

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_OnOff">
  <attribute name="val" type="ST_OnOff"/>
</complexType>
```

### 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.

[Example: Consider the following WordprocessingML run which has font kerning properties specified:

```
<w:r>
  <w:rPr>
    <w:sz w:val="22" />
    <w:kern w:val="28" />
  </w:rPr>
</w:r>
```

Even though font kerning is turned on via the kern element, the contents of this run shall not be kerned because that settings only applied to font sizes of 14 points (28 half-points) or larger. If the kern element's val attribute was less than or equal to the sz element's val attribute, then kerning would be applied:

```
<w:r>
  <w:rPr>
    <w:sz w:val="22" />
    <w:kern w:val="22" />
  </w:rPr>
</w:r>
```

*end example]*

#### Parent Elements

rPr (§2.7.8.1); rPr (§2.3.1.29); rPr (§2.5.2.26); rPr (§2.3.2.25); rPr (§2.3.2.26); rPr (§2.7.4.4); rPr (§2.3.1.30); rPr (§2.9.26); rPr (§2.5.2.27); rPr (§2.7.5.2)

#### Attributes

#### Description

Attributes	Description
val (Half Point Measurement)	<p>Specifies a positive measurement specified in half-points (1/144 of an inch).</p> <p>The contents of this attribute value are interpreted based on the context of the parent XML element.</p> <p>[<i>Example:</i> Consider the following WordprocessingML fragment:</p> <pre data-bbox="451 499 792 594">&lt;w:rPr&gt;   &lt;w:sz w:val="28" /&gt; &lt;/w:rPr&gt;</pre> <p>The value of the val attribute is the font size of the run's contents.</p> <p>However, consider the following fragment:</p> <pre data-bbox="451 783 824 877">&lt;w:rPr&gt;   &lt;w:kern w:val="30" /&gt; &lt;/w:rPr&gt;</pre> <p>In this case, the value in the val attribute is the minimum size for which font characters shall be automatically kerned.</p> <p>In each case, the value is interpreted in the context of the parent element. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_HpsMeasure simple type (§2.18.48).</p>

1 The following XML Schema fragment defines the contents of this element:

```
2 <complexType name="CT_HpsMeasure">
3   <attribute name="val" type="ST_HpsMeasure" use="required"/>
4 </complexType>
```

### 5 2.3.2.18 lang (Languages for Run Content)

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

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

11 [*Example:* Consider a run which contains both Latin and complex script characters in its contents. If those  
12 contents should be interpreted as French (Canada) and Hebrew, respectively, that requirement would be  
13 specified as follows in the resulting WordprocessingML:

```
14 <w:r>
15   <w:rPr>
```

```

1      <w:lang w:val="fr-CA" w:bidirectional="he-IL" />
2      </w:rPr>
3  </w:r>

```

4 The resulting run specifies that any complex script contents shall be spell and grammar checked as if they were  
5 Hebrew, and any Latin character contents shall be spell and grammar checked as if they were French (Canada).  
6 *end example]*

Parent Elements
rPr (§2.7.8.1); rPr (§2.3.1.29); rPr (§2.5.2.26); rPr (§2.3.2.25); rPr (§2.3.2.26); rPr (§2.7.4.4); rPr (§2.3.1.30); rPr (§2.9.26); rPr (§2.5.2.27); rPr (§2.7.5.2)

7

Attributes	Description
bidi (Complex Script Language)	<p>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.</p> <p>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.</p> <p>[<i>Example:</i> Consider a run which contains complex script characters in its contents. If those contents should be interpreted as Hebrew, that requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>           &lt;w:r&gt;             &lt;w:rPr&gt;               &lt;w:lang w:bidirectional="he-IL" /&gt;             &lt;/w:rPr&gt;           &lt;/w:r&gt; </pre> <p>The resulting run specifies that any complex script contents shall be spell and grammar checked using a Hebrew dictionary and grammar engine, if one is available. <i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_Lang simple type (§2.18.51).</p>
eastAsia (East Asian Language)	<p>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.</p> <p>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.</p> <p>[<i>Example:</i> Consider a run which contains East Asian characters in its contents. If those contents should be interpreted as Korean, that requirement would be specified as</p>

Attributes	Description
	<p>follows in the resulting WordprocessingML:</p> <pre>&lt;w:r&gt;   &lt;w:rPr&gt;     &lt;w:lang w:bidl="ko-KR" /&gt;   &lt;/w:rPr&gt; &lt;/w:r&gt;</pre> <p>The resulting run specifies that any complex script contents shall be spell and grammar checked using a Korean dictionary and grammar engine, if one is available. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_Lang simple type (§2.18.51).</p>
val (Latin Language)	<p>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.</p> <p>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.</p> <p>[<i>Example</i>: Consider a run which contains Latin characters in its contents. If those contents should be interpreted as English (Canada), that requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:r&gt;   &lt;w:rPr&gt;     &lt;w:lang w:bidl="en-CA" /&gt;   &lt;/w:rPr&gt; &lt;/w:r&gt;</pre> <p>The resulting run specifies that any complex script contents shall be spell and grammar checked using a English (Canada) dictionary and grammar engine, if one is available. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_Lang simple type (§2.18.51).</p>

1 The following XML Schema fragment defines the contents of this element:

```
2 <complexType name="CT_Language">
3   <attribute name="val" type="ST_Lang" use="optional"/>
4   <attribute name="eastAsia" type="ST_Lang" use="optional"/>
5   <attribute name="bidl" type="ST_Lang" use="optional"/>
6 </complexType>
```

### 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.

[*Example:* Consider a basic WordprocessingML paragraph with a pair of runs. This run would be expressed as follows:

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>Text</w:t>
      </w:r>
      <w:fldSimple w:instr="AUTHOR">
        <w:r>
          <w:t>Author Name</w:t>
        </w:r>
      </w:fldSimple>
    </w:p>
  </w:body>
</w:document>
```

The r element is the container for all of the content in the run, which in this example includes both a run in the paragraph and a run within a simple field. *end example*]

#### Parent Elements

customXml (§2.5.1.5); del (§2.13.5.12); fldSimple (§2.16.21); hyperlink (§2.16.24); ins (§2.13.5.20); moveFrom (§2.13.5.21); moveTo (§2.13.5.26); p (§2.3.1.22); rt (§2.3.3.23); rubyBase (§2.3.3.26); sdtContent (§2.5.2.35); smartTag (§2.5.1.9)

#### Child Elements

#### Subclause

annotationRef (Comment Information Block)	§2.13.4.1
br (Break)	§2.3.3.1
commentReference (Comment Content Reference Mark)	§2.13.4.5
continuationSeparator (Continuation Separator Mark)	§2.11.1
cr (Carriage Return)	§2.3.3.4
dayLong (Date Block - Long Day Format)	§2.3.3.5
dayShort (Date Block - Short Day Format)	§2.3.3.6
delInstrText (Deleted Field Code)	§2.16.13

Child Elements	Subclause
delText (Deleted Text)	§2.3.3.7
drawing (DrawingML Object)	§2.3.3.9
endnoteRef (Endnote Reference Mark)	§2.11.6
endnoteReference (Endnote Reference)	§2.11.7
fldChar (Complex Field Character)	§2.16.18
footnoteRef (Footnote Reference Mark)	§2.11.13
footnoteReference (Footnote Reference)	§2.11.14
instrText (Field Code)	§2.16.25
lastRenderedPageBreak (Position of Last Calculated Page Break)	§2.3.3.13
monthLong (Date Block - Long Month Format)	§2.3.3.15
monthShort (Date Block - Short Month Format)	§2.3.3.16
noBreakHyphen (Non Breaking Hyphen Character)	§2.3.3.18
object (Inline Embedded Object)	§2.3.3.19
pgNum (Page Number Block)	§2.3.3.20
pict (VML Object)	§2.3.3.21
ptab (Absolute Position Tab Character)	§2.3.3.22
rPr (Run Properties)	§2.3.2.25
ruby (Phonetic Guide)	§2.3.3.24
separator (Footnote/Endnote Separator Mark)	§2.11.23
softHyphen (Optional Hyphen Character)	§2.3.3.28
sym (Symbol Character)	§2.3.3.29
t (Text)	§2.3.3.30
tab (Tab Character)	§2.3.3.31
yearLong (Date Block - Long Year Format)	§2.3.3.32
yearShort (Date Block - Short Year Format)	§2.3.3.33

1

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</p>

Attributes	Description
	<p>in this document.</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.57).</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 (§2.18.57).</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 (§2.18.57).</p>

1 The following XML Schema fragment defines the contents of this element:

```

2 <complexType name="CT_R">
3   <sequence>
4     <group ref="EG_RPr" minOccurs="0"/>
5     <group ref="EG_RunInnerContent" minOccurs="0" maxOccurs="unbounded"/>
6   </sequence>
7   <attribute name="rsidRPr" type="ST_LongHexNumber"/>
8   <attribute name="rsidDel" type="ST_LongHexNumber"/>
9   <attribute name="rsidR" type="ST_LongHexNumber"/>
10 </complexType>

```

#### 11 2.3.2.24 rFonts (Run Fonts)

12 This element specifies the fonts which shall be used to display the text contents of this run. Within a single run,  
13 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.

