarked paragraph in full context from anywhere in the document.

2.16.5.59 REVNUM

Syntax:

REVNUM

Description: Retrieves the document's revision number (which indicates the number of times the document has been saved), as recorded in the **Revision** element of the Core File Properties part.

Field Value: The document's revision number.

Switches: None.

2.16.5.60 SAVEDATE

Syntax:

SAVEDATE [*switches*]

Description: Retrieves the date and time on which the document was last saved, as recorded in the **DateModified** element of the Core File Properties part. By default, the Gregorian calendar is used and the *date-and-time-formatting-switch* used is implementation-defined. For a document that has never been saved, the date and time corresponds to 0000-00-00T00:00:00 local time and each text component is XXX.

Field Value: The date and time on which the document was last saved.

Switches: Zero or one *date-and-time-formatting-switch* and zero or one of the following *field-specific-switches*.

\h	Use the Hijri/Lunar calendar.
\s	Use the Saka Era calendar.

2.16.5.61 SECTION

Syntax:

SECTI ON [*switches*]

Description: Retrieves the number of the current section.

Field Value: The number of the current section.

Switches: Zero or more *general-formatting-switches*.

2.16.5.62 SECTIONPAGES

Syntax:

SECTIONPAGES [*switches*]

Description: Retrieves the number of the current page within the current section.

Field Value: The number of the current page within the current section.

Switches: Zero or more *general-formatting-switches*.

2.16.5.63 SEQ

Syntax:

SEQ *identifier* [*field-argument*] [*switches*]

Description: Sequentially numbers chapters, tables, figures, and other user-defined lists of items in a document. If an item and its SEQ field are added, deleted, or moved, updating the remaining SEQ fields in the document reflects the new sequence. A SEQ field in a header, footer, annotation, or footnote shall NOT affect the sequence numbering that results from SEQ fields in the document text.

[*Note:* The LISTNUM field also produces automatic numbering and may be a better alternative when creating a complex numbered list. *end note*]

identifier is the name assigned to the series of items that are to be numbered. *identifier* shall start with a Latin letter and shall consist of no more than 40 Latin letters, Arabic digits, and underscores. (See the TOC field (§2.16.5.75) switches \c and \s for uses of *identifier*.)

text in *field-argument* specifies a bookmark name that refers to an item elsewhere in the document rather than in the current location.

Field Value: The next number in the sequence.

Switches: Zero or one of the *numeric-formatting-switches*, or zero or more of the following *field-specific-switches*. If no *numeric-formatting-switch* is present, * Arabic is used.

\c	Repeats the closest preceding sequence number. [<i>Note:</i> This is useful for inserting chapter numbers in headers or footers. <i>end note</i>]
\h	Hides the field result unless a <i>general-formatting-switch</i> is also present. [<i>Note:</i> This switch can be used to refer to a SEQ field in a cross-reference without printing the number. <i>end note</i>]
\n	Inserts the next sequence number for the specified item. This is the default.
\r <i>field-argument</i>	Resets the sequence number to the integer number specified by <i>text</i> in this switch's <i>field-argument</i> .
\s <i>field-argument</i>	Resets the sequence number to the built-in (integer) heading level specified by <i>text</i> in this switch's <i>field-argument</i> .

2.16.5.64 SET

Syntax:

SET *field-argument-1* *field-argument-2*

field-argument-1:
field-argument

field-argument-2:
field-argument

Description: Defines the bookmark name specified by *field-argument-1* to represent the information specified by *field-argument-2*.

Field Value: None.

Switches: None.

2.16.5.65 SKIPIF

Syntax:

SKI P *Expression-1* *Operator* *Expression-2*

Expression-1:

expression

Expression-2:

expression

Description: Compares the values designated by *Expression-1* and *Expression-2* using the operator designated by *Operator*. If the comparison is true, SKIPIF cancels the current merge document, moves to the next in the data source, and starts a new merge document. If the comparison is false, the current merge document is continued. *Operator* can be any one of the six relational and equality operators specified for *operator* (§2.16.3.3).

Field Value: None.

Switches: None.

2.16.5.66 STYLeref

Syntax:

STYLeref *field-argument* [*switches*]

Description: Inserts the nearest piece of text prior to this field that is formatted by the style whose name is specified by *text* in *field-argument*. The style can be a paragraph style or a character style.

When this field is used in a header or footer, it results in the first or the last text formatted with the specified style on the current page, allowing for dictionary-style headers or footers.

The location at which a STYLeref field is inserted determines the direction searched for the style, as follows:

- In document text, by default, the search goes backward from the STYLeref field. If the style isn't found, the search goes forward from the STYLeref field.
- In footnotes, annotations, and endnotes, the search goes backward from the footnote, annotation, or endnote reference mark. If the style isn't found, the search goes forward from the reference mark.
- In headers and footers in a printed document, the search is applied to the current page, by default, from top to bottom, for the specified style. If the style isn't found, the search goes from the top of the page to the beginning of the document, and then from the bottom of the page to the end of the document. If the \l switch is used, the search goes from the bottom of the page to the beginning and then to the end of the document.
- In headers and footers in an electronic document, the search goes on in the section that contains the STYLeref field, from the beginning, for the specified style. If the style isn't found, the search goes from the end of the section to the end of the document.

Field Value: The nearest piece of text prior to this field that is formatted by the style whose name is specified by *text* in *field-argument*.

Switches: Zero or more of the following *field-specific-switches*.

\l	Inserts the nearest text following the field.
\n	Inserts the paragraph number of the referenced paragraph exactly as it appears in the document.
\p	Inserts the relative position of the referenced paragraph as being "above" or "below".
\r	Inserts the paragraph number of the referenced paragraph exactly in relative context.
\t	When used with the \n, \r, or \w switch, causes non-delimiter and non-numerical text to be suppressed.

\w	Inserts the paragraph number of the referenced paragraph in full context, from anywhere in the document.
----	--

2.16.5.67 SUBJECT

Syntax:

SUBJECT [*field-argument*] [*switch*]

Description: Retrieves, and optionally sets, the document's subject, as recorded in the **Subject** element of the Core File Properties part or, if *field-argument* is present, the subject specified by *text* in *field-argument*. Specifying a *field-argument* shall change **Subject** to *text*.

Field Value: The document's subject.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.68 SYMBOL

Syntax:

SYMBOL *field-argument* [*switches*]

Description: Retrieves the character whose code point value is specified in decimal or hexadecimal (by using a leading 0x or 0X) by *text* in *field-argument*. The formatting switches override any formatting applied directly to the result. The XML generated for a complex field implementation shall not have the optional field value stored.

Field Value: The specified character.

Switches: Zero or more of the following *field-specific-switches*.

\a	Interprets <i>text</i> in <i>field-argument</i> as the value of an ANSI character.
\f <i>field-argument</i>	Interprets <i>text</i> in the switch's <i>field-argument</i> as the name of the font from which the character whose value is specified by <i>text</i> in the field's <i>field-argument</i> . By default, the font used is that for the current text run.
\h	Inserts the symbol without affecting the line spacing of the paragraph. If large symbols are inserted with this switch, text above the symbol may be overwritten.
\j	Interprets <i>text</i> in <i>field-argument</i> as the value of a SHIFT-JIS character.
\s <i>field-argument</i>	Interprets <i>text</i> in the switch's <i>field-argument</i> as the integral font size in points.
\u	Interprets <i>text</i> in <i>field-argument</i> as the value of a Unicode character.

2.16.5.69 TA

Syntax:

TA [*switches*]

Description: Defines the text and page number for a table of authorities entry, which is used by a TOA field (§2.16.5.74).

Field Value: None.

Switches: Zero or one of the following *field-specific-switches*.

\b	Applies bold formatting to the page number for the entry. If the table of authorities style for the entry already has bold formatting, \b removes it.
\c <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the integral entry category, which is a number that corresponds to the order of categories. The number determines how citations are grouped in tables of authorities. If \c is omitted, category 1 is the default.
\i	Applies italic formatting to the page number for the entry. If the table of authorities' style for the entry already has italic formatting, \i removes it.

<code>\l field-argument</code>	<i>text</i> in this switch's <i>field-argument</i> defines the long citation for the entry.
<code>\r field-argument</code>	Inserts as the entry's page number the range of pages marked by the bookmark specified by <i>text</i> in this switch's <i>field-argument</i> .
<code>\s field-argument</code>	<i>text</i> in this switch's <i>field-argument</i> defines the short citation for the entry.

2.16.5.70 TC

Syntax:

`TC field-argument [switches]`

Description: Defines the text and page number for a table of contents (including a table of figures) entry, which is used by a TOC field (§2.16.5.75). The text of the entry is *text* in *field-argument*.

Field Value: None.

Switches: Zero or one of the following *field-specific-switches*.

<code>\f field-argument</code>	The type of items collected in a particular contents list. Use a unique Type identifier (typically a letter from A-Z) for each type of list. For example, to build a list of illustrations, mark each illustration with a field such as TC "Illustration 1" <code>\f i</code> , where <i>i</i> indicates only illustration entries. If no type is specified, the entry is listed in a table of contents.
<code>\l field-argument</code>	The level of the TC entry. If no level is specified, level 1 is assumed.
<code>\n</code>	Omits the page number for the entry.

2.16.5.71 TEMPLATE

Syntax:

`TEMPLATE [switch]`

Description: Retrieves the disk file name of the template used by the current document.

Field Value: The disk file name of the template used by the current document.

Switches: One of the following *general-formatting-switches*: `* Caps`, `* FirstCap`, `* Lower`, or `* Upper`, and zero or one of the following *field-specific-switches*.

<code>\p</code>	Include the full file path name.
-----------------	----------------------------------

2.16.5.72 TIME

Syntax:

`TIME [switch]`

Description: Retrieves the current date and time. The Gregorian calendar is always used. By default, the *date-and-time-formatting-switch* used is implementation-defined.

Field Value: The current date and time.

Switches: Zero or one *date-and-time-formatting-switch*.

[Note: For some *date-and-time-formatting-switches*, the DATE (§2.16.5.18) and TIME fields can produce the same result.
end note]

2.16.5.73 TITLE

Syntax:

TITLE [*field-argument*] [*switch*]

Description: Retrieves, and optionally sets, the document's title, as recorded in the **Title** element of the Core File Properties part or, if *field-argument* is present, the name specified by *text* in *field-argument*. Specifying a *field-argument* shall change **Title** to *text*.

Field Value: The document's title.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.74 TOA

Syntax:

TOA [*switches*]

Description: Builds a table of authorities (that is, a list of the references in a legal document, such as references to cases, statutes, and rules, along with the numbers of the pages on which the references appear) using the entries specified by TA fields (§2.16.5.69).

Field Value: The table of authorities.

Switches: Zero or more of the following *field-specific-switches*.

\b <i>field-argument</i>	Includes entries only from the portion of the document marked by the bookmark specified by <i>text</i> in this switch's <i>field-argument</i> .
\c <i>field-argument</i>	<i>Includes the entries whose integral category is that specified by text in this switch's field-argument.</i>
\d <i>field-argument</i>	Used in conjunction with \s to specify the character sequence that separates the sequence numbers and page numbers. If \d is omitted, a hyphen (-) is used.
\e <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the character sequence that separates a table of authorities entry and its page number. If \e is not specified, a tab stop with leader dots is used.
\f	Removes the formatting of the entry text in the document from the entry in the table of authorities.
\g <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the character sequence that separates the pages in a page range. If \g is omitted, an en dash (–) is used.
\h	Includes the category heading for the entries in a table of authorities.
\l <i>field-argument</i>	<i>text</i> in this switch's <i>field-argument</i> specifies the character sequence that separates multiple page references. If \l is omitted, a comma (,) and space are used.
\p	Replaces five or more different page references to the same authority with "passim", which is used to indicate that a word or passage occurs frequently in the work cited.
\s <i>field-argument</i>	Includes a case or section number before the page number. The entry shall be numbered with a SEQ field (§2.16.5.63), and <i>text</i> in this switch's <i>field-argument</i> shall match the identifier in the SEQ field.

2.16.5.75 TOC

Syntax:TOC [*switches*]

Description: Builds a table of contents (which can also be a table of figures) using the entries specified by TC fields (§2.16.5.70), their heading levels, and specified styles, and inserts that table at this place in the document. Each table entry is a separate paragraph.

Field Value: The table of contents.

Switches: Zero or more of the following *field-specific-switches*.

<code>\a <i>field-argument</i></code>	Includes captioned items, but omits caption labels and numbers. The identifier designated by <i>text</i> in this switch's <i>field-argument</i> corresponds to the caption label. Use <code>\c</code> to build a table of captions with labels and numbers.
<code>\b <i>field-argument</i></code>	Includes entries only from the portion of the document marked by the bookmark named by <i>text</i> in this switch's <i>field-argument</i> .
<code>\c <i>field-argument</i></code>	Includes figures, tables, charts, and other items that are numbered by a SEQ field (§2.16.5.63). The sequence identifier designated by <i>text</i> in this switch's <i>field-argument</i> , which corresponds to the caption label, shall match the identifier in the corresponding SEQ field.
<code>\d <i>field-argument</i></code>	When used with <code>\s</code> , the <i>text</i> in this switch's <i>field-argument</i> defines the separator between sequence and page numbers. The default separator is a hyphen (-).
<code>\f <i>field-argument</i></code>	Includes only those TC fields whose identifier exactly matches the <i>text</i> in this switch's <i>field-argument</i> (which is typically a letter).
<code>\h</code>	Makes the table of contents entries hyperlinks.
<code>\l <i>field-argument</i></code>	Includes TC fields that assign entries to one of the levels specified by <i>text</i> in this switch's <i>field-argument</i> as a range having the form <i>startLevel-endLevel</i> , where <i>startLevel</i> and <i>endLevel</i> are integers, and <i>startLevel</i> has a value equal-to or less-than <i>endLevel</i> . TC fields that assign entries to lower levels are skipped.
<code>\n <i>field-argument</i></code>	Without <i>field-argument</i> , omits page numbers from the table of contents. Page numbers are omitted from all levels unless a range of entry levels is specified by <i>text</i> in this switch's <i>field-argument</i> . A range is specified as for <code>\l</code> .
<code>\o <i>field-argument</i></code>	Uses paragraphs formatted with all or the specified range of built-in heading styles. Headings in a style range are specified by <i>text</i> in this switch's <i>field-argument</i> using the notation specified as for <code>\l</code> , where each integer corresponds to the style with a style ID of HeadingX (e.g. 1 corresponds to Heading1). If no heading range is specified, all heading levels used in the document are listed.
<code>\p <i>field-argument</i></code>	<i>text</i> in this switch's <i>field-argument</i> specifies a sequence of characters that separate an entry and its page number. The default is a tab with leader dots.
<code>\s <i>field-argument</i></code>	For entries numbered with a SEQ field (§2.16.5.63), adds a prefix to the page number. The prefix depends on the type of entry. <i>text</i> in this switch's <i>field-argument</i> shall match the identifier in the SEQ field.
<code>\t <i>field-argument</i></code>	Uses paragraphs formatted with styles other than the built-in heading

	styles. <i>text</i> in this switch's <i>field-argument</i> specifies those styles as a set of comma-separated doublets, with each doublet being a comma-separated set of style name and table of content level. \t can be combined with \o.
\u	Uses the applied paragraph outline level.
\w	Preserves tab entries within table entries.
\x	Preserves newline characters within table entries.
\z	Hides tab leader and page numbers in Web layout view.

2.16.5.76 USERADDRESS

Syntax:

USERADDRESS [*field-argument*] [*switch*]

Description: Retrieves the current user's postal address or, if *field-argument* is present, the address specified by *text* in *field-argument*. Specifying a field-argument shall not change the address of the current user.

Field Value: A postal address.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.77 USERINITIALS

Syntax:

USERINITIALS [*field-argument*] [*switch*]

Description: Retrieves the current user's initials or, if *field-argument* is present, the initials specified by *text* in *field-argument*. Specifying a field-argument shall not change the initials of the current user.

Field Value: The set of initials.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.78 USERNAME

Syntax:

USERNAME [*field-argument*] [*switch*]

Description: Retrieves the current user's name or, if *field-argument* is present, the name specified by *text* in *field-argument*. Specifying a field-argument shall not change the name of the current user.

Field Value: The name.

Switches: One of the following *general-formatting-switches*: * Caps, * FirstCap, * Lower, or * Upper.

2.16.5.79 XE

Syntax:

XE *field-argument* [*switches*]

Description: Defines the text and page number for an index entry, which is used by an INDEX field (§2.16.5.35). The text of the entry is *text* in *field-argument*. To indicate a subentry, the main entry text and the subentry text shall be separated by a colon (:). Subentries beyond one level are permitted.

Field Value: None.

Switches: Zero or one of the following *field-specific-switches*.

\b	Applies bold formatting to the entry's page number. However, if the index style for that entry is already bold, this switch removes that formatting for that entry.
\f <i>field-argument</i>	The <i>text</i> in this switch's <i>field-argument</i> defines an index entry type. If an

	INDEX field has the same \f switch and <i>field-argument</i> , this entry is included in the resulting index; otherwise, it is excluded.
\i	Applies italic formatting to the entry's page number. However, if the index style for that entry is already italic, this switch removes that formatting for that entry.
\r <i>field-argument</i>	Instead of the entry's page number, uses the range of pages marked by the bookmark specified by <i>text</i> in this switch's <i>field-argument</i> .
\t <i>field-argument</i>	Uses <i>text</i> from <i>field-argument</i> in place of a page number. [Note: Useful for "See ..." or "See also ..." entries. <i>end note</i>]
\y <i>field-argument</i>	Specifies that the <i>text</i> from <i>field-argument</i> defines the yomi (first phonetic character for sorting indexes) for the index entry.

2.16.6 calcOnExit (Recalculate Fields When Current Field Is Modified)

This element specifies that the current contents of all fields within the current WordprocessingML document shall be recalculated from their field codes when the contents of the parent form field are modified. [Note: It is at the discretion of an application to determine the scope of a single modification, for example, when the user moves the insertion point in a user interface, or after each keystroke, etc. *end note*]

If this element is omitted, then modification of the contents of the current field shall not result in all fields in the current document being recalculated.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.16.7 checkBox (Checkbox Form Field Properties)

This element specifies a set of properties which shall be associated with the parent FORMCHECKBOX checkbox form field (§2.16.5.26) within the document.

If the parent form field is not a checkbox (i.e. its field code does not have a value of FORMCHECKBOX), then these properties may be ignored.

2.16.8 checked (Checkbox Form Field State)

This element specifies the current state for a checkbox form field. This value shall be used to specify the current value for a checkbox as explicitly chosen for that checkbox, as opposed its default value, which is specified using the **default** element (§2.16.12).

If this element is omitted, then the parent form field checkbox has no state, and its state shall be determined based on the value of the default element in the checkbox form field properties.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.16.9 ddList (Drop-Down List Form Field Properties)

This element specifies a set of properties which shall be associated with the parent FORMDROPDOWN drop-down list form field (§2.16.5.27) within the document.

If the parent form field is not a drop-down list (i.e. its field code does not have a value of FORMDROPDOWN), then these properties may be ignored.

2.16.10 default (Default Text Box Form Field String)

This element specifies the default string for the parent text box form field. This string is the content which shall be displayed in the document story within this form field if its current run contents are empty (i.e. there is not actual content within the text box). If the **type** (§2.16.34) of the current form field is **calculation**, then this string shall hold the calculation to be performed.

If this element is omitted, then the current text box form field shall not have a default value.

Attributes	Description
val (String Value)	See 2.3.1.27

2.16.11 default (Default Drop-Down List Item Index)

This element specifies the zero-based index of the default entry for the parent drop-down list form field. This index value is the value within the drop-down list which shall be displayed in the document story within this form field if no element is selected (i.e. the **result** element (§2.16.29) is omitted).

If this element is omitted, then the current drop-down list form field shall have a default value of 0 (its first entry). If the attribute value references an index value which does not exist (i.e. a negative number or a number that exceeds the number of items in the drop-down list), then this value may be ignored and the current drop-down list form field shall have a default value of 0 (its first entry).

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.16.12 default (Default Checkbox Form Field State)

This element specifies the default checkbox state for the parent checkbox form field. This value determines the checkbox state when its current run contents are empty (i.e. there is not actual content within the drop-down list).

If this element is omitted, then the current checkbox form field shall have a default value of 0 (unchecked).

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.16.13 delInstrText (Deleted Field Code)

This element specifies that this run contains deleted field codes (§2.16.5) within a complex field in the document. The **delInstrText** element shall be used for all runs containing field codes which are part of a region of text that is contained in a deleted region using the **del** element (§2.13.5.12).

If this element is not contained within a **del** element, then the document is invalid. If this element is contained within a run which is not part of a complex field's field codes, then it should be handled as regular deleted text.

Attributes	Description
space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	See 2.3.3.7

2.16.14 enabled (Form Field Enabled)

This element specifies whether the parent form field shall behave as though it is enabled or disabled when it is displayed in the document. This setting shall have no effect on the behavior of this form field unless the document's Settings part specifies that the **documentProtection** element for the current document is in a state allowing the filling in of form fields. If this element is omitted, then the parent form field shall be in its enabled state when the document settings specify that the document allows the filling-in of form fields.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.16.15 entryMacro (Script Function to Execute on Form Field Entry)

This element specifies a subroutine in a scripting language which should be executed when the when the run contents of the parent form field are entered. The language and location of this subroutine may be determined using any method desired by an application. [Note: It is at the discretion of an application to determine the scope and timing of "entering" a form field, for example, when the user moves the insertion point in a user interface or upon each operation by an application without a user interface, etc. *end note*]

If this element is omitted, then no subroutine shall be associated with entering the run contents of the parent form field. If this element specifies a macro which cannot be located or is not supported by an application, then its value may be ignored, but shall not be lost upon resaving the file.

Attributes	Description
val (Name of Script Function)	Specifies the name of a single scripting subroutine which shall be associated with the parent element. Its use is specifies based on the context of the parent XML element. The possible values for this attribute are defined by the ST_MacroName simple type (§2.18.57).

2.16.16 exitMacro (Script Function to Execute on Form Field Exit)

This element specifies a subroutine in a scripting language which should be executed when the when the run contents of the parent form field are exited. The language and location of this subroutine may be determined using any method desired by an application. [Note: It is at the discretion of an application to determine the scope and timing of "exiting" a form field, for example, when the user moves the insertion point in a user interface or upon each operation by an application without a user interface, etc. *end note*]

If this element is omitted, then no subroutine shall be associated with exiting the run contents of the parent form field. If this element specifies a macro which cannot be located or is not supported by an application, then its value may be ignored, but shall not be lost upon resaving the file.

Attributes	Description
val (Name of Script Function)	See 2.16.15

2.16.17 ffData (Form Field Properties)

This element specifies a set of properties which shall be associated with the parent form field within the document. This form field may be of any of the following types (with the associated field codes in parentheses):

- Checkbox (FORMCHECKBOX)
- Drop-down List (FORMDROPDOWN)
- Text box (FORMTEXT)

If this element is present and the field codes for the document do not specify a form field of one of these types, then the document shall be considered invalid.

If this element is omitted, then the properties associated with the parent form field shall be determined based on their default values.

2.16.18 fldChar (Complex Field Character)

This element specifies the presence of a complex field character at the current location in the parent run. A *complex field character* is a special character which delimits the start and end of a complex field or separates its field codes from its current field result.

A complex field is defined via the use of the two required complex field characters: a *start character*, which specifies the beginning of a complex field within the document content; and an *end character*, which specifies the end of a complex field. This syntax allows multiple fields to be embedded (or "nested") within each other in a document.

