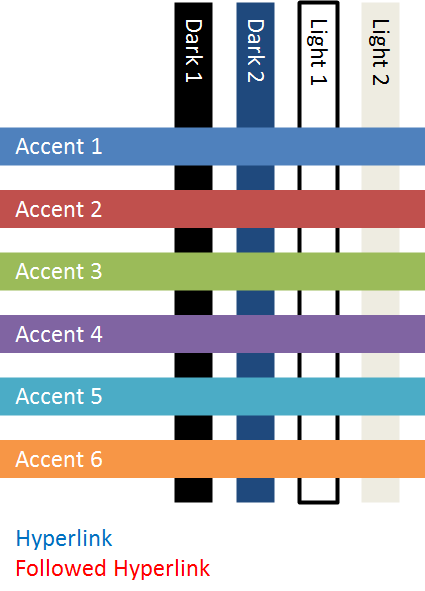
#### [Styles](Styles.docx)

The elements in this section compose the basic definition of a style, including its associated colors, [effect](effect.docx) styles, line styles, [fill](fill.docx) styles, background styles, and font scheme.

##### accent1 (Accent 1)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

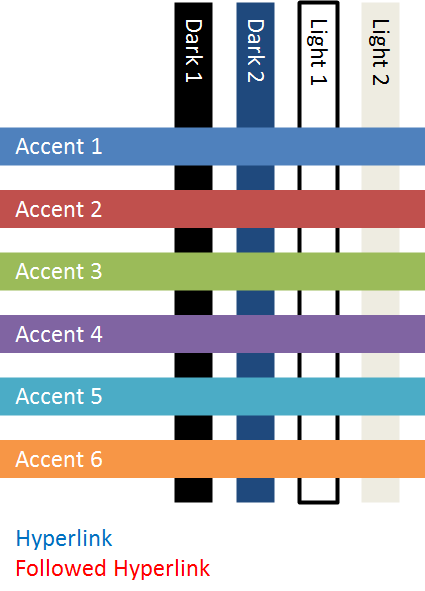
</sequence>

</complexType>

##### accent2 (Accent 2)

This element defines a color that happens to be the accent 2 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

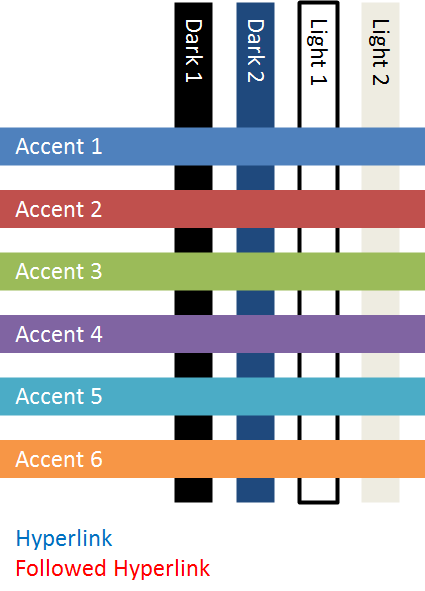
</sequence>

</complexType>

##### accent3 (Accent 3)

This element defines a color that happens to be the accent 3 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

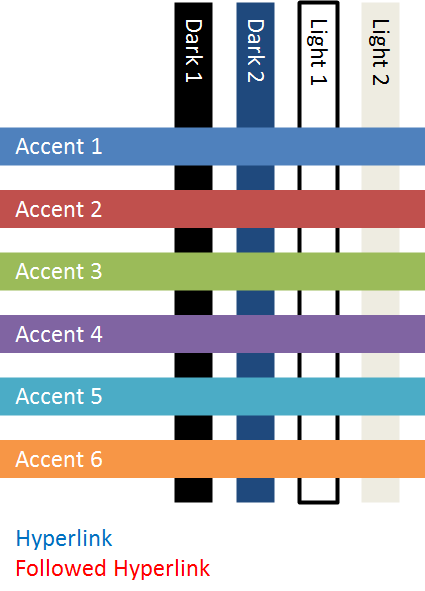
</sequence>

</complexType>

##### accent4 (Accent 4)

This element defines a color that happens to be the accent 4 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