[*Example:* Consider a single text run with both Arabic and English text, as follows:

English العربية

This content may be expressed in a single WordprocessingML run:

```
<w:r>
  <w:t>English العربية</w:t>
</w:r>
```

Although it is in the same run, the contents are in different font faces by specifying a different font for ASCII and CS characters in the run:

```
<w:r>
  <w:rPr>
    <w:rFonts w:ascii="Courier New" w:cs="Times New Roman" />
  </w:rPr>
  <w:t>English العربية</w:t>
</w:r>
```

This text run shall therefore use the Courier New font for all characters in the ASCII range, and shall use the Times New Roman font for all characters in the Complex Script range. *end example*]

#### Parent Elements

rPr (§2.7.8.1); rPr (§2.3.1.29); rPr (§2.5.2.26); rPr (§2.3.2.25); rPr (§2.3.2.26); rPr (§2.7.4.4); rPr (§2.3.1.30); rPr (§2.9.26); rPr (§2.5.2.27); rPr (§2.7.5.2)

#### Attributes

#### Description

ascii (ASCII Font)

Specifies a font which shall be used to format all characters in the ASCII range (0 - 127) within the parent run.

If the `asciiTheme` attribute is also specified, then this attribute shall be ignored and that



Attributes	Description
	<p>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>[<i>Example</i>: Consider a run of ASCII text which shall be displayed using the Courier New font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:rPr&gt;   &lt;w:rFonts w:ascii="Courier New" /&gt; &lt;/w:rPr&gt;</pre> <p>The ascii attribute specifies that the run shall use the Courier New font for all text in the ASCII range. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.89).</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>[<i>Example</i>: Consider a run of ASCII text which shall be displayed using the majorASCII theme font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:rPr&gt;   &lt;w:rFonts w:asciiTheme="majorAscii" /&gt; &lt;/w:rPr&gt;</pre> <p>The ascii attribute specifies that the run shall use the majorAscii theme font as defined in the document's themes part for all text in the ASCII range. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_Theme simple type (§2.18.103).</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>

Attributes	Description
	<p>If the <code>csTheme</code> 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>[<i>Example</i>: Consider a run of Arabic text which shall be displayed using the Arial Unicode MS font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:rPr&gt;   &lt;w:rFonts w:cs="Arial Unicode MS" /&gt; &lt;/w:rPr&gt;</pre> <p>The <code>cs</code> attribute specifies that the run shall use the Arial Unicode MS font for all text in a complex script range. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the <code>ST_String</code> simple type (§2.18.89).</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 <code>cs</code> 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 <code>cs</code> attribute.</p> <p>[<i>Example</i>: Consider a run of Arabic text which shall be displayed using the <code>majorBidi</code> theme font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:rPr&gt;   &lt;w:rFonts w:cstheme="majorBidi" /&gt; &lt;/w:rPr&gt;</pre> <p>The <code>cstheme</code> attribute specifies that the run shall use the <code>majorBidi</code> theme font as defined in the document's themes part for all text in a complex script range. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the <code>ST_Theme</code> simple type (§2.18.103).</p>

Attributes	Description
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>[<i>Example</i>: Consider a run of Japanese text which shall be displayed using the MS Mincho font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:rPr&gt;   &lt;w:rFonts w:eastAsia="MS Mincho" /&gt; &lt;/w:rPr&gt;</pre> <p>The eastAsia attribute specifies that the run shall use the MS Mincho font for all text in an East Asian range. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.89).</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>[<i>Example</i>: Consider a run of Japanese text which shall be displayed using the minorEastAsia theme font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre>&lt;w:rPr&gt;   &lt;w:rFonts w:eastAsiaTheme="minorEastAsia" /&gt; &lt;/w:rPr&gt;</pre> <p>The eastAsiaTheme attribute specifies that the run shall use the minorEastAsia theme font as defined in the document's themes part for all text in an East Asian range. <i>end example</i>]</p>

Attributes	Description
	The possible values for this attribute are defined by the ST_Theme simple type (§2.18.103).
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>[Example: Consider a run of text which falls into a high ANSI range, and shall be displayed using the Bauhaus 93 font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre data-bbox="451 905 1015 999">&lt;w:rPr&gt;   &lt;w:rFonts w:hAnsi="Bauhaus 93" /&gt; &lt;/w:rPr&gt;</pre> <p>The hAnsi attribute specifies that the run shall use the Bauhaus 93 font for all text in a high ANSI range. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.89).</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> <p>[Example: Consider a run of text which falls into a high ANSI range, and shall be displayed using the minorHAnsi theme font. This requirement would be specified as follows in the resulting WordprocessingML:</p> <pre data-bbox="451 1801 1096 1898">&lt;w:rPr&gt;   &lt;w:rFonts w:hAnsiTheme="minorHAnsi" /&gt; &lt;/w:rPr&gt;</pre>

Attributes	Description
	<p>The hAnsiTheme attribute specifies that the run shall use the minorHAnsi theme font as defined in the document's themes part for all text in a high ANSI range. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_Theme simple type (§2.18.103).</p>
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. [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. <i>end note</i>]</p> <p>If this attribute is omitted, then this ambiguity may be resolved by any means available.</p> <p>[Example: Consider the run representing the paragraph mark glyph, which is not stored as a physical character. Since this could therefore be formatted with any of the fonts specified for the run, this ambiguity is resolved using the following WordprocessingML:</p> <pre data-bbox="451 1045 1003 1213">&lt;w:pPr&gt;   &lt;w:rPr&gt;     &lt;w:rFonts w:hint="eastAsia" /&gt;   &lt;/w:rPr&gt; &lt;/w:pPr&gt;</pre> <p>The hint attribute specifies that the run shall use the eastAsia font (theme or not, whichever is in use for East Asian text) as defined for this range. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_Hint simple type (§2.18.47).</p>

1 The following XML Schema fragment defines the contents of this element:

```

2 <complexType name="CT_Fonts">
3   <attribute name="hint" type="ST_Hint"/>
4   <attribute name="ascii" type="ST_String"/>
5   <attribute name="hAnsi" type="ST_String"/>
6   <attribute name="eastAsia" type="ST_String"/>
7   <attribute name="cs" type="ST_String"/>
8   <attribute name="asciiTheme" type="ST_Theme"/>
9   <attribute name="hAnsiTheme" type="ST_Theme"/>
10  <attribute name="eastAsiaTheme" type="ST_Theme"/>
11  <attribute name="cstheme" type="ST_Theme"/>
12 </complexType>
```

Child Elements	Subclause
rPrChange (Revision Information for Run Properties)	§2.13.5.32
rStyle (Referenced Character Style)	§2.3.2.27
rtl (Right To Left Text)	§2.3.2.28
shadow (Shadow)	§2.3.2.29
shd (Run Shading)	§2.3.2.30
smallCaps (Small Caps)	§2.3.2.31
snapToGrid (Use Document Grid Settings For Inter-Character Spacing)	§2.3.2.32
spacing (Character Spacing Adjustment)	§2.3.2.33
specVanish (Paragraph Mark Is Always Hidden)	§2.3.2.34
strike (Single Strikethrough)	§2.3.2.35
sz (Font Size)	§2.3.2.36
szCs (Complex Script Font Size)	§2.3.2.37
u (Underline)	§2.3.2.38
vanish (Hidden Text)	§2.3.2.39
vertAlign (Subscript/Superscript Text)	§2.3.2.40
w (Expanded/Compressed Text)	§2.3.2.41
webHidden (Web Hidden Text)	§2.3.2.42

