Page Content Ground Truth and Storage (Part of the PAGE Format)

Table of Contents

top

Schema Document Properties

Target Namespace	**http://schema.primaresearch.org/PAGE/gts/pagecontent/2013-07-15**
Element and Attribute Namespaces	- Global element and attribute declarations belong to this schema's target namespace. - By default, local element declarations belong to this schema's target namespace. - By default, local attribute declarations have no namespace.

Declared Namespaces

Prefix	Namespace
Default namespace	http://www.w3.org/2001/XMLSchema
xml	http://www.w3.org/XML/1998/namespace
pc	**http://schema.primaresearch.org/PAGE/gts/pagecontent/2013-07-15**

Schema Component Representation

```
<schema targetNamespace="http://schema.primaresearch.org/PAGE/gts/
pagecontent/2013-07-15" elementFormDefault="qualified">
   ...
</schema>
```

top

Global Declarations

Element: PcGts

Name	PcGts
Type	pc:PcGtsType
Nillable	no
Abstract	no
Documentation	Page Content - Ground Truth and Storage

XML Instance Representation

```
<pc:PcGts
 pcGtsId="ID [0..1]">
   <pc:Metadata> pc:MetadataType </pc:Metadata> [1]
   <pc:Page> pc:PageType </pc:Page> [1]
</pc:PcGts>
```

Schema Component Representation

```
<element name="PcGts" type=" pc:PcGtsType "/>
```

top

Global Definitions

Complex Type: AdvertRegionType

Super-types:	RegionType < **AdvertRegionType** (by extension)

Sub-types:	None

Name	AdvertRegionType
Abstract	no
Documentation	Regions containing advertisements.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ?"
 comments="string [0..1]"
 orientation="float [0..1] ?"
 bgColour=" pc:ColourSimpleType [0..1] ?">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="AdvertRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: AlternativeImageType

Super-types:	None
Sub-types:	None

Name	AlternativeImageType
Abstract	no

XML Instance Representation

```
<...
 filename="string [1]"
 comments="string [0..1]"/>
```

Schema Component Representation

```
<complexType name="AlternativeImageType">
   <attribute name="filename" type=" string " use="required"/>
   <attribute name="comments" type=" string "/>
</complexType>
```

top

Complex Type: BaselineType

Super-types:	None
Sub-types:	None

Name	BaselineType
Abstract	no

XML Instance Representation

```
<...
 points=" pc:PointsType [1]"/>
```

Schema Component Representation

```
<complexType name="BaselineType">
   <attribute name="points" type=" pc:PointsType " use="required"/>
</complexType>
```

top

Complex Type: BorderType

Super-types:	None
Sub-types:	None

Name	BorderType
Abstract	no
Documentation	Border of the actual page (if the scanned image contains parts not belonging to the page).

XML Instance Representation

```
<...>
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
</...>
```

Schema Component Representation

```
<complexType name="BorderType">
   <sequence>
      <element name="Coords" type=" pc:CoordsType "/>
   </sequence>
</complexType>
```

top

Complex Type: ChartRegionType

Super-types:	RegionType < **ChartRegionType** (by extension)

Sub-types:	None

Name	ChartRegionType
Abstract	no
Documentation	Regions containing charts or graphs of any type, should be marked as chart regions.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ?"
 comments="string [0..1]"
 orientation="float [0..1] ?"
 type=" pc:ChartTypeSimpleType [0..1] ?"
 numColours="int [0..1] ?"
 bgColour=" pc:ColourSimpleType [0..1] ?"
 embText="boolean [0..1] ?">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="ChartRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="type" type=" pc:ChartTypeSimpleType " use="optional"/>
         <attribute name="numColours" type=" int " use="optional"/>
         <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
         <attribute name="embText" type=" boolean " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: ChemRegionType

Super-types:	RegionType < **ChemRegionType** (by extension)
Sub-types:	None

Name	ChemRegionType
Abstract	no

Documentation	Regions containing chemical formulas.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ? "
 comments="string [0..1]"
 orientation="float [0..1] ? "
 bgColour=" pc:ColourSimpleType [0..1] ? ">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
     <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
     <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
     <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
     <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
     <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
     <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
     <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
     <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
     <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
     <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
     <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
     <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
     <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="ChemRegionType">
  <complexContent>
    <extension base=" pc:RegionType ">
      <attribute name="orientation" type=" float " use="optional"/>
      <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

top

Complex Type: CoordsType

Super-types:	None
Sub-types:	None

Name	CoordsType
Abstract	no

XML Instance Representation

```
<...
 points=" pc:PointsType [1] ? "/>
```

Schema Component Representation

```
<complexType name="CoordsType">
  <attribute name="points" type=" pc:PointsType " use="required"/>
</complexType>
```

top

Complex Type: GlyphType

Super-types:	None
Sub-types:	None

Name	GlyphType
Abstract	no

XML Instance Representation

```
<...
 id="ID [1]"
 ligature="boolean [0..1]"
 symbol="boolean [0..1]"
 production=" pc:ProductionSimpleType [0..1] ?"
 custom="string [0..1] ?"
 comments="string [0..1]">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   <pc:TextEquiv> pc:TextEquivType </pc:TextEquiv> [0..1]
   <pc:TextStyle> pc:TextStyleType </pc:TextStyle> [0..1]
</...>
```

Schema Component Representation

```
<complexType name="GlyphType">
   <sequence>
      <element name="Coords" type=" pc:CoordsType "/>
      <element name="TextEquiv" type=" pc:TextEquivType " minOccurs="0"
      maxOccurs="1"/>
      <element name="TextStyle" type=" pc:TextStyleType " minOccurs="0"/>
   </sequence>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="ligature" type=" boolean " use="optional"/>
   <attribute name="symbol" type=" boolean " use="optional"/>
   <attribute name="production" type=" pc:ProductionSimpleType "/>
   <attribute name="custom" type=" string "/>
   <attribute name="comments" type=" string "/>
</complexType>
```

top

Complex Type: GraphicRegionType

Super-types:	RegionType < **GraphicRegionType** (by extension)
Sub-types:	None

Name	GraphicRegionType
Abstract	no
Documentation	Regions containing simple graphics, such as a company logo, should be marked as graphic regions.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ?"
 comments="string [0..1]"
 orientation="float [0..1] ?"
 type=" pc:GraphicsTypeSimpleType [0..1] ? "
 numColours="int [0..1] ? "
 embText="boolean [0..1] ? ">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
     <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
     <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
     <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
     <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
     <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
     <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
     <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
     <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
     <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
     <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
     <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
     <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
     <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="GraphicRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="type" type=" pc:GraphicsTypeSimpleType " use="optional"/>
         <attribute name="numColours" type=" int " use="optional"/>
         <attribute name="embText" type=" boolean " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: ImageRegionType

Super-types:	RegionType < **ImageRegionType** (by extension)
Sub-types:	None

Name	ImageRegionType
Abstract	no
Documentation	An image is considered to be more intricate and complex than a graphic. These can be photos or drawings.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ? "
 comments="string [0..1]"
 orientation="float [0..1] ? "
 colourDepth=" pc:ColourDepthSimpleType [0..1] ? "
 bgColour=" pc:ColourSimpleType [0..1] ? "
 embText="boolean [0..1] ? ">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="ImageRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="colourDepth" type=" pc:ColourDepthSimpleType "
         use="optional"/>
         <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
         <attribute name="embText" type=" boolean " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: LayersType

Super-types:	None
Sub-types:	None

Name	LayersType
Abstract	no
Documentation	Can be used to express the z-index of overlapping regions. An element with a greater z-index is always in front of another element with lower z-index.