As well, because a complex field may specify both its field codes and its current result within the document, these two items are separated by the optional *separator character*, which defines the end of the field codes and the beginning of the field contents. The omission of this character shall be used to specify that the contents of the field are entirely field codes (i.e. the field has no result).

If a complex field character is located in an inappropriate location in a WordprocessingML document, then its presence shall be ignored and no field shall be present in the resulting document when displayed. Also, if a complex field is not closed before the end of a document story, then no field shall be generated and each individual run shall be processed as if the field characters did not exist (i.e. the contents of all field code run content shall not be displayed, and the field results shall be displayed as literal text).

Attributes	Description
dirty (Field Result Invalidated)	<p>Specifies that this field has been flagged by an application to indicate that its current results are invalid (stale) due to other modifications made to the document, and these contents should be updated before they are displayed if this functionality is supported by the next processing application.</p> <p>[<i>Rationale</i>: This functionality allows applications with limited subsets of the full functionality of this Office Open XML Standard to process Word Open XML documents without needing to understand and update all fields based on the semantics for their field codes.</p> <p>For example, an application can add a new paragraph and flag the table of contents as dirty, without needing to understand anything about how to recalculate that field's content. <i>end rationale</i>]</p> <p>If this attribute is omitted, then its value shall be assumed to be <code>false</code>. If the type of the current field character is not <code>start</code>, then this setting may be ignored. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>
fldCharType (Field Character Type)	<p>Specifies the type of the current complex field character in the document. The possible values for this attribute are defined by the ST_FldCharType simple type (§2.18.32).</p>
fldLock (Field Should Not Be Recalculated)	<p>Specifies that the parent complex field shall not have its field result recalculated, even if an application attempts to recalculate the results of all fields in the document or a recalculation is explicitly requested.</p> <p>If this attribute is omitted, then its value shall be assumed to be <code>false</code>. If the type of the current field character is not <code>start</code>, then this setting may be ignored. The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>

2.16.19 fldData (Custom Field Data)

This element specifies custom field data which shall be associated with the parent field. No information or semantics are applied to the contents of this data by this Office Open XML Standard, and therefore this field may be used as desired to

store additional application-specific data with the field. However, applications should not lose the contents of this custom data if they do not understand or utilize it (i.e. the information should continue to be saved with the file). If this element is omitted, then no custom field data is stored with the parent field. If the type attribute of the current field character is not `start`, then this setting may be ignored.

Attributes	Description
space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	See 2.3.3.7

2.16.20 fldData (Custom Field Data)

This element specifies custom field data which shall be associated with the parent field. No information or semantics are applied to the contents of this data by this Office Open XML Standard, and therefore this field may be used as desired to store additional application-specific data with the field. However, applications should not lose the contents of this custom data if they do not understand or utilize it (i.e. the information should continue to be saved with the file). If this element is omitted, then no custom field data is stored with the parent field.

Attributes	Description
space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	See 2.3.3.7

2.16.21 fldSimple (Simple Field)

This element specifies the presence of a simple field at the current location in the document. The semantics of this field are defined via its field codes (§2.16.5).

Attributes	Description
dirty (Field Result Invalidated)	See 2.16.18
fldLock (Field Should Not Be Recalculated)	See 2.16.18
instr (Field Codes)	See 2.16.18

2.16.22 format (Text Box Form Field Formatting)

This element specifies the field formatting which shall be applied to the contents of the parent form field whenever those contents are modified. The type of formatting which is applied to the field depends on the value of its **type** element (§2.16.34), as follows:

- When the **type** is equal to `currentDate`, `currentTime`, or `date`, a date formatting string using the syntax defined in §2.16.4.1

- When the **type** is equal to `calculated` or `number`, a number formatting string using the syntax defined in §2.16.4.2
- When the type is equal to `regular`, a text formatting string defined as follows:

Argument	Description
Uppercase	All letters are uppercase.
Lowercase	All letters are lowercase.
First capital	Capitalizes the first letter of the first word.
Title case	Capitalizes the first letter of each word.

Attributes	Description
val (String Value)	See 2.3.1.27

2.16.23 **helpText (Associated Help Text)**

This element specifies optional help text which shall be associated with the parent form field. The method or user interface by which this help text may be surfaced is not defined by this Office Open XML Standard.

If this element is omitted, then no help text shall be associated with the current form field.

Attributes	Description
type (Help Text Type)	Specifies the type of help text which is specified by this element, defined by the simple type below. If this attribute is omitted, then its value shall be assumed to be <code>text</code> . The possible values for this attribute are defined by the <code>ST_InfoTextType</code> simple type (§2.18.48).
val (Help Text Value)	Specifies the help text for the current form field. Based on the value of the type attribute, the contents of this field shall be interpreted as follows: <ul style="list-style-type: none"> When the type attribute value is <code>text</code>, contains the literal help text for the form field. When the type attribute value is <code>autoText</code> , contains the name of a glossary document entry which contains the help text for the form field. The possible values for this attribute are defined by the <code>ST_FFHelpTextVal</code> simple type (§2.18.28).

2.16.24 **hyperlink (Hyperlink)**

This element specifies the presence of a hyperlink at the current location in the document.

Attributes	Description
anchor (Hyperlink Anchor)	Specifies the name of a bookmark in the current document which shall be the target of this hyperlink. If this attribute is omitted, then the default behavior shall be to navigate to the start of the document. If a hyperlink target is also specified using the <code>r:id</code> attribute, then this attribute shall be ignored. If no bookmark exists in the current document with the given bookmark name, then the default behavior shall be to navigate to the start of the document. The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.18.88).
docLocation (Location in Target Document)	Specifies a location in the target of the hyperlink that has no bookmarks. The method by which the contents of this attribute are linked to document text is outside the scope of this Office Open XML Standard. If this attribute is omitted, then no location shall be associated with the parent hyperlink. If the

Attributes	Description												
	<p>anchor attribute is also specified, then this attribute may be ignored when the hyperlink is invoked.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>												
history (Add To Viewed Hyperlinks)	<p>Specifies whether the target of the parent hyperlink (as specified via the r:id attribute) shall be added to a list of viewed hyperlinks when it is invoked.</p> <p>If this attribute is omitted, then its value shall be assumed to be false.</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.66).</p>												
id (Hyperlink Target) Namespace: .../officeDocument/2006/relationships	<p>Specifies the ID of the relationship whose target shall be used as the target for this hyperlink.</p> <p>If this attribute is omitted, then there shall be no external hyperlink target for the current hyperlink - a location in the current document may still be target via the anchor attribute. If this attribute exists, it shall supersede the value in the anchor attribute.</p> <p>The possible values for this attribute are defined by the ST_RelationshipId simple type (Error! Reference source not found.).</p>												
tgtFrame (Hyperlink Target Frame)	<p>Specifies a frame within the parent HTML frameset for the target of the parent hyperlink when one exists. All values specified by this element shall be handled as follows:</p> <table border="1"> <thead> <tr> <th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>_top</td><td>Open hyperlink target in the full region of the current window.</td></tr> <tr> <td>_self</td><td>Open hyperlink target in the same frame as the hyperlink appears.</td></tr> <tr> <td>_parent</td><td>Open hyperlink target in the parent of the current frame, or the current frame if this frame has no parent.</td></tr> <tr> <td>_blank</td><td>Open hyperlink target in a new window.</td></tr> <tr> <td>all other values</td><td> <p>Open hyperlink target in the frame with the specified name. If no frame exists with this name, open in the current frame.</p> <p>If this string does not begin with an alphabetic character, it shall be ignored.</p> </td></tr> </tbody> </table> <p>If this attribute is omitted, then no target frame information shall be associated with the parent hyperlink. If the current document is not part of a frameset, then this information may be ignored.</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.88).</p>	Value	Description	_top	Open hyperlink target in the full region of the current window.	_self	Open hyperlink target in the same frame as the hyperlink appears.	_parent	Open hyperlink target in the parent of the current frame, or the current frame if this frame has no parent.	_blank	Open hyperlink target in a new window.	all other values	<p>Open hyperlink target in the frame with the specified name. If no frame exists with this name, open in the current frame.</p> <p>If this string does not begin with an alphabetic character, it shall be ignored.</p>
Value	Description												
_top	Open hyperlink target in the full region of the current window.												
_self	Open hyperlink target in the same frame as the hyperlink appears.												
_parent	Open hyperlink target in the parent of the current frame, or the current frame if this frame has no parent.												
_blank	Open hyperlink target in a new window.												
all other values	<p>Open hyperlink target in the frame with the specified name. If no frame exists with this name, open in the current frame.</p> <p>If this string does not begin with an alphabetic character, it shall be ignored.</p>												
tooltip (Associated String)	<p>Specifies a string which may be surfaced in a user interface as associated with the parent hyperlink. The method by which this string is surfaced by an application is outside the scope of this Office Open XML Standard.</p> <p>If this attribute is omitted, then no associated string shall be linked to the parent hyperlink in the</p>												

Attributes	Description
	document. The possible values for this attribute are defined by the ST_String simple type (§2.18.88).

2.16.25 instrText (Field Code)

This element specifies that this run contains field codes (§2.16.5) within a complex field in the document.

If this element is contained within a run which is not part of a complex field's field codes, then it and its contents should be treated as regular text. If this element is contained within a **del** element, then the document is invalid.

Attributes	Description
space (Content Contains Significant Whitespace) Namespace: http://www.w3.org/XML/1998/namespace	See 2.3.3.7

2.16.26 listEntry (Drop-Down List Entry)

This element specifies the presence of a single drop-down list entry within the parent drop-down list form field in the document. The order of appearance of the series of **listEntry** elements in the WordprocessingML markup shall dictate the order of the entries in the drop-down list when it is displayed.

Attributes	Description
val (String Value)	See 2.3.1.27

2.16.27 maxLength (Text Box Form Field Maximum Length)

This element specifies the maximum length of text which should be allowed within the parent text box form field before any formatting specified by the **format** element (§2.16.22). If the current contents of this field exceed the specified value when the document is loaded, that violation shall not result in an error, but the application shall prevent the addition of any additional characters until the contents are brought below that limit.

If this element is omitted, then there shall be no limit on the number of characters in the parent text box form field.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.16.28 name (Form Field Name)

This element specifies the name of the current form field.

Attributes	Description
val (Form Field Name Value)	Specifies the name of the form field. If this attribute is omitted, then the parent form field shall have no name. The possible values for this attribute are defined by the ST_FFName simple type (§2.18.29).

2.16.29 result (Drop-Down List Selection)

This element specifies the zero-based index of the currently selected entry for the parent drop-down list form field. If this element is omitted, then the current drop-down list form field shall have the selection specified by the value of the **default** element (§2.16.11). If the attribute value references an index value which does not exist (i.e. a negative number or a number that exceeds the number of items in the drop-down list), then this value may be ignored and the current drop-down list form field shall have the selection specified by the value of the **default** element.

Attributes	Description
val (Decimal Number Value)	See 2.3.1.10

2.16.30 size (Checkbox Form Field Size)

This element specifies the exact size for the parent checkbox form field. The resulting field shall be displayed in this point size regardless of the size specified by the formatting of its corresponding content in the document via the style hierarchy.

Attributes	Description
val (Half Point Measurement)	See 2.3.2.17

2.16.31 sizeAuto (Automatically Size Form Field)

This element specifies that the parent checkbox form field shall be formatted using the point size which is applied to its field characters via the style hierarchy.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.16.32 statusText (Associated Status Text)

This element specifies optional status text which shall be associated with the parent form field. The method or user interface by which this status text may be surfaced is not defined by this Office Open XML Standard. If this element is omitted, then no status text shall be associated with the current form field.

Attributes	Description
type (Status Text Type)	Specifies the type of status text which is specified by this element, defined by the simple type below. If this attribute is omitted, then its value shall be assumed to be text . The possible values for this attribute are defined by the ST_InfoTextType simple type (§2.18.48).
val (Status Text Value)	Specifies the status text for the current form field. Based on the value of the type attribute, the contents of this field shall be interpreted as follows: <ul style="list-style-type: none"> When the type attribute value is text, contains the literal status text for the form field. When the type attribute value is autoText , contains the name of a glossary document entry which contains the status text for the form field. The possible values for this attribute are defined by the ST_FFStatusTextVal simple type (§2.18.30).

2.16.33 textInput (Text Box Form Field Properties)

This element specifies a set of properties which shall be associated with the parent FORMTEXT text box form field (§2.16.5.28) within the document.

If the parent form field is not a text box (i.e. its field code does not have a value of FORMTEXT), then these properties may be ignored.

2.16.34 type (Text Box Form Field Type)

This element specifies the type of the contents of the current text box form field. This element shall not be used to prevent the successful loading of any contents in the field, but shall be used to parse the formatting specified in the **format** element (§2.16.22) and should be used to prevent the addition of illegal content when its contents are edited by an application.

If this element is omitted, then its default value shall be assumed to be `regular`.

Attributes	Description
val (Text Box Form Field Type Values)	Specifies the type of the text box form field, as defined by the simple type referenced below. The possible values for this attribute are defined by the ST_FFTextType simple type (§2.18.31).

2.17 Miscellaneous Topics

This section covers topics not covered elsewhere within the WordprocessingML documentation.

2.17.1 Text Box Content

All VML-based drawing objects (except for connectors) support the addition of rich WordprocessingML content within their extents. When WordprocessingML contents have been added to a VML drawing object, the resulting text is contained within a *text box*.

When WordprocessingML content is contained within a text box, it is allowed within the object by specifying the VML **textbox** element (§Error! Reference source not found.), which contains within it a single **txbxContent** element that contains all of the desired WordprocessingML content.

2.17.1.1 txbxContent (Rich Text Box Content Container)

This element specifies that its contents shall be any rich WordprocessingML content, and that this content is the rich contents of a drawing object defined using the Vector Markup Language (VML) syntax (§Error! Reference source not found.).

If this element contains within any of its contents any of the following content, then the document shall be considered non-conformant:

- References to other WordprocessingML document stories (comments, footnotes, endnotes)
- Additional **txbxContent** elements (as part of nested VML objects)

2.17.2 Subdocuments

Within a WordprocessingML document, it is sometimes necessary to break a large document into two or more separate WordprocessingML document files, allowing each of these files to be distributed, edited, and handled independently.

When a WordprocessingML document is comprised of other WordprocessingML documents in this way, the resulting documents are called a master document and its subdocuments.

- A *subdocument* is a WordprocessingML document - there is no specific information in a document which classifies it as such, other than that it is incorporated into another document.
- A *master document* is a document which incorporates one or more subdocuments (as well as optional WordprocessingML content) to create a larger document

2.17.2.1 **subDoc** (Anchor for Subdocument Location)

This element specifies a location within a master document for the insertion of the contents of a specified subdocument. The specified subdocument's contents should appear at the specified location within the master document as needed, but shall remain part of the separate file specified by the subdocument location. The location of the subdocument shall be specified by the relationship whose Id attribute matches the id attribute on this element.

If the relationship type of the relationship specified by this element is not

<http://schemas.openxmlformats.org/officeDocument/2006/subDocument>, is not present, or does not have a TargetMode attribute value of External, then the document shall be considered non-conformant.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.17.3 External Content Import

When generating WordprocessingML documents, it is sometimes necessary to include existing document content (henceforth called *external content*) within the document. External content in a document is typically included because it was stored in a format other than the WordprocessingML format defined by this Office Open XML Standard.

In order to facilitate the inclusion of such content without requiring its conversion as a prerequisite to its inclusion in a document, WordprocessingML includes the facility for applications to implement the import of external content in any format as part of a WordprocessingML document. This functionality, called external content import, allows the inclusion of content of an arbitrary content type within the WordprocessingML package, which shall then be opened and merged into the main document when the package is consumed by applications which understand that content type.

2.17.3.1 **altChunk** (Anchor for Imported External Content)

This element specifies a location within a document for the insertion of the contents of a specified file containing external content to be imported into the main WordprocessingML document. The specified file's contents should appear at the specified location within the document, and may henceforth be emitted as regular WordprocessingML without distinction to its origin. The location of the external content to be imported shall be specified by the relationship whose Id attribute matches the id attribute on this element.

If the relationship type of the relationship specified by this element is not

<http://schemas.openxmlformats.org/officeDocument/2006/afChunk>, is not present, or does not have a TargetMode attribute value of Internal, then the document shall be considered non-conformant. If an application cannot process external content of the content type specified by the targeted part, then it should ignore the specified alternate content but continue to process the file. If possible, it should also provide some indication that unknown content was not imported.

Attributes	Description
id (Relationship to Part) Namespace: .../officeDocument/2006/relationships	See 2.3.3.17

2.17.3.2 **altChunkPr** (External Content Import Properties)

This element specifies the set of properties which shall be applied to the import of the external content specified by the parent **altChunk** element. Within this Office Open XML Standard, only one property is specified.

2.17.3.3 **matchSrc (Keep Source Formatting on Import)**

This element specifies if any style definitions present in the imported content shall be overridden by identical styles present in the host WordprocessingML document. If this element's `val` attribute is `true`, then any style exists in both the imported content and main document shall be maintained on the imported content by redefining the style name and/or ID as needed. Conversely, if this element's `val` attribute is `false`, any style which exists in both the imported content and main document shall apply the style form the main document in place of the style in the imported content.

If this element is omitted, then styles from the main document shall override identical styles from the imported content.

Attributes	Description
val (On/Off Value)	See §2.3.1.1

2.17.4 **Roundtripping Alternate Content**

Office Open XML defines a mechanism for the storage of content which is not defined by this Office Open XML Standard, for example extensions developed by future software applications which leverage the Open XML formats. This mechanism allows for the storage of a series of alternative representations of content, of which the consuming application should use the first alternative whose requirements are met.

These alternate content blocks may occur at any location within a WordprocessingML document, and applications shall handle and process them appropriately (taking the appropriate choice).

However, WordprocessingML does not explicitly define a set of locations where applications shall attempt to store and roundtrip all non-taken choices whenever possible. This behavior is therefore application-defined.

2.18 **Simple Types**

This is the complete list of simple types in the <http://schemas.openxmlformats.org/wordprocessingml/2006/main> namespace.

2.18.1 **ST_AlgClass (Cryptographic Algorithm Classes)**

This simple type specifies the possible classes of cryptographic algorithm used by protection. [*Note: The initial version of this Office Open XML Standard only supports a single version - hash - but future versions may expand this as necessary. end note*]

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
hash (Hashing)	Specifies that the algorithm is a hashing function, which creates a hash value for user-supplied input that is very difficult to reverse-engineer.

2.18.2 **ST_AlgType (Cryptographic Algorithm Types)**

This simple type specifies the possible values for the type of cryptographic algorithm used by protection. [*Note: The initial version of this Office Open XML Standard only supports a single type - typeAny - but future versions may expand this as necessary. end note*]

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
-------------------	-------------

Enumeration Value	Description
typeAny (Any Type)	Specifies that any type of cryptographic algorithm type may be used.

2.18.3 ST_AnnotationVMerge (Table Cell Vertical Merge Revision Type)

This simple type specifies the possible values for the vertical merge setting which applied to a table cell by a cell merge (or split) revision.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
cont (Vertically Merged Cell)	Specifies that the revision resulted in this cell being vertically merged with the cell above it.
rest (Vertically Split Cell)	Specifies that the revision resulted in this cell being vertically split from the one above it.

2.18.4 ST_Border (Border Styles)



This simple type specifies the types of borders which can be specified for WordprocessingML objects which have a border. Borders can be separated into two types:








- *Line borders*, which specify a pattern to be used when drawing a line around the specified object.
- *Art borders*, which specify a repeated image to be used when drawing a border around the specified object.

Line borders may be specified on any object which allows a border, however, art borders may only be used as a border at the page level - the borders under the **pgBorders** element (§2.6.10).










This simple type's contents are a restriction of the XML Schema string datatype.





The following are possible enumeration values for this type:

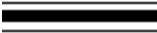





Enumeration Value	Description
apples (Apples Art Border)	Specifies an art border consisting of a repeated image of an apple, as follows (showing two repetitions): 
archedScallops (Arched Scallops Art Border)	Specifies an art border consisting of a repeated image of a shell pattern, as follows (showing two repetitions): 
babyPacifier (Baby Pacifier Art Border)	Specifies an art border consisting of a repeated image of a baby pacifier, as follows (showing two repetitions):








Enumeration Value	Description
	
babyRattle (Baby Rattle Art Border)	<p>Specifies an art border consisting of a repeated image of a baby rattle, as follows (showing two repetitions):</p> 
balloons3Colors (Three Color Balloons Art Border)	<p>Specifies an art border consisting of a repeated image of a set of balloons, as follows (showing two repetitions):</p> 
balloonsHotAir (Hot Air Balloons Art Border)	<p>Specifies an art border consisting of a repeated image of a hot air balloon, as follows (showing two repetitions):</p> 
basicBlackDashes (Black Dash Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (repeated twice):</p>  <p>If the border is on the top or bottom, this image is as follows (repeated twice):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
basicBlackDots (Black Dot Art Border)	<p>Specifies an art border consisting of a repeating image of a black dot on a white background.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>

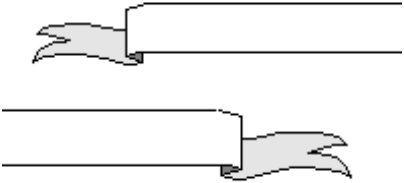




Enumeration Value	Description
	<p>• •</p> <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p> <p>• •</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> <p>•</p>
basicBlackSquares (Black Square Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (repeated twice):</p> <p>■ ■</p> <p>If the border is on the top or bottom, this image is as follows (repeated twice):</p> <p>■ ■</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> <p>■</p>
basicThinLines (Thin Line Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows:</p> <p> </p> <p>If the border is on the top or bottom, this image is as follows:</p> <p>====</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p>








Enumeration Value	Description
	
basicWhiteDashes (White Dash Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (repeated twice):</p>   <p>If the border is on the top or bottom, this image is as follows (repeated twice):</p>   <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
basicWhiteDots (White Dot Art Border)	<p>Specifies an art border consisting of a repeating image of a white dot on a black background.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
basicWhiteSquares (White Square Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (repeated twice):</p>







Enumeration Value	Description
	<p>□ □</p> <p>If the border is on the top or bottom, this image is as follows (repeated twice):</p> <p>□ □</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> <p>□</p>
basicWideInline (Wide Inline Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (showing for the left, flipped horizontally for the right border):</p>  <p>If the border is on the top or bottom, this image is as follows (showing for the top, flipped vertically for the bottom border):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
basicWideMidline (Wide Midline Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (showing for the left, flipped horizontally for the right border):</p>  <p>If the border is on the top or bottom, this image is as follows</p>


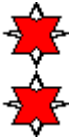





Enumeration Value	Description
	<p>(showing for the top, flipped vertically for the bottom border):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
basicWideOutline (Wide Outline Art Border)	<p>Specifies an art border consisting of a repeating image of a black and white background.</p> <p>If the border is on the left or right, this image is as follows (showing for the left, flipped horizontally for the right border):</p>  <p>If the border is on the top or bottom, this image is as follows (showing for the top, flipped vertically for the bottom border):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
bats (Bats Art Border)	<p>Specifies an art border consisting of a repeated image of bats, as follows (showing two repetitions):</p> 
birds (Birds Art Border)	<p>Specifies an art border consisting of repeating images of birds.</p> <p>If the border is on the left or right, no border is displayed.</p> <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>