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_RPr">
  <sequence>
    <group ref="EG_RPrContent" minOccurs="0"/>
  </sequence>
</complexType>
```

## 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.

[Example: Consider the following information stored in a document's font table part:

```

1    <w:fonts>
2      <w:font w:name="Times New Roman">
3        <w:panose1 w:val="02020603050405020304" />
4        <w:charset w:val="00" />
5        <w:family w:val="roman" />
6        <w:pitch w:val="variable" />
7        <w:sig w:usb0="20002A87" w:usb1="80000000" w:usb2="00000008"
8    w:usb3="00000000" w:csb0="000001FF" w:csb1="00000000" />
9        <w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-
10    89EF99B2457E}" />
11      </w:font>
12    </w:fonts>

```

The font table contains information about the Times New Roman font; specifically, information used to locate a substitute font when it is not available and a relationship to the embedded form of the regular form of the font. *end example*]

## 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

[*Example:* Consider a font to be embedded whose first 32 bytes are as follows:

00	01	00	00	00	12	01	00	00	04	00	20	44	53	49	47
A3	0E	BF	F3	00	01	36	8C	00	00	14	DC	4C	54	53	48

To obfuscate this font for storage:

- Generate a GUID (e.g. 001B70DC-AA60-4AD5-90EC-18A0948E1EAE)
- Reverse its order (e.g. AE1E8E94-A018-EC90-D54A-60AADC701B00)
- XOR the GUID with the first and second 16 bytes

The resulting 32 bytes would be:

AE	1F	8E	94	A0	0A	ED	90	D5	4E	60	8A	98	23	52	47
0D	10	31	67	A0	19	DA	1C	D5	4A	74	76	90	24	48	48

*end example]*

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 term to attempt to locate the font, and use the first font for which is locates a match.

If this element is omitted, then no alternate names are present for the parent font.

[*Example:* Consider the following information stored for a single font:

```
<w:font w:name="SimSun">
  <w:altName w:val="Arial Unicode MS" />
  ...
</w:font>
```

The altName element specifies that when no font with a name of SimSun (the primary font name) can be located, that applications should attempt to locate a font with the name Arial Unicode MS before doing substitution based on the font metrics. *end example]*

Parent Elements
font (§2.8.2.10)

Attributes	Description
val (String Value)	<p>Specifies that its contents will contain a string.</p> <p>The contents of this string are interpreted based on the context of the parent XML element.</p> <p>[<i>Example:</i> Consider the following WordprocessingML fragment:</p>



Attributes	Description
	<pre>&lt;w:pPr&gt;   &lt;w:pStyle w:val="heading1" /&gt; &lt;/w:pPr&gt;</pre> <p>The value of the val attribute is the ID of the associated paragraph style's styleId.</p> <p>However, consider the following fragment:</p> <pre>&lt;w:sdtPr&gt;   &lt;w:alias w:val="SDT Title Example" /&gt;   ... &lt;/w:sdtPr&gt;</pre> <p>In this case, the decimal number in the val attribute is the caption of the parent structured document tag. In each case, the value is interpreted in the context of the parent element. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.89).</p>

1 The following XML Schema fragment defines the contents of this element:

```
2 <complexType name="CT_String">
3   <attribute name="val" type="ST_String" use="required"/>
4 </complexType>
```

### 5 2.8.2.2 charset (Character Set Supported By Font)

6 This element specifies the character set which is supported by the parent font. This information may be used as  
7 defined in font substitution logic to locate an appropriate substitute font when this font is not available. This  
8 information is determined by querying the font when present and shall not be modified when the font is not  
9 available.

10 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.

Value	Description
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.

[*Example:* Consider the following information stored for a single font:

```
<w:font w:name="SimSun">
  <w:charset w:val="86" />
  ...
</w:font>
```

The charset element specifies via its val attribute value of 86 that this font uses the GB-2312 character set. *end example]*

Parent Elements
font (§2.8.2.10)

Attributes	Description
val (Value)	<p>Specifies a value specified as single octet (two digit) hexadecimal number whose contents are interpreted based on the context of the parent XML element.</p> <p>[<i>Example:</i> Consider the following value for an attribute of type ST_UCharHexNumber:</p> <pre>&lt;w:... w:val="BE"/&gt;</pre> <p>This value is valid, as it contains two hexadecimal digits, an encoding of an octet of the actual decimal number value. <i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_UCharHexNumber simple</p>

Attributes	Description
	type (§2.18.106).

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_UcharHexNumber">
  <attribute name="val" type="ST_UcharHexNumber" use="required"/>
</complexType>
```

### 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.

[*Example:* Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```
<w:font w:name="Arial">
  ...
  <w:embedBold r:id="rId10" />
</w:font>
```

The embedBold element specifies that the embedded font targeted with the relationship with ID rId10 may be used to retrieve the bold form of the embedded Arial font. *end example*]

Parent Elements
font (§2.8.2.10)

Attributes	Description
fontKey (Embedded Font Obfuscation Key)	<p>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.</p> <p>If this attribute is omitted, then no key is provided for this font.</p> <p>[<i>Example:</i> Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt;   ...   &lt;w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" /&gt;</pre>

Attributes	Description
	<p><code>&lt;/w:font&gt;</code></p> <p>The fontKey attribute has a value of {302EE813-EB4A-4642-A93A-89EF99B2457E}, therefore the embedded Arial font targeted with the relationship with ID rId10 may be retrieved if needed by using this key and the algorithm above. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_Guid simple type (§2.18.39).</p>
<p>id (Relationship to Part)</p> <p>Namespace: .../officeDocument/2006/relationships</p>	<p>Specifies the relationship ID to a specified part.</p> <p>The specified relationship shall match the type required by the parent element:</p> <ul style="list-style-type: none"> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer">http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer</a> for the footerReference element</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/header">http://schemas.openxmlformats.org/officeDocument/2006/relationships/header</a> for the headerReference element</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/font">http://schemas.openxmlformats.org/officeDocument/2006/relationships/font</a> for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings">http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings</a> for the printerSettings element</li> </ul> <p>[Example: Consider an XML element which has the following id attribute:</p> <pre>&lt;... r:id="rId10" /&gt;</pre> <p>The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).</p>
<p>subsetting (Embedded Font Is Subsetting)</p>	<p>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.</p> <p>If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetting.</p> <p>[Example: Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt; ... &lt;w:embedRegular r:id="rId10" w:subsetting="true" /&gt; &lt;/w:font&gt;</pre>

Attributes	Description
	<p>The subsetted attribute has a value of <code>true</code>, therefore the embedded Arial font targeted with the relationship with ID <code>rId10</code> shall be treated as a subsetted font. <i>end example</i></p> <p>The possible values for this attribute are defined by the <code>ST_OnOff</code> simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```

<complexType name="CT_FontRel">
  <complexContent>
    <extension base="CT_Rel">
      <attribute name="fontKey" type="ST_Guid"/>
      <attribute name="subsetted" type="ST_OnOff"/>
    </extension>
  </complexContent>
</complexType>