XML Instance Representation

```
<...>
   Start Sequence [1..*]
      <pc:Layer> pc:LayerType </pc:Layer> [1]
   End Sequence
</...>
```

Schema Component Representation

```
<complexType name="LayersType">
   <sequence minOccurs="1" maxOccurs="unbounded">
      <element name="Layer" type=" pc:LayerType "/>
   </sequence>
</complexType>
```

top

Complex Type: LayerType

Super-types:	None

Sub-types:	None

Name	LayerType
Abstract	no

XML Instance Representation

```
<...
 id="ID [1]"
 zIndex="int [1]"
 caption="string [0..1]">
   Start Sequence [1..*]
      <pc:RegionRef> pc:RegionRefType </pc:RegionRef> [1]
   End Sequence
</...>
```

Schema Component Representation

```
<complexType name="LayerType">
   <sequence minOccurs="1" maxOccurs="unbounded">
      <element name="RegionRef" type=" pc:RegionRefType "/>
   </sequence>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="zIndex" type=" int " use="required"/>
   <attribute name="caption" type=" string "/>
</complexType>
```

top

Complex Type: LineDrawingRegionType

Super-types:	RegionType < **LineDrawingRegionType** (by extension)
Sub-types:	None

Name	LineDrawingRegionType
Abstract	no
Documentation	A line drawing is a single colour illustration without solid areas.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ?"
 comments="string [0..1]"
 orientation="float [0..1] ?"
 penColour=" pc:ColourSimpleType [0..1] ?"
 bgColour=" pc:ColourSimpleType [0..1] ?"
 embText="boolean [0..1] ?">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="LineDrawingRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="penColour" type=" pc:ColourSimpleType " use="optional"/>
         <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
         <attribute name="embText" type=" boolean " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: MathsRegionType

Super-types:	RegionType < **MathsRegionType** (by extension)
Sub-types:	None

Name	MathsRegionType
Abstract	no
Documentation	Regions containing equations and mathematical symbols should be marked as maths regions.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1]  ? "
 comments="string [0..1]"
 orientation="float [0..1]  ? "
 bgColour=" pc:ColourSimpleType [0..1]  ? ">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="MathsRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
```

```
        <attribute name="orientation" type=" float " use="optional"/>
        <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: MetadataType

Super-types:	None
Sub-types:	None

Name	MetadataType
Abstract	no

XML Instance Representation

```
<...>
   <pc:Creator> string </pc:Creator> [1]
   <pc:Created> dateTime </pc:Created> [1] ?
   <pc:LastChange> dateTime </pc:LastChange> [1] ?
   <pc:Comments> string </pc:Comments> [0..1]
</...>
```

Schema Component Representation

```
<complexType name="MetadataType">
   <sequence>
      <element name="Creator" type=" string "/>
      <element name="Created" type=" dateTime "/>
      <element name="LastChange" type=" dateTime "/>
      <element name="Comments" type=" string " minOccurs="0" maxOccurs="1"/>
   </sequence>
</complexType>
```

top

Complex Type: MusicRegionType

Super-types:	RegionType < **MusicRegionType** (by extension)
Sub-types:	None

Name	MusicRegionType
Abstract	no
Documentation	Regions containing musical notations.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ?"
 comments="string [0..1]"
 orientation="float [0..1] ?"
 bgColour=" pc:ColourSimpleType [0..1] ?">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
     <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
     <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
     <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
     <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
     <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
     <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
     <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
     <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
     <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
     <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
     <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
     <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="MusicRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: NoiseRegionType

Super-types:	RegionType < **NoiseRegionType** (by extension)
Sub-types:	None

Name	NoiseRegionType
Abstract	no
Documentation	Noise regions are regions where no real data lies, only false data created by artifacts on the document or scanner noise.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1]  ? "
 comments="string [0..1]">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
```

```
  End Choice
</...>
```

Schema Component Representation

```
<complexType name="NoiseRegionType">
  <complexContent>
    <extension base=" pc:RegionType "/>
  </complexContent>
</complexType>
```

top

Complex Type: OrderedGroupIndexedType

Super-types:	None
Sub-types:	None

Name	OrderedGroupIndexedType
Abstract	no
Documentation	Indexed group containing ordered elements

XML Instance Representation

```
<...
 id="ID [1]"
 index="int [1] ?"
 caption="string [0..1]">
  Start Choice [1..*]
    <pc:RegionRefIndexed> pc:RegionRefIndexedType </pc:RegionRefIndexed> [1]
    <pc:OrderedGroupIndexed> pc:OrderedGroupIndexedType </pc:OrderedGroupIndexed>
    [1]
    <pc:UnorderedGroupIndexed> pc:UnorderedGroupIndexedType </
    pc:UnorderedGroupIndexed> [1]
  End Choice
</...>
```

Schema Component Representation

```
<complexType name="OrderedGroupIndexedType">
  <choice minOccurs="1" maxOccurs="unbounded">
    <element name="RegionRefIndexed" type=" pc:RegionRefIndexedType "/>
    <element name="OrderedGroupIndexed" type=" pc:OrderedGroupIndexedType "/>
    <element name="UnorderedGroupIndexed" type=" pc:UnorderedGroupIndexedType "/>
  </choice>
  <attribute name="id" type=" ID " use="required"/>
  <attribute name="index" type=" int " use="required"/>
  <attribute name="caption" type=" string "/>
</complexType>
```

top

Complex Type: OrderedGroupType

Super-types:	None
Sub-types:	None

Name	OrderedGroupType

Abstract	no
Documentation	Numbered group (contains ordered elements)

XML Instance Representation

```
<...
 id="ID [1]"
 caption="string [0..1]">
   Start Choice [1..*]
     <pc:RegionRefIndexed> pc:RegionRefIndexedType </pc:RegionRefIndexed> [1]
     <pc:OrderedGroupIndexed> pc:OrderedGroupIndexedType </pc:OrderedGroupIndexed>
     [1]
     <pc:UnorderedGroupIndexed> pc:UnorderedGroupIndexedType </
     pc:UnorderedGroupIndexed> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="OrderedGroupType">
   <choice minOccurs="1" maxOccurs="unbounded">
     <element name="RegionRefIndexed" type=" pc:RegionRefIndexedType "/>
     <element name="OrderedGroupIndexed" type=" pc:OrderedGroupIndexedType "/>
     <element name="UnorderedGroupIndexed" type=" pc:UnorderedGroupIndexedType "/>
   </choice>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="caption" type=" string "/>
</complexType>
```

top

Complex Type: PageType

Super-types:	None
Sub-types:	None

Name	PageType
Abstract	no

XML Instance Representation

```
<...
 imageFilename="string [1]"
 imageWidth="int [1]"
 imageHeight="int [1]"
 custom="string [0..1] ?"
 type=" pc:PageTypeSimpleType [0..1] ?">
   <pc:AlternativeImage> pc:AlternativeImageType </pc:AlternativeImage> [0..*] ?
   <pc:Border> pc:BorderType </pc:Border> [0..1]
   <pc:PrintSpace> pc:PrintSpaceType </pc:PrintSpace> [0..1]
   <pc:ReadingOrder> pc:ReadingOrderType </pc:ReadingOrder> [0..1] ?
   <pc:Layers> pc:LayersType </pc:Layers> [0..1] ?
   <pc:Relations> pc:RelationsType </pc:Relations> [0..1]
   Start Choice [0..*]
     <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
     <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
     <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
     <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
     <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
     <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
     <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
```