Enumeration Value	Description
	
birdsFlight (Birds Flying Art Border)	<p>Specifies an art border consisting of a repeated image of birds flying, as follows (showing two repetitions):</p> 
cabins (Cabin Art Border)	<p>Specifies an art border consisting of a repeated image of a cabin, as follows (showing two repetitions):</p> 
cakeSlice (Cake Art Border)	<p>Specifies an art border consisting of a repeated image of a piece of cake, as follows (showing two repetitions):</p> 
candyCorn (Candy Corn Art Border)	<p>Specifies an art border consisting of a repeated image of candy corn, as follows (showing two repetitions):</p> 
celticKnotwork (Knot Work Art Border)	<p>Specifies an art border consisting of a repeated image of a knot work pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 








Enumeration Value	Description
certificateBanner (Certificate Banner Art Border)	<p>Specifies an art border consisting of a banner.</p> <p>If the border is on the left or right, no border is displayed.</p> <p>If the border is on the top, this image is as follows (showing each end):</p>  <p>If this border is on the bottom, then the ends shall be flipped vertically.</p>
chainLink (Chain Link Art Border)	<p>Specifies an art border consisting of a repeating image of a chain link pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p> 
champagneBottle (Champagne Bottle Art Border)	<p>Specifies an art border consisting of a repeated image of a champagne bottle, as follows (showing two repetitions):</p> 
checkedBarBlack (Black and White Bar Art Border)	<p>Specifies an art border consisting of repeating images of a compass.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions on the left, the right would be flipped horizontally):</p> 






Enumeration Value	Description
	<p>If the border is on the top or bottom, this image is as follows (showing two repetitions on top, the bottom would be flipped vertically):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
checkedBarColor (Color Checked Bar Art Border)	<p>Specifies an art border consisting of a repeating image of a colored pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
checkered (Checkerboard Art Border)	<p>Specifies an art border consisting of a repeated image of a checkerboard, as follows (showing two repetitions):</p> 
christmasTree (Christmas Tree Art Border)	<p>Specifies an art border consisting of a repeated image of a Christmas tree, as follows (showing two repetitions):</p> 
circlesLines (Circles And Lines Art Border)	<p>Specifies an art border consisting of repeating images of lines and circles.</p>







Enumeration Value	Description
	<p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
circlesRectangles (Circles and Rectangles Art Border)	<p>Specifies an art border consisting of a repeated image of a rectangular pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
classicalWave (Wave Art Border)	<p>Specifies an art border consisting of a repeated image of a wave, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
clocks (Clocks Art Border)	<p>Specifies an art border consisting of a repeated image of a</p>






Enumeration Value	Description
	<p>clock, as follows (showing two repetitions):</p> 
<p>compass (Compass Art Border)</p>	<p>Specifies an art border consisting of repeating images of a compass.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
<p>confetti (Confetti Art Border)</p>	<p>Specifies an art border consisting of a repeated image of confetti, as follows (showing two repetitions):</p> 
<p>confettiGrays (Confetti Art Border)</p>	<p>Specifies an art border consisting of a repeated image of confetti, as follows (showing two repetitions):</p> 
<p>confettiOutline (Confetti Art Border)</p>	<p>Specifies an art border consisting of a repeated image of confetti, as follows (showing two repetitions):</p> 
<p>confettiStreamers (Confetti Streamers Art Border)</p>	<p>Specifies an art border consisting of a repeated image of confetti streamers, as follows (showing two repetitions):</p>





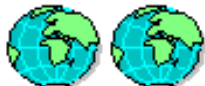


Enumeration Value	Description
	
confettiWhite (Confetti Art Border)	<p>Specifies an art border consisting of a repeated image of confetti, as follows (showing two repetitions):</p> 
cornerTriangles (Corner Triangle Art Border)	<p>Specifies an art border consisting of a repeated image of a line as follows:</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
couponCutoutDashes (Dashed Line Art Border)	<p>Specifies an art border consisting of a dashed line, as follows:</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>If the top border is of this type, the border shall use the following image in the top left corner:</p> 
couponCutoutDots (Dotted Line Art Border)	<p>Specifies an art border consisting of a dotted line, as follows:</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>If the top border is of this type, the border shall use the following image in the top left corner:</p>  <p>If the bottom border is of this type, the border shall use the following image in the bottom right corner:</p>




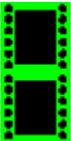



Enumeration Value	Description
	
crazyMaze (Maze Art Border)	<p>Specifies an art border consisting of a repeated image of a maze-like pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
creaturesButterfly (Butterfly Art Border)	<p>Specifies an art border consisting of a repeated image of a butterfly, as follows (showing two repetitions):</p> 
creaturesFish (Fish Art Border)	<p>Specifies an art border consisting of a repeated image of a fish, as follows (showing two repetitions):</p> 
creaturesInsects (Insects Art Border)	<p>Specifies an art border consisting of repeating images of insects.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the</p>








Enumeration Value	Description
	<p>intersection shall use the following image:</p> 
creaturesLadyBug (Ladybug Art Border)	<p>Specifies an art border consisting of a repeated image of a ladybug, as follows (showing two repetitions):</p> 
crossStitch (Cross-stitch Art Border)	<p>Specifies an art border consisting of repeating images of a cross-stitch pattern, as follows (showing two repetitions):</p> 
cup (Cupid Art Border)	<p>Specifies an art border consisting of a repeated image of Cupid, as follows (showing two repetitions):</p> 
dashDotStroked (Dash Dot Strokes Line Border)	<p>Specifies a line border consisting of a line with a series of alternating thin and thick strokes around the parent object.</p>
dashed (Dashed Line Border)	<p>Specifies a line border consisting of a dashed line around the parent object.</p>
dashSmallGap (Dashed Line Border)	<p>Specifies a line border consisting of a dashed line with small gaps around the parent object.</p>
decoArch (Archway Art Border)	<p>Specifies an art border consisting of repeating images of an archway.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>








Enumeration Value	Description
	 <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
decoArchColor (Color Archway Art Border)	<p>Specifies an art border consisting of repeating images of a color archway.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
decoBlocks (Blocks Art Border)	<p>Specifies an art border consisting of repeating images of a series of blocks.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>



Enumeration Value	Description
	 <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated accordingly:</p> 
diamondsGray (Gray Diamond Art Border)	<p>Specifies an art border consisting of a repeated image of diamonds, as follows (showing two repetitions):</p> 
dotDash (Dot Dash Line Border)	<p>Specifies a line border consisting of a alternating dotted and dashed line around the parent object.</p>
dotDotDash (Dot Dot Dash Line Border)	<p>Specifies a line border consisting of a alternating dotted, dotted, dashed line around the parent object.</p>
dotted (Dotted Line Border)	<p>Specifies a line border consisting of a dotted line around the parent object.</p>
double (Double Line Border)	<p>Specifies a line border consisting of a double line around the parent object.</p>
doubleD (Double D Art Border)	<p>Specifies an art border consisting of repeating images of a pattern.</p> <p>If the border is on the left, this image is as follows (showing two repetitions):</p>  <p>If the border is on the right, this image is rotated as follows (showing two repetitions):</p> 





Enumeration Value	Description
	<p>If the border is on the top, this image is rotated as follows (showing two repetitions):</p>  <p>If the border is on the bottom, this image is rotated as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
doubleDiamonds (Diamond Art Border)	<p>Specifies an art border consisting of a repeated image of a series of diamonds, as follows (showing two repetitions):</p> 
doubleWave (Double Wave Line Border)	<p>Specifies a line border consisting of a double wavy line around the parent object.</p>
earth1 (Earth Art Border)	<p>Specifies an art border consisting of a repeated image of Earth, as follows (showing two repetitions):</p> 
earth2 (Earth Art Border)	<p>Specifies an art border consisting of a repeated image of Earth, as follows (showing two repetitions):</p> 
eclipsingSquares1 (Shadowed Square Art Border)	<p>Specifies an art border consisting of a repeated image of a shadowed square, as follows (showing two repetitions):</p> 
eclipsingSquares2 (Shadowed Square Art Border)	<p>Specifies an art border consisting of a repeated image of a shadowed square, as follows (showing two repetitions):</p>








Enumeration Value	Description
	
eggsBlack (Painted Egg Art Border)	<p>Specifies an art border consisting of a repeated image of a painted egg, as follows (showing two repetitions):</p> 
fans (Fans Art Border)	<p>Specifies an art border consisting of a repeated image of fans, as follows (showing two repetitions):</p> 
film (Film Reel Art Border)	<p>Specifies an art border consisting of repeating images of a film reel.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
firecrackers (Firecracker Art Border)	<p>Specifies an art border consisting of a repeated image of a firecracker, as follows (showing two repetitions):</p> 
flowersBlockPrint (Flowers Art Border)	<p>Specifies an art border consisting of a repeated image of flowers, as follows (showing two repetitions):</p>








Enumeration Value	Description
	
flowersDaisies (Daisy Art Border)	<p>Specifies an art border consisting of a repeated image of a daisy, as follows (showing two repetitions):</p> 
flowersModern1 (Flowers Art Border)	<p>Specifies an art border consisting of a repeated image of flowers, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p>
flowersModern2 (Flowers Art Border)	<p>Specifies an art border consisting of a repeated image of flowers, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p>
flowersPansy (Pansy Art Border)	<p>Specifies an art border consisting of a repeated image of a pansy, as follows (showing two repetitions):</p> 
flowersRedRose (Red Rose Art Border)	<p>Specifies an art border consisting of a repeated image of a red rose, as follows (showing two repetitions):</p> 
flowersRoses (Roses Art Border)	<p>Specifies an art border consisting of a repeated image of a rose, as follows (showing two repetitions):</p> 
flowersTeacup (Flowers in a Teacup Art Border)	<p>Specifies an art border consisting of a repeated image of flowers in a teacup, as follows (showing two repetitions):</p>







Enumeration Value	Description
	
flowersTiny (Small Flower Art Border)	<p>Specifies an art border consisting of a repeated image of small flowers, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p>
gems (Gems Art Border)	<p>Specifies an art border consisting of a repeated image of a square pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image (shown from the left side - this image shall be flipped horizontally for the right side):</p> 
gingerbreadMan (Gingerbread Man Art Border)	<p>Specifies an art border consisting of a repeated image of a gingerbread man, as follows (showing two repetitions):</p> 
gradient (Triangle Gradient Art Border)	<p>Specifies an art border consisting of a repeated image of a triangle with a gradient pattern as follows (showing two repetitions):</p> 
handmade1 (Handmade Art Border)	<p>Specifies an art border consisting of an image with a handmade appearance.</p> <p>On the top (bottom is flipped vertically), this image shall be a stretched version of the following:</p> 









Enumeration Value	Description
	<p>On the left (right is flipped horizontally), this image shall be a stretched version of the following:</p> 
<p>handmade2 (Handmade Art Border)</p>	<p>Specifies an art border consisting of an image with a handmade appearance.</p> <p>On the top (bottom is flipped vertically), this image shall be a stretched version of the following:</p>  <p>On the left (right is flipped horizontally), this image shall be a stretched version of the following:</p>




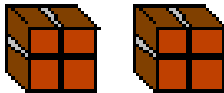


Enumeration Value	Description
	
heartBalloon (Heart-Shaped Balloon Art Border)	<p>Specifies an art border consisting of a repeated image of a heart-shaped balloon, as follows (showing two repetitions):</p> 
heartGray (Gray Heart Art Border)	<p>Specifies an art border consisting of a repeated image of a heart, as follows (showing two repetitions):</p> 
hearts (Hearts Art Border)	<p>Specifies an art border consisting of a repeated image of hearts, as follows (showing two repetitions):</p> 
heebieJeebies (Pattern Art Border)	<p>Specifies an art border consisting of a repeated image of a pattern as follows (showing two repetitions):</p>

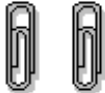





Enumeration Value	Description
	
holly (Holly Art Border)	Specifies an art border consisting of a repeated image of holly, as follows (showing two repetitions): 
houseFunky (House Art Border)	Specifies an art border consisting of a repeated image of a house, as follows (showing two repetitions): 
hypnotic (Circular Art Border)	Specifies an art border consisting of a repeated image of a series of circles, as follows (showing two repetitions):  As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents. At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately: 
iceCreamCones (Ice Cream Cone Art Border)	Specifies an art border consisting of a repeated image of an ice cream cone, as follows (showing two repetitions): 
inset (Inset Line Border)	Specifies a line border consisting of an inset set of lines around the parent object.
lightBulb (Light Bulb Art Border)	Specifies an art border consisting of a repeated image of a light bulb as follows (showing two repetitions): 
lightning1 (Lightning Art Border)	Specifies an art border consisting of a repeated image of lightning, as follows (showing two repetitions):

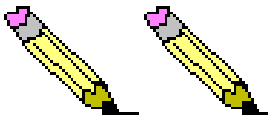







Enumeration Value	Description
	
lightning2 (Lightning Art Border)	<p>Specifies an art border consisting of a repeated image of a lightning pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
mapleLeaf (Maple Leaf Art Border)	<p>Specifies an art border consisting of a repeated image of a black and white image of a maple leaf, as follows (showing two repetitions):</p> 
mapleMuffins (Muffin Art Border)	<p>Specifies an art border consisting of a repeated image of a muffin, as follows (showing two repetitions):</p> 
mapPins (Map Pins Art Border)	<p>Specifies an art border consisting of a repeated image of a map pin, as follows (showing two repetitions):</p> 
marquee (Marquee Art Border)	<p>Specifies an art border consisting of a repeated image of a pattern as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated</p>








Enumeration Value	Description
	<p>appropriately:</p> 
marqueeToothed (Marquee Art Border)	<p>Specifies an art border consisting of a repeated image of a pattern as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
moons (Moon Art Border)	<p>Specifies an art border consisting of repeating images of phases of the moon.</p> <p>If the border is on the top, this image is as follows (showing two repetitions):</p>  <p>If the border is on the left, this image is as follows (showing two repetitions):</p>  <p>If the border is on the right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the bottom, this image is as follows (showing two repetitions):</p>








Enumeration Value	Description
	 <p>At the top-left corner where two borders of this type intersect, the intersection shall use the following image:</p>  <p>At the top-right corner where two borders of this type intersect, the intersection shall use the following image:</p>  <p>At the bottom-left corner where two borders of this type intersect, the intersection shall use the following image:</p>  <p>At the bottom-right corner where two borders of this type intersect, the intersection shall use the following image:</p> 
mosaic (Mosaic Art Border)	<p>Specifies an art border consisting of a repeated image of a mosaic pattern, as follows (showing two repetitions):</p> 
musicNotes (Musical Note Art Border)	<p>Specifies an art border consisting of a repeated image of a musical note, as follows (showing two repetitions):</p> 
nil (No Border)	Specifies that no border shall be applied to the current item.
none (No Border)	Specifies that no border shall be applied to the current item.
northwest (Patterned Art Border)	<p>Specifies an art border consisting of a repeated image of a pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p>








Enumeration Value	Description
	
outset (Outset Line Border)	Specifies a line border consisting of an outset set of lines around the parent object.
ovals (Oval Art Border)	<p>Specifies an art border consisting of a repeated image of a series of ovals, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
packages (Package Art Border)	<p>Specifies an art border consisting of a repeated image of a package, as follows (showing two repetitions):</p> 
palmsBlack (Black Palm Tree Art Border)	<p>Specifies an art border consisting of a repeated image of a black and white palm tree, as follows (showing two repetitions):</p> 
palmsColor (Color Palm Tree Art Border)	<p>Specifies an art border consisting of a repeated image of a color palm tree, as follows (showing two repetitions):</p> 
paperClips (Paper Clip Art Border)	<p>Specifies an art border consisting of a repeated image of a paper clip, as follows (showing two repetitions):</p>


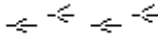




Enumeration Value	Description
	
<p>papyrus (Papyrus Art Border)</p>	<p>Specifies an art border consisting of repeating images of ovals and an art corner.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
<p>partyFavor (Party Favor Art Border)</p>	<p>Specifies an art border consisting of a repeated image of a party favor, as follows (showing two repetitions):</p> 
<p>partyGlass (Party Glass Art Border)</p>	<p>Specifies an art border consisting of a repeated image of a party glass, as follows (showing two repetitions):</p> 
<p>pencils (Pencils Art Border)</p>	<p>Specifies an art border consisting of a repeated image of a pencil, as follows (showing two repetitions):</p>








Enumeration Value	Description
	
people (Character Art Border)	<p>Specifies an art border consisting of a repeated image of a character, as follows (showing two repetitions):</p> 
peopleHats (Character With Hat Art Border)	<p>Specifies an art border consisting of a repeated image of a character with a hat, as follows (showing two repetitions):</p> 
peopleWaving (Waving Character Border)	<p>Specifies an art border consisting of a repeated image of a character waving, as follows (showing two repetitions):</p> 
poinsettias (Poinsettia Art Border)	<p>Specifies an art border consisting of a repeated image of a poinsettia, as follows (showing two repetitions):</p> 
postageStamp (Postage Stamp Art Border)	<p>Specifies an art border consisting of repeating images of a stamp-like pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
pumpkin1 (Pumpkin Art Border)	<p>Specifies an art border consisting of a repeated image of a pumpkin, as follows (showing two repetitions):</p> 




Enumeration Value	Description
pushPinNote1 (Push Pin Art Border)	<p>Specifies an art border consisting of a black line, as follows:</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>If the top border is of this type, the border shall use the following image in the top left corner:</p> 
pushPinNote2 (Push Pin Art Border)	<p>Specifies an art border consisting of a repeated image of a black line.</p> <p>If the border is on the top, left or right, then the image is as follows (showing two repetitions):</p>  <p>If the border is on the bottom, then the image is as follows (showing two repetitions):</p>  <p>If the top border is of this type, the border shall use the following image in the top left corner:</p> 
pyramids (Pyramid Art Border)	<p>Specifies an art border consisting of repeating images of a pyramid pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
pyramidsAbove (Pyramid Art Border)	<p>Specifies an art border consisting of a repeated image of a pyramid viewed from above, as follows (showing two repetitions):</p>



Enumeration Value	Description
	
quadrants (Quadrants Art Border)	<p>Specifies an art border consisting of a repeating image of a colored pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
rings (Rings Art Border)	<p>Specifies an art border consisting of a repeated image of a ring, as follows (showing two repetitions):</p> 
safari (Safari Art Border)	<p>Specifies an art border consisting of a repeated image of a print pattern, as follows (showing two repetitions):</p> 
sawtooth (Saw tooth Art Border)	<p>Specifies an art border consisting of repeating images of a saw tooth pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p>




Enumeration Value	Description
	
sawtoothGray (Gray Saw tooth Art Border)	<p>Specifies an art border consisting of repeating images of a saw tooth pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
scaredCat (Scared Cat Art Border)	<p>Specifies an art border consisting of a repeated image of a frightened cat, as follows (showing two repetitions):</p> 
seattle (Umbrella Art Border)	<p>Specifies an art border consisting of a repeated image of an umbrella, as follows (showing two repetitions):</p> 
shadowedSquares (Shadowed Squares Art Border)	<p>Specifies an art border consisting of a repeated image of squares with a drop shadow, as follows (showing two repetitions):</p> 
sharksTeeth (Shark Tooth Art Border)	<p>Specifies an art border consisting of a repeating image of a shark tooth pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows</p>








Enumeration Value	Description
	<p>(showing two repetitions on the top, the bottom shall be rotated 180 degrees):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the left and right border.</p>
shorebirdTracks (Bird Tracks Art Border)	<p>Specifies an art border consisting of a repeated image of bird tracks, as follows</p>  <p>As well, this art border shall be rotated such that the bottom of each image is always nearest the text extents.</p>
single (Single Line Border)	<p>Specifies a line border consisting of a single line around the parent object.</p>
skyrocket (Rocket Art Border)	<p>Specifies an art border consisting of a repeated image of a rocket, as follows (showing two repetitions):</p> 
snowflakeFancy (Snowflake Art Border)	<p>Specifies an art border consisting of a repeated image of a snowflake, as follows (showing two repetitions):</p> 
snowflakes (Snowflake Art Border)	<p>Specifies an art border consisting of a repeated image of a snowflake, as follows (showing two repetitions):</p> 
sombrero (Sombrero Art Border)	<p>Specifies an art border consisting of a repeated image of a sombrero, as follows (showing two repetitions):</p> 
southwest (Southwest-themed Art Border)	<p>Specifies an art border consisting of a repeated image of a pattern as follows (showing two repetitions):</p>








Enumeration Value	Description
	 <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
stars (Stars Art Border)	<p>Specifies an art border consisting of a repeated image of stars, as follows (showing two repetitions):</p> 
stars3d (3-D Stars Art Border)	<p>Specifies an art border consisting of a repeated image of three-dimensional stars, as follows (showing two repetitions):</p> 
starsBlack (Stars Art Border)	<p>Specifies an art border consisting of a repeated image of black stars, as follows (showing three repetitions):</p> 
starsShadowed (Stars With Shadows Art Border)	<p>Specifies an art border consisting of a repeated image of stars with a shadow effect, as follows (showing two repetitions):</p> 
starsTop (Stars On Top Art Border)	<p>Specifies an art border consisting of repeating images of stars on the top of the page.</p> <p>If the border is on the left or right, no border is displayed.</p> <p>If the border is on the top, this image is as follows:</p>  <p>If the border is on the bottom, this image is as follows:</p>







Enumeration Value	Description
	
sun (Sun Art Border)	Specifies an art border consisting of a repeated image of the sun, as follows (showing two repetitions): 
swirligig (Whirligig Art Border)	Specifies an art border consisting of a repeated image as follows (showing two repetitions): 
thick (Single Line Border)	Specifies a line border consisting of a single line around the parent object.
thickThinLargeGap (Thick, Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a large sized intermediate gap around the parent object.
thickThinMediumGap (Thick, Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a medium sized intermediate gap around the parent object.
thickThinSmallGap (Thick, Thin Line Border)	Specifies a line border consisting of a thick line contained within a thin line with a small intermediate gap around the parent object.
thinThickLargeGap (Thin, Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a large sized intermediate gap between each around the parent object.
thinThickMediumGap (Thin, Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a medium sized intermediate gap between each around the parent object.
thinThickSmallGap (Thin, Thick Line Border)	Specifies a line border consisting of a thin line contained within a thick line contained within a thick thin with a small intermediate gap between each around the parent object.