```

#### 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.

[*Example:* Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```

<w:font w:name="Arial">
  ...
  <w:embedBoldItalic r:id="rId11" />
</w:font>

```

The `embedBoldItalic` element specifies that the embedded font targeted with the relationship with ID `rId11` may be used to retrieve the bold italic form of the embedded Arial font. *end example*

Parent Elements
font (§2.8.2.10)

Attributes	Description
fontKey (Embedded Font Obfuscation Key)	<p>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.</p> <p>If this attribute is omitted, then no key is provided for this font.</p> <p>[<i>Example:</i> Consider a WordprocessingML document in which the Arial font has been</p>

Attributes	Description
	<p>embedded in the file. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt; ... &lt;w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" /&gt; &lt;/w:font&gt;</pre> <p>The fontKey attribute has a value of {302EE813-EB4A-4642-A93A-89EF99B2457E}, therefore the embedded Arial font targeted with the relationship with ID rId10 may be retrieved if needed by using this key and the algorithm above. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_Guid simple type (§2.18.39).</p>
<p>id (Relationship to Part)</p> <p>Namespace: .../officeDocument/2006/relationships</p>	<p>Specifies the relationship ID to a specified part.</p> <p>The specified relationship shall match the type required by the parent element:</p> <ul style="list-style-type: none"> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer">http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer</a> for the footerReference element</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/header">http://schemas.openxmlformats.org/officeDocument/2006/relationships/header</a> for the headerReference element</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/font">http://schemas.openxmlformats.org/officeDocument/2006/relationships/font</a> for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings">http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings</a> for the printerSettings element</li> </ul> <p>[Example: Consider an XML element which has the following id attribute:</p> <pre>&lt;... r:id="rId10" /&gt;</pre> <p>The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).</p>
<p>subsetting (Embedded Font Is Subsetting)</p>	<p>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.</p> <p>If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetting.</p> <p>[Example: Consider a WordprocessingML document in which the Arial font has been</p>

Attributes	Description
	<p>embedded in the file after subsetting. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt; ... &lt;w:embedRegular r:id="rId10" w:subsetting="true" /&gt; &lt;/w:font&gt;</pre> <p>The subsetting attribute has a value of true, therefore the embedded Arial font targeted with the relationship with ID rId10 shall be treated as a subsetting font. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.67).</p>

1 The following XML Schema fragment defines the contents of this element:

```
2 <complexType name="CT_FontRel">
3   <complexContent>
4     <extension base="CT_Rel">
5       <attribute name="fontKey" type="ST_Guid"/>
6       <attribute name="subsetting" type="ST_OnOff"/>
7     </extension>
8   </complexContent>
9 </complexType>
```

### 10 2.8.2.5 `embedItalic` (Italic Font Style Embedding)

11 This element specifies information about the embedded font storage for the italic form of a font, when it is  
12 embedded. This form is used when italics are applied to a text run.

13 If this element is omitted, then no italic form of the font is stored in the document.

14 [Example: Consider a WordprocessingML document in which the Arial font has been embedded in the file. This  
15 status would be specified using the following WordprocessingML:

```
16 <w:font w:name="Arial">
17   ...
18   <w:embedItalic r:id="rId12" />
19 </w:font>
```

20 The embedItalic element specifies that the embedded font targeted with the relationship with ID rId12 may be  
21 used to retrieve the italic form of the embedded Arial font. *end example*

Parent Elements
font (§2.8.2.10)

22

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

Attributes	Description
<b>fontKey</b> (Embedded Font Obfuscation Key)	<p>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.</p> <p>If this attribute is omitted, then no key is provided for this font.</p> <p>[<i>Example:</i> Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt; ... &lt;w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" /&gt; &lt;/w:font&gt;</pre> <p>The fontKey attribute has a value of {302EE813-EB4A-4642-A93A-89EF99B2457E}, therefore the embedded Arial font targeted with the relationship with ID rId10 may be retrieved if needed by using this key and the algorithm above. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_Guid simple type (§2.18.39).</p>
<b>id</b> (Relationship to Part)  Namespace: .../officeDocument/2006/relationships	<p>Specifies the relationship ID to a specified part.</p> <p>The specified relationship shall match the type required by the parent element:</p> <ul style="list-style-type: none"> <li>• http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer for the footerReference element</li> <li>• http://schemas.openxmlformats.org/officeDocument/2006/relationships/header for the headerReference element</li> <li>• http://schemas.openxmlformats.org/officeDocument/2006/relationships/font for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements</li> <li>• http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings for the printerSettings element</li> </ul> <p>[<i>Example:</i> Consider an XML element which has the following id attribute:</p> <pre>&lt;... r:id="rId10" /&gt;</pre> <p>The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).</p>
<b>subsetted</b> (Embedded Font Is Subsetted)	<p>Specifies that the embedded font targeted by the id attribute has been subsetted. <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</p>



Attributes	Description
	<p>file from becoming unnecessarily large from the use of a small number of glyphs from a large embedded font.</p> <p>If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetted.</p> <p>[<i>Example:</i> Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt;   ...   &lt;w:embedRegular r:id="rId10" w:subsetting="true" /&gt; &lt;/w:font&gt;</pre> <p>The subsetting attribute has a value of true, therefore the embedded Arial font targeted with the relationship with ID rId10 shall be treated as a subsetted font. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_FontRel">
  <complexContent>
    <extension base="CT_Rel">
      <attribute name="fontKey" type="ST_Guid"/>
      <attribute name="subsetting" type="ST_OnOff"/>
    </extension>
  </complexContent>
</complexType>
```

### 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.

[*Example:* Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:

```
<w:font w:name="Arial">
  ...
  <w:embedRegular r:id="rId13" />
</w:font>
```

The embedRegular element specifies that the embedded font targeted with the relationship with ID rId13 may be used to retrieve the regular form of the embedded Arial font. *end example*]

Parent Elements
font (§2.8.2.10)

1

Attributes	Description
fontKey (Embedded Font Obfuscation Key)	<p>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.</p> <p>If this attribute is omitted, then no key is provided for this font.</p> <p>[<i>Example:</i> Consider a WordprocessingML document in which the Arial font has been embedded in the file. This status would be specified using the following WordprocessingML:</p> <pre>&lt;w:font w:name="Arial"&gt; ...   &lt;w:embedRegular r:id="rId10" w:fontKey="{302EE813-EB4A-4642-A93A-89EF99B2457E}" /&gt; &lt;/w:font&gt;</pre> <p>The fontKey attribute has a value of {302EE813-EB4A-4642-A93A-89EF99B2457E}, therefore the embedded Arial font targeted with the relationship with ID rId10 may be retrieved if needed by using this key and the algorithm above. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_Guid simple type (§2.18.39).</p>
id (Relationship to Part)  Namespace: .../officeDocument/2006/relationships	<p>Specifies the relationship ID to a specified part.</p> <p>The specified relationship shall match the type required by the parent element:</p> <ul style="list-style-type: none"> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer">http://schemas.openxmlformats.org/officeDocument/2006/relationships/footer</a> for the footerReference element</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/header">http://schemas.openxmlformats.org/officeDocument/2006/relationships/header</a> for the headerReference element</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/font">http://schemas.openxmlformats.org/officeDocument/2006/relationships/font</a> for the embedBold, embedBoldItalic, embedItalic, or embedRegular elements</li> <li>• <a href="http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings">http://schemas.openxmlformats.org/officeDocument/2006/relationships/printerSettings</a> for the printerSettings element</li> </ul> <p>[<i>Example:</i> Consider an XML element which has the following id attribute:</p> <pre>&lt;... r:id="rId10" /&gt;</pre> <p>The markup specifies the associated relationship part with relationship ID rId1 contains the corresponding relationship information for the parent XML element. <i>end example</i>]</p>

Attributes	Description
	The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).
subsetting (Embedded Font Is Subsetting)	<p>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.</p> <p>If this attribute is omitted, then the embedded font target by the id attribute shall not be handled as though it is subsetting.</p> <p>[<i>Example:</i> Consider a WordprocessingML document in which the Arial font has been embedded in the file after subsetting. This status would be specified using the following WordprocessingML:</p> <pre data-bbox="451 793 1289 926"> &lt;w:font w:name="Arial"&gt; ...   &lt;w:embedRegular r:id="rId10" w:subsetting="true" /&gt; &lt;/w:font&gt; </pre> <p>The subsetting attribute has a value of true, therefore the embedded Arial font targeted with the relationship with ID rId10 shall be treated as a subsetting font. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```

<complexType name="CT_FontRel">
  <complexContent>
    <extension base="CT_Rel">
      <attribute name="fontKey" type="ST_Guid"/>
      <attribute name="subsetting" type="ST_OnOff"/>
    </extension>
  </complexContent>
</complexType>

```

### 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 embedding is turned on. If the embedTrueTypeFonts element is omitted or false, then this setting has no effect.

[*Example:* Consider a WordprocessingML document that specifies that it shall embed fonts, including common system fonts. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:embedTrueTypeFonts w:val="true" />
<w:embedSystemFonts w:val="true"/>
```