```
    <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
    <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
    <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
    <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
    <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
    <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
  End Choice
</...>
```

Schema Component Representation

```
<complexType name="PageType">
  <sequence>
    <element name="AlternativeImage" type=" pc:AlternativeImageType "
    minOccurs="0" maxOccurs="unbounded"/>
    <element name="Border" type=" pc:BorderType " minOccurs="0" maxOccurs="1"/>
    <element name="PrintSpace" type=" pc:PrintSpaceType " minOccurs="0"
    maxOccurs="1"/>
    <element name="ReadingOrder" type=" pc:ReadingOrderType " minOccurs="0"
    maxOccurs="1"/>
    <element name="Layers" type=" pc:LayersType " minOccurs="0" maxOccurs="1"/>
    <element name="Relations" type=" pc:RelationsType " minOccurs="0"/>
    <choice minOccurs="0" maxOccurs="unbounded">
      <element name="TextRegion" type=" pc:TextRegionType "/>
      <element name="ImageRegion" type=" pc:ImageRegionType "/>
      <element name="LineDrawingRegion" type=" pc:LineDrawingRegionType "/>
      <element name="GraphicRegion" type=" pc:GraphicRegionType "/>
      <element name="TableRegion" type=" pc:TableRegionType "/>
      <element name="ChartRegion" type=" pc:ChartRegionType "/>
      <element name="SeparatorRegion" type=" pc:SeparatorRegionType "/>
      <element name="MathsRegion" type=" pc:MathsRegionType "/>
      <element name="ChemRegion" type=" pc:ChemRegionType "/>
      <element name="MusicRegion" type=" pc:MusicRegionType "/>
      <element name="AdvertRegion" type=" pc:AdvertRegionType "/>
      <element name="NoiseRegion" type=" pc:NoiseRegionType "/>
      <element name="UnknownRegion" type=" pc:UnknownRegionType "/>
    </choice>
  </sequence>
  <attribute name="imageFilename" type=" string " use="required"/>
  <attribute name="imageWidth" type=" int " use="required"/>
  <attribute name="imageHeight" type=" int " use="required"/>
  <attribute name="custom" type=" string "/>
  <attribute name="type" type=" pc:PageTypeSimpleType "/>
</complexType>
```

top

Complex Type: PcGtsType

Super-types:	None
Sub-types:	None

Name	PcGtsType
Abstract	no

XML Instance Representation

```
<...
 pcGtsId="ID [0..1]">
  <pc:Metadata> pc:MetadataType </pc:Metadata> [1]
  <pc:Page> pc:PageType </pc:Page> [1]
```

```
</...>
```

Schema Component Representation

```
<complexType name="PcGtsType">
   <sequence>
      <element name="Metadata" type=" pc:MetadataType "/>
      <element name="Page" type=" pc:PageType "/>
   </sequence>
   <attribute name="pcGtsId" type=" ID " use="optional"/>
</complexType>
```

top

Complex Type: PrintSpaceType

Super-types:	None
Sub-types:	None

Name	PrintSpaceType
Abstract	no
Documentation	Determines the effective area on the paper of a printed page. Its size is equal for all pages of a book (exceptions: titlepage, multipage pictures). It contains all living elements (except marginals) like body type, footnotes, headings, running titles. It does not contain pagenumber (if not part of running title), marginals, signature mark, preview words.

XML Instance Representation

```
<...>
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
</...>
```

Schema Component Representation

```
<complexType name="PrintSpaceType">
   <sequence>
      <element name="Coords" type=" pc:CoordsType "/>
   </sequence>
</complexType>
```

top

Complex Type: ReadingOrderType

Super-types:	None
Sub-types:	None

Name	ReadingOrderType
Abstract	no
Documentation	Definition of the reading order within the page. To express a reading order between elements they have to be included in an OrderedGroup. Groups may contain further groups.

XML Instance Representation

```
<...>
   Start Choice [1]
```

```
    <pc:OrderedGroup> pc:OrderedGroupType </pc:OrderedGroup> [1]
    <pc:UnorderedGroup> pc:UnorderedGroupType </pc:UnorderedGroup> [1]
  End Choice
</...>
```

Schema Component Representation

```
<complexType name="ReadingOrderType">
  <choice minOccurs="1" maxOccurs="1">
    <element name="OrderedGroup" type=" pc:OrderedGroupType "/>
    <element name="UnorderedGroup" type=" pc:UnorderedGroupType "/>
  </choice>
</complexType>
```

top

Complex Type: RegionRefIndexedType

Super-types:	None
Sub-types:	None

Name	RegionRefIndexedType
Abstract	no
Documentation	Numbered region

XML Instance Representation

```
<...
 index="int [1] ? "
 regionRef="IDREF [1]"/>
```

Schema Component Representation

```
<complexType name="RegionRefIndexedType">
  <attribute name="index" type=" int " use="required"/>
  <attribute name="regionRef" type=" IDREF " use="required"/>
</complexType>
```

top

Complex Type: RegionRefType

Super-types:	None
Sub-types:	None

Name	RegionRefType
Abstract	no

XML Instance Representation

```
<...
 regionRef="IDREF [1]"/>
```

Schema Component Representation

```
<complexType name="RegionRefType">
  <attribute name="regionRef" type=" IDREF " use="required"/>
</complexType>
```

top

Complex Type: RegionType

Super-types: None

Sub-types:

- TextRegionType (by extension)
- ImageRegionType (by extension)
- LineDrawingRegionType (by extension)
- GraphicRegionType (by extension)
- TableRegionType (by extension)
- ChartRegionType (by extension)
- SeparatorRegionType (by extension)
- MathsRegionType (by extension)
- ChemRegionType (by extension)
- MusicRegionType (by extension)
- AdvertRegionType (by extension)
- NoiseRegionType (by extension)
- UnknownRegionType (by extension)

Name	RegionType
Abstract	yes

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1]  ? "
 comments="string [0..1]">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="RegionType" abstract="true">
   <sequence>
      <element name="Coords" type=" pc:CoordsType "/>
      <choice minOccurs="0" maxOccurs="unbounded">
         <element name="TextRegion" type=" pc:TextRegionType "/>
         <element name="ImageRegion" type=" pc:ImageRegionType "/>
         <element name="LineDrawingRegion" type=" pc:LineDrawingRegionType "/>
         <element name="GraphicRegion" type=" pc:GraphicRegionType "/>
         <element name="TableRegion" type=" pc:TableRegionType "/>
         <element name="ChartRegion" type=" pc:ChartRegionType "/>
         <element name="SeparatorRegion" type=" pc:SeparatorRegionType "/>
         <element name="MathsRegion" type=" pc:MathsRegionType "/>
         <element name="ChemRegion" type=" pc:ChemRegionType "/>
         <element name="MusicRegion" type=" pc:MusicRegionType "/>
```

```
        <element name="AdvertRegion" type=" pc:AdvertRegionType "/>
        <element name="NoiseRegion" type=" pc:NoiseRegionType "/>
        <element name="UnknownRegion" type=" pc:UnknownRegionType "/>
      </choice>
   </sequence>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="custom" type=" string "/>
   <attribute name="comments" type=" string "/>
</complexType>
```

top

Complex Type: RelationsType

Super-types:	None
Sub-types:	None

Name	RelationsType
Abstract	no
Documentation	Container for one-to-one relations between layout objects (for example: DropCap - paragraph, caption - image)