Enumeration Value	Description
thinThickThinLargeGap (Thin, Thick, Thin Line Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a large sized intermediate gap around the parent object.
thinThickThinMediumGap (Thin, Thick, Thin Line Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a medium sized intermediate gap around the parent object.
thinThickThinSmallGap (Thin, Thick, Thin Line Border)	Specifies a line border consisting of a thin line contained within a thick line, contained within a thin line with a small intermediate gap around the parent object.
threeDEmboss (3D Embossed Line Border)	Specifies a line border consisting of three staged gradient lines around the parent object, getting darker towards the object.
threeDEngrave (3D Engraved Line Border)	Specifies a line border consisting of three staged gradient lines around the parent object, getting darker away from the object.
tornPaper (Torn Paper Art Border)	<p>Specifies an art border consisting of repeating images of stars on the top of the page.</p> <p>If the border is on the left or right, no border is displayed.</p> <p>If the border is on the top or bottom, this image is as follows (shown on top, flipped vertically on bottom):</p> 
tornPaperBlack (Black Torn Paper Art Border)	<p>Specifies an art border consisting of an image with a torn appearance.</p> <p>On the top (bottom border shall be flipped vertically), this image shall be a stretched version of the following:</p>  <p>On the left (the right side shall be flipped horizontally), this image shall be a stretched version of the following:</p>








Enumeration Value	Description
	
trees (Tree Art Border)	<p>Specifies an art border consisting of a repeated image of a tree, as follows (showing two repetitions):</p> 
triangleParty (Triangle Art Border)	<p>Specifies an art border consisting of a repeated image of triangles, as follows (showing two repetitions):</p> 
triangles (Triangles Art Border)	<p>Specifies an art border consisting of a repeating image of a triangle pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions on the left, the right shall be rotated 180 degrees):</p>







Enumeration Value	Description
	 <p>If the border is on the top or bottom, this image is as follows (showing two repetitions on the top, the bottom shall be rotated 180 degrees):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
tribal1 (Tribal Art Border One)	<p>Specifies an art border consisting of a repeated image of a tribal pattern as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
tribal2 (Tribal Art Border Two)	<p>Specifies an art border consisting of a repeated image of a tribal pattern as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 







Enumeration Value	Description
tribal3 (Tribal Art Border Three)	<p>Specifies an art border consisting of a repeated image of a tribal pattern as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
tribal4 (Tribal Art Border Four)	<p>Specifies an art border consisting of a repeated image of a tribal pattern as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
tribal5 (Tribal Art Border Five)	<p>Specifies an art border consisting of a repeating image of a tribal pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 

Enumeration Value	Description
tribal6 (Tribal Art Border Six)	<p>Specifies an art border consisting of a repeated image of a tribal pattern as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
triple (Triple Line Border)	<p>Specifies a line border consisting of a triple line around the parent object.</p>
twistedLines1 (Twisted Lines Art Border)	<p>Specifies an art border consisting of a repeated image of twisted lines, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
twistedLines2 (Twisted Lines Art Border)	<p>Specifies an art border consisting of a repeated image of twisted lines, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
vine (Vine Art Border)	<p>Specifies an art border consisting of a repeating image of a</p>

Enumeration Value	Description
	<p>vine pattern.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p>  <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
wave (Wavy Line Border)	Specifies a line border consisting of a wavy line around the parent object.
waveline (Wavy Line Art Border)	<p>Specifies an art border consisting of a repeated image of wavy lines, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
weavingAngles (Weaving Angles Art Border)	Specifies an art border consisting of a repeated image of weaving angles, as follows (showing two repetitions):

Enumeration Value	Description
	 <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
weavingBraid (Weaving Braid Art Border)	<p>Speci Specifies an art border consisting of repeating images of a weaving pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
weavingRibbon (Weaving Ribbon Art Border)	<p>Specifies an art border consisting of repeating images of a weaving ribbon, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated accordingly:</p> 
weavingStrips (Weaving Strips Art Border)	<p>Specifies an art border consisting of repeating images of weaving strips, as follows (showing two repetitions):</p> 

Enumeration Value	Description
	<p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated accordingly:</p> 
whiteFlowers (White Flowers Art Border)	<p>Specifies an art border consisting of a repeating image of white flowers.</p> <p>If the border is on the left or right, this image is as follows (showing two repetitions):</p>  <p>If the border is on the top or bottom, this image is as follows (showing two repetitions):</p> 
woodwork (Woodwork Art Border)	<p>Specifies an art border consisting of repeating images of a woodwork pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image:</p> 
xIllusions (Crisscross Art Border)	<p>Specifies an art border consisting of repeating images of a crisscross pattern, as follows (showing two repetitions):</p> 
zanyTriangles (Triangle Art Border)	<p>Specifies an art border consisting of repeating images of a</p>

Enumeration Value	Description
	<p>triangle pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated appropriately:</p> 
zigZag (Zigzag Art Border)	<p>Specifies an art border consisting of repeating images of a zigzag pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated accordingly:</p> 
zigZagStitch (Zigzag stitch)	<p>Specifies an art border consisting of a repeating image of a zigzag pattern, as follows (showing two repetitions):</p>  <p>As well, this art border shall be rotated such that the bottom of the image above is always nearest the text extents.</p> <p>At any corner where two borders of this type intersect, the intersection shall use the following image, rotated accordingly:</p> 

2.18.5 ST_BrClear (Line Break Text Wrapping Restart Location)

This simple type specifies the set of possible restart locations which may be used as to determine the next available line when a break's type attribute has a value of `textWrapping`. This property only affects the restart location when the

current run is being displayed on a line which does not span the full text extents due to the presence of a floating object (see enumeration values for details).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
all (Restart On Next Full Line)	<p>Specifies that the text wrapping break shall advance the text to the next line in the WordprocessingML document which spans the full width of the line (i.e. the next line which is not interrupted by any floating objects when those objects are positioned on the page at display time.</p> <p>[<i>Note</i>: This setting is typically used to place a single line of text next to a floating object for use as a caption. <i>end note</i>]</p>
left (Restart In Next Text Region When In Leftmost Position)	<p>Specifies that the text wrapping break shall behave as follows:</p> <p>If this line is broken into multiple regions (a floating object in the center of the page has text wrapping on both sides:</p> <ul style="list-style-type: none"> • If this is the leftmost region of text flow on this line, advance the text to the next position on the line • Otherwise, treat this as a text wrapping break of type all. <p>If this line is not broken into multiple regions, then treat this break as a text wrapping break of type none.</p> <p>If the parent paragraph is right to left, then these behaviors are also reversed.</p> <p>[<i>Note</i>: This break type is used to control the text wrapping on the left side of a floating image without preventing text from appearing on the opposite side. <i>end note</i>]</p>
none (Restart On Next Line)	<p>Specifies that the text wrapping break shall advance the text to the next line in the WordprocessingML document, regardless of its position left to right or the presence of any floating objects which intersect with the line,</p> <p>This is the setting for a typical line break in a document.</p>
right (Restart In Next Text Region When In Rightmost Position)	<p>Specifies that the text wrapping break shall behave as follows:</p> <p>If this line is broken into multiple regions (a floating object in the center of the page has text wrapping on both sides:</p> <ul style="list-style-type: none"> • If this is the rightmost region of text flow on this line, advance the text to the next position on the next line • Otherwise, treat this as a text wrapping break of type all.

Enumeration Value	Description
	<p>If this line is not broken into multiple regions, then treat this break as a text wrapping break of type none.</p> <p>If the parent paragraph is right to left, then these behaviors are also reversed.</p> <p>[<i>Note:</i> This break type is used to control the text wrapping on the right side of a floating image without preventing text from appearing on the opposite side. <i>end note</i>]</p>

2.18.6 ST_BrType (Break Types)

This simple type specifies the possible types of break characters in a WordprocessingML document. The break type determines the next location where text shall be placed after this manual break is applied to the text contents (see enumeration values for details).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
column (Column Break)	<p>Specifies that the current break shall restart itself on the next column available on the current page when the document is displayed in page view.</p> <p>If the current section is not divided into columns, or the column break occurs in the last column on the current page when displayed, then the restart location for text shall be the next page in the document.</p>
page (Page Break)	<p>Specifies that the current break shall restart itself on the next page of the document when the document is displayed in page view.</p> <p>Page breaks shall be ignored when present in frames unless the showBreaksInFrames element (§2.15.3.42) is present in the document's compatibility settings.</p>
textWrapping (Line Break)	<p>Specifies that the current break shall restart itself on the next line in the document when the document is displayed in page view.</p> <p>The determine of the next line shall be done subject to the value of the clear attribute on the specified break character.</p>

2.18.7 ST_CalendarType (Calendar Types)

This simple type specifies the possible types of calendars which may be used within the context of a WordprocessingML document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
gregorian (Gregorian)	Specifies that the Gregorian calendar shall be used. This calendar may be localized into the appropriate language as desired.
gregorianXlitEnglish (Gregorian transliterated English)	Specifies that the Gregorian transliterated English calendar shall be used.
gregorianXlitFrench (Gregorian transliterated French)	Specifies that the Gregorian transliterated French calendar shall be used.
hebrew (Hebrew)	Specifies that the Hebrew lunar calendar shall be used.
hijri (Hijri)	Specifies that the Hijri lunar calendar shall be used.
japan (Japanese Emperor Era)	Specifies that the Japanese Emperor Era calendar shall be used.
korea (Korean Tangun Era)	Specifies that the Korean Tangun Era calendar shall be used.
saka (Saka Era)	Specifies that the Saka Era calendar shall be used.
taiwan (Taiwan)	Specifies that the Taiwanese calendar shall be used.
thai (Thai)	Specifies that the Thai calendar shall be used.

2.18.8 ST_CaptionPos (Automatic Caption Positioning Values)

This simple type specifies the possible values may be used for the position of an automatically inserted caption on an object within this document. These values specify the position a given caption shall be take relative to the object it is used to label.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
above (Position Caption Above Object)	Specifies that an automatically inserted caption shall be positioned above the object that it is used to label.
below (Position Caption Below Object)	Specifies that an automatically inserted caption shall be positioned below the object that it is used to label.
left (Position Caption Left Of Object)	Specifies that an automatically inserted caption shall be positioned to the left of the object that it is used to label (the position where text typed immediately before the object would appear).
right (Position Caption Right Of Object)	Specifies that an automatically inserted caption shall be positioned to the right of the object that it is used to label (the position where text typed immediately after the object would appear).

2.18.9 ST_ChapterSep (Chapter Separator Types)

This simple type specifies the character which shall be used to separate the chapter number from the page number for page numbers in a given section, when chapter numbers are being displayed.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
colon (Colon Chapter Separator)	Specifies that a colon character shall be used to separate the chapter number from the page number when page numbers are displayed.
emDash (Em Dash Chapter Separator)	Specifies that an em dash character shall be used to separate the chapter number from the page number when page numbers are displayed.
enDash (En Dash Chapter Separator)	Specifies that an en dash character shall be used to separate the chapter number from the page number when page numbers are displayed.
hyphen (Hyphen Chapter Separator)	Specifies that a non-breaking hyphen character shall be used to separate the chapter number from the page number when page numbers are displayed.
period (Period Chapter Separator)	Specifies that a period character shall be used to separate the chapter number from the page number when page numbers are displayed.

2.18.10 ST_CharacterSpacing (Character-Level Whitespace Compression Settings)

This simple type specifies the possible ways in which full-width characters in the current WordprocessingML document may be compressed to remove additional whitespace when the contents of this document are displayed, specifically by specifying the set(s) of characters which may be compressed to remove additional whitespace.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
compressPunctuation (Compress Whitespace From Punctuation Characters)	Specifies that only whitespace characters shall have whitespace compression applied to them.
compressPunctuationAndJapaneseKana (Compress Whitespace From Both Japanese Kana And Punctuation Characters)	Specifies that whitespace and Japanese kana characters shall have whitespace compression applied to them.
doNotCompress (Do Not Compress Whitespace)	Specifies that characters shall not have whitespace compression applied to them.

2.18.11 ST_Cnf (Conditional Formatting Bitmask)

This simple type specifies the format for the set of conditional formatting properties that have been applied to this object.

These properties are expressed using a string serialization of a binary bitmask for each of the following properties (reading from the first character position right):

- First Row - Is this the first row of the table?
- Last Row - Is this the last row of the table?
- First Column - Does this belong to the first column of the table?
- Last Column - Does this belong to the last column of the table?
- Band 1 Vertical - Does this belong to a column which should receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered columns (e.g. 1,3,5,...)
- Band 2 Vertical - Does this belong to a column which should receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered columns (e.g. 2,4,6...)
- Band 1 Horizontal - Does this receive band 1 formatting? This property specifies whether the cell should receive the formatting specified for odd-numbered rows (e.g. 1,3,5,...)
- Band 2 Horizontal - Does this receive band 2 formatting? This property specifies whether the cell should receive the formatting specified for even-numbered rows (e.g. 2,4,6...)
- NE Cell - Is this part of the top-right corner of the table?
- NW Cell - Is this part of the top-left corner of the table?
- SE Cell - Is this part of the bottom-right corner of the table?
- SW Cell - Is this part of the bottom-left corner of the table?

For each of these properties, a value of 1 in the specified character position in the string means that the value is true, a value of 0 means false. All values must be specified.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 12 characters.
- This simple type's contents must match the following regular expression pattern: [01]*.

2.18.12 ST_ColorSchemeIndex (Theme Color Reference)

This simple type specifies the possible set of theme color stored in the document's Theme part which can be referenced by document content. This reference is used to map the use of the theme colors in the ST_ThemeColor enumeration to the theme colors in the theme part.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
accent1 (Accent 1 Theme Color Reference)	Specifies a reference to the accent1 theme color in the document's Theme part.
accent2 (Accent 2 Theme Color Reference)	Specifies a reference to the accent2 theme color in the document's Theme part.
accent3 (Accent 3 Theme Color Reference)	Specifies a reference to the accent3 theme color in the document's Theme part.
accent4 (Accent4 Theme Color Reference)	Specifies a reference to the accent4 theme color in the document's Theme part.
accent5 (Accent5 Theme Color Reference)	Specifies a reference to the accent5 theme color in the document's Theme part.
accent6 (Accent 6 Theme Color Reference)	Specifies a reference to the accent6 theme color in the document's Theme part.
dark1 (Dark 1 Theme Color Reference)	Specifies a reference to the dk1 theme color in the document's Theme part.

Enumeration Value	Description
dark2 (Dark 2 Theme Color Reference)	Specifies a reference to the dk2 theme color in the document's Theme part.
followedHyperlink (Followed Hyperlink Theme Color Reference)	Specifies a reference to the folHlink theme color in the document's Theme part.
hyperlink (Hyperlink Theme Color Reference)	Specifies a reference to the hlink theme color in the document's Theme part.
light1 (Light 1 Theme Color Reference)	Specifies a reference to the lt1 theme color in the document's Theme part.
light2 (Light 2 Theme Color Reference)	Specifies a reference to the lt2 theme color in the document's Theme part.

2.18.13 ST_CombineBrackets (Two Lines in One Enclosing Character Type)

This simple type specifies the type of bracket character which shall be used to enclose the two lines in one text within the current run when displayed

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
angle (Angle Brackets)	Specifies that angle bracket characters shall be used to enclose the contents of the current run's two lines in one text.
curly (Curly Brackets)	Specifies that curly bracket characters shall be used to enclose the contents of the current run's two lines in one text.
none (No Enclosing Brackets)	Specifies that no characters shall be used to enclose the contents of the current run's two lines in one text.
round (Round Brackets)	Specifies that round bracket characters shall be used to enclose the contents of the current run's two lines in one text.
square (Square Brackets)	Specifies that square bracket characters shall be used to enclose the contents of the current run's two lines in one text.

2.18.14 ST_CryptProv (Cryptographic Provider Types)

This simple type specifies the possible types of cryptographic providers which may be used.

This simple type's contents are a restriction of the XML Schema dateTime datatype.

2.18.15 ST_DecimalNumber (Decimal Number Value)

This simple type specifies that its contents will contain a whole decimal number (positive or negative), whose contents are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema integer datatype.

2.18.16 ST_DisplacedByCustomXml (Location of Custom XML Markup Displacing an Annotation)

This simple type specifies the possible values for the location of a single custom XML element's start and/or end tag relative to the location of an annotation tag in document order. This enumeration shall be used to specify that the parent annotation's placement shall be directly linked with the location of the physical presentation of a custom XML element in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
next (Displaced by Next Custom XML Markup Tag)	Specifies that this annotation anchor shall be displaced by the physical representation of the next element of custom XML markup in the document. If no custom XML markup exists in the same paragraph and after this anchor, then this setting shall be ignored.
prev (Displaced by Previous Custom XML Markup Tag)	Specifies that this annotation anchor shall be displaced by the physical representation of the previous element of custom XML markup in the document. If no custom XML markup exists in the same paragraph and directly before this anchor, then this setting shall be ignored.

2.18.17 ST_DocGrid (Document Grid Types)

Specifies the type of the current document grid, which defines the grid behavior.

The grid can define a grid which snaps all East Asian characters to grid positions, but leaves Latin text with its default spacing; a grid which adds the specified character pitch to all characters on each row; or a grid which affects only the line pitch for the current section.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
default (No Document Grid)	Specifies that no document grid shall be applied to the contents of the current section in the document.
lines (Line Grid Only)	Specifies that the parent section shall have additional line pitch added to each line within it (as specified on the docGrid element (§2.6.5)) in order to maintain the specified number of lines per page.
linesAndChars (Line and Character Grid)	Specifies that the parent section shall have both the

Enumeration Value	Description
	<p>additional line pitch and character pitch added to each line and character within it (as specified on the docGrid element (§2.6.5)) in order to maintain a specific number of lines per page and characters per line.</p> <p>When this value is set, the input specified via the user interface may be allowed in exact number of line/character pitch units.</p>
snapToChars (Character Grid Only)	<p>Specifies that the parent section shall have both the additional line pitch and character pitch added to each line and character within it (as specified on the docGrid element (§2.6.5)) in order to maintain a specific number of lines per page and characters per line.</p> <p>When this value is set, the input specified via the user interface may be restricted to the number of lines per page and characters per line, with the consumer or producer translating this information based on the current font data to get the resulting line and character pitch values</p>

2.18.18 ST_DocPartBehavior (Insertion Behavior Types)

This simple type specifies the possible sets of behaviors which may be applied to the contents of a single glossary document entry (§2.12.5) when it is added to the main document story of a WordprocessingML document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
content (Insert Content At Specified Location)	<p>Specifies that when the glossary document entry is inserted into the main document contents of the document, it shall be inserted normally as defined above.</p> <p>This includes ensuring that the final paragraph which is included in the part is not inserted, and its run content is added to the paragraph into which the current part is being inserted.</p>
p (Ensure Entry Is In New Paragraph)	Specifies that the glossary document entry shall be added into its own unique paragraph, by failing to remove the last paragraph from the entry's contents when they are added to the document.
pg (Ensure Entry Is On New Page)	Specifies that the glossary document entry shall be added into its own new page, by preceding the entry with a blank paragraph whose only content is a page break character.

2.18.19 ST_DocPartGallery (Entry Gallery Types)

This simple type specifies possible settings for the predefined gallery into which a glossary document part shall be classified. This classification, although its enumeration values may be interpreted to imply semantics around the contents of the parent glossary document entry, shall only be used to classify and sort this entry (via an application or a user interface).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
any (All Galleries)	Specifies that this glossary document entry shall be associated with all possible gallery classification values.
autoTxt (AutoText Gallery)	Specifies that this glossary document entry shall be associated with the AutoText gallery classification.
bib (Bibliography Gallery)	Specifies that this glossary document entry shall be associated with the Bibliography gallery classification.
coverPg (Cover Page Gallery)	Specifies that this glossary document entry shall be associated with the Cover Page gallery classification.
custAutoTxt (Custom AutoText Gallery)	Specifies that this glossary document entry shall be associated with the Custom AutoText gallery classification.
custBib (Custom Bibliography Gallery)	Specifies that this glossary document entry shall be associated with the Custom Bibliography gallery classification.
custCoverPg (Custom Cover Page Gallery)	Specifies that this glossary document entry shall be associated with the Custom Cover Page gallery classification.
custEq (Custom Equation Gallery)	Specifies that this glossary document entry shall be associated with the Custom Equation gallery classification.
custFtrs (Custom Footer Gallery)	Specifies that this glossary document entry shall be associated with the Custom Footer gallery classification.
custHdrs (Custom Header Gallery)	Specifies that this glossary document entry shall be associated with the Custom Header gallery classification.
custom1 (Custom 1 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 1 gallery classification.
custom2 (Custom 2 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 2 gallery classification.
custom3 (Custom 3 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 3 gallery classification.
custom4 (Custom 4 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 4 gallery classification.
custom5 (Custom 5 Gallery)	Specifies that this glossary document entry shall be associated with the Custom 5 gallery classification.
custPgNum (Custom Page Number Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number gallery classification.
custPgNumB (Custom Page Number At Bottom Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number At Bottom gallery classification.

Enumeration Value	Description
custPgNumMargins (Custom Page Number At Margins Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number At Margins gallery classification.
custPgNumT (Custom Page Number At Top Gallery)	Specifies that this glossary document entry shall be associated with the Custom Page Number At Top gallery classification.
custQuickParts (Custom Quick Parts Gallery)	Specifies that this glossary document entry shall be associated with the Custom Quick Parts gallery classification.
custTblOfContents (Custom Table of Contents Gallery)	Specifies that this glossary document entry shall be associated with the Custom Table of Contents gallery classification.
custTbIs (Custom Table Gallery)	Specifies that this glossary document entry shall be associated with the Custom Tables gallery classification.
custTextBox (Custom Text Box Gallery)	Specifies that this glossary document entry shall be associated with the Custom Text Box gallery classification.
custWatermarks (Custom Watermark Gallery)	Specifies that this glossary document entry shall be associated with the Custom Watermark gallery classification.
default (No Gallery Classification)	Specifies that this glossary document entry shall not have a gallery classification.
docParts (Document Parts Gallery)	Specifies that this glossary document entry shall be associated with the Document Parts gallery classification.
eq (Equations Gallery)	Specifies that this glossary document entry shall be associated with the Equations gallery classification.
ftIs (Footers Gallery)	Specifies that this glossary document entry shall be associated with the Footers gallery classification.
hdrs (Headers Gallery)	Specifies that this glossary document entry shall be associated with the Headers gallery classification.
pgNum (Page Numbers Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers gallery classification.
pgNumB (Page Numbers At Bottom Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers At Bottom gallery classification.
pgNumMargins (Page Numbers At Margins Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers At Margins gallery classification.
pgNumT (Page Numbers At Top Gallery)	Specifies that this glossary document entry shall be associated with the Page Numbers At Top gallery classification.
placeholder (Structured Document Tag Placeholder Text Gallery)	Specifies that this glossary document entry shall be associated with the Structured Document Tag Placeholder Text gallery classification.
tblOfContents (Table of Contents Gallery)	Specifies that this glossary document entry shall be

Enumeration Value	Description
	associated with the Table of Contents gallery classification.
tbls (Table Gallery)	Specifies that this glossary document entry shall be associated with the Tables gallery classification.
txtBox (Text Box Gallery)	Specifies that this glossary document entry shall be associated with the Text Box gallery classification.
watermarks (Watermark Gallery)	Specifies that this glossary document entry shall be associated with the Watermark gallery classification.

2.18.20 ST_DocPartType (Entry Types)

This simple type specifies the possible types which may be applied to the properties of a single glossary document entry (§2.12.5). Each of these types may, based on their values, influence the visibility and behavior of the parent glossary document entry.

This simple type's contents are a restriction of the XML Schema string datatype.
The following are possible enumeration values for this type:

Enumeration Value	Description
autoExp (Automatically Replace Name With Content)	Specifies that the type of the current glossary document entry shall allow the entry to be automatically inserted into the document whenever its name is entered into an application.
bbPlcHdr (Structured Document Tag Placeholder Text)	Specifies that the type of the current glossary document entry shall be structured document tag placeholder text.
formFld (Form Field Help Text)	Specifies that the type of the current glossary document entry shall be form field help text.
none (No Type)	Specifies no type information for the current glossary document entry.
normal (Normal)	Specifies that the type of the current glossary document entry shall be normal (i.e. a regular glossary document entry).
speller (AutoCorrect Entry)	Specifies that the type of the current glossary document entry shall be associated with the spelling and grammar tools.
toolbar (AutoText User Interface Entry)	Specifies that the type of the current glossary document entry shall be associated with a special grouping of entries associated with a single piece of user interface.

2.18.21 ST_DocProtect (Document Protection Types)

This simple type specifies the possible set of editing restrictions which may be enforced on a given WordprocessingML document.