The embedSystemFonts element's val attribute has a value of true specifying that common system fonts should be included in this document when they are used. *end example]*

Parent Elements
settings (§2.15.1.78)

Attributes	Description
val (On/Off Value)	<p>Specifies a binary value for the property defined by the parent XML element.</p> <p>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.</p> <p>A value of off, 0, or false specifies that the property shall be explicitly turned off.</p> <p>[<i>Example:</i> For example, consider the following on/off property:</p> <pre>&lt;w:... w:val="off"/&gt;</pre> <p>The val attribute explicitly declares that the property is turned off. <i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_OnOff">
  <attribute name="val" type="ST_OnOff"/>
</complexType>
```

### 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.

[*Example:* Consider a WordprocessingML document that specifies that it shall embed fonts, including common system fonts. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:embedTrueTypeFonts w:val="true" />
```

```
<w:embedSystemFonts w:val="true"/>
```

The embedTrueType element's val attribute has a value of true specifying that fonts should be embedded in this document when they are used. *end example*

Parent Elements
settings (§2.15.1.78)

Attributes	Description
val (On/Off Value)	<p>Specifies a binary value for the property defined by the parent XML element.</p> <p>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.</p> <p>A value of off, 0, or false specifies that the property shall be explicitly turned off.</p> <p>[Example: For example, consider the following on/off property:</p> <pre>&lt;w:... w:val="off" /&gt;</pre> <p>The val attribute explicitly declares that the property is turned off. <i>end example</i></p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_OnOff">
```

```
  <attribute name="val" type="ST_OnOff"/>
```

```
</complexType>
```

### 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.

[Example: Consider the following information stored for a single font:

```
<w:font w:name="Calibri">
```

```
  <w:family w:val="swiss" />
```

```
  ...
```

```
</w:font>
```

1 The family element specifies via its val attribute value of swiss that this font is part of the Swiss family. *end*  
2 *example*]

Parent Elements
font (§2.8.2.10)

3

Attributes	Description
val (Font Family Value)	<p>Specifies the font family for the parent font.</p> <p>[<i>Example</i>: Consider the following information stored for a single font:</p> <pre>&lt;w:font w:name="Times New Roman"&gt;   &lt;w:family w:val="roman" /&gt;   ... &lt;/w:font&gt;</pre> <p>The val attribute value of swiss that this font is part of the Roman family. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_FontFamily simple type (§2.18.34).</p>

4 The following XML Schema fragment defines the contents of this element:

5 <complexType name="CT_FontFamily"> 6   <attribute name="val" type="ST_FontFamily" use="required"/> 7 </complexType>
---

8 **2.8.2.10 font (Properties for a Single Font)**

9 This element specifies the properties for one of the fonts used in this document. A font element shall be written  
10 out for each font face used in the document, and includes:

- 11
- 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
- 13

14 [*Example*: Consider the following information stored for a single font:

15 <w:font w:name="Times New Roman"> 16   <w:panose1 w:val="02020603050405020304" /> 17   <w:charset w:val="00" /> 18   <w:family w:val="roman" /> 19   <w:pitch w:val="variable" /> 20   <w:sig w:usb0="20002A87" w:usb1="80000000" w:usb2="00000008" w:usb3="00000000" 21   w:csb0="000001FF" w:csb1="00000000" /> 22 </w:font>
--

- 1 The font element contains information about the Times New Roman font; specifically, information used to  
 2 locate a substitute font if it is not available. *end example*]

3

Parent Elements
fonts (§2.8.2.11)

4

Child Elements	Subclause
altName (Alternate Names for Font)	§2.8.2.1
charset (Character Set Supported By Font)	§2.8.2.2
embedBold (Bold Style Font Style Embedding)	§2.8.2.3
embedBoldItalic (Bold Italic Font Style Embedding)	§2.8.2.4
embedItalic (Italic Font Style Embedding)	§2.8.2.5
embedRegular (Regular Font Style Embedding)	§2.8.2.6
family (Font Family)	§2.8.2.9
notTrueType (Raster or Vector Font)	§2.8.2.12
panose1 (Pansose-1 Typeface Classification Number)	§2.8.2.13
pitch (Font Pitch)	§2.8.2.14
sig (Supported Unicode Subranges and Code Pages)	§2.8.2.16

Attributes	Description
name (Primary Font Name)	<p>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.</p> <p>[<i>Example:</i> Consider the following information stored for a single font:</p> <pre>&lt;w:font w:name="Times New Roman"&gt; ... &lt;/w:font&gt;</pre> <p>The name attribute specifies that the information contained in this element shall be used to look up information about all uses of the Times New Roman font in the document contents. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_String simple type (§2.18.89).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Font">
  <sequence>
    <element name="altName" type="CT_String" minOccurs="0" maxOccurs="1"/>
    <element name="panose1" type="CT_Panose" minOccurs="0" maxOccurs="1"/>
    <element name="charset" type="CT_UcharHexNumber" minOccurs="0" maxOccurs="1"/>
    <element name="family" type="CT_FontFamily" minOccurs="0" maxOccurs="1"/>
    <element name="notTrueType" type="CT_OnOff" minOccurs="0" maxOccurs="1"/>
    <element name="pitch" type="CT_Pitch" minOccurs="0" maxOccurs="1"/>
    <element name="sig" type="CT_FontSig" minOccurs="0" maxOccurs="1"/>
    <element name="embedRegular" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
    <element name="embedBold" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
    <element name="embedItalic" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
    <element name="embedBoldItalic" type="CT_FontRel" minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="name" type="ST_String" use="required"/>
</complexType>
```

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.

[Example: Consider the following information stored in a font table part:

```
<w:fonts>
  <w:font w:name="Times New Roman">
    ...
  </w:font>
  <w:font w:name="Arial">
    ...
  </w:font>
</w:fonts>
```

The fonts element contains information about all fonts used in the document - in this example, the Times New Roman and Arial fonts. *end example]*

Parent Elements	
Root element of WordprocessingML Font Table part	

Child Elements	Subclause
font (Properties for a Single Font)	§2.8.2.10



The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_FontsList">
  <sequence>
    <element name="font" type="CT_Font" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

### 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.

[*Example:* Consider the following information stored for a single font:

```
<w:font w:name="JonsFont">
  <w:notTrueType w:val="true" />
  ...
</w:font>
```

The notTrueType element specifies via its val attribute value of true that this font is a raster or vector font. *end example]*

#### Parent Elements

font (§2.8.2.10)

Attributes	Description
val (On/Off Value)	<p>Specifies a binary value for the property defined by the parent XML element.</p> <p>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.</p> <p>A value of off, 0, or false specifies that the property shall be explicitly turned off.</p> <p>[<i>Example:</i> For example, consider the following on/off property:</p> <pre>&lt;w:... w:val="off"/&gt;</pre> <p>The val attribute explicitly declares that the property is turned off. <i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_OnOff simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_OnOff">
  <attribute name="val" type="ST_OnOff"/>
</complexType>
```

### 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.

[*Example:* Consider the following information stored for a single font:

```
<w:font w:name="Times New Roman">
  <w:panose1 w:val="02020603050405020304" />
  ...
</w:font>
```

The panose1 element specifies its Panose-1 number via its val attribute value of 02020603050405020304. *end example]*

#### Parent Elements

font (§2.8.2.10)

#### 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.

[*Example:* Consider the following information stored for a single font:

```
<w:panose1 w:val="020F0603050405020304" />
```

The val attribute specifies that the digits in the Panose-1 number are: 2,15,6,3,5,2,3,4. *end example]*

The possible values for this attribute are defined by the ST\_Panose simple type (§2.18.72).

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Panose">
  <attribute name="val" type="ST_Panose" use="required"/>
</complexType>
```

#### 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.

[*Example:* Consider the following information stored for a single font:

```
<w:font w:name="Courier New">
  <w:pitch w:val="fixed" />
  ...
</w:font>
```

The pitch element specifies via its val attribute value of fixed that this is a fixed width font. *end example]*

Parent Elements
font (§2.8.2.10)