XML Instance Representation

```
<...>
   Start Sequence [1..*]
      <pc:Relation> pc:RelationType </pc:Relation> [1]
   End Sequence
</...>
```

Schema Component Representation

```
<complexType name="RelationsType">
   <sequence minOccurs="1" maxOccurs="unbounded">
      <element name="Relation" type=" pc:RelationType "/>
   </sequence>
</complexType>
```

top

Complex Type: RelationType

Super-types:	None
Sub-types:	None

Name	RelationType
Abstract	no
Documentation	One-to-one relation between to layout object. Use 'link' for loose relations and 'join' for strong relations (where something is fragmented for instance). Examples for 'link': caption - image floating - paragraph paragraph - paragraph (when a pragraph is split across columns and the last word of the first paragraph DOES NOT continue in the second paragraph) drop-cap - paragraph (when the drop-cap is a whole word) Examples for 'join': word - word (separated word at the end of a line) drop-cap - paragraph (when the drop-cap is not a whole word) paragraph - paragraph (when a pragraph is split across columns and the last word of the first paragraph DOES continue in the second paragraph)

XML Instance Representation

```
<...
 type=" string (value comes from list: {'link'|'join'}) [1]"
 custom="string [0..1] ? "
 comments="string [0..1]">
   Start Sequence [2..2]
      <pc:RegionRef> pc:RegionRefType </pc:RegionRef> [1]
   End Sequence
</...>
```

Schema Component Representation

```
<complexType name="RelationType">
   <sequence minOccurs="2" maxOccurs="2">
      <element name="RegionRef" type=" pc:RegionRefType "/>
   </sequence>
   <attribute name="type" use="required">
      <simpleType>
         <restriction base=" string ">
            <enumeration value="link"/>
            <enumeration value="join"/>
         </restriction>
      </simpleType>
   </attribute>
   <attribute name="custom" type=" string "/>
   <attribute name="comments" type=" string "/>
</complexType>
```

top

Complex Type: SeparatorRegionType

Super-types:	RegionType < **SeparatorRegionType** (by extension)
Sub-types:	None

Name	SeparatorRegionType
Abstract	no
Documentation	Separators are lines that lie between columns and paragraphs and can be used to logically separate different articles from each other.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ? "
 comments="string [0..1]"
 orientation="float [0..1] ? "
 colour=" pc:ColourSimpleType [0..1] ? ">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
```

```
     <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
     <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
  End Choice
</...>
```

Schema Component Representation

```
<complexType name="SeparatorRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="colour" type=" pc:ColourSimpleType " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: TableRegionType

Super-types:	RegionType < **TableRegionType** (by extension)
Sub-types:	None

Name	TableRegionType
Abstract	no
Documentation	Tabular data in any form is represented with a table region. Rows and columns may or may not have separator lines; these lines are not separator regions.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ?"
 comments="string [0..1]"
 orientation="float [0..1] ?"
 rows="int [0..1] ?"
 columns="int [0..1] ?"
 lineColour=" pc:ColourSimpleType [0..1] ?"
 bgColour=" pc:ColourSimpleType [0..1] ?"
 lineSeparators="boolean [0..1] ?"
 embText="boolean [0..1] ?">
  <pc:Coords> pc:CoordsType </pc:Coords> [1]
  Start Choice [0..*]
     <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
     <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
     <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
     <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
     <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
     <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
     <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
     <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
     <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
     <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
     <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
     <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
     <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
  End Choice
</...>
```

Schema Component Representation

```
<complexType name="TableRegionType">
   <complexContent>
      <extension base=" pc:RegionType ">
         <attribute name="orientation" type=" float " use="optional"/>
         <attribute name="rows" type=" int " use="optional"/>
         <attribute name="columns" type=" int " use="optional"/>
         <attribute name="lineColour" type=" pc:ColourSimpleType " use="optional"/>
         <attribute name="bgColour" type=" pc:ColourSimpleType " use="optional"/>
         <attribute name="lineSeparators" type=" boolean " use="optional"/>
         <attribute name="embText" type=" boolean " use="optional"/>
      </extension>
   </complexContent>
</complexType>
```

top

Complex Type: TextEquivType

Super-types:	None
Sub-types:	None

Name	TextEquivType
Abstract	no

XML Instance Representation

```
<...
 conf=" float (0 < value < 1) [0..1] ? ">
   <pc:PlainText> string </pc:PlainText> [0..1] ?
   <pc:Unicode> string </pc:Unicode> [1] ?
</...>
```

Schema Component Representation

```
<complexType name="TextEquivType">
   <sequence>
      <element name="PlainText" type=" string " minOccurs="0"/>
      <element name="Unicode" type=" string "/>
   </sequence>
   <attribute name="conf">
      <simpleType>
         <restriction base=" float ">
            <minExclusive value="0"/>
            <maxExclusive value="1"/>
         </restriction>
      </simpleType>
   </attribute>
</complexType>
```

top

Complex Type: TextLineType

Super-types:	None
Sub-types:	None

Name	TextLineType

Abstract	no

XML Instance Representation

```
<...
 id="ID [1]"
 primaryLanguage=" pc:LanguageSimpleType [0..1] ? "
 production=" pc:ProductionSimpleType [0..1] ? "
 custom="string [0..1] ? "
 comments="string [0..1]">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   <pc:Baseline> pc:BaselineType </pc:Baseline> [0..1] ?
   <pc:Word> pc:WordType </pc:Word> [0..*]
   <pc:TextEquiv> pc:TextEquivType </pc:TextEquiv> [0..1]
   <pc:TextStyle> pc:TextStyleType </pc:TextStyle> [0..1]
</...>
```

Schema Component Representation

```
<complexType name="TextLineType">
   <sequence>
      <element name="Coords" type=" pc:CoordsType "/>
      <element name="Baseline" type=" pc:BaselineType " minOccurs="0"/>
      <element name="Word" type=" pc:WordType " minOccurs="0"
      maxOccurs="unbounded"/>
      <element name="TextEquiv" type=" pc:TextEquivType " minOccurs="0"
      maxOccurs="1"/>
      <element name="TextStyle" type=" pc:TextStyleType " minOccurs="0"/>
   </sequence>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="primaryLanguage" type=" pc:LanguageSimpleType "/>
   <attribute name="production" type=" pc:ProductionSimpleType "/>
   <attribute name="custom" type=" string "/>
   <attribute name="comments" type=" string "/>
</complexType>
```

top

Complex Type: TextRegionType

Super-types:	RegionType < **TextRegionType** (by extension)
Sub-types:	None

Name	TextRegionType
Abstract	no
Documentation	Pure text is represented as a text region. This includes drop capitals, but practically ornate text may be considered as a graphic.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ? "
 comments="string [0..1]"
 orientation="float [0..1] ? "
 type=" pc:TextTypeSimpleType [0..1] ? "
 leading="int [0..1] ? "
 readingDirection=" pc:ReadingDirectionSimpleType [0..1] ? "
 readingOrientation="float [0..1] ? "
 indented="boolean [0..1] ? "
 align=" pc:AlignSimpleType [0..1] ? "
```

```
primaryLanguage=" pc:LanguageSimpleType [0..1]  ? "
secondaryLanguage=" pc:LanguageSimpleType [0..1]  ? "
primaryScript=" pc:ScriptSimpleType [0..1]  ? "
secondaryScript=" pc:ScriptSimpleType [0..1]  ? "
production=" pc:ProductionSimpleType [0..1]">
  <pc:Coords> pc:CoordsType </pc:Coords> [1]
  Start Choice [0..*]
    <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
    <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
    <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
    <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
    <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
    <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
    <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
    <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
    <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
    <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
    <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
    <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
    <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
  End Choice
  <pc:TextLine> pc:TextLineType </pc:TextLine> [0..*]
  <pc:TextEquiv> pc:TextEquivType </pc:TextEquiv> [0..1]
  <pc:TextStyle> pc:TextStyleType </pc:TextStyle> [0..1]
</...>
```