This simple type's contents are a restriction of the XML Schema string datatype.
The following are possible enumeration values for this type:

Enumeration Value	Description
-------------------	-------------

Enumeration Value	Description
comments (Allow Editing of Comments)	Specifies that the edits made to this document shall be restricted to: <ul style="list-style-type: none"> The insertion and deletion of comments within the document The editing of regions delimited by range permissions which match the editing rights of the user account which is performing the editing.
forms (Allow Editing of Form Fields)	Specifies that the edits made to this document shall be restricted to: <ul style="list-style-type: none"> The editing of form fields in sections where the formProt element (§2.6.6) has a value of true. No restrictions in sections where the formProt element has a value of false.
none (No Editing Restrictions)	Specifies that no editing restrictions have been applied to the document.
readOnly (Allow No Editing)	Specifies that the edits made to this document shall be restricted to: <ul style="list-style-type: none"> The editing of regions delimited by range permissions which match the editing rights of the user account which is performing the editing.
trackedChanges (Allow Editing With Revision Tracking)	Specifies that the edits made to this document shall be tracked as revisions. This value shall imply the presence of the trackRevisions element (§2.15.1.90), and applications shall not allow that element's state to be changed to false .

2.18.22 ST_DocType (Document Classification Values)

This simple type specifies the possible classifications may be used for a WordprocessingML document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
eMail (E-Mail Message)	Specifies that this document shall be classified as an e-mail message.
letter (Letter)	Specifies that this document shall be classified as a letter.
notSpecified (Default Document)	Specifies that this document shall be classified as a default document.

2.18.23 ST_DropCap (Text Frame Drop Cap Location)

This simple type specifies the location which shall be used to position a drop cap text frame when the contents of that text frame are displayed in the anchor paragraph at display time.

[Note: Although a drop cap is simply a text frame, the values of this simple type are used to determine how the cap should be positioned relative to the following non-frame paragraph in relative terms (see enumeration values), rather than relying on absolute sizing. *end note*]

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
drop (Drop Cap Inside Margin)	Specifies that the drop cap text frame shall be positioned inside the text margin on the anchor paragraph when this text frame is displayed in the document.
margin (Drop Cap Outside Margin)	Specifies that the drop cap text frame shall be positioned outside of the text margin on the anchor paragraph when this text frame is displayed in the document.
none (Not Drop Cap)	Specifies that this text frame is not a drop cap text frame.

2.18.24 ST_EdGrp (Range Permission Editing Group)

This simple type specifies the set of possible aliases (or editing groups) which may be used as aliases to determine if the current user shall be allowed to edit a single range defined by a range permission within a document. This mechanism simply provides a set of predefined editing groups which may be associated with user accounts by applications in any desired manner.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
administrators (Administrator Group)	Specifies that users associated with the Administrators group shall be allowed to edit range permissions using this editing group when document protection is enabled.
contributors (Contributors Group)	Specifies that users associated with the Contributors group shall be allowed to edit range permissions using this editing group when document protection is enabled.
current (Current Group)	Specifies that users associated with the Current group shall be allowed to edit range permissions using this editing group when document protection is enabled.
editors (Editors Group)	Specifies that users associated with the Editors group shall be allowed to edit range permissions using this editing group when document protection is enabled.
everyone (All Users Have Editing Permissions)	Specifies that all users that open the document shall be allowed to edit range permissions using this editing group when document protection is enabled.
none (No Users Have Editing Permissions)	Specifies that none of the users that open the document shall be allowed to edit range permissions using this editing group when document protection is enabled.
owners (Owners Group)	Specifies that users associated with the Owners group shall be allowed to edit range permissions using this editing group when document protection is enabled.

2.18.25 ST_EdnPos (Endnote Positioning Location)

This simple type specifies the possible positions of endnotes in a document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
docEnd (Endnotes Positioned at End of Document)	Specifies that all endnotes shall be placed at the end of the current document, regardless of which section they are referenced within.
sectEnd (Endnotes Positioned at End of Section)	Specifies that endnotes shall be placed at the end of the section in which they are referenced. An endnote which is never referenced is never displayed.

2.18.26 ST_EighthPointMeasure (Measurement in Eighths of a Point)

This simple type specifies that its contents will contain a positive whole number, whose contents consist of a measurement in eighths of a point (equivalent to 1/576th of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber simple type (§2.18.107).

2.18.27 ST_Em (Emphasis Mark Type)

This simple type specifies possible types of emphasis marks which may be displayed for each non-space character in a run. This character is rendered above or below the character glyph as specified by enumeration values.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
circle (Circle Emphasis Mark Above Characters)	Specifies that the emphasis mark is a circle character which shall be rendered above each character in this run using Unicode character 0x02DA when the language of the text is not Traditional Chinese. For that language Unicode character 0x3002 shall be used instead, positioned beneath the characters.
comma (Comma Emphasis Mark Above Characters)	Specifies that the emphasis mark is a comma character which shall be rendered above each character in this run, using Unicode character 0x3001.
dot (Dot Emphasis Mark Above Characters)	Specifies that the emphasis mark is a dot character which shall be rendered above each character in this run using Unicode character 0x02D9 whenever the language of the text is not Japanese, Simplified Chinese, or Traditional Chinese. For those three languages, the emphasis mark shall be rendered as follows: <ul style="list-style-type: none"> Japanese = Unicode character 0xFF0E (dot beneath characters) Simplified Chinese = Unicode character 0xFF0E (dot beneath characters) Traditional Chinese = Unicode character 0x2027
none (No Emphasis Mark)	Specifies that there shall be no emphasis mark for any

Enumeration Value	Description
	character in this run.
underDot (Dot Emphasis Mark Below Characters)	Specifies that the emphasis mark is a dot character which shall be rendered below each character in this run using Unicode character 0xFF0E.

2.18.28 ST_FFHelpTextVal (Help Text Value)

This simple type specifies the format of optional help text which may be associated with the parent form field.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 256 characters.

2.18.29 ST_FFName (Form Field Name Value)

This simple type specifies the format of the name which may be associated with the parent form field.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 65 characters.

2.18.30 ST_FFStatusTextVal (Status Text Value)

This simple type specifies the format of optional status text which may be associated with the parent form field.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 140 characters.

2.18.31 ST_FFTextType (Text Box Form Field Type Values)

This simple type specifies the possible types of the contents of a text box form field.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
calculated (Field Calculation)	Specifies that the contents of this text box form field shall be the result of the field calculation specified by the corresponding default element (§2.16.10). This field should not be directly editable when the editing of form fields is enabled.
currentDate (Current Date Display)	Specifies that the contents of this text box form field shall be the current date when the field is updated.
currentTime (Current Time Display)	Specifies that the contents of this text box form field shall be the current time when the field is updated.
date (Date)	Specifies that the contents of this text box form field shall be treated as a date.
number (Number)	Specifies that the contents of this text box form field shall be treated as a number value.

Enumeration Value	Description
regular (Text Box)	Specifies that this text form field is a plain text field (no additional content restrictions).

2.18.32 ST_FldCharType (Complex Field Character Type)

This simple type specifies the possible values for the type of a single complex field character in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
begin (Start Character)	Specifies that the character is a start character, which defines the start of a complex field.
end (End Character)	Specifies that the character is an end character, which defines the end of a complex field.
separate (Separator Character)	Specifies that the character is a separator character, which defines the end of the field codes and the start of the field result for a complex field.

2.18.33 ST_FontFamily (Font Family Value)

This simple type specifies possible values for the font family of a font.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
auto (No Font Family)	Specifies that information about a font's font family does not exist.
decorative (Novelty Font)	Specifies the Novelty font family.
modern (Monospace Font)	Specifies a monospace font with or without serifs (monospace fonts are usually modern).
roman (Proportional Font With Serifs)	Specifies a proportional font with serifs.
script (Script Font)	Specifies a script font designed to mimic the appearance of handwriting.
swiss (Proportional Font Without Serifs)	Specifies a proportional font without serifs.

2.18.34 ST_FrameLayout (Frameset Layout Order)

This simple type specifies the possible order in which the frames (and nested framesets) in a frameset may be displayed.

When a frameset is created, it can only contain frames which are stacked in one direction:

- Vertically (one on top of another)
- Horizontally (one next to another)

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
-------------------	-------------

Enumeration Value	Description
cols (Stack Frames Horizontally)	Specifies that the frames in the frameset shall be stacked horizontally next to each other in left to right order.
none (Do Not Stack Frames)	Specifies that no frames shall be shown in the frameset.
rows (Stack Frames Vertically)	Specifies that the frames in the frameset shall be stacked vertically next to each other in top to bottom order.

2.18.35 ST_FrameScrollbar (Frame Scrollbar Visibility)

This simple type specifies the possible settings for when a scrollbar shall be visible for the contents of the current frame.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
auto (Automatically Show Scrollbar As Needed)	Specifies that the scrollbar for a frame shall automatically be hidden and/or displayed as needed based on the length of the contents.
off (Never Show Scrollbar)	Specifies that the scrollbar for a frame shall always be hidden.
on (Always Show Scrollbar)	Specifies that the scrollbar for a frame shall always be displayed (even when not needed).

2.18.36 ST_FtnEdn (Footnote or Endnote Type)

This simple type specifies the possible types of footnotes and endnotes which may be specified in a WordprocessingML document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
continuationNotice (Continuation Notice Separator)	<p>Specifies that this footnote or endnote is a continuation notice footnote or endnote.</p> <p><i>Continuation notice footnotes and endnotes</i> are used when the footnotes or endnotes exceed the length allowed on a single page. When this happens, this footnote or endnote shall be placed on the bottom of each page where the note shall continue to indicate that fact to the reader.</p>
continuationSeparator (Continuation Separator)	<p>Specifies that this footnote or endnote is a continuation separator footnote or endnote.</p> <p><i>Continuation separator footnotes and endnotes</i> are used when the footnotes or endnotes exceed the length allowed on a single page. When this happens, this footnote or endnote shall be placed between the main text contents and the continued footnotes/endnotes on all subsequent pages of the document.</p>

Enumeration Value	Description
normal (Normal Footnote/Endnote)	Specifies that this footnote or endnote is a normal footnote or endnote, and can be referenced by main document content.
separator (Separator)	Specifies that this footnote or endnote is a separator footnote or endnote. <i>Separator footnotes and endnotes</i> are used to indicate the separation between the main document's content and the footnotes or endnotes to indicate that fact to the reader.

2.18.37 ST_FtnPos (Footnote Positioning Location)

This simple type specifies the position of footnotes in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
beneathText (Footnotes Positioned Beneath Text)	Specifies that footnotes shall be displayed immediately after the last line of text on the page on which the note reference mark appears.
docEnd (Footnotes Positioned At End of Document)	Specifies that all footnotes shall be placed at the end of the current document, regardless of which section they are referenced within.
pageBottom (Footnotes Positioned at Page Bottom)	Specifies that footnotes shall be displayed at the bottom margin of the page on which the note reference mark appears.
sectEnd (Footnotes Positioned At End of Section)	Specifies that all footnotes shall be placed at the end of the section in which they are referenced. A footnote which is never referenced is never displayed.

2.18.38 ST_Guid (128-Bit GUID)

This simple type specifies that its values shall be a 128-bit globally unique identifier (GUID) value.

This simple type's contents are a restriction of the XML Schema token datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must match the following regular expression pattern: `\{[0-9A-F]{8}-[0-9A-F]{4}-[0-9A-F]{4}-[0-9A-F]{12}\}`.

2.18.39 ST_HAnchor (Horizontal Anchor Location)

This simple type specifies the horizontal position to which the parent object has been anchored in the document. This anchor position shall be used as the base location to determine the final horizontal position of the object in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
-------------------	-------------

Enumeration Value	Description
margin (Relative To Margin)	<p>Specifies that the parent object shall be horizontally anchored to the text margins.</p> <p>This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the text margin.</p>
page (Relative to Page)	<p>Specifies that the parent object shall be horizontally anchored to the page edge.</p> <p>This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the edge of the page.</p>
text (Relative to Text Extents)	<p>Specifies that the parent object shall be horizontally anchored to the text extents.</p> <p>This shall be used to specify that any horizontal positioning values shall be calculated with respect to the location of the edge of the text in the anchor paragraph (including text indentations on that paragraph within the text margins).</p>

2.18.40 ST_HdrFtr (Header or Footer Type)

This simple type specifies the possible types of headers and footers which may be specified for a given header or footer reference in a document. This value determines the page(s) on which the current header or footer shall be displayed.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
default (Default Header or Footer)	<p>Specifies that this header or footer shall appear on every page in this section which is not overridden with a specific even or first page header/footer.</p> <p>In a section with all three types specified, this type shall be used on all odd numbered pages (counting from the first page in the section, not the section numbering).</p>
even (Even Numbered Pages Only)	<p>Specifies that this header or footer shall appear on all even numbered pages in this section (counting from the first page in the section, not the section numbering).</p> <p>The appearance of this header or footer is contingent on the setting of the evenAndOddHeaders element (§2.10.1).</p>
first (First Page Only)	<p>Specifies that this header or footer shall appear on the first page in this section.</p> <p>The appearance of this header or footer is contingent on the setting of the titlePg element (§2.10.6).</p>

2.18.41 ST_HeightRule (Height Rule)

This simple type specifies the logic which shall be used to calculate the height of the parent object when it is displayed in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
atLeast (Minimum Height)	Specifies that the height of the parent object shall be at least the value specified, but may be expanded to fit its content as needed.
auto (Determine Height Based On Contents)	Specifies that the height of the parent object shall be automatically determined by the size of its contents, with no predetermined minimum or maximum size.
exact (Exact Height)	Specifies that the height of the parent object shall be exactly the value specified, regardless of the size of the contents of the object. If the contents are too large for the specified height, then they shall be clipped.

2.18.42 ST_HexColor (Color Value)

This simple type specifies that its contents will contain one of the following:

- A color values in RRGGBB format (**ST_HexColorRGB**)
- The enumeration value **auto** (**ST_HexColorAuto**)

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type is defined as a union of the following types:

- TheST_HexColorAuto simple type (§2.18.43).
- TheST_HexColorRGB simple type (§2.18.44).

2.18.43 ST_HexColorAuto ('Automatic' Color Value)

This simple type specifies that its contents will contain the enumeration value **auto**. This value shall be used to specify an automatically determined color value, the meaning of which is interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
auto (Automatically Determined Color)	Specifies that the color value may automatically be defined when this document is processed, based on the display context.

2.18.44 ST_HexColorRGB (Hexadecimal Color Value)

This simple type specifies that its contents shall contain a color value in RRGGBB hexadecimal format. This specifies that each of the red, green, and blue color values form 0-255 will be encoded as a two-digit hexadecimal number.

This simple type's contents are a restriction of the XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 3 characters.

2.18.45 ST_HighlightColor (Text Highlight Colors)

This simple type specifies the possible values for highlighting colors which may be applied as a background behind the contents of a text run.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
black (Black Highlighting Color)	Specifies that the text highlighting color for this run shall be black. The hexadecimal RGB value for this setting shall be 000000.
blue (Blue Highlighting Color)	Specifies that the text highlighting color for this run shall be blue. The hexadecimal RGB value for this setting shall be 0000FF.
cyan (Cyan Highlighting Color)	Specifies that the text highlighting color for this run shall be cyan. The hexadecimal RGB value for this setting shall be 00FFFF.
darkBlue (Dark Blue Highlighting Color)	Specifies that the text highlighting color for this run shall be dark blue. The hexadecimal RGB value for this setting shall be 000080.
darkCyan (Dark Cyan Highlighting Color)	Specifies that the text highlighting color for this run shall be dark cyan. The hexadecimal RGB value for this setting shall be 008080.
darkGray (Dark Gray Highlighting Color)	Specifies that the text highlighting color for this run shall be dark gray. The hexadecimal RGB value for this setting shall be 808080.
darkGreen (Dark Green Highlighting Color)	Specifies that the text highlighting color for this run shall be dark green. The hexadecimal RGB value for this setting shall be 008000.
darkMagenta (Dark Magenta Highlighting Color)	Specifies that the text highlighting color for this run shall be dark magenta. The hexadecimal RGB value for this setting shall be 800080.
darkRed (Dark Red Highlighting Color)	Specifies that the text highlighting color for this run shall be dark red. The hexadecimal RGB value for this setting shall be 800000.
darkYellow (Dark Yellow Highlighting Color)	Specifies that the text highlighting color for this run shall be dark cyan.

Enumeration Value	Description
	The hexadecimal RGB value for this setting shall be 808000.
green (Green Highlighting Color)	Specifies that the text highlighting color for this run shall be green. The hexadecimal RGB value for this setting shall be 00FF00.
lightGray (Light Gray Highlighting Color)	Specifies that the text highlighting color for this run shall be light gray. The hexadecimal RGB value for this setting shall be C0C0C0.
magenta (Magenta Highlighting Color)	Specifies that the text highlighting color for this run shall be magenta. The hexadecimal RGB value for this setting shall be FF00FF.
none (No Text Highlighting)	Specifies that this text run shall have no text highlighting applied to its contents.
red (Red Highlighting Color)	Specifies that the text highlighting color for this run shall be red. The hexadecimal RGB value for this setting shall be FF0000.
white (White Highlighting Color)	Specifies that the text highlighting color for this run shall be white. The hexadecimal RGB value for this setting shall be FFFFFFFF.
yellow (Yellow Highlighting Color)	Specifies that the text highlighting color for this run shall be yellow. The hexadecimal RGB value for this setting shall be FFFF00.

2.18.46 ST_Hint (Font Type Hint)

Specifies the font type which shall be used to format any ambiguous characters in the current run.

There are certain characters which are not explicitly stored in the document, and may be mapped into multiple categories of the four mentioned above. This attribute shall be used to arbitrate that conflict, and determine how ambiguities in this run shall be handled. [*Note*: This is primarily used to handle the formatting on the paragraph mark glyph, and other characters that are not stored as text in the WordprocessingML document. *end note*]

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
cs (Complex Script Font)	Specifies that the font hint for this text run shall be to use the Complex Script font defined on the run via the style hierarchy.
default (High ANSI Font)	Specifies that the font hint for this text run shall be to use the High ANSI font defined on the run via the style hierarchy.

Enumeration Value	Description
eastAsia (East Asian Font)	Specifies that the font hint for this text run shall be to use the East Asian font defined on the run via the style hierarchy.

2.18.47 ST_HpsMeasure (Measurement in Half-Points)

This simple type specifies that its contents will contain a positive whole number, whose contents consist of a measurement in half-points (equivalent to 1/144th of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber simple type (§2.18.107).

2.18.48 ST_InfoTextType (Help or Status Text Type)

This simple type specifies the possible values for the type of help or status text which may be associated with a form field.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
autoText (Glossary Document Entry)	Specifies that the value specified by the parent XML element's val attribute shall be interpreted as the name of a glossary document entry whose contents contain the help or status text.
text (Literal Text)	Specifies that the value specified by the parent XML element's val attribute shall be interpreted as the literal text for the help or status text.

2.18.49 ST_Jc (Horizontal Alignment Type)

This simple type specifies all types of alignment which are available to be applied to objects in a WordprocessingML document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
both (Justified)	Specifies that the text shall be justified between both of the text margins in the document. The lowKashida setting shall also be applied to Arabic text when this setting is applied. This type of justification shall only affect the inter-word spacing on each line, and not the inter-character spacing between each word when justifying its contents.
center (Align Center)	Specifies that the text shall be centered on the line between both of the text margins in the document.
distribute (Distribute All Characters Equally)	Specifies that the text shall be justified between both of the text margins in the document.

Enumeration Value	Description
	This type of justification shall equally affect the inter-word spacing on each line as well as the inter-character spacing between each word when justifying its contents - that is, an equal amount of additional character pitch shall be added to all characters on the line.
highKashida (Widest Kashida Length)	<p>Specifies that the kashida length for text in the current paragraph shall be extended to its widest possible length.</p> <p>This setting only affects <i>kashidas</i>, which are special characters used to extend the joiner between two Arabic characters. [Note: They are typically used to improve the appearance of justified text by visually lengthening words rather than increasing the spacing between words. <i>end note</i>]</p>
left (Align Left)	Specifies that the text shall be aligned on the left text margin in the document.
lowKashida (Low Kashida Length)	<p>Specifies that the kashida length for text in the current paragraph shall be extended to a slightly longer length. This setting shall also be applied to Arabic text when the both setting is applied.</p> <p>This setting only affects <i>kashidas</i>, which are special characters used to extend the joiner between two Arabic characters. [Note: They are typically used to improve the appearance of justified text by visually lengthening words rather than increasing the spacing between words. <i>end note</i>]</p>
mediumKashida (Medium Kashida Length)	<p>Specifies that the kashida length for text in the current paragraph shall be extended to a medium length determined by the consumer.</p> <p>This setting only affects <i>kashidas</i>, which are special characters used to extend the joiner between two Arabic characters. [Note: They are typically used to improve the appearance of justified text by visually lengthening words rather than increasing the spacing between words. <i>end note</i>]</p>
numTab (Align to List Tab)	<p>Specifies that the text shall be aligned to the list tab, which is the tab stop after the numbering for the current paragraph.</p> <p>If the current paragraph has no numbering, this setting has</p>

Enumeration Value	Description
	no effect. [Note: This justification style is used for backwards compatibility with earlier word processors, and should be deprecated in favor of hanging paragraph indentation. <i>end note</i>]
right (Align Right)	Specifies that the text shall be aligned on the right text margin in the document.
thaiDistribute (Thai Language Justification)	Specifies that the text shall be justified with an optimization for Thai. This type of justification shall affect both the inter-word spacing on each line, and the inter-character spacing between each word when justifying its contents, unlike both justification. This difference is created in that the inter-character space is increased slightly in order to ensure that the additional space created by the justification is reduced. [Note: This setting is different from justification in that the reduction in inter-character spacing would be inappropriate in Western languages. <i>end note</i>]

2.18.50 ST_Lang (Language Reference)

This simple type specifies that its contents will contain one of the following:

- A hexadecimal language code (**ST_LangCode**)
- An *ISO 639-1* letter code plus a dash plus an *ISO 3166-1 alpha-2* letter code (**ST_String**)

The contents of this language are interpreted based on the context of the parent XML element.

This simple type is defined as a union of the following types:

- TheST_LangCode simple type (§2.18.51).
- TheST_String simple type (§2.18.88).