Attributes	Description
val (Value)	<p>Specifies the font pitch for the font.</p> <p>[<i>Example:</i> Consider the following information stored for a single font:</p> <pre>&lt;w:pitch w:val="variable" /&gt;</pre> <p>The val attribute value of variable specifies that this is a variable width font. <i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_Pitch simple type (§2.18.73).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_Pitch">
  <attribute name="val" type="ST_Pitch" use="required"/>
</complexType>
```

#### 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.

[*Example:* Consider a WordprocessingML document that specifies that it shall subset embedded fonts. This requirement would be specified using the following WordprocessingML in the document settings part:

```
<w:embedTrueTypeFonts w:val="true" />
<w:saveSubsetFonts w:val="true"/>
```

The `embedSystemFonts` element's `val` attribute has a value of `true` specifying fonts should be subsetted in this document when they are embedded. *end example*]

Parent Elements
settings (§2.15.1.78)

Attributes	Description
val (On/Off Value)	<p>Specifies a binary value for the property defined by the parent XML element.</p> <p>A value of <code>on</code>, <code>1</code>, or <code>true</code> 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.</p> <p>A value of <code>off</code>, <code>0</code>, or <code>false</code> specifies that the property shall be explicitly turned off.</p> <p>[<i>Example:</i> For example, consider the following on/off property:</p> <pre>&lt;w:... w:val="off"/&gt;</pre> <p>The <code>val</code> attribute explicitly declares that the property is turned off. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the <code>ST_OnOff</code> simple type (§2.18.67).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_OnOff">
  <attribute name="val" type="ST_OnOff"/>
</complexType>
```

### 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.

[*Example:* Consider the following information stored for a single font:

```
<w:font w:name="Times New Roman">
  <w:sig w:usb0="20002A87" w:usb1="80000000" w:usb2="00000008" w:usb3="00000000"
  w:csb0="000001FF" w:csb1="00000000" />
  ...
</w:font>
```

The sig element specifies the supported code pages and Unicode sub ranges via its attributes. For example, the code pages supported are:

- Latin 1
- Latin 2: Eastern Europe
- Cyrillic
- Greek
- Turkish
- Baltic

*end example]*

#### Parent Elements

font (§2.8.2.10)

#### Attributes

csb0 (Lower 32 Bits of Code Page Bit Field)

#### Description

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.

Each bit in this 32 bits represents the following code page:

Bit	Description
0	Latin 1
1	Latin 2: Eastern Europe
2	Cyrillic
3	Greek
4	Turkish
5	Hebrew
6	Arabic

Attributes	Description																				
	<table border="1" data-bbox="415 243 1205 730"> <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 data-bbox="415 768 1092 800">[Example: Consider font information specified as follows:</p> <pre data-bbox="456 842 967 972"> &lt;w:font w:name="Lucida Console"&gt;   &lt;w:sig w:csb0="0000001F" ... /&gt;   ... &lt;/w:font&gt; </pre> <p data-bbox="415 1014 1380 1077">The csb0 attribute value of 0000001F specifies that the following code pages are supported by this font:</p> <ul data-bbox="456 1087 786 1262" style="list-style-type: none"> <li>• Latin 1</li> <li>• Latin 2: Eastern Europe</li> <li>• Cyrillic</li> <li>• Greek</li> <li>• Turkish</li> </ul> <p data-bbox="415 1304 574 1335"><i>end example]</i></p> <p data-bbox="415 1377 1484 1440">The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.57).</p>	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
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 data-bbox="415 1461 1455 1566">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 data-bbox="415 1608 1105 1640">Each bit in this 32 bits represents the following code page:</p> <table border="1" data-bbox="415 1640 1205 1877"> <tr> <th data-bbox="415 1640 561 1682">Bit</th><th data-bbox="561 1640 1205 1682">Description</th></tr> <tr> <td data-bbox="415 1682 561 1734">0 to 15</td><td data-bbox="561 1682 1205 1734">Reserved for OEM</td></tr> <tr> <td data-bbox="415 1734 561 1776">16</td><td data-bbox="561 1734 1205 1776">IBM Greek</td></tr> <tr> <td data-bbox="415 1776 561 1818">17</td><td data-bbox="561 1776 1205 1818">MS-DOS Russian</td></tr> <tr> <td data-bbox="415 1818 561 1877">18</td><td data-bbox="561 1818 1205 1877">MS-DOS Nordic</td></tr> </table>	Bit	Description	0 to 15	Reserved for OEM	16	IBM Greek	17	MS-DOS Russian	18	MS-DOS Nordic										
Bit	Description																				
0 to 15	Reserved for OEM																				
16	IBM Greek																				
17	MS-DOS Russian																				
18	MS-DOS Nordic																				

Attributes	Description																										
	<table border="1" data-bbox="415 243 1205 873"> <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 data-bbox="415 915 1092 945">[Example: Consider font information specified as follows:</p> <pre data-bbox="453 984 967 1115"> &lt;w:font w:name="Lucida Console"&gt;   &lt;w:sig w:csb1="00000000" ... /&gt;   ... &lt;/w:font&gt; </pre> <p data-bbox="415 1157 1468 1224">The csb1 attribute value of 00000000 specifies that none of the specified code pages are supported by this font. <i>end example</i>]</p> <p data-bbox="415 1266 1479 1333">The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.57).</p>	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
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)	<p data-bbox="415 1350 1417 1417">Specifies the first 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard.</p> <p data-bbox="415 1459 1092 1488">[Example: Consider font information specified as follows:</p> <pre data-bbox="453 1528 984 1659"> &lt;w:font w:name="Times New Roman"&gt;   &lt;w:sig w:usb0="20002A87" ... /&gt;   ... &lt;/w:font&gt; </pre> <p data-bbox="415 1701 1429 1768">The usb0 attribute value of 20002A87 specifies that the first 32 bits of the bitfield are 001000000000000000010101010000111, which corresponds to:</p> <ul data-bbox="461 1774 745 1877" style="list-style-type: none"> <li>• Basic Latin</li> <li>• Latin-1 Supplement</li> <li>• Latin Extended-A</li> </ul>																										

Attributes	Description
	<ul style="list-style-type: none"> <li>• Basic Greek</li> <li>• Cyrillic</li> <li>• Basic Hebrew</li> <li>• Basic Arabic</li> <li>• Latin Extended Additional</li> </ul> <p><i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.57).</p>
usb1 (Second 32 Bits of Unicode Subset Bitfield)	<p>Specifies the second 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard.</p> <p>[<i>Example:</i> Consider font information specified as follows:</p> <pre>&lt;w:font w:name="Times New Roman"&gt;   &lt;w:sig w:usb1="80000000" ... /&gt;   ... &lt;/w:font&gt;</pre> <p>The usb0 attribute value of 80000000 specifies that the first 32 bits of the bitfield are 10000000000000000000000000000000, which corresponds to:</p> <ul style="list-style-type: none"> <li>• Arabic Presentation Forms-A</li> </ul> <p><i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.57).</p>
usb2 (Third 32 Bits of Unicode Subset Bitfield)	<p>Specifies the third 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard.</p> <p>[<i>Example:</i> Consider font information specified as follows:</p> <pre>&lt;w:font w:name="Times New Roman"&gt;   &lt;w:sig w:usb2="00000008" ... /&gt;   ... &lt;/w:font&gt;</pre> <p>The usb0 attribute value of 80000000 specifies that the first 32 bits of the bitfield are 00000000000000000000000000001000, which corresponds to:</p> <ul style="list-style-type: none"> <li>• Arabic Presentation Forms-B</li> </ul> <p><i>end example]</i></p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type</p>



Attributes	Description
	(§2.18.57).
usb3 (Fourth 32 Bits of Unicode Subset Bitfield)	<p>Specifies the fourth 32 bits of the 128-bit Unicode subset bit field (USB). Subranges are ordered in accordance with the ISO 10646 standard.</p> <p>[<i>Example:</i> Consider font information specified as follows:</p> <pre>&lt;w:font w:name="Times New Roman"&gt;   &lt;w:sig w:usb3="00000000" ... /&gt;   ... &lt;/w:font&gt;</pre> <p>The usb3 attribute value of 00000000 specifies that the first 32 bits of the bitfield are 00000000000000000000000000000000, which corresponds to no subranges. <i>end example</i>]</p> <p>The possible values for this attribute are defined by the ST_LongHexNumber simple type (§2.18.57).</p>