Schema Component Representation

```
<complexType name="TextRegionType">
  <complexContent>
    <extension base=" pc:RegionType ">
      <sequence>
        <element name="TextLine" type=" pc:TextLineType " minOccurs="0"
        maxOccurs="unbounded"/>
        <element name="TextEquiv" type=" pc:TextEquivType " minOccurs="0"
        maxOccurs="1"/>
        <element name="TextStyle" type=" pc:TextStyleType " minOccurs="0"
        maxOccurs="1"/>
      </sequence>
      <attribute name="orientation" type=" float " use="optional"/>
      <attribute name="type" type=" pc:TextTypeSimpleType " use="optional"/>
      <attribute name="leading" type=" int " use="optional"/>
      <attribute name="readingDirection" type=" pc:ReadingDirectionSimpleType "
      use="optional"/>
      <attribute name="readingOrientation" type=" float " use="optional"/>
      <attribute name="indented" type=" boolean " use="optional"/>
      <attribute name="align" type=" pc:AlignSimpleType "/>
      <attribute name="primaryLanguage" type=" pc:LanguageSimpleType "
      use="optional"/>
      <attribute name="secondaryLanguage" type=" pc:LanguageSimpleType "
      use="optional"/>
      <attribute name="primaryScript" type=" pc:ScriptSimpleType "
      use="optional"/>
      <attribute name="secondaryScript" type=" pc:ScriptSimpleType "
      use="optional"/>
      <attribute name="production" type=" pc:ProductionSimpleType "/>
    </extension>
  </complexContent>
</complexType>
```

top

Complex Type: TextStyleType

Super-types:	None
Sub-types:	None

Name	TextStyleType
Abstract	no
Documentation	Monospace (fixed-pitch, non-proportional) or proportional font

XML Instance Representation

```
<...
 fontFamily="string [0..1]  ? "
 serif="boolean [0..1]  ? "
 monospace="boolean [0..1]"
 fontSize="float [0..1]  ? "
 kerning="int [0..1]  ? "
 textColour=" pc:ColourSimpleType [0..1]"
 bgColour=" pc:ColourSimpleType [0..1]  ? "
 reverseVideo="boolean [0..1]  ? "
 bold="boolean [0..1]"
 italic="boolean [0..1]"
 underlined="boolean [0..1]"
 subscript="boolean [0..1]"
 superscript="boolean [0..1]"
 strikethrough="boolean [0..1]"
 smallCaps="boolean [0..1]"
 letterSpaced="boolean [0..1]"/>
```

Schema Component Representation

```
<complexType name="TextStyleType">
   <attribute name="fontFamily" type=" string "/>
   <attribute name="serif" type=" boolean "/>
   <attribute name="monospace" type=" boolean "/>
   <attribute name="fontSize" type=" float "/>
   <attribute name="kerning" type=" int "/>
   <attribute name="textColour" type=" pc:ColourSimpleType "/>
   <attribute name="bgColour" type=" pc:ColourSimpleType "/>
   <attribute name="reverseVideo" type=" boolean "/>
   <attribute name="bold" type=" boolean "/>
   <attribute name="italic" type=" boolean "/>
   <attribute name="underlined" type=" boolean "/>
   <attribute name="subscript" type=" boolean "/>
   <attribute name="superscript" type=" boolean "/>
   <attribute name="strikethrough" type=" boolean "/>
   <attribute name="smallCaps" type=" boolean "/>
   <attribute name="letterSpaced" type=" boolean "/>
</complexType>
```

top

Complex Type: UnknownRegionType

Super-types:	RegionType < **UnknownRegionType** (by extension)
Sub-types:	None

Name	UnknownRegionType
Abstract	no

Documentation	To be used if the region type cannot be ascertained.

XML Instance Representation

```
<...
 id="ID [1]"
 custom="string [0..1] ? "
 comments="string [0..1]">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   Start Choice [0..*]
      <pc:TextRegion> pc:TextRegionType </pc:TextRegion> [1]
      <pc:ImageRegion> pc:ImageRegionType </pc:ImageRegion> [1]
      <pc:LineDrawingRegion> pc:LineDrawingRegionType </pc:LineDrawingRegion> [1]
      <pc:GraphicRegion> pc:GraphicRegionType </pc:GraphicRegion> [1]
      <pc:TableRegion> pc:TableRegionType </pc:TableRegion> [1]
      <pc:ChartRegion> pc:ChartRegionType </pc:ChartRegion> [1]
      <pc:SeparatorRegion> pc:SeparatorRegionType </pc:SeparatorRegion> [1]
      <pc:MathsRegion> pc:MathsRegionType </pc:MathsRegion> [1]
      <pc:ChemRegion> pc:ChemRegionType </pc:ChemRegion> [1]
      <pc:MusicRegion> pc:MusicRegionType </pc:MusicRegion> [1]
      <pc:AdvertRegion> pc:AdvertRegionType </pc:AdvertRegion> [1]
      <pc:NoiseRegion> pc:NoiseRegionType </pc:NoiseRegion> [1]
      <pc:UnknownRegion> pc:UnknownRegionType </pc:UnknownRegion> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="UnknownRegionType">
   <complexContent>
      <extension base=" pc:RegionType "/>
   </complexContent>
</complexType>
```

top

Complex Type: UnorderedGroupIndexedType

Super-types:	None
Sub-types:	None

Name	UnorderedGroupIndexedType
Abstract	no
Documentation	Indexed group containing unordered elements

XML Instance Representation

```
<...
 id="ID [1]"
 index="int [1] ? "
 caption="string [0..1]">
   Start Choice [1..*]
      <pc:RegionRef> pc:RegionRefType </pc:RegionRef> [1]
      <pc:OrderedGroup> pc:OrderedGroupType </pc:OrderedGroup> [1]
      <pc:UnorderedGroup> pc:UnorderedGroupType </pc:UnorderedGroup> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="UnorderedGroupIndexedType">
   <choice minOccurs="1" maxOccurs="unbounded">
```

```
      <element name="RegionRef" type=" pc:RegionRefType "/>
      <element name="OrderedGroup" type=" pc:OrderedGroupType "/>
      <element name="UnorderedGroup" type=" pc:UnorderedGroupType "/>
   </choice>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="index" type=" int " use="required"/>
   <attribute name="caption" type=" string "/>
</complexType>
```

top

Complex Type: UnorderedGroupType

Super-types:	None
Sub-types:	None

Name	UnorderedGroupType
Abstract	no
Documentation	Numbered group (contains unordered elements)

XML Instance Representation

```
<...
 id="ID [1]"
 caption="string [0..1]">
   Start Choice [1..*]
      <pc:RegionRef> pc:RegionRefType </pc:RegionRef> [1]
      <pc:OrderedGroup> pc:OrderedGroupType </pc:OrderedGroup> [1]
      <pc:UnorderedGroup> pc:UnorderedGroupType </pc:UnorderedGroup> [1]
   End Choice
</...>
```