2.18.51 ST_LangCode (Two Digit Hexadecimal Language Code)

This simple type specifies that its contents will contain a two digit hexadecimal language code defined as follows:

Language Code	Language - Country/Region
1025	Arabic - Saudi Arabia
1026	Bulgarian
1027	Catalan
1028	Chinese - Taiwan
1029	Czech
1030	Danish
1031	German - Germany
1032	Greek

Language Code	Language - Country/Region
1033	English - United States
1034	Spanish - Spain (Traditional Sort)
1035	Finnish
1036	French - France
1037	Hebrew
1038	Hungarian
1039	Icelandic
1040	Italian - Italy
1041	Japanese
1042	Korean
1043	Dutch - Netherlands
1044	Norwegian (Bokmål)
1045	Polish
1046	Portuguese - Brazil
1047	Rhaeto-Romanic
1048	Romanian
1049	Russian
1050	Croatian
1051	Slovak
1052	Albanian - Albania
1053	Swedish
1054	Thai
1055	Turkish
1056	Urdu
1057	Indonesian
1058	Ukrainian
1059	Belarusian
1060	Slovenian
1061	Estonian
1062	Latvian
1063	Lithuanian
1064	Tajik
1065	Farsi
1066	Vietnamese
1067	Armenian - Armenia
1068	Azeri (Latin)
1069	Basque

Language Code	Language - Country/Region
1070	Sorbian
1071	FYRO Macedonian
1072	Sutu
1073	Tsonga
1074	Tswana
1075	Venda
1076	Xhosa
1077	Zulu
1078	Afrikaans - South Africa
1079	Georgian
1080	Faroese
1081	Hindi
1082	Maltese
1083	Sami (Lappish)
1084	Gaelic (Scotland)
1085	Yiddish
1086	Malay - Malaysia
1087	Kazakh
1088	Kyrgyz (Cyrillic)
1089	Swahili
1090	Turkmen
1091	Uzbek (Latin)
1092	Tatar
1093	Bengali (India)
1094	Punjabi
1095	Gujarati
1096	Oriya
1097	Tamil
1098	Telugu
1099	Kannada
1100	Malayalam
1101	Assamese
1102	Marathi
1103	Sanskrit
1104	Mongolian (Cyrillic)
1105	Tibetan - People's Republic of China
1106	Welsh

Language Code	Language - Country/Region
1107	Khmer
1108	Lao
1109	Burmese
1110	Galician
1111	Konkani
1112	Manipuri
1113	Sindhi - India
1114	Syriac
1115	Sinhalese - Sri Lanka
1116	Cherokee - United States
1117	Inuktitut
1118	Amharic - Ethiopia
1119	Tamazight (Arabic)
1120	Kashmiri (Arabic)
1121	Nepali
1122	Frisian - Netherlands
1123	Pashto
1124	Filipino
1125	Divehi
1126	Edo
1127	Fulfulde - Nigeria
1128	Hausa - Nigeria
1129	Ibibio - Nigeria
1130	Yoruba
1131	Quecha - Bolivia
1132	Sepedi
1136	Igbo - Nigeria
1137	Kanuri - Nigeria
1138	Oromo
1139	Tigrigna - Ethiopia
1140	Guarani - Paraguay
1141	Hawaiian - United States
1142	Latin
1143	Somali
1144	Yi
1145	Papiamentu
1152	Uighur - China

Language Code	Language - Country/Region
1153	Maori - New Zealand
1279	HID (Human Interface Device)
2049	Arabic - Iraq
2052	Chinese - People's Republic of China
2055	German - Switzerland
2057	English - United Kingdom
2058	Spanish - Mexico
2060	French - Belgium
2064	Italian - Switzerland
2067	Dutch - Belgium
2068	Norwegian (Nynorsk)
2070	Portuguese - Portugal
2072	Romanian - Moldava
2073	Russian - Moldava
2074	Serbian (Latin)
2077	Swedish - Finland
2080	Urdu - India
2092	Azeri (Cyrillic)
2108	Gaelic (Ireland)
2110	Malay - Brunei Darussalam
2115	Uzbek (Cyrillic)
2117	Bengali (Bangladesh)
2118	Punjabi (Pakistan)
2128	Mongolian (Mongolian)
2129	Tibetan - Bhutan
2137	Sindhi - Pakistan
2143	Tamazight (Latin)
2144	Kashmiri
2145	Nepali - India
2155	Quecha - Ecuador
2163	Tigrigna - Eritrea
3073	Arabic - Egypt
3076	Chinese - Hong Kong SAR
3079	German - Austria
3081	English - Australia
3082	Spanish - Spain (Modern Sort)
3084	French - Canada

Language Code	Language - Country/Region
3098	Serbian (Cyrillic)
3179	Quecha - Peru
4097	Arabic - Libya
4100	Chinese - Singapore
4103	German - Luxembourg
4105	English - Canada
4106	Spanish - Guatemala
4108	French - Switzerland
4122	Croatian (Bosnia/Herzegovina)
5121	Arabic - Algeria
5124	Chinese - Macao SAR
5127	German - Liechtenstein
5129	English - New Zealand
5130	Spanish - Costa Rica
5132	French - Luxembourg
5146	Bosnian (Bosnia/Herzegovina)
6145	Arabic - Morocco
6153	English - Ireland
6154	Spanish - Panama
6156	French - Monaco
7169	Arabic - Tunisia
7177	English - South Africa
7178	Spanish - Dominican Republic
7180	French - West Indies
8193	Arabic - Oman
8201	English - Jamaica
8202	Spanish - Venezuela
8204	French - Reunion
9217	Arabic - Yemen
9225	English - Caribbean
9226	Spanish - Colombia
9228	French - Democratic Rep. of Congo
10241	Arabic - Syria
10249	English - Belize
10250	Spanish - Peru
10252	French - Senegal
11265	Arabic - Jordan

Language Code	Language - Country/Region
11273	English - Trinidad
11274	Spanish - Argentina
11276	French - Cameroon
12289	Arabic - Lebanon
12297	English - Zimbabwe
12298	Spanish - Ecuador
12300	French - Cote d'Ivoire
13313	Arabic - Kuwait
13321	English - Philippines
13322	Spanish - Chile
13324	French - Mali
14337	Arabic - U.A.E.
14345	English - Indonesia
14346	Spanish - Uruguay
14348	French - Morocco
15361	Arabic - Bahrain
15369	English - Hong Kong SAR
15370	Spanish - Paraguay
15372	French - Haiti
16385	Arabic - Qatar
16393	English - India
16394	Spanish - Bolivia
17417	English - Malaysia
17418	Spanish - El Salvador
18441	English - Singapore
18442	Spanish - Honduras
19466	Spanish - Nicaragua
20490	Spanish - Puerto Rico
21514	Spanish - United States
58378	Spanish - Latin America
58380	French - North Africa
Any other value	Undefined. Shall not be used.

The contents of this language are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 2 characters.

2.18.52 ST_LevelSuffix (Content Between Numbering Symbol and Paragraph Text)

This simple type specifies the types of content which shall be possible between a given numbering level's text and the text of every numbered paragraph which references that numbering level.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
nothing (Nothing Between Numbering and Text)	Specifies that no character shall be displayed between the numbering level's text and the contents of the paragraph when displaying the numbered paragraph.
space (Space Between Numbering and Text)	Specifies that a space character shall be displayed between the numbering level's text and the contents of the paragraph when displaying the numbered paragraph.
tab (Tab Between Numbering and Text)	Specifies that a tab character shall be displayed between the numbering level's text and the contents of the paragraph when displaying the numbered paragraph. This tab shall follow normal tab stop rules to determine its length.

2.18.53 ST_LineNumberRestart (Line Numbering Restart Position)

This simple type specifies when the line numbering in the parent section shall be reset to its restart value. The line numbering increments for each line (even if the line number itself is not displayed) until it reaches the restart point specified by this element.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
continuous (Continue Line Numbering From Previous Section)	Specifies that line numbering for the parent section shall continue from the line numbering from the end of the previous section, if any.
newPage (Restart Line Numbering on Each Page)	Specifies that line numbering for the parent section shall restart to the starting value whenever a new page is displayed.
newSection (Restart Line Numbering for Each Section)	Specifies that line numbering for the parent section shall restart to the starting value whenever the parent begins.

2.18.54 ST_LineSpacingRule (Line Spacing Rule)

This simple type specifies the logic which shall be used to calculate the line spacing of the parent object when it is displayed in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
atLeast (Minimum Line Height)	Specifies that the height of the line shall be at least the

Enumeration Value	Description
	value specified, but may be expanded to fit its content as needed.
auto (Automatically Determined Line Height)	Specifies that the line spacing of the parent object shall be automatically determined by the size of its contents, with no predetermined minimum or maximum size.
exact (Exact Line Height)	<p>Specifies that the height of the line shall be exactly the value specified, regardless of the size of the contents of the contents.</p> <p>If the contents are too large for the specified height, then they shall be clipped as necessary.</p>

2.18.55 ST_Lock (Locking Types)

This simple type specifies the possible set of locking behaviors which may be applied to the contents of the parent structured document tag when the contents of this documents are edited by an application (whether through a user interface or directly).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
contentLocked (Contents Cannot Be Edited At Runtime)	<p>Specifies that the editing restriction applied to the parent structured document tag shall be as follows:</p> <ul style="list-style-type: none"> • This structured document tag's contents shall not be editable • This structured document tag may be deleted in its entirety (but only entirely, no sub portion of it may be deleted)
sdtContentLocked (Contents Cannot Be Edited At Runtime And SDT Cannot Be Deleted)	<p>Specifies that the editing restriction applied to the parent structured document tag shall be as follows:</p> <ul style="list-style-type: none"> • This structured document tag's contents shall not be editable • This structured document tag shall not be deleted in its entirety
sdtLocked (SDT Cannot Be Deleted)	<p>Specifies that the editing restriction applied to the parent structured document tag shall be as follows:</p> <ul style="list-style-type: none"> • This structured document tag's contents shall be editable • This structured document tag shall not be deleted in its entirety
unlocked (No Locking)	<p>Specifies that no special locking behaviors shall be applied to the parent structured document tag.</p> <p>The default behaviors as specified on the lock element (§2.5.2.22) shall be used.</p>

2.18.56 ST_LongHexNumber (Four Digit Hexadecimal Number Value)

This simple type specifies a number value specified as a four octet (eight digit) hexadecimal number), whose contents are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 4 characters.

2.18.57 ST_MacroName (Script Subroutine Name Value)

This simple type specifies a subroutine in a scripting language which may be executed based on the context of the parent XML element. The language and location of this subroutine may be determined using any method desired by an application.

This simple type's contents are a restriction of the XML Schema string datatype.

This simple type also specifies the following restrictions:

- This simple type's contents have a maximum length of 33 characters.

2.18.58 ST_MailMergeDataType (Mail Merge Data Source Type Values)

This simple type specifies the possible values for the types of external data sources to be connected to via the Dynamic Data Exchange (DDE) system (such as a spreadsheet or database), or the alternative method of data access if the Dynamic Data Exchange system is not used. This setting is purely a suggestion of the data source access mechanism which shall be used, and may be ignored in favor of an alternative mechanism if one is present.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
database (Database Data Source)	Specifies that a given merged WordprocessingML document has been connected to a database via the Dynamic Data Exchange (DDE) system.
native (Office Data Source Object Data Source)	Specifies that a given merged WordprocessingML document has been connected to an external data source via the Office Data Source Object (ODSO) interface.
odbc (Open Database Connectivity Data Source)	Specifies that a given merged WordprocessingML document has been connected to an external data source via the Open Database Connectivity interface.
query (Query Data Source)	Specifies that a given merged WordprocessingML document has been connected to an external data source using an external query tool.
spreadsheet (Spreadsheet Data Source)	Specifies that a given merged WordprocessingML document has been connected to a database via the Dynamic Data Exchange (DDE) system.
textFile (Text File Data Source)	Specifies that a given merged WordprocessingML document has been connected to a text file via the Dynamic Data Exchange (DDE) system.

2.18.59 ST_MailMergeDest (Merged Document Destination Types)

This simple type specifies the possible results which may be generated when a mail merge is carried out on a given WordprocessingML source document. In other words, this element is used to specify what is to be done with the merged

documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
email (Send Merged Documents as E-mail Messages)	Specifies that conforming hosting applications shall generate emails using the documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.
fax (Send Merged Documents as Faxes)	Specifies that conforming hosting applications shall generate faxes using the documents that result from populating the fields within a given merged WordprocessingML document with data from the specified external data source.
newDocument (Send Merged Documents to New Documents)	Specifies that conforming hosting applications shall generate new documents by populating the fields within a given merged WordprocessingML document with data from the specified external data source.
printer (Send Merged Documents to Printer)	Specifies that conforming hosting applications shall print the documents that result from populating the fields within a given merged WordprocessingML document with external data from the specified external data source.

2.18.60 ST_MailMergeDocType (Source Document Types)

This simple type specifies the possible types for a given WordprocessingML source document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
catalog (Catalog Source Document)	Specifies that the mail merge source document is of the catalog type.
email (E-Mail Source Document)	Specifies that the mail merge source document is of the e-mail message type.
envelopes (Envelope Source Document)	Specifies that the mail merge source document is of the envelope type.
fax (Fax Source Document)	Specifies that the mail merge source document is of the fax type.
formLetters (Form Letter Source Document)	Specifies that the mail merge source document is of the form letter type.
mailingLabels (Mailing Label Source Document)	Specifies that the mail merge source document is of the mailing label type.

2.18.61 ST_MailMergeOdsoFMDFieldType (Merge Field Mapping Types)

This simple type specifies the possible types used to indicate if a given mail merge field has been mapped to a column in the given external data source.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
dbColumn (Field Mapping to Data Source Column)	Specifies that the mail merge field has been mapped to a column in the given external data source.
null (Field Not Mapped)	Specifies that the mail merge field has not been mapped to a column in the given external data source.

2.18.62 ST_MailMergeSourceType (Mail Merge ODSO Data Source Types)

This simple type specifies the type of external data source to be connected to via as part of the ODSO connection information for this mail merge. This setting is purely a suggestion of the data source type which is being used for this mail merge, and may be ignored in favor of an alternative mechanism if one is present.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
addressBook (Address Book Data Source)	Specifies that a given merged WordprocessingML document has been connected to an address book of contacts.
database (Database Data Source)	Specifies that a given merged WordprocessingML document has been connected to a database.
document1 (Alternate Document Format Data Source)	Specifies that a given merged WordprocessingML document has been connected to another document format supported by the producing application. The format of this document is application-defined and outside the scope of this Office Open XML Standard.
document2 (Alternate Document Format Data Source Two)	Specifies that a given merged WordprocessingML document has been connected to another document format supported by the producing application. The format of this document is application-defined and outside the scope of this Office Open XML Standard.
email (E-Mail Program Data Source)	Specifies that a given merged WordprocessingML document has been connected to an e-mail application.
legacy (Legacy Document Format Data Source)	Specifies that a given merged WordprocessingML document has been connected to a legacy document format supported by the producing application. The format of this legacy document is application-defined and outside the scope of this Office Open XML Standard.
master (Aggregate Data Source)	Specifies that a given merged WordprocessingML document has been connected to a data source which aggregates other data sources.
native (Native Data Source)	Specifies that a given merged WordprocessingML document has been connected to another document format native to

Enumeration Value	Description
	the producing application. The format of this document is application-defined and outside the scope of this Office Open XML Standard.
text (Text File Data Source)	Specifies that a given merged WordprocessingML document has been connected to a text file.

2.18.63 ST_Merge (Merged Cell Type)

This element specifies the way in which a cell shall be included in a merged group of cells (horizontally or vertically) within the parent table.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
continue (Continue Merged Region)	Specifies that the current cell continues a previously existing merged group of cells in the parent table. If the previous cell in the document (horizontally or vertically) does not either begin or continue a set of merged cells, then this value shall be ignored (i.e. a group of merged cells must start with a merge whose ST_Merge value is restart).
restart (Start/Restart Merged Region)	Specifies that the current cell starts (or restarts) a group of merged cells in the parent table. After this value, all following cells which have a value of continue shall be merged into this merged cell group.

2.18.64 ST_MultiLevelType (Numbering Definition Type)

This simple type specifies the possible types of numbering which may be defined by a given abstract numbering type. This information shall only be used by a consumer to determine user interface behaviors for this numbering definition, and shall not be used to limit the behavior of the list (i.e. a list with multiple levels marked as **singleLevel** shall not be prevented from using levels 2 through 9).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
hybridMultilevel (Hybrid Multilevel Numbering Definition)	Specifies that this numbering definition defines a numbering format consisting of a multiple levels, each of a potentially different type (bullets vs. level text).
multilevel (Multilevel Numbering Definition)	Specifies that this numbering definition defines a numbering format consisting of a multiple levels, each of the same type (bullets vs. level text).
singleLevel (Single Level Numbering Definition)	Specifies that this numbering definition defines a numbering format consisting of a single level only.

2.18.65 ST_NumberFormat (Numbering Format)

This simple type specifies the numbering format which shall be used for a group of automatically numbered objects,

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
aiueo (AIUEO Order Hiragana)	Specifies that the sequence shall consist of hiragana characters in the traditional a-i-u-e-o order.
aiueoFullWidth (Full-Width AIUEO Order Hiragana)	Specifies that the sequence shall consist of full-width hiragana characters in the traditional a-i-u-e-o order.
arabicAbjad (Arabic Abjad Numerals)	Specifies that the sequence shall consist of ascending Abjad numerals.
arabicAlpha (Arabic Alphabet)	Specifies that the sequence shall consist of characters in the Arabic alphabet.
bullet (Bullet)	Specifies that the sequence shall consist of bullet characters.
cardinalText (Cardinal Text)	Specifies that the sequence shall consist of cardinal text of the run language.
chicago (Chicago Manual of Style)	Specifies that the sequence shall consist of characters as defined in the Chicago Manual of Style.
chineseCounting (Chinese Counting System)	Specifies that the sequence shall consist of ascending numbers from the Chinese counting system.
chineseCountingThousand (Chinese Counting Thousand System)	Specifies that the sequence shall consist of sequential numbers from the Chinese counting thousand system.
chineseLegalSimplified (Chinese Legal Simplified Format)	Specifies that the sequence shall consist of sequential numbers from the Chinese simplified legal format.
chosung (Korean Chosung Numbering)	Specifies that the sequence shall consist of sequential

Enumeration Value	Description
	numbers from the Korean Chosung format.
decimal (Decimal Numbers)	Specifies that the sequence shall consist of decimal numbering.
decimalEnclosedCircle (Decimal Numbers Enclosed in a Circle)	<p>Specifies that the sequence shall consist of decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character.</p> <p>Once the specified sequence reaches 21, the numbers may be replaced with non-enclosed equivalents.</p>
decimalEnclosedCircleChinese (Decimal Numbers Enclosed in a Circle)	<p>Specifies that the sequence shall consist of decimal numbering enclosed in a circle, using the enclosed alphanumeric glyph character.</p> <p>Once the specified sequence reaches 11, the numbers may be replaced with non-enclosed equivalents.</p>
decimalEnclosedFullstop (Decimal Numbers Followed by a Period)	<p>Specifies that the sequence shall consist of decimal numbering followed by a period, using the enclosed alphanumeric glyph character.</p> <p>Once the specified sequence reaches 21, the numbers may be replaced with non-enclosed equivalents.</p>
decimalEnclosedParen (Decimal Numbers Enclosed in Parenthesis)	<p>Specifies that the sequence shall consist of decimal numbering enclosed in parenthesis, using the enclosed alphanumeric glyph character.</p> <p>Once the specified sequence reaches 21, the numbers may be replaced with non-enclosed equivalents.</p>
decimalFullWidth (Double Byte Arabic Numerals)	Specifies that the sequence shall consist of double-byte Arabic numbering.
decimalFullWidth2 (Double Byte Arabic Numerals Alternate)	Specifies that the sequence shall consist of an alternative set of double-byte Arabic numbering, if one exists in the run font.

Enumeration Value	Description
decimalHalfWidth (Single Byte Arabic Numerals)	Specifies that the sequence shall consist of single-byte Arabic numbering.
decimalZero (Initial Zero Arabic Numerals)	Specifies that the sequence shall consist of Arabic numbering with a zero added to numbers one through nine.
ganada (Korean Ganada Numbering)	Specifies that the sequence shall consist of sequential numbers from the Korean Ganada format.
hebrew1 (Hebrew Numerals)	Specifies that the sequence shall consist of Hebrew numerals.
hebrew2 (Hebrew Alphabet)	Specifies that the sequence shall consist of the Hebrew alphabet.
hex (Hexadecimal Numbering)	Specifies that the sequence shall consist of hexadecimal numbering.
hindiConsonants (Hindi Consonants)	Specifies that the sequence shall consist of Hindi consonants.
hindiCounting (Hindi Counting System)	Specifies that the sequence shall consist of sequential numbers from the Hindi counting system.
hindiNumbers (Hindi Numbers)	Specifies that the sequence shall consist of Hindi numbers.
hindiVowels (Hindi Vowels)	Specifies that the sequence shall consist of Hindi vowels.
ideographDigital (Ideographs)	Specifies that the sequence shall consist of sequential numerical ideographs enclosed in a circle, using the appropriate character.
ideographEnclosedCircle (Ideographs Enclosed in a Circle)	Specifies that the sequence shall consist of sequential numerical ideographs enclosed in a circle, using the appropriate character.

Enumeration Value	Description
	Once the specified sequence reaches 11, the numbers may be replaced with non-enclosed equivalents.
ideographLegalTraditional (Traditional Legal Ideograph Format)	Specifies that the sequence shall consist of sequential numerical traditional legal ideographs.
ideographTraditional (Traditional Ideograph Format)	Specifies that the sequence shall consist of sequential numerical traditional ideographs.
ideographZodiac (Zodiac Ideograph Format)	Specifies that the sequence shall consist of sequential zodiac ideographs.
ideographZodiacTraditional (Traditional Zodiac Ideograph Format)	Specifies that the sequence shall consist of sequential traditional zodiac ideographs.
iroha (Iroha Ordered Katakana)	Specifies that the sequence shall consist of the iroha.
irohaFullWidth (Full-Width Iroha Ordered Katakana)	Specifies that the sequence shall consist of the full-width forms of the iroha.
japaneseCounting (Japanese Counting System)	Specifies that the sequence shall consist of sequential numbers from the Japanese counting system.
japaneseDigitalTenThousand (Japanese Digital Ten Thousand Counting System)	Specifies that the sequence shall consist of sequential numbers from the Japanese digital ten thousand counting system.
japaneseLegal (Japanese Legal Numbering)	Specifies that the sequence shall consist of sequential numbers from the Japanese legal counting system.
koreanCounting (Korean Counting System)	Specifies that the sequence shall consist of sequential numbers from the Korean counting system.
koreanDigital (Korean Digital Counting System)	Specifies that the sequence shall consist of sequential

Enumeration Value	Description
	numbers from the Korean digital counting system.
koreanDigital2 (Korean Digital Counting System Alternate)	Specifies that the sequence shall consist of sequential numbers from the Korean digital counting system.
koreanLegal (Korean Legal Numbering)	Specifies that the sequence shall consist of sequential numbers from the Korean legal numbering system.
lowerLetter (Lowercase Latin Alphabet)	Specifies that the sequence shall consist of the letters of the Latin alphabet in lower case.
lowerRoman (Lowercase Roman Numerals)	Specifies that the sequence shall consist of lowercase roman numerals.
none (No Numbering)	Specifies that the sequence shall not display any numbering.
numberInDash (Number With Dashes)	Specifies that the sequence shall consist of the Arabic numbering surrounded by dash characters.
ordinal (Ordinal)	Specifies that the sequence shall consist of ordinals of the run language.
ordinalText (Ordinal Text)	Specifies that the sequence shall consist of ordinal text of the run language.
russianLower (Lowercase Russian Alphabet)	Specifies that the sequence shall consist of the letters of the Russian alphabet in lower case.
russianUpper (Uppercase Russian Alphabet)	Specifies that the sequence shall consist of the letters of the Russian alphabet in upper case.
taiwaneseCounting (Taiwanese Counting System)	Specifies that the sequence shall consist of sequential numbers from the Taiwanese counting system.
taiwaneseCountingThousand (Taiwanese Counting Thousand System)	Specifies that the sequence shall consist of sequential numbers from the Taiwanese counting thousand system.

Enumeration Value	Description
taiwaneseDigital (Taiwanese Digital Counting System)	Specifies that the sequence shall consist of sequential numbers from the Taiwanese digital counting system.
thaiCounting (Thai Counting System)	Specifies that the sequence shall consist of sequential numbers from the Thai counting system.
thaiLetters (Thai Letters)	Specifies that the sequence shall consist of Thai letters.
thaiNumbers (Thai Numerals)	Specifies that the sequence shall consist of Thai numerals.
upperLetter (Uppercase Latin Alphabet)	Specifies that the sequence shall consist of the letters of the Latin alphabet in upper case.
upperRoman (Uppercase Roman Numerals)	Specifies that the sequence shall consist of uppercase roman numerals.
vietnameseCounting (Vietnamese Numerals)	Specifies that the sequence shall consist of Vietnamese numerals.

2.18.66 ST_OnOff (On/Off Value)

This simple type specifies a set of values for any binary (on or off) property defined in a WordprocessingML document. A value of **on**, **1**, or **true** specifies that the property shall be turned on. This is the default value for this attribute, and is implied when the parent element is present, but this attribute is omitted.