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_FontSig">
  <attribute name="usb0" use="required" type="ST_LongHexNumber"/>
  <attribute name="usb1" use="required" type="ST_LongHexNumber"/>
  <attribute name="usb2" use="required" type="ST_LongHexNumber"/>
  <attribute name="usb3" use="required" type="ST_LongHexNumber"/>
  <attribute name="csb0" use="required" type="ST_LongHexNumber"/>
  <attribute name="csb1" use="required" type="ST_LongHexNumber"/>
</complexType>
```

## 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.

[*Example:* The following two paragraphs each contain numbering as defined by WordprocessingML: the first uses an Arabic numeral, the second a symbol character:

9. This is a paragraph with numbering information.

- This is also a paragraph with numbering information.

*end example*]

The basis for all numbering in WordprocessingML is specified via two structures:

- abstract numbering definitions
- numbering definition instances

information, this tag mainly focuses on referencing to other parts of the presentation document. This is accomplished via the relationship identification attribute that is required for all specified tags.

Parent Elements
custDataLst (§4.4.1.16)

Attributes	Description
id (Relationship ID)  Namespace: .../officeDocument /2006/relationships	This attribute specifies the relationship identifier for the customer data tag. This allows for a link to a resource that is external from the current XML document but still contained within the presentation document.  The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_TagsData">  
  <attribute ref="r:id" use="required"/>  
</complexType>
```

### 4.3 Presentation

The Presentation portion of the PresentationML framework houses a set of elements that describe the storing of presentation-wide and view-specific properties. The presentation-wide properties are those that pertain to the entire presentation. The view-specific properties assist the generating application and viewing application by storing parameters that pertain to the final delivery of the presentation.

#### 4.3.1 Presentation Properties

This section contains all presentation-level properties that pertain to a presentation document:

##### 4.3.1.1 bold (Bold Embedded Font)

This element specifies a bold embedded font that is linked to a parent typeface. Once specified, this bold version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[Example: Consider the following embedded font with a bold version specified.

```
<p:embeddedFont>  
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>  
  <p:bold r:id="rId2"/>  
</p:embeddedFont>
```

end example]

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

Parent Elements
embeddedFont (§4.3.1.8)

Attributes	Description
id (Relationship Identifier)  Namespace: .../officeDocument /2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation.  The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_EmbeddedFontDataId">  
  <attribute ref="r:id" use="required"/>  
</complexType>
```

#### 4.3.1.2 boldItalic (Bold Italic Embedded Font)

This element specifies a bold italic embedded font that is linked to a parent typeface. Once specified, this bold italic version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[*Example*: Consider the following embedded font with a bold italic version specified.

```
<p:embeddedFont>  
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>  
  <p:boldItalic r:id="rId2"/>  
</p:embeddedFont>
```

*end example*]

[*Note*: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note*]

Parent Elements
embeddedFont (§4.3.1.8)

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

Attributes	Description
id (Relationship Identifier)  Namespace: .../officeDocument/2006/relationships	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation.  The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).

1 The following XML Schema fragment defines the contents of this element:

```
2 <complexType name="CT_EmbeddedFontDataId">
3   <attribute ref="r:id" use="required"/>
4 </complexType>
```

5 **4.3.1.3** browse (Browse Slide Show Mode)

6 This element specifies that the presentation slide show should be viewed in a single window or browse mode,  
7 instead of full screen.

8 [Example: Consider the following presentation that is set to be viewed in a browse mode.

```
9 <p:presentationPr xmlns:a="" xmlns:r="" xmlns:p="">
10   <p:showPr>
11     ..
12     <p:browse showScrollbar="0"/>
13     ..
14   </p:showPr>
15 </p:presentationPr>
```

16 end example]

Parent Elements
showPr (§4.3.1.28)

17

Attributes	Description
showScrollbar (Show Scroll Bar in Window)	Specifies whether to show the scroll bar in the viewing window.  The possible values for this attribute are defined by the XML Schema boolean datatype.

18 The following XML Schema fragment defines the contents of this element:

```
19 <complexType name="CT_ShowInfoBrowse">
20   <attribute name="showScrollbar" type="xsd:boolean" use="optional" default="true"/>
21 </complexType>
```

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_TextListStyle">
  <sequence>
    <element name="defPPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv11pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv12pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv13pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv14pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv15pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv16pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv17pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv18pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="lv19pPr" type="CT_TextParagraphProperties" minOccurs="0" maxOccurs="1"/>
    <element name="extLst" type="CT_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

4.3.1.8 **embeddedFont (Embedded Font)**

This element specifies an embedded font. Once specified, this font is available for use within the presentation. Within a font specification there can be regular, bold, italic and boldItalic versions of the font specified. The actual font data for each of these is referenced using a relationships file that contains links to all available fonts. This font data contains font information for each of the characters to be made available in each version of the font.

[Example: Consider the following embedded font.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:regular r:id="rId2"/>
</p:embeddedFont>
```

end example]

[Note: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. end note]

Parent Elements
embeddedFontLst (§4.3.1.9)

Child Elements	Subclause
bold (Bold Embedded Font)	§4.3.1.1
boldItalic (Bold Italic Embedded Font)	§4.3.1.2
font (Embedded Font Name)	§4.3.1.10
italic (Italic Embedded Font)	§4.3.1.14

Child Elements	Subclause
regular (Regular Embedded Font)	§4.3.1.27

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_EmbeddedFontListEntry">
  <sequence>
    <element name="font" type="a:CT_TextFont" minOccurs="1" maxOccurs="1"/>
    <element name="regular" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
    <element name="bold" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
    <element name="italic" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
    <element name="boldItalic" type="CT_EmbeddedFontDataId" minOccurs="0" maxOccurs="1"/>
  </sequence>
</complexType>
```

4.3.1.9 [embeddedFontLst \(Embedded Font List\)](#)

This element specifies a list of fonts that are embedded within the corresponding presentation. The font data for these fonts is stored alongside the other document parts within the document container. The actual font data is referenced within the embeddedFont element.

Parent Elements
presentation (§4.3.1.24)

Child Elements	Subclause
embeddedFont (Embedded Font)	§4.3.1.8

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_EmbeddedFontList">
  <sequence>
    <element name="embeddedFont" type="CT_EmbeddedFontListEntry" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

4.3.1.10 [font \(Embedded Font Name\)](#)

This element specifies specific properties describing an embedded font. Once specified, this font is available for use within the presentation. Within a font specification there can be regular, bold, italic and boldItalic versions of the font specified. The actual font data for each of these is referenced using a relationships file that contains links to all available fonts. This font data contains font information for each of the characters to be made available in each version of the font.

[Example: Consider the following embedded font.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
```

1       <p:regular r:id="rId2"/>  
2       </p:embeddedFont>

3   *end example]*

4   [*Note:* Not all characters for a typeface must be stored. It is up to the generating application to determine which  
5   characters are to be stored in the corresponding font data files. *end note]*

Parent Elements
embeddedFont (§4.3.1.8)

6

Attributes	Description
charset (Similar Character Set)  Namespace: .../drawingml/2006/main	Specifies the most similar character set to the one being used. This is useful if the generating application cannot use the current font and must choose a similar one.  The possible values for this attribute are defined by the XML Schema byte datatype.
panose (Panose Setting)  Namespace: .../drawingml/2006/main	Specifies the panose standard setting that will be used to determine the closest matching font by any generating application that employs this method.  The possible values for this attribute are defined by the ST_Panose simple type (§5.1.12.37).
pitchFamily (Similar Font Family)  Namespace: .../drawingml/2006/main	Specifies the most similar font family to the one being used. This is useful if the generating application cannot use the current font and must choose a similar one.  The possible values for this attribute are defined by the XML Schema byte datatype.
typeface (Text Typeface)  Namespace: .../drawingml/2006/main	Specifies the typeface, or name of the font that is to be used for a bullet. The typeface used here should be selected from the font list of the generating application.  The possible values for this attribute are defined by the ST_TextTypeface simple type (§5.1.12.81).