Schema Component Representation

```
<complexType name="UnorderedGroupType">
   <choice minOccurs="1" maxOccurs="unbounded">
      <element name="RegionRef" type=" pc:RegionRefType "/>
      <element name="OrderedGroup" type=" pc:OrderedGroupType "/>
      <element name="UnorderedGroup" type=" pc:UnorderedGroupType "/>
   </choice>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="caption" type=" string "/>
</complexType>
```

top

Complex Type: WordType

Super-types:	None
Sub-types:	None

Name	WordType
Abstract	no

XML Instance Representation

```
<...
 id="ID [1]"
 language=" pc:LanguageSimpleType [0..1] ? "
 production=" pc:ProductionSimpleType [0..1]  ? "
 custom="string [0..1]  ? "
 comments="string [0..1]">
   <pc:Coords> pc:CoordsType </pc:Coords> [1]
   <pc:Glyph> pc:GlyphType </pc:Glyph> [0..*]
   <pc:TextEquiv> pc:TextEquivType </pc:TextEquiv> [0..1]
   <pc:TextStyle> pc:TextStyleType </pc:TextStyle> [0..1]
</...>
```

Schema Component Representation

```
<complexType name="WordType">
   <sequence>
      <element name="Coords" type=" pc:CoordsType "/>
      <element name="Glyph" type=" pc:GlyphType " minOccurs="0"
      maxOccurs="unbounded"/>
      <element name="TextEquiv" type=" pc:TextEquivType " minOccurs="0"
      maxOccurs="1"/>
      <element name="TextStyle" type=" pc:TextStyleType " minOccurs="0"/>
   </sequence>
   <attribute name="id" type=" ID " use="required"/>
   <attribute name="language" type=" pc:LanguageSimpleType "/>
   <attribute name="production" type=" pc:ProductionSimpleType "/>
   <attribute name="custom" type=" string "/>
   <attribute name="comments" type=" string "/>
</complexType>
```

top

Simple Type: AlignSimpleType

Super-types:	string < **AlignSimpleType** (by restriction)
Sub-types:	None

Name	AlignSimpleType
Content	• Base XSD Type: string • *value* comes from list: {'left'|'centre'|'right'|'justify'}

Schema Component Representation

```
<simpleType name="AlignSimpleType">
   <restriction base=" string ">
      <enumeration value="left"/>
      <enumeration value="centre"/>
      <enumeration value="right"/>
      <enumeration value="justify"/>
   </restriction>
</simpleType>
```

top

Simple Type: ChartTypeSimpleType

Super-types:	string < **ChartTypeSimpleType** (by restriction)
Sub-types:	None

Name	ChartTypeSimpleType

Content	• Base XSD Type: string • *value* comes from list: {'bar'|'line'|'pie'|'scatter'|'surface'|'other'}

Schema Component Representation

```
<simpleType name="ChartTypeSimpleType">
   <restriction base=" string ">
      <enumeration value="bar"/>
      <enumeration value="line"/>
      <enumeration value="pie"/>
      <enumeration value="scatter"/>
      <enumeration value="surface"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: ColourDepthSimpleType

Super-types:	string < **ColourDepthSimpleType** (by restriction)
Sub-types:	None

Name	ColourDepthSimpleType
Content	• Base XSD Type: string • *value* comes from list: {'bilevel'|'greyscale'|'colour'|'other'}

Schema Component Representation

```
<simpleType name="ColourDepthSimpleType">
   <restriction base=" string ">
      <enumeration value="bilevel"/>
      <enumeration value="greyscale"/>
      <enumeration value="colour"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: ColourSimpleType

Super-types:	string < **ColourSimpleType** (by restriction)
Sub-types:	None

Name	ColourSimpleType
Content	• Base XSD Type: string • *value* comes from list: {'black'|'blue'|'brown'|'cyan'|'green'|'grey'|'indigo'|'magenta'|'orange'|'pink'|'red'|'turquoise'|'violet'|'white'|'yellow'|'other'}

Schema Component Representation

```
<simpleType name="ColourSimpleType">
```

```
   <restriction base=" string ">
      <enumeration value="black"/>
      <enumeration value="blue"/>
      <enumeration value="brown"/>
      <enumeration value="cyan"/>
      <enumeration value="green"/>
      <enumeration value="grey"/>
      <enumeration value="indigo"/>
      <enumeration value="magenta"/>
      <enumeration value="orange"/>
      <enumeration value="pink"/>
      <enumeration value="red"/>
      <enumeration value="turquoise"/>
      <enumeration value="violet"/>
      <enumeration value="white"/>
      <enumeration value="yellow"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: GraphicsTypeSimpleType

Super-types:	string < **GraphicsTypeSimpleType** (by restriction)
Sub-types:	None

Name	GraphicsTypeSimpleType
Content	• Base XSD Type: string • *value* comes from list: {'logo'|'letterhead'|'decoration'|'frame'|'handwritten-annotation'|'stamp'|'signature'|'barcode'|'paper-grow'|'punch-hole'|'other'}

Schema Component Representation

```
<simpleType name="GraphicsTypeSimpleType">
   <restriction base=" string ">
      <enumeration value="logo"/>
      <enumeration value="letterhead"/>
      <enumeration value="decoration"/>
      <enumeration value="frame"/>
      <enumeration value="handwritten-annotation"/>
      <enumeration value="stamp"/>
      <enumeration value="signature"/>
      <enumeration value="barcode"/>
      <enumeration value="paper-grow"/>
      <enumeration value="punch-hole"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: LanguageSimpleType

Super-types:	string < **LanguageSimpleType** (by restriction)