A value of **off**, **0**, or **false** specifies that the property shall be explicitly turned off.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
0 (False)	Specifies that the binary state of this property is off (parent property is explicitly not applied).
1 (True)	Specifies that the binary state of this property is on (parent property is explicitly applied).
false (False)	Specifies that the binary state of this property is off (parent property is explicitly not applied).
off (False)	Specifies that the binary state of this property is off (parent property is explicitly not applied).

Enumeration Value	Description
on (True)	Specifies that the binary state of this property is on (parent property is explicitly applied).
true (True)	Specifies that the binary state of this property is on (parent property is explicitly applied).

2.18.67 ST_PageBorderDisplay (Page Border Display Options)

This simple type specifies the pages in the parent section on which the page border shall be printed.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
allPages (Display Page Border on All Pages)	Specifies that the page border shall be displayed on all pages in the parent section.
firstPage (Display Page Border on First Page)	Specifies that the page border shall be displayed on only the first page in the parent section.
notFirstPage (Display Page Border on All Pages Except First)	Specifies that the page border shall be displayed on only the first page in the parent section.

2.18.68 ST_PageBorderOffset (Page Border Positioning Base)

This simple type specifies how the relative positioning of the page borders shall be calculated.

If the value of this attribute is **text**, then the space attribute on each page border shall be interpreted as the distance from the text margins that shall be left before the page border.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
page (Page Border Is Positioned Relative to Page Edges)	Specifies that the space attribute on each page border shall be interpreted as the distance from the edge of the page that shall be left before the page border.
text (Page Border Is Positioned Relative to Text Extents)	Specifies that the space attribute on each page border shall be interpreted as the distance from the edge of the text extents (text margins) that shall be left before the page border..

2.18.69 ST_PageBorderZOrder (Page Border Z-Order)

This simple type specifies whether the page border is positioned above or below intersecting texts and objects in this document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
back (Page Border Behind Text)	Specifies that the page border shall be rendered beneath any text or object which intersects it - effectively placing it

Enumeration Value	Description
	at the lowest z-order on the page.
front (Page Border Ahead of Text)	Specifies that the page border shall be rendered above any text or object which intersects it - effectively placing it at the highest z-order on the page.

2.18.70 ST_PageOrientation (Page Orientation)

This simple type specifies the orientation of all pages in the parent section. This information is used to determine the actual paper size to use when printing the file.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
landscape (Landscape Mode)	Specifies that pages in this section shall be printed in landscape mode, which prints the page contents with a 90 degree rotation with respect to the normal page orientation.
portrait (Portrait Mode)	Specifies that pages in this section shall be printed in portrait mode.

2.18.71 ST_Panose (Panose-1 Number)

This simple type specifies a number consisting of 10 hexadecimal digits which defines the Panose-1 classification number a font.

This simple type's contents are a restriction of the XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 10 characters.

2.18.72 ST_Pitch (Font Pitch Value)

This simple type specifies the possible values for the font pitch of a font.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
default (Default)	Specifies that no information is available about the pitch of a font.
fixed (Fixed Width)	Specifies that this is a fixed width font.
variable (Proportional Width)	Specifies that this is a proportional width font.

2.18.73 ST_PixelsMeasure (Measurement in Pixels)

This simple type specifies that its contents will contain a positive whole number, whose contents consist of a measurement in pixels.

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber simple type (§2.18.107).

2.18.74 ST_PointMeasure (Measurement in Points)

This simple type specifies that its contents will contain a positive whole number, whose contents consist of a measurement in points (equivalent to 1/72nd of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber simple type (§2.18.107).

2.18.75 ST_Proof (Proofing State Values)

This simple type specifies the values which may be used to indicate the status of a given hosting application's grammar and spell checking when a given WordprocessingML document was last saved.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
clean (Check Completed)	Specifies that the given proofing engine completed checking the document when it was last saved.
dirty (Check Not Completed)	Specifies that the given proofing engine did not complete checking the document when it was last saved.

2.18.76 ST_ProofErr (Proofing Error Type)

This simple type specifies the possible values for the types of proofing error markers which can appear in the contents of a WordprocessingML document to indicate the last known state of any spell- and grammar-checking performed on the contents of this document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
gramEnd (End of Region Marked as Grammatical Error)	Specifies that this proofing error marker shall indicate the start of a region to be marked as a grammatical error in the document.
gramStart (Start of Region Marked as Grammatical Error)	Specifies that this proofing error marker shall indicate the end of a region to be marked as a grammatical error in the document.
spellEnd (End of Region Marked as Spelling Error)	Specifies that this proofing error marker shall indicate the end of a region to be marked as a spelling error in the document.
spellStart (Start of Region Marked as Spelling Error)	Specifies that this proofing error marker shall indicate the start of a region to be marked as a spelling error in the document.

2.18.77 ST_PTabAlignment (Absolute Position Tab Alignment)

This simple type specifies the alignment of an absolutely positioned tab character in a document. This alignment value determines the position on the line to which this absolute tab shall advance, as well as the alignment of the text entered after the alignment tab character position.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
center (Center)	Specifies that the positional tab should be center aligned on the line relative to the specified base (the text margins with or without indents), and that the text at that location shall be center aligned.
left (Left)	Specifies that the positional tab should be left aligned on the line relative to the specified base (the text margins with or without indents), and that the text at that location shall be left aligned.
right (Right)	Specifies that the positional tab should be right aligned on the line relative to the specified base (the text margins with or without indents), and that the text at that location shall be right aligned.

2.18.78 ST_PTabLeader (Absolute Position Tab Leader Character)

This simple type specifies the characters which may be used to fill in the space created by a positional tab. This character shall be repeated as required to completely fill the tab spacing generated by the positional tab character.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
dot (Dot Leader Character)	Specifies that the leader character for this positional tab stop shall be a dot.
hyphen (Hyphen Leader Character)	Specifies that the leader character for this positional tab stop shall be a hyphen.
middleDot (Centered Dot Leader Character)	Specifies that the leader character for this positional tab stop shall be a centered dot.
none (No Leader Character)	Specifies that there shall be no leader character for this positional tab.
underscore (Underscore Leader Character)	Specifies that the leader character for this positional tab stop shall be an underscore.

2.18.79 ST_PTabRelativeTo (Absolute Position Tab Positioning Base)

Specifies the possible extents which may be used to calculate the absolute positioning of this positional tab character.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
indent (Relative To Indents)	Specifies that the absolute positioning of the tab shall be relative to the indents.
margin (Relative To Text Margins)	Specifies that the absolute positioning of the tab shall be relative to the margins.

2.18.80 ST_RestartNumber (Footnote/Endnote Numbering Restart Locations)

This simple type specifies the possible values for when the automatic numbering of footnotes or endnotes shall be restarted.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
continuous (Continue Numbering From Previous Section)	Specifies that the numbering of footnotes or endnotes shall continue from the previous section in the document.
eachPage (Restart Numbering On Each Page)	Specifies that the numbering of footnotes or endnotes shall be restarted to its starting value for each unique page in the document.
eachSect (Restart Numbering For Each Section)	Specifies that the numbering of footnotes or endnotes shall be restarted to its starting value for each unique section in the document.

2.18.81 ST_RubyAlign (Phonetic Guide Text Alignment)

This simple type specifies the possible alignment settings which may be used to determine the placement of phonetic guide text with respect to the base text when this phonetic guide is displayed.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
center (Center)	Specifies that the phonetic guide text shall be centered with respect to the base text in this document.
distributeLetter (Distribute All Characters)	<p>Specifies that the phonetic guide text shall be distributed with respect to the base text in this document.</p> <p>This type of justification shall equally affect the inter-word spacing on each line as well as the inter-character spacing between each word when justifying its contents - that is, an equal amount of additional character pitch shall be added to</p>

Enumeration Value	Description
	all characters on the line.
distributeSpace (Distribute all Characters w/ Additional Space On Either Side)	<p>Specifies that the phonetic guide text shall be distributed with respect to the base text in this document, with additional space added to the guide text to ensure it is indented with respect to the base text.</p> <p>This type of justification shall equally affect the inter-word spacing on each line as well as the inter-character spacing between each word when justifying its contents - that is, an equal amount of additional character pitch shall be added to all characters on the line. As well, an additional space is added before and after the guide text to ensure it is indented with respect to the base text.</p>
left (Left Aligned)	Specifies that the phonetic guide text shall be left aligned with respect to the base text in this document.
right (Right Aligned)	Specifies that the phonetic guide text shall be right aligned with respect to the base text in this document.
rightVertical (Vertically Aligned to Right of Base Text)	Specifies that the phonetic guide text shall be right aligned with respect to the base text in this document, and shall always be displayed vertically and to the right of the base text, regardless of the alignment of the base text.

2.18.82 ST_SdtDateMappingType (Date Storage Format Types)

This simple type specifies then possible types of translations which may be performed on the displayed date in a date picker structured document tag when the current contents are saved into the associated custom XML data via the **dataBinding** element (§2.5.2.6).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
date (XML Schema Date Format)	Specifies that the date specified in the parent date picker structured document tag shall be converted to the xsd:date format when stored in a mapped XML element.
dateTime (XML Schema DateTime Format)	Specifies that the date specified in the parent date picker structured document tag shall be converted to the xsd:dateTime format when stored in a mapped XML element.

Enumeration Value	Description
text (Same As Display)	Specifies that no translation shall be performed on the displayed date when stored in a mapped XML element - the mapped contents shall be the same as the displayed contents.

2.18.83 ST_SectionMark (Section Type)

Specifies the type of the current section.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
continuous (Continuous Section Break)	Specifies a continuous section break, which begin the new section on the following paragraph. This means that continuous section breaks might not specify certain page-level section properties, since they must be inherited from the following section. These breaks, however, can specify other section properties, such as line numbering and footnote/endnote settings.
evenPage (Even Page Section Break)	Specifies an even page section break, which begins the new section on the next even-numbered page, leaving the next odd page blank if necessary.
nextColumn (Column Section Break)	Specifies a column section break, which begins the new section on the following column on the page.
nextPage (Next Page Section Break)	Specifies a next page section break, which begins the new section on the following page.
oddPage (Odd Page Section Break)	Specifies an odd page section break, which begins the new section on the next odd-numbered page, leaving the next even page blank if necessary.



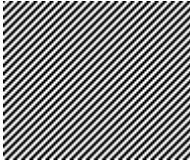

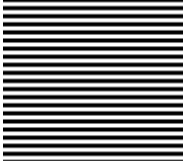
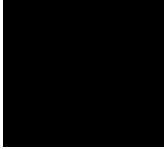
2.18.84 ST_Shd (Shading Patterns)

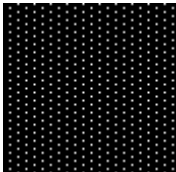
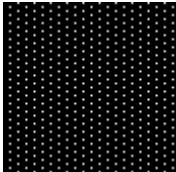
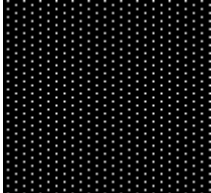
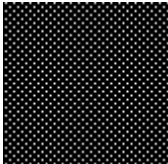
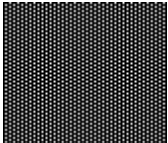
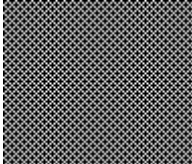
This simple type specifies the pattern which shall be used to lay the pattern color over the background color for a shading. This pattern consists of a mask which is applied over the background shading color to get the locations where the pattern color should be shown. Each of these possible masks are shown in the enumeration values located below. In each example, black has been used as the fill color, and white has been used as the pattern color.

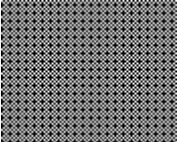
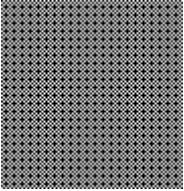
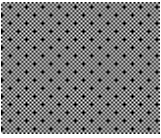
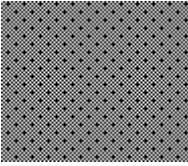
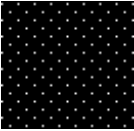
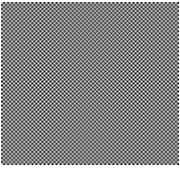
This simple type's contents are a restriction of the XML Schema string datatype.


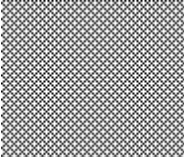
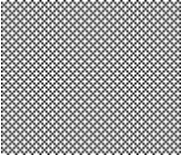
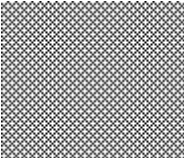
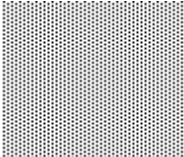

The following are possible enumeration values for this type:

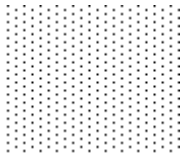
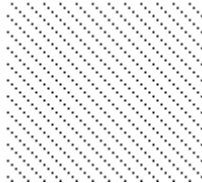

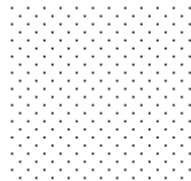


Enumeration Value	Description
clear (No Pattern)	Specifies that there shall be no pattern used on the current shaded region (i.e. the pattern shall be a complete fill with the background color), as follows:

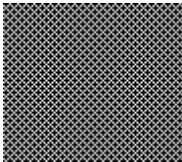

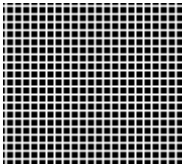
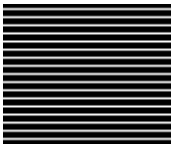
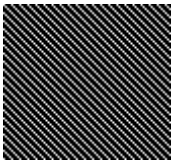
Enumeration Value	Description
	
diagCross (Diagonal Cross Pattern)	Specifies that the pattern used on the current shaded region shall be a series of diagonal crosses, as follows: 
diagStripe (Diagonal Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of diagonal stripes, as follows: 
horzCross (Horizontal Cross Pattern)	Specifies that the pattern used on the current shaded region shall be a series of horizontal crosses, as follows: 
horzStripe (Horizontal Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of horizontal stripes, as follows: 
nil (No Pattern)	Specifies that there shall be no pattern used on the current shaded region (i.e. the pattern shall be a complete fill with the background color), as follows: 
pct10 (10% Fill Pattern)	Specifies that the pattern used for the current shaded region shall be a 10% fill pattern, as follows:

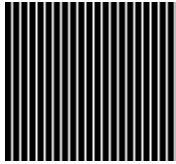
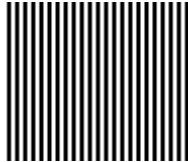
Enumeration Value	Description
	
pct12 (12.5% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 12.5% fill pattern, as follows:</p> 
pct15 (15% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 15% fill pattern, as follows:</p> 
pct20 (20% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 20% fill pattern, as follows:</p> 
pct25 (25% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 25% fill pattern, as follows:</p> 
pct30 (30% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 30% fill pattern, as follows:</p> 
pct35 (35% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 35% fill pattern, as follows:</p>

Enumeration Value	Description
	
pct37 (37.5% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 37.5% fill pattern, as follows:</p> 
pct40 (40% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 40% fill pattern, as follows:</p> 
pct45 (45% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 45% fill pattern, as follows:</p> 
pct5 (5% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 5% fill pattern, as follows:</p> 
pct50 (50% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 50% fill pattern, as follows:</p> 
pct55 (55% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 55% fill pattern, as follows:</p>

Enumeration Value	Description
	
pct60 (60% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 60% fill pattern, as follows:</p> 
pct62 (62.5% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 62.5% fill pattern, as follows:</p> 
pct65 (65% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 65% fill pattern, as follows:</p> 
pct70 (70% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 70% fill pattern, as follows:</p> 
pct75 (75% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 75% fill pattern, as follows:</p> 
pct80 (80% Fill Pattern)	<p>Specifies that the pattern used for the current shaded region shall be a 80% fill pattern, as follows:</p>

Enumeration Value	Description
	
pct85 (85% Fill Pattern)	Specifies that the pattern used for the current shaded region shall be a 85% fill pattern, as follows: 
pct87 (87.5% Fill Pattern)	Specifies that the pattern used for the current shaded region shall be a 87.5% fill pattern, as follows: 
pct90 (90% Fill Pattern)	Specifies that the pattern used for the current shaded region shall be a 90% fill pattern, as follows: 
pct95 (95% Fill Pattern)	Specifies that the pattern used for the current shaded region shall be a 95% fill pattern, as follows: 
reverseDiagStripe (Reverse Diagonal Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of reverse diagonal stripes, as follows: 
solid (100% Fill Pattern)	Specifies that the pattern used for the current shaded region shall be a 100% fill pattern, as follows:

Enumeration Value	Description
thinDiagCross (Thin Diagonal Cross Pattern)	Specifies that the pattern used on the current shaded region shall be a series of thin diagonal crosses, as follows: 
thinDiagStripe (Thin Diagonal Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of thin diagonal stripes, as follows: 
thinHorzCross (Thin Horizontal Cross Pattern)	Specifies that the pattern used on the current shaded region shall be a series of thin horizontal crosses, as follows: 
thinHorzStripe (Thin Horizontal Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of thin horizontal stripes, as follows: 
thinReverseDiagStripe (Thin Reverse Diagonal Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of thin reverse diagonal stripes, as follows: 
thinVertStripe (Thin Vertical Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of thin vertical stripes, as follows:

Enumeration Value	Description
	
vertStripe (Vertical Stripe Pattern)	Specifies that the pattern used on the current shaded region shall be a series of vertical stripes, as follows: 

2.18.85 ST_ShortHexNumber (Two Digit Hexadecimal Number Value)

This simple type specifies a number value specified as a two octet hexadecimal number), whose contents are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 2 characters.

2.18.86 ST_SignedHpsMeasure (Signed Measurement in Half-Points)

This simple type specifies that its contents will contain a positive or negative whole number, whose contents consist of a measurement in half-points (equivalent to 1/144th of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema integer datatype.

2.18.87 ST_SignedTwipsMeasure (Signed Measurement in Twentieths of a Point)

This simple type specifies that its contents will contain a positive or negative whole number, whose contents consist of a measurement in twentieths of a point (equivalent to 1/1440th of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema integer datatype.

2.18.88 ST_String (String)

This simple type specifies that its contents will contain a string. The contents of this string are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema string datatype.

2.18.89 ST_StyleType (Style Types)

This simple type specifies the possible values for the types of style definitions defined within a WordprocessingML document. WordprocessingML supports six types of style definitions:

- Paragraph styles
- Character styles
- Table styles

- Numbering styles
- Linked styles (paragraph + character)
- Default paragraph + character properties

Each of the first four types corresponds to a different value below, and therefore defines the type of the current style.

[Note: The last two types are unique in that they are not simply a style type: a linked style is a pairing of a character and paragraph style via the **link** element (§2.7.3.6); and the document default properties are defined via the **docDefaults** element (§2.7.4.1). *end note*]

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
character (Character Style)	Specifies that the parent style definition is a character style.
numbering (Numbering Style)	Specifies that the parent style definition is a numbering style.
paragraph (Paragraph Style)	Specifies that the parent style definition is a paragraph style.
table (Table Style)	Specifies that the parent style definition is a table style.

2.18.90 ST_TabJc (Custom Tab Stop Type)

This simple type specifies the available types of custom tab stop, which determines the behavior of the tab stop and the alignment which shall be applied to text entered at the current custom tab stop.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
bar (Bar Tab)	Specifies that the current tab is a bar tab. A <i>bar tab</i> is a tab which does not result in a custom tab stop in the parent paragraph (this tab stop location shall be skipped when positioning custom tab characters), but instead shall be used to draw a vertical line (or bar) at this location in the parent paragraph.
center (Centered Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is centered (i.e. all text runs following this tab stop and preceding the next tab stop shall be centered around the tab stop location).
clear (No Tab Stop)	Specifies that the current tab stop is cleared and shall be removed and ignored when processing the contents of this document.
decimal (Decimal Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is aligned around the first decimal character in the following text runs. All text runs before the first decimal character shall be before the tab stop, all text runs after it shall be after the tab stop location.
left (Left Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is left aligned (i.e. all text runs following this tab stop and preceding the next

Enumeration Value	Description
	tab stop shall be left aligned with respect to the tab stop location).
num (List Tab)	Specifies that the current tab is a list tab, which is the tab stop between the numbering and the paragraph contents in a numbered paragraph. [Note: This justification style is used for backwards compatibility with earlier word processors, and should be deprecated in favor of hanging paragraph indentation. <i>end note</i>]
right (Right Tab)	Specifies that the current tab stop shall result in a location in the document where all following text is right aligned (i.e. all text runs following this tab stop and preceding the next tab stop shall be right aligned with respect to the tab stop location).

2.18.91 ST_TabTlc (Custom Tab Stop Leader Character)

This simple type specifies the characters which may be used to fill in the space created by a tab which ends at this custom tab stop. The chosen character shall be repeated as required to completely fill the tab spacing generated by the tab character.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
dot (Dotted leader line)	Specifies that the leader character for this custom tab stop shall be a dot.
heavy (Heavy solid leader line)	Specifies that the leader character for this custom tab stop shall be a heavy solid line, or an underscore. [Note: This setting is used for backwards compatibility with earlier word processors, and should be deprecated in favor of other leader characters. It may be displayed using underscores if desired. <i>end note</i>]
hyphen (Dashed tab stop leader line)	Specifies that the leader character for this custom tab stop shall be a hyphen.
middleDot (Middle dot leader line)	Specifies that the leader character for this custom tab stop shall be a centered dot.
none (No tab stop leader)	Specifies that there shall be no leader character for this

Enumeration Value	Description
	custom tab.
underscore (Solid leader line)	Specifies that the leader character for this custom tab stop shall be an underscore.

2.18.92 ST_TargetScreenSz (Target Screen Sizes for Generated Web Pages)

This simple type specifies possible ideal minimum target screen sizes (width by height, specified in pixels) for which web pages generated may be optimized when saving this document as a web page.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
1024x768 (Optimize for 1024x768)	Specifies that web pages produced from this document should be optimized for a screen size of 1024x768.
1152x882 (Optimize for 1152x882)	Specifies that web pages produced from this document should be optimized for a screen size of 1152x882.
1152x900 (Optimize for 1152x900)	Specifies that web pages produced from this document should be optimized for a screen size of 1152x900.
1280x1024 (Optimize for 1280x1024)	Specifies that web pages produced from this document should be optimized for a screen size of 1280x1024.
1600x1200 (Optimize for 1600x1200)	Specifies that web pages produced from this document should be optimized for a screen size of 1600x1200.
1800x1440 (Optimize for 1800x1440)	Specifies that web pages produced from this document should be optimized for a screen size of 1800x1440.
1920x1200 (Optimize for 1920x1200)	Specifies that web pages produced from this document should be optimized for a screen size of 1920x1200.
544x376 (Optimize for 544x376)	Specifies that web pages produced from this document should be optimized for a screen size of 544x376.
640x480 (Optimize for 640x480)	Specifies that web pages produced from this document should be optimized for a screen size of 640x480.
720x512 (Optimize for 720x512)	Specifies that web pages produced from this document should be optimized for a screen size of 720x512.
800x600 (Optimize for 800x600)	Specifies that web pages produced from this document should be optimized for a screen size of 800x600.

2.18.93 ST_TblLayoutType (Table Layout Type)

This simple type defines the possible types of layout algorithms which may be used to lay out a table within a WordprocessingML document.

These algorithms are defined in the following paragraphs (noting, of course, that implementations are free to implement more efficient versions of each).

Fixed Width Table Layout - This method of table layout uses the preferred widths on the table items to generate the final sizing of the table, but does not change that size regardless of the contents of each table cell, hence the table is fixed width.