7   The following XML Schema fragment defines the contents of this element:

8   <complexType name="CT\_TextFont">  
9     <attribute name="typeface" type="ST\_TextTypeface"/>  
10    <attribute name="panose" type="ST\_Panose" use="optional"/>  
11    <attribute name="pitchFamily" type="xsd:byte" use="optional" default="0"/>  
12    <attribute name="charset" type="xsd:byte" use="optional" default="1"/>  
13 </complexType>

Attributes	Description
Notes)	The possible values for this attribute are defined by the XML Schema boolean datatype.
title (HTML Output Title)	Specifies a title for the HTML output file. The possible values for this attribute are defined by the XML Schema string datatype.

The following XML Schema fragment defines the contents of this element:

```

<complexType name="CT_HtmlPublishProperties">
  <sequence>
    <group ref="EG_SlideListChoice" minOccurs="1" maxOccurs="1"/>
    <element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="showSpeakerNotes" type="xsd:boolean" use="optional" default="true"/>
  <attribute name="pubBrowser" type="ST_HtmlPublishWebBrowserSupport" use="optional"
    default="v3v4"/>
  <attribute name="title" type="xsd:string" use="optional" default=""/>
  <attribute ref="r:id" use="required"/>
</complexType>

```

#### 4.3.1.14 *italic* (Italic Embedded Font)

This element specifies an italic embedded font that is linked to a parent typeface. Once specified, this italic version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[*Example:* Consider the following embedded font with a italic version specified.

```

<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:italic r:id="rId2"/>
</p:embeddedFont>

```

*end example]*

[*Note:* Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note]*

Parent Elements
embeddedFont (§4.3.1.8)

Attributes	Description
id (Relationship Identifier)	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation.



Attributes	Description
Namespace: .../officeDocument /2006/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).

1 The following XML Schema fragment defines the contents of this element:

```
2 <complexType name="CT_EmbeddedFontDataId">
3   <attribute ref="r:id" use="required"/>
4 </complexType>
```

#### 5 4.3.1.15 kinsoku (Kinsoku Settings)

6 This element specifies the presentation-wide kinsoku settings that define the line breaking behaviour of East  
7 Asian text within the corresponding presentation.

Parent Elements
presentation (§4.3.1.24)

8

Attributes	Description
invalEndChars (Invalid Kinsoku End Characters)	Specifies the characters that are not valid for ending a line of text with. The possible values for this attribute are defined by the XML Schema string datatype.
invalStChars (Invalid Kinsoku Start Characters)	Specifies the characters that are not valid for starting a line of text with. The possible values for this attribute are defined by the XML Schema string datatype.
lang (Language)	Specifies the corresponding East Asian language that these settings apply to. The possible values for this attribute are defined by the XML Schema string datatype.

9 The following XML Schema fragment defines the contents of this element:

```
10 <complexType name="CT_Kinsoku">
11   <attribute name="lang" type="xsd:string" use="optional"/>
12   <attribute name="invalStChars" type="xsd:string" use="required"/>
13   <attribute name="invalEndChars" type="xsd:string" use="required"/>
14 </complexType>
```

#### 15 4.3.1.16 kiosk (Kiosk Slide Show Mode)

16 This element specifies that the presentation slide show should be viewed in a full-screen kiosk mode. A  
17 presentation viewed in kiosk mode should have user input disabled and will restart after a specified interval.

18 [Example: Consider the following presentation that is set to be viewed in a looping kiosk mode.

```
19 <p:presentationPr xmlns:a="" xmlns:r="" xmlns:p="">
20   <p:showPr loop="1" showNarration="1">
```

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_PrintProperties">
  <sequence>
    <element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="prnWhat" type="ST_PrintWhat" use="optional" default="slides"/>
  <attribute name="clrMode" type="ST_PrintColorMode" use="optional" default="clr"/>
  <attribute name="hiddenSlides" type="xsd:boolean" use="optional" default="false"/>
  <attribute name="scaleToFitPaper" type="xsd:boolean" use="optional" default="false"/>
  <attribute name="frameSlides" type="xsd:boolean" use="optional" default="false"/>
</complexType>
```

4.3.1.27 regular (Regular Embedded Font)

This element specifies a regular embedded font that is linked to a parent typeface. Once specified, this regular version of the given typeface name is available for use within the presentation. The actual font data is referenced using a relationships file that contains links to all fonts available. This font data contains font information for each of the characters to be made available.

[Example: Consider the following embedded font with a regular version specified.

```
<p:embeddedFont>
  <p:font typeface="MyFont" pitchFamily="34" charset="0"/>
  <p:regular r:id="rId2"/>
</p:embeddedFont>
```

*end example]*

[Note: Not all characters for a typeface must be stored. It is up to the generating application to determine which characters are to be stored in the corresponding font data files. *end note]*

Parent Elements

embeddedFont (§4.3.1.8)

Attributes

Description

id (Relationship Identifier)	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location of this embedded font that is referenced in a presentation.
Namespace: .../officeDocument /2006/relationships	The possible values for this attribute are defined by the ST_RelationshipId simple type (§7.8.2.1).

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_EmbeddedFontDataId">
  <attribute ref="r:id" use="required"/>
</complexType>
```

#### 4.3.1.28 **showPr** (Presentation-wide Show Properties)

- This element functions as a parent element within which all presentation-wide show properties are contained.  
All properties and their corresponding settings are defined within the child elements.

Parent Elements
presentationPr (§4.3.1.25)

Child Elements	Subclause
browse (Browse Slide Show Mode)	§4.3.1.3
custShow (Custom Show)	§4.2.2
extLst (Extension List)	§4.2.5
kiosk (Kiosk Slide Show Mode)	§4.3.1.16
penClr (Pen Color for Slide Show)	§4.3.1.21
present (Presenter Slide Show Mode)	§4.3.1.23
sldAll (All Slides)	§4.2.7
sldRg (Slide Range)	§4.2.8

Attributes	Description
loop (Loop Slide Show)	Specifies whether the slide show should be set to loop at the end.  The possible values for this attribute are defined by the XML Schema boolean datatype.
showAnimation (Show Animation in Slide Show)	Specifies whether slide show animation should be shown when presenting.  The possible values for this attribute are defined by the XML Schema boolean datatype.
showNarration (Show Narration in Slide Show)	Specifies whether slide show narration should be played when presenting.  The possible values for this attribute are defined by the XML Schema boolean datatype.
useTimings (Use Timings in Slide Show)	Specifies whether slide transition timings should be used to advance slides when presenting.  The possible values for this attribute are defined by the XML Schema boolean datatype.

The following XML Schema fragment defines the contents of this element:

```
<complexType name="CT_ShowProperties">
  <sequence minOccurs="0" maxOccurs="1">
    <group ref="EG_ShowType" minOccurs="0" maxOccurs="1"/>
    <group ref="EG_SlideListChoice" minOccurs="0" maxOccurs="1"/>
    <element name="penClr" type="a:CT_Color" minOccurs="0" maxOccurs="1"/>
    <element name="extLst" type="CT_ExtensionList" minOccurs="0" maxOccurs="1"/>
  </sequence>
  <attribute name="loop" type="xsd:boolean" use="optional" default="false"/>
  <attribute name="showNarration" type="xsd:boolean" use="optional" default="false"/>
  <attribute name="showAnimation" type="xsd:boolean" use="optional" default="true"/>
  <attribute name="useTimings" type="xsd:boolean" use="optional" default="true"/>
</complexType>
```

4.3.1.29 sldId (Slide ID)

This element specifies a presentation slide that is available within the corresponding presentation. A slide contains the information that is specific to a single slide such as slide-specific shape and text information.

[Example: Consider the following specification of a slide master within a presentation

```
<p:presentation xmlns:a="" xmlns:r="" xmlns:p="" embedTrueTypeFonts="1">
  ..
  <p:sldIdLst>
    <p:sldId id="256" r:id="rId3"/>
    <p:sldId id="257" r:id="rId4"/>
    <p:sldId id="258" r:id="rId5"/>
    <p:sldId id="259" r:id="rId6"/>
    <p:sldId id="260" r:id="rId7"/>
  </p:sldIdLst>
  ..
</p:presentation>
```

end example]

Parent Elements
sldIdLst (§4.3.1.30)

Child Elements	Subclause
extLst (Extension List)	§4.2.5

Attributes	Description
id (Relationship Identifier)	Specifies the relationship identifier that is used in conjunction with a corresponding relationship file to resolve the location within a presentation of the sld element defining this slide.