Sub-types:	None

<table>
<tr><td><b>Name</b></td><td>LanguageSimpleType</td></tr>
<tr><td><b>Content</b></td><td>• Base XSD Type: string
• <i>value</i> comes from list: {'Abkhaz'|'Afar'|'Afrikaans'|'Akan'|'Albanian'|'Amharic'|'Arabic'|'Aragonese'|'Armenian'|'Assamese'|'Avaric'|'Avestan'|'Aymara'|'Azerbaijani'|'Bambara'|'Bashkir'|'Basque'|'Belarusian'|'Bengali'|'Bihari'|'Bislama'|'Bosnian'|'Breton'|'Bulgarian'|'Burmese'|'Cambodian'|'Cantonese'|'Catalan'|'Chamorro'|'Chechen'|'Chichewa'|'Chinese'|'Chuvash'|'Cornish'|'Corsican'|'Cree'|'Croatian'|'Czech'|'Danish'|'Divehi'|'Dutch'|'Dzongkha'|'English'|'Esperanto'|'Estonian'|'Ewe'|'Faroese'|'Fijian'|'Finnish'|'French'|'Fula'|'Gaelic'|'Galician'|'Ganda'|'Georgian'|'German'|'Greek'|'Guaraní'|'Gujarati'|'Haitian'|'Hausa'|'Hebrew'|'Herero'|'Hindi'|'Hiri Motu'|'Hungarian'|'Icelandic'|'Ido'|'Igbo'|'Indonesian'|'Interlingua'|'Interlingue'|'Inuktitut'|'Inupiaq'|'Irish'|'Italian'|'Japanese'|'Javanese'|'Kalaallisut'|'Kannada'|'Kanuri'|'Kashmiri'|'Kazakh'|'Khmer'|'Kikuyu'|'Kinyarwanda'|'Kirundi'|'Komi'|'Kongo'|'Korean'|'Kurdish'|'Kwanyama'|'Kyrgyz'|'Lao'|'Latin'|'Latvian'|'Limburgish'|'Lingala'|'Lithuanian'|'Luba-Katanga'|'Luxembourgish'|'Macedonian'|'Malagasy'|'Malay'|'Malayalam'|'Maltese'|'Manx'|'Māori'|'Marathi'|'Marshallese'|'Mongolian'|'Nauru'|'Navajo'|'Ndonga'|'Nepali'|'North Ndebele'|'Northern Sami'|'Norwegian'|'Norwegian Bokmål'|'Norwegian Nynorsk'|'Nuosu'|'Occitan'|'Ojibwe'|'Old Church Slavonic'|'Oriya'|'Oromo'|'Ossetian'|'Pāli'|'Panjabi'|'Pashto'|'Persian'|'Polish'|'Portuguese'|'Punjabi'|'Quechua'|'Romanian'|'Romansh'|'Russian'|'Samoan'|'Sango'|'Sanskrit'|'Sardinian'|'Serbian'|'Shona'|'Sindhi'|'Sinhala'|'Slovak'|'Slovene'|'Somali'|'South Ndebele'|'Southern Sotho'|'Spanish'|'Sundanese'|'Swahili'|'Swati'|'Swedish'|'Tagalog'|'Tahitian'|'Tajik'|'Tamil'|'Tatar'|'Telugu'|'Thai'|'Tibetan'|'Tigrinya'|'Tonga'|'Tsonga'|'Tswana'|'Turkish'|'Turkmen'|'Twi'|'Uighur'|'Ukrainian'|'Urdu'|'Uzbek'|'Venda'|'Vietnamese'|'Volapük'|'Walloon'|'Welsh'|'Western Frisian'|'Wolof'|'Xhosa'|'Yiddish'|'Yoruba'|'Zhuang'|'Zulu'|'other'}</td></tr>
</table>

Schema Component Representation

```
<simpleType name="LanguageSimpleType">
  <restriction base=" string ">
    <enumeration value="Abkhaz"/>
    <enumeration value="Afar"/>
    <enumeration value="Afrikaans"/>
    <enumeration value="Akan"/>
    <enumeration value="Albanian"/>
    <enumeration value="Amharic"/>
    <enumeration value="Arabic"/>
    <enumeration value="Aragonese"/>
    <enumeration value="Armenian"/>
    <enumeration value="Assamese"/>
    <enumeration value="Avaric"/>
    <enumeration value="Avestan"/>
    <enumeration value="Aymara"/>
    <enumeration value="Azerbaijani"/>
    <enumeration value="Bambara"/>
    <enumeration value="Bashkir"/>
    <enumeration value="Basque"/>
    <enumeration value="Belarusian"/>
    <enumeration value="Bengali"/>
    <enumeration value="Bihari"/>
    <enumeration value="Bislama"/>
    <enumeration value="Bosnian"/>
    <enumeration value="Breton"/>
    <enumeration value="Bulgarian"/>
    <enumeration value="Burmese"/>
    <enumeration value="Cambodian"/>
    <enumeration value="Cantonese"/>
    <enumeration value="Catalan"/>
    <enumeration value="Chamorro"/>
    <enumeration value="Chechen"/>
    <enumeration value="Chichewa"/>
```

```
      <enumeration value="Chinese"/>
      <enumeration value="Chuvash"/>
      <enumeration value="Cornish"/>
      <enumeration value="Corsican"/>
      <enumeration value="Cree"/>
      <enumeration value="Croatian"/>
      <enumeration value="Czech"/>
      <enumeration value="Danish"/>
      <enumeration value="Divehi"/>
      <enumeration value="Dutch"/>
      <enumeration value="Dzongkha"/>
      <enumeration value="English"/>
      <enumeration value="Esperanto"/>
      <enumeration value="Estonian"/>
      <enumeration value="Ewe"/>
      <enumeration value="Faroese"/>
      <enumeration value="Fijian"/>
      <enumeration value="Finnish"/>
      <enumeration value="French"/>
      <enumeration value="Fula"/>
      <enumeration value="Gaelic"/>
      <enumeration value="Galician"/>
      <enumeration value="Ganda"/>
      <enumeration value="Georgian"/>
      <enumeration value="German"/>
      <enumeration value="Greek"/>
      <enumeration value="Guaraní"/>
      <enumeration value="Gujarati"/>
      <enumeration value="Haitian"/>
      <enumeration value="Hausa"/>
      <enumeration value="Hebrew"/>
      <enumeration value="Herero"/>
      <enumeration value="Hindi"/>
      <enumeration value="Hiri Motu"/>
      <enumeration value="Hungarian"/>
      <enumeration value="Icelandic"/>
      <enumeration value="Ido"/>
      <enumeration value="Igbo"/>
      <enumeration value="Indonesian"/>
      <enumeration value="Interlingua"/>
      <enumeration value="Interlingue"/>
      <enumeration value="Inuktitut"/>
      <enumeration value="Inupiaq"/>
      <enumeration value="Irish"/>
      <enumeration value="Italian"/>
      <enumeration value="Japanese"/>
      <enumeration value="Javanese"/>
      <enumeration value="Kalaallisut"/>
      <enumeration value="Kannada"/>
      <enumeration value="Kanuri"/>
      <enumeration value="Kashmiri"/>
      <enumeration value="Kazakh"/>
      <enumeration value="Khmer"/>
      <enumeration value="Kikuyu"/>
      <enumeration value="Kinyarwanda"/>
      <enumeration value="Kirundi"/>
      <enumeration value="Komi"/>
      <enumeration value="Kongo"/>
      <enumeration value="Korean"/>
      <enumeration value="Kurdish"/>
      <enumeration value="Kwanyama"/>
      <enumeration value="Kyrgyz"/>
      <enumeration value="Lao"/>
      <enumeration value="Latin"/>
      <enumeration value="Latvian"/>
      <enumeration value="Limburgish"/>
      <enumeration value="Lingala"/>
      <enumeration value="Lithuanian"/>
      <enumeration value="Luba-Katanga"/>
      <enumeration value="Luxembourgish"/>
      <enumeration value="Macedonian"/>
      <enumeration value="Malagasy"/>
      <enumeration value="Malay"/>
      <enumeration value="Malayalam"/>
      <enumeration value="Maltese"/>
      <enumeration value="Manx"/>
      <enumeration value="Māori"/>
      <enumeration value="Marathi"/>
      <enumeration value="Marshallese"/>
      <enumeration value="Mongolian"/>
      <enumeration value="Nauru"/>
      <enumeration value="Navajo"/>
      <enumeration value="Ndonga"/>
      <enumeration value="Nepali"/>
      <enumeration value="North Ndebele"/>
      <enumeration value="Northern Sami"/>
      <enumeration value="Norwegian"/>
      <enumeration value="Norwegian Bokmål"/>
      <enumeration value="Norwegian Nynorsk"/>
      <enumeration value="Nuosu"/>
      <enumeration value="Occitan"/>
      <enumeration value="Ojibwe"/>
      <enumeration value="Old Church Slavonic"/>
      <enumeration value="Oriya"/>
      <enumeration value="Oromo"/>
      <enumeration value="Ossetian"/>
      <enumeration value="Pāli"/>
      <enumeration value="Panjabi"/>
      <enumeration value="Pashto"/>
      <enumeration value="Persian"/>
      <enumeration value="Polish"/>
      <enumeration value="Portuguese"/>
      <enumeration value="Punjabi"/>
      <enumeration value="Quechua"/>
      <enumeration value="Romanian"/>
      <enumeration value="Romansh"/>
      <enumeration value="Russian"/>
      <enumeration value="Samoan"/>
      <enumeration value="Sango"/>
      <enumeration value="Sanskrit"/>
      <enumeration value="Sardinian"/>
      <enumeration value="Serbian"/>
      <enumeration value="Shona"/>
      <enumeration value="Sindhi"/>
      <enumeration value="Sinhala"/>
      <enumeration value="Slovak"/>
      <enumeration value="Slovene"/>
      <enumeration value="Somali"/>
      <enumeration value="South Ndebele"/>
      <enumeration value="Southern Sotho"/>
      <enumeration value="Spanish"/>
      <enumeration value="Sundanese"/>
      <enumeration value="Swahili"/>
      <enumeration value="Swati"/>
      <enumeration value="Swedish"/>
      <enumeration value="Tagalog"/>
      <enumeration value="Tahitian"/>
      <enumeration value="Tajik"/>
      <enumeration value="Tamil"/>
      <enumeration value="Tatar"/>
      <enumeration value="Telugu"/>
      <enumeration value="Thai"/>
      <enumeration value="Tibetan"/>
      <enumeration value="Tigrinya"/>
      <enumeration value="Tonga"/>
      <enumeration value="Tsonga"/>
      <enumeration value="Tswana"/>
      <enumeration value="Turkish"/>
      <enumeration value="Turkmen"/>
      <enumeration value="Twi"/>
      <enumeration value="Uighur"/>
      <enumeration value="Ukrainian"/>
      <enumeration value="Urdu"/>
      <enumeration value="Uzbek"/>
      <enumeration value="Venda"/>
      <enumeration value="Vietnamese"/>
      <enumeration value="Volapük"/>
      <enumeration value="Walloon"/>
      <enumeration value="Welsh"/>
      <enumeration value="Western Frisian"/>
      <enumeration value="Wolof"/>
      <enumeration value="Xhosa"/>
      <enumeration value="Yiddish"/>
      <enumeration value="Yoruba"/>
      <enumeration value="Zhuang"/>
      <enumeration value="Zulu"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: PageTypeSimpleType

Super-types:	string < **PageTypeSimpleType** (by restriction)
Sub-types:	None

Name	PageTypeSimpleType
Content	• Base XSD Type: string • *value* comes from list: {'front-cover'|'back-cover'|'title'|'table-of-contents'|'index'|'content'|'blank'|'other'}

Schema Component Representation