[*Guidance*: Although an application may choose to use a different process, this layout could be performed as follows:

- The table grid is used to create the set of shared columns in the table and their initial widths as defined in the **tblGrid** element (§2.4.44)
- The table's total width is defined based on the **tblW** property (§2.4.61) – if it is set to `auto` or `nil`, then the width is not yet determined and will be specified using the row and cell information.
- The first table row is read and the initial number of grid units before the row starts is skipped. The width of the skipped grid columns is set using the **wBefore** property (§2.4.83).
- The first cell is placed on the grid, and the width of the specified grid column span set by **gridSpan** (§2.4.13) is set based on the **tcW** property (§2.4.68).
- Each additional cell is placed on the grid.
- If at any stage, the preferred width requested for the cells exceeds the preferred width of the table, then each grid column is proportionally reduced in size to fit the table width.
- If the grid is exceeded (e.g. **tblGrid** specifies three grid columns, but the second cell has a **gridSpan** of three), the grid is dynamically increased with a default width for the new grid column.
- For each subsequent row, cells are placed on the grid, and each grid column is adjusted to be the maximum value of the requested widths (if the widths do not agree) by adding width to the last cell that ends with that grid column. Again, if at any point, the space requested for the cells exceeds the width of the table, then each grid column is proportionally reduced in size to fit the table width.

end guidance

The resulting table shall be displayed regardless of its contents to the size requested.

AutoFit Table Layout - This method of table layout uses the preferred widths on the table items to generate the final sizing of the table, but then uses the contents of each cell to determine final column widths.

[*Guidance*: This layout may be performed in any manner available to an application, but one algorithm as follows may be used:

- Perform the steps above to lay out the fixed width version of the table.
- Calculate the minimum content width - the width of the cell's contents including all possible line breaking locations (or the cell's width, if the width of the content is smaller), and the maximum content width - the width of the cell's contents (assuming no line breaking not generated by explicit line breaks).
- The minimum and maximum content width of all cells that span a single grid column is the minimum and maximum content width of that column.
- For cells which span multiple grid columns, enlarge all cells which it spans as needed to meet that cell's minimum width.
- If any cell in a grid column has a preferred width, the first such width overrides the maximum width of the column's contents.
- Place the text in the cells in the table, respecting the minimum content width of each cell's content. If a cell's minimum content width exceeds the cell's current width, preferences are overridden as follows:
- First, override the column widths by making all other grid columns proportionally smaller until each it at its minimum width. This cell may then grow to any width between its own minimum and maximum width.
- Next, override the preferred table width until the table reaches the page width.
- Finally, force a line break in each cell's contents as needed

end guidance

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
autofit (AutoFit Table Layout)	Specifies that this table shall use an AutoFit table layout algorithm.
fixed (Fixed Width Table Layout)	Specifies that this table shall use the fixed width table layout algorithm described above.

2.18.94 ST_TblOverlap (Table Overlap Setting)

This simple type contains the possible settings for a floating table which shall be used to determine if the table can overlap with other floating tables when displayed in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
never (Floating Table Cannot Overlap)	Specifies that the parent table, if floating, shall never be displayed in a state where it would be overlapping another floating table in the document. If two floating tables intersect and this option is set on either of them, then one or both tables shall be adjusted as needed to ensure that the table whose value is never is not overlapped when displayed.
overlap (Floating Table Can Overlap)	Specifies that the parent table, if floating, may be displayed in a state where it would be overlapping another floating table in the document.

2.18.95 ST_TblStyleOverrideType (Conditional Table Style Formatting Types)

This simple type specifies possible values for the sections of the table to which the current conditional formatting properties shall be applied when this table style is used.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
band1Horz (Banded Row Conditional Formatting)	Specifies that the table formatting applies to odd numbered groupings of rows.
band1Vert (Banded Column Conditional Formatting)	Specifies that the table formatting applies to odd numbered groupings of columns.
band2Horz (Even Row Stripe Conditional Formatting)	Specifies that the table formatting applies to even numbered groupings of rows.
band2Vert (Even Column Stripe Conditional Formatting)	Specifies that the table formatting applies to even numbered groupings of columns.
firstCol (First Column Conditional Formatting)	Specifies that the table formatting applies to the first column.
firstRow (First Row Conditional Formatting)	Specifies that the table formatting applies to the first row. Any subsequent row which has the tblHeader element present (§2.4.46) shall also use this conditional format.
lastCol (Last table column formatting)	Specifies that the table formatting applies to the last column.
lastRow (Last table row formatting)	Specifies that the table formatting applies to the last row.
neCell (Top right table cell formatting)	Specifies that the table formatting applies to the top right

Enumeration Value	Description
	cell.
nwCell (Top left table cell formatting)	Specifies that the table formatting applies to the top left cell.
seCell (Bottom right table cell formatting)	Specifies that the table formatting applies to the bottom right cell.
swCell (Bottom left table cell formatting)	Specifies that the table formatting applies to the bottom left cell.
wholeTable (Whole table formatting)	Specifies that the conditional formatting applies to the whole table.

2.18.96 ST_TblWidth (Table Width Units)

This simple type specifies the possible values for the units of the width property being defined by a specific table width property. These properties are used to define various properties of a table, including: cell spacing, preferred width, and table margins.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
auto (Automatically Determined Width)	Specifies that the value for the measurement of the current table width property in the parent table shall be automatically determined by the table layout algorithm when the table is displayed (this width can be adjusted as appropriate). If this value is inappropriate for the current measurement (i.e. this measurement is not affected by that algorithm), then this type and the associated value may be ignored.
dx a (Width in Twentieths of a Point)	Specifies that the value for the measurement of the current table width property in the parent table shall be interpreted as twentieths of a point (1/1440 of an inch).
nil (No Width)	Specifies that the current width is zero, regardless of any width value specified on the parent element.
pct (Width in Fiftieths of a Percent)	Specifies that the value for the measurement of the current table width property in the parent table shall be interpreted as fiftieths of a percent. These percentages shall be calculated relative to the extents specified by the parent XML element. If this value is inappropriate for the current measurement (i.e. this measurement is not part of the width of the table), then this type and the associated value may be ignored.

2.18.97 ST_TextAlignment (Vertical Text Alignment Types)

This simple type specifies the type of vertical alignment which shall be used to align the characters on each line in the parent object.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
auto (Automatically Determine Alignment)	Specifies that all text in the parent object shall be aligned automatically when displayed.
baseline (Align Text at Baseline)	Specifies that all text in the parent object shall be aligned to the baseline of each character when displayed.
bottom (Align Text at Bottom)	Specifies that all text in the parent object shall be aligned to the bottom of each character when displayed.
center (Align Text at Center)	Specifies that all text in the parent object shall be aligned to the center of each character when displayed.
top (Align Text at Top)	Specifies that all text in the parent object shall be aligned to the top of each character when displayed.

2.18.98 ST_TextboxTightWrap (Lines To Tight Wrap Within Text Box)

This simple type specifies the lines in the parent paragraph which shall allow the text to be tight wrapped to the paragraph (and not the containing text box) extents when displaying the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
allLines (Tight Wrap All Lines)	Specifies that all lines in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
firstAndLastLine (Tight Wrap First and Last Lines)	Specifies that only the first and last lines in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
firstLineOnly (Tight Wrap First Line)	Specifies that only the first line in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
lastLineOnly (Tight Wrap Last Line)	Specifies that only the last line in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.
none (Do Not Tight Wrap)	Specifies that no lines in the paragraph shall allow surrounding text to be tight wrapped to their extents and not the containing text box's extents.

2.18.99 ST_TextDirection (Text Flow Direction)

This simple type specifies the direction of the text flow for the parent object.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
-------------------	-------------

Enumeration Value	Description
btLr (Bottom to Top, Left to Right)	<p>Specifies that text in the parent object shall flow from bottom to top vertically, then from left to right horizontally on the page.</p> <p>This means that vertical lines are filled before the text expands horizontally.</p>
lRtB (Left to Right, Top to Bottom)	<p>Specifies that text in the parent object shall flow from left to right horizontally, then top to bottom vertically on the page.</p> <p>This means that horizontal lines are filled before the text expands vertically.</p>
lRtBV (Left to Right, Top to Bottom Rotated)	<p>Specifies that text in the parent object shall flow from left to right horizontally, then top to bottom vertically on the page.</p> <p>This means that horizontal lines are filled before the text expands vertically.</p> <p>This flow is also rotated such that any East Asian text shall be rotated 270 degrees when displayed on a page.</p>
tbLrV (Top to Bottom, Left to Right Rotated)	<p>Specifies that text in the parent object shall flow from top to bottom vertically, then left to right horizontally on the page.</p> <p>This means that vertical lines are filled before the text expands horizontally.</p> <p>This flow is also rotated such that all text is rotated 90 degrees when displayed on a page.</p>
tbRI (Top to Bottom, Right to Left)	<p>Specifies that text in the parent object shall flow from right to left horizontally, then top to bottom vertically on the page.</p> <p>This means that horizontal lines are filled before the text expands vertically.</p>
tbRIV (Top to Bottom, Right to Left Rotated)	<p>Specifies that text in the parent object shall flow from top to bottom vertically, then right to left horizontally on the page.</p> <p>This means that vertical lines are filled before the text expands horizontally.</p> <p>This flow is also rotated such that all text is rotated 90 degrees when displayed on a page.</p>

2.18.100 ST_TextEffect (Animated Text Effects)

This simple type specifies the possible types of animated text effect which may be applied to a text run when it is displayed..

This simple type's contents are a restriction of the XML Schema string datatype.
The following are possible enumeration values for this type:

Enumeration Value	Description
antsBlack (Black Dashed Line Animation)	Specifies that this text shall be surrounded by an animated black dashed line border.
antsRed (Marching Red Ants)	Specifies that this text shall be surrounded by an animated red dashed line border.
blinkBackground (Blinking Background Animation)	Specifies that this text shall be surrounded by a background color which alternates between black and white.
lights (Colored Lights Animation)	Specifies that this text shall be surrounded by a border consisting of a series of colored lights, which constantly change colors in sequence.
none (No Animation)	Specifies that this text shall have no animated text effect.
shimmer (Shimmer Animation)	Specifies that this text shall be animated by alternating between normal and blurry states.
sparkle (Sparkling Lights Animation)	Specifies that this text shall have a background consisting of a random pattern of colored lights, which constantly change colors in sequence.

2.18.101 ST_TextScale (Text Expansion/Compression Percentage)

This simple type specifies that the percentage by which the contents of a run shall be expanded or compressed with respect to its normal (100%) character width, with a minimum width of 1% and maximum width of 600%.

This simple type's contents are a restriction of the XML Schema integer datatype.

This simple type also specifies the following restrictions:

- This simple type has a minimum value of greater than or equal to 0.
- This simple type has a maximum value of less than or equal to 600.

2.18.102 ST_Theme (Theme Font)

This simple type specifies a theme font type which may be referenced as a theme font within the parent run properties.

This theme font is a reference to one of the predefined theme fonts, located in the document's Theme part, which allows for font information to be set centrally in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
majorAscii (Major ASCII Theme Font)	Specifies that the current font is a reference to the major theme font for the ASCII range.
majorBidi (Major Complex Script Theme Font)	Specifies that the current font is a reference to the major theme font for the Complex Script range.
majorEastAsia (Major East Asian Theme Font)	Specifies that the current font is a reference to the major theme font for the East Asian range.
majorHAnsi (Major High ANSI Theme Font)	Specifies that the current font is a reference to the major theme font for the High ANSI range.
minorAscii (Minor ASCII Theme Font)	Specifies that the current font is a reference to the minor theme font for the ASCII range.
minorBidi (Minor Complex Script Theme Font)	Specifies that the current font is a reference to the minor

Enumeration Value	Description
	theme font for the Complex Script range.
minorEastAsia (Minor East Asian Theme Font)	Specifies that the current font is a reference to the minor theme font for the East Asian range.
minorHAnsi (Minor High ANSI Theme Font)	Specifies that the current font is a reference to the minor theme font for the High ANSI range.

2.18.103 ST_ThemeColor (Theme Color)

This simple type specifies a theme color to be applied to the current object. The specified theme color is a reference to one of the predefined theme colors, located in the document's Theme part, which allows color information to be set centrally in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
accent1 (Accent 1 Theme Color)	Specifies that the color to be used shall be the accent1 theme color.
accent2 (Accent 2 Theme Color)	Specifies that the color to be used shall be the accent2 theme color.
accent3 (Accent 3 Theme Color)	Specifies that the color to be used shall be the accent3 theme color.
accent4 (Accent 4 Theme Color)	Specifies that the color to be used shall be the accent4 theme color.
accent5 (Accent 5 Theme Color)	Specifies that the color to be used shall be the accent5 theme color.
accent6 (Accent 6 Theme Color)	Specifies that the color to be used shall be the accent6 theme color.
background1 (Background 1 Theme Color)	Specifies that the color to be used shall be the background1 theme color.
background2 (Background 2 Theme Color)	Specifies that the color to be used shall be the background2 theme color.
dark1 (Dark 1 Theme Color)	Specifies that the color to be used shall be the dark1 theme color.
dark2 (Dark 2 Theme Color)	Specifies that the color to be used shall be the dark2 theme color.
followedHyperlink (Followed Hyperlink Theme Color)	Specifies that the color to be used shall be the followedHyperlink theme color.
hyperlink (Hyperlink Theme Color)	Specifies that the color to be used shall be the hyperlink theme color.
light1 (Light 1 Theme Color)	Specifies that the color to be used shall be the light1 theme color.
light2 (Light 2 Theme Color)	Specifies that the color to be used shall be the light1 theme color.

Enumeration Value	Description
none (No Theme Color)	Specifies that no theme color shall be applied to the current object.
text1 (Text 1 Theme Color)	Specifies that the color to be used shall be the text1 theme color.
text2 (Text 2 Theme Color)	Specifies that the color to be used shall be the text2 theme color.

2.18.104 ST_TwipsMeasure (Measurement in Twentieths of a Point)

This simple type specifies that its contents will contain a positive whole number, whose contents consist of a measurement in twentieths of a point (equivalent to 1/1440th of an inch).

The contents of this measurement are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the ST_UnsignedDecimalNumber simple type (§2.18.107).

2.18.105 ST_UcharHexNumber (Two Digit Hexadecimal Number Value)

This simple type specifies a number value specified as a two digit (one octet) hexadecimal number), whose contents are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema hexBinary datatype.

This simple type also specifies the following restrictions:

- This simple type's contents must have a length of exactly 1 characters.

2.18.106 ST_Underline (Underline Patterns)

This simple type specifies the types of patterns which may be used to create the underline applied beneath the text in a run.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
dash (Dashed Underline)	Specifies an underline consisting of a dashed line beneath all characters in this run.
dashDotDotHeavy (Thick Dash-Dot-Dot Underline)	Specifies an underline consisting of a series of thick dash, dot, dot characters beneath all characters in this run.
dashDotHeavy (Thick Dash-Dot Underline)	Specifies an underline consisting of a series of thick dash, dot characters beneath all characters in this run.
dashedHeavy (Thick Dashed Underline)	Specifies an underline consisting of a series of thick dashes beneath all characters in this run.
dashLong (Long Dashed Underline)	Specifies an underline consisting of long dashed characters

Enumeration Value	Description
	beneath all characters in this run.
dashLongHeavy (Thick Long Dashed Underline)	Specifies an underline consisting of thick long dashed characters beneath all characters in this run.
dotDash (Dash-Dot Underline)	Specifies an underline consisting of a series of dash, dot characters beneath all characters in this run.
dotDotDash (Dash-Dot-Dot Underline)	Specifies an underline consisting of a series of dash, dot, dot characters beneath all characters in this run.
dotted (Dotted Underline)	Specifies an underline consisting of a series of dot characters beneath all characters in this run.
dottedHeavy (Thick Dotted Underline)	Specifies an underline consisting of a series of thick dot characters beneath all characters in this run.
double (Double Underline)	Specifies an underline consisting of two lines beneath all characters in this run.
none (No Underline)	Specifies no underline beneath this run.
single (Single Underline)	Specifies an underline consisting of a single line beneath all characters in this run.
thick (Thick Underline)	Specifies an underline consisting of a single thick line beneath all characters in this run.
wave (Wave Underline)	Specifies an underline consisting of a single wavy line beneath all characters in this run.
wavyDouble (Double Wave Underline)	Specifies an underline consisting of a pair of wavy lines beneath all characters in this run.

Enumeration Value	Description
wavyHeavy (Heavy Wave Underline)	Specifies an underline consisting of a single thick wavy line beneath all characters in this run.
words (Underline Non-Space Characters Only)	Specifies an underline consisting of a single line beneath all non-space characters in the run. There shall be no underline beneath any space character (breaking or non-breaking).

2.18.107 ST_UnsignedDecimalNumber (Unsigned Decimal Number Value)

This simple type specifies that its contents will contain a positive whole decimal number, whose contents are interpreted based on the context of the parent XML element.

This simple type's contents are a restriction of the XML Schema unsignedLong datatype.

2.18.108 ST_VAnchor (Vertical Anchor Location)

This simple type specifies the vertical position to which the parent object has been anchored in the document. This anchor position shall be used as the base location to determine the final vertical position of the object in the document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
margin (Relative To Margin)	Specifies that the parent object shall be vertically anchored to the text margins. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the text margin.
page (Relative To Page)	Specifies that the parent object shall be vertically anchored to the page edge. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the edge of the page.
text (Relative To Vertical Text Extents)	Specifies that the parent object shall be vertically anchored to the text extents. This shall be used to specify that any vertical positioning values shall be calculated with respect to the location of the top edge of the text in the anchor paragraph.

2.18.109 ST_VerticalAlignRun (Vertical Positioning Location)

This simple type specifies possible values for the alignment of the contents of this run in relation to the default appearance of the run's text. This allows the text to be repositioned as subscript or superscript without altering the font size of the run properties.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
baseline (Regular Vertical Positioning)	Specifies that the text in the parent run shall be located at the baseline and presented in the same size as surrounding text.
subscript (Subscript)	Specifies that this text should be subscript. This setting shall lower the text in this run below the baseline and change it to a smaller size, if a smaller size is available.
superscript (Superscript)	Specifies that this text should be superscript. This setting shall raise the text in this run above the baseline and change it to a smaller size, if a smaller size is available.

2.18.110 ST_VerticalJc (Vertical Alignment Type)

This simple type specifies the vertical alignment for text between the top and bottom margins of the parent container (page or table cell).

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
both (Vertical Justification)	Specifies that the text shall be vertically justified between the top and bottom margins of the parent object, by adding additional line spacing to each paragraph as required. This setting is only applied for the content of the section which is displayed on full pages. If the content does not use the full page (e.g. another section begins on the same page, or the document ends mid-page), then the value shall be ignored when rendering that page (returning to the default value of top) This value is only valid for page justification settings, and shall be ignored when specified on a table cell (returning to the default value of top).
bottom (Align Bottom)	Specifies that the text shall be vertically aligned to the bottom margin of the parent object, by moving all text to the bottom text extent within the parent object as required.
center (Align Center)	Specifies that the text shall be vertically aligned to the center of the parent object..
top (Align Top)	Specifies that the text shall be vertically aligned to the top margin of the parent object, by moving all text to the top text extent within the parent object as required.

2.18.111 ST_View (Document View Values)

This simple type defines the possible views which may be used to determine how WordprocessingML documents may be rendered when displayed by an application.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
masterPages (Master Document View)	Specifies that a given WordprocessingML document shall be rendered in a view optimized for outlining or creating long documents. [Note: This setting may be interpreted as functionally equivalent to the <code>outline</code> setting, as it only remains separate to support legacy applications. <i>end note</i>]
none (Default View)	Specifies that a given WordprocessingML document shall be rendered in the default view of the application.
normal (Draft View)	Specifies that a given WordprocessingML document shall be rendered in a view optimized for outlining or creating long documents.
outline (Outline View)	Specifies that a given WordprocessingML document shall be rendered in a view optimized for outlining or creating long documents.
print (Print Layout View)	Specifies that this document shall be opened in a view that displays the document as it will print.
web (Web Page View)	Specifies that a given WordprocessingML document shall be rendered in a view mimicking the way this document would be displayed in a web page.

2.18.112 ST_Wrap (Text Wrapping around Text Frame Type)

This simple type specifies the type of text wrapping which shall be allowed around a text frame within a document.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
around (Allow Text Wrapping Around Frame)	Specifies that text shall be allowed to wrap around the remaining space on each line around this text frame in the document.
auto (Default Text Wrapping Around Frame)	Specifies that text shall have the default application-defined behavior of the application displaying the WordprocessingML document with regard to the text wrapping displayed around the frame.
none (No Text Wrapping Around Frame)	Specifies that text shall not be allowed to wrap around the remaining space on each lines around this text frame. Any text content shall therefore be placed on the next line following this text frame which does not intersect with the

Enumeration Value	Description
	frame's extents.
notBeside (No Text Wrapping Beside Frame)	Specifies that text shall not be allowed to wrap around the remaining space on each line around this text frame. Any text content shall therefore be placed on the next line following this text frame which does not intersect with the frame's extents.
through (Through Text Wrapping Around Frame)	Specifies that text shall be allowed to wrap around the remaining space on each line around this text frame in the document.
tight (Tight Text Wrapping Around Frame)	Specifies that text shall be allowed to tightly wrap around the remaining space on each line around this text frame in the document.

2.18.113 ST_XAlign (Horizontal Alignment Location)

This simple type specifies the set of possible relative horizontal positions for the parent floating object. This relative position is specified relative to the horizontal anchor specified by the parent object.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
center (Centered Horizontally)	Specifies that the parent object shall be centered with respect to the anchor settings.
inside (Inside)	Specifies that the parent object shall be inside of the anchor object.
left (Left Aligned Horizontally)	Specifies that the parent object shall be left aligned with respect to the anchor settings.
outside (Outside)	Specifies that the parent object shall be outside of the anchor object.
right (Right Aligned Horizontally)	Specifies that the parent object shall be right aligned with respect to the anchor settings.

2.18.114 ST_YAlign (Vertical Alignment Location)

This simple type specifies the set of possible relative vertical positions for the parent floating object. This relative position is specified relative to the vertical anchor specified by the parent object.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
bottom (Bottom)	Specifies that the parent object shall be vertically aligned to the bottom edge of the anchor object .
center (Centered Vertically)	Specifies that the parent object shall be vertically centered with respect to the anchor object.
inline (In line With Text)	Specifies that the parent object shall be vertically aligned in line with the surrounding text (i.e. shall not allow any text wrapping around it when positioned in the document.
inside (Inside Anchor Extents)	Specifies that the parent object shall be vertically aligned to the edge of the anchor object, and positioned inside that object.
outside (Outside Anchor Extents)	Specifies that the parent object shall be vertically aligned to the edge of the anchor object, and positioned outside that object.
top (Top)	Specifies that the parent object shall be vertically aligned to the top edge of the anchor object .

2.18.115 ST_Zoom (Magnification Preset Values)

This simple type specifies the type of magnification settings which may be applied to a given document on open.

This simple type's contents are a restriction of the XML Schema string datatype.

The following are possible enumeration values for this type:

Enumeration Value	Description
bestFit (Display Page Width)	Specifies that the magnification setting shall be adjusted to ensure the width of the current page matches the available window width.
fullPage (Display One Full Page)	Specifies that the magnification setting shall be adjusted to ensure that one full page can be seen at a time.
none (No Preset Magnification)	Specifies that no preset magnification is present, and the last known cached setting shall be used.
textFit (Display Text Width)	Specifies that the magnification setting shall be adjusted to ensure the width of the text extents on the current page matches the available window width.