```
<simpleType name="PageTypeSimpleType">
   <restriction base=" string ">
      <enumeration value="front-cover"/>
      <enumeration value="back-cover"/>
      <enumeration value="title"/>
      <enumeration value="table-of-contents"/>
      <enumeration value="index"/>
      <enumeration value="content"/>
      <enumeration value="blank"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: PointsType

Super-types:	string < **PointsType** (by restriction)
Sub-types:	None

Name	PointsType
Content	• Base XSD Type: string • *pattern* = ([0-9]+,[0-9]+)+([0-9]+,[0-9]+)
Documentation	Point list with format "x1,y1 x2,y2 ..."

Schema Component Representation

```
<simpleType name="PointsType">
   <restriction base=" string ">
      <pattern value="([0-9]+,[0-9]+ )+([0-9]+,[0-9]+)"/>
   </restriction>
</simpleType>
```

top

Simple Type: ProductionSimpleType

Super-types:	string < **ProductionSimpleType** (by restriction)
Sub-types:	None

Name	ProductionSimpleType
Content	• Base XSD Type: string • *value* comes from list: {'printed'|'typewritten'|'handwritten-cursive'|'handwritten-printscript'|'medieval-manuscript'|'other'}
Documentation	Text production type

Schema Component Representation

```
<simpleType name="ProductionSimpleType">
   <restriction base=" string ">
      <enumeration value="printed"/>
      <enumeration value="typewritten"/>
      <enumeration value="handwritten-cursive"/>
      <enumeration value="handwritten-printscript"/>
      <enumeration value="medieval-manuscript"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: ReadingDirectionSimpleType

Super-types:	string < **ReadingDirectionSimpleType** (by restriction)
Sub-types:	None

Name	ReadingDirectionSimpleType

Content	- Base XSD Type: string - *value* comes from list: {'left-to-right'|'right-to-left'|'top-to-bottom'|'bottom-to-top'}

Schema Component Representation

```
<simpleType name="ReadingDirectionSimpleType">
   <restriction base=" string ">
      <enumeration value="left-to-right"/>
      <enumeration value="right-to-left"/>
      <enumeration value="top-to-bottom"/>
      <enumeration value="bottom-to-top"/>
   </restriction>
</simpleType>
```

top

Simple Type: ScriptSimpleType

Super-types:	string < **ScriptSimpleType** (by restriction)
Sub-types:	None

Name	ScriptSimpleType
Content	- Base XSD Type: string - *value* comes from list: {'Arabic'|'Bengali'|'Chinese-simplified'|'Chinese-traditional'|'Cyrillic'|'Devangari'|'Ethiopic'|'Greek'|'Gujarati'|'Gurmukhi'|'Hebrew'|'Latin'|'Thai'|'other'}

Schema Component Representation

```
<simpleType name="ScriptSimpleType">
   <restriction base=" string ">
      <enumeration value="Arabic"/>
      <enumeration value="Bengali"/>
      <enumeration value="Chinese-simplified"/>
      <enumeration value="Chinese-traditional"/>
      <enumeration value="Cyrillic"/>
      <enumeration value="Devangari"/>
      <enumeration value="Ethiopic"/>
      <enumeration value="Greek"/>
      <enumeration value="Gujarati"/>
      <enumeration value="Gurmukhi"/>
      <enumeration value="Hebrew"/>
      <enumeration value="Latin"/>
      <enumeration value="Thai"/>
      <enumeration value="other"/>
   </restriction>
</simpleType>
```

top

Simple Type: TextTypeSimpleType

Super-types:	string < **TextTypeSimpleType** (by restriction)
Sub-types:	None

Name	TextTypeSimpleType

Content

- Base XSD Type: string

- *value* comes from list: {'paragraph'|'heading'|'caption'|'header'|'footer'|'page-number'|'drop-capital'|'credit'|'floating'|'signature-mark'|'catch-word'|'marginalia'|'footnote'|'footnote-continued'|'endnote'|'TOC-entry'|'other'}

Schema Component Representation

```
<simpleType name="TextTypeSimpleType">
  <restriction base=" string ">
    <enumeration value="paragraph"/>
    <enumeration value="heading"/>
    <enumeration value="caption"/>
    <enumeration value="header"/>
    <enumeration value="footer"/>
    <enumeration value="page-number"/>
    <enumeration value="drop-capital"/>
    <enumeration value="credit"/>
    <enumeration value="floating"/>
    <enumeration value="signature-mark"/>
    <enumeration value="catch-word"/>
    <enumeration value="marginalia"/>
    <enumeration value="footnote"/>
    <enumeration value="footnote-continued"/>
    <enumeration value="endnote"/>
    <enumeration value="TOC-entry"/>
    <enumeration value="other"/>
  </restriction>
</simpleType>
```

top

Generated by xs3p (old link) . Last modified: 12/16/2013 16:22:03