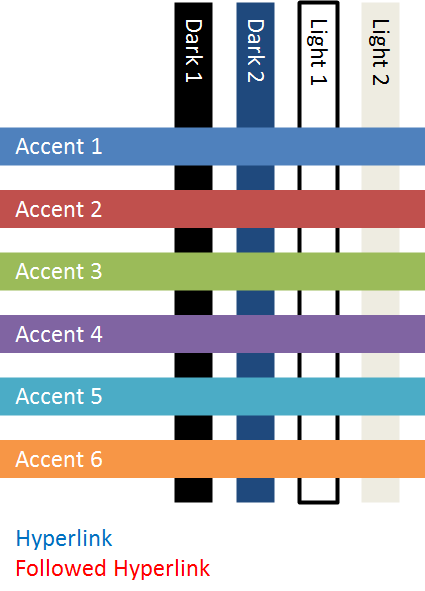
</sequence>

</complexType>

##### accent5 (Accent 5)

This element defines a color that happens to be the accent 5 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

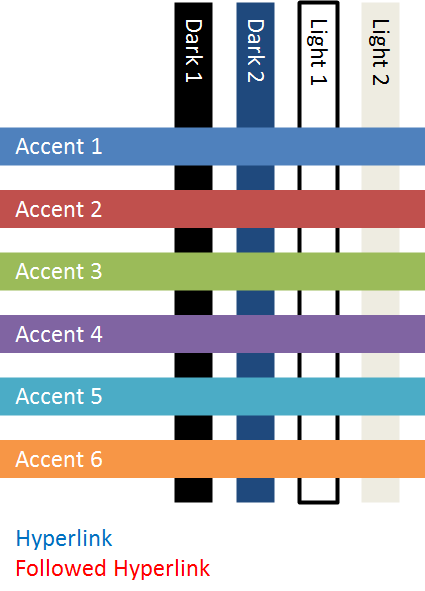
</sequence>

</complexType>

##### accent6 (Accent 6)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

</sequence>

</complexType>

##### bgFillStyleLst (Background Fill Style List)

This element defines a list of background fills that are used within a theme. The background fills consist of three fills, arranged in [order](order.docx) from subtle to moderate to intense.

[Example: Consider the following example of a background [fill](fill.docx) style list within DrawingML:

<bgFillStyleLst>  
 <[solidFill](solidFill.docx)>  
…  
 </[solidFill](solidFill.docx)>  
 <[gradFill](gradFill.docx) rotWithShape="1">  
…  
 </[gradFill](gradFill.docx)>  
 <[blipFill](blipFill.docx)>  
…  
 </[blipFill](blipFill.docx)>  
</bgFillStyleLst>

In this example, we see that the list contains a solid [fill](fill.docx) for the subtle [fill](fill.docx), a gradient [fill](fill.docx) for the moderate [fill](fill.docx) and an image [fill](fill.docx) for the intense background fill. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| fmtScheme (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [blipFill](blipFill.docx) (Picture Fill) | § |
| [gradFill](gradFill.docx) (Gradient Fill) | § |
| [grpFill](grpFill.docx) (Group Fill) | § |
| [noFill](noFill.docx) (No Fill) | § |
| [pattFill](pattFill.docx) (Pattern Fill) | § |
| [solidFill](solidFill.docx) (Solid Fill) | § |

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

<complexType [name](name.docx)="CT\_BackgroundFillStyleList">

<sequence>

<group ref="EG\_FillProperties" minOccurs="3" maxOccurs="unbounded"/>

</sequence>

</complexType>

##### custClr (Custom color)

This element defines a custom color. The custom colors are used within a custom color list to define custom colors that are extra colors that can be appended to a theme. This is useful within corporate scenarios where there is a set corporate color palette from which to work.

|  |
| --- |
| Parent Elements |
| [custClrLst](custClrLst.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [name](name.docx) (Name) | The name of the color shown in the color picker.  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](name.docx)="CT\_CustomColor">

<sequence>

<group ref="EG\_ColorChoice" minOccurs="1" maxOccurs="1"/>

</sequence>

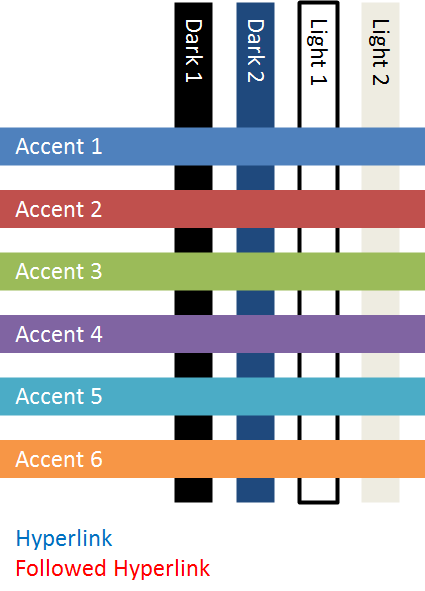
<attribute name="name" type="xsd:string" use="optional" default=""/>

</complexType>

##### dk1 (Dark 1)

This element defines a color that happens to be the dark 1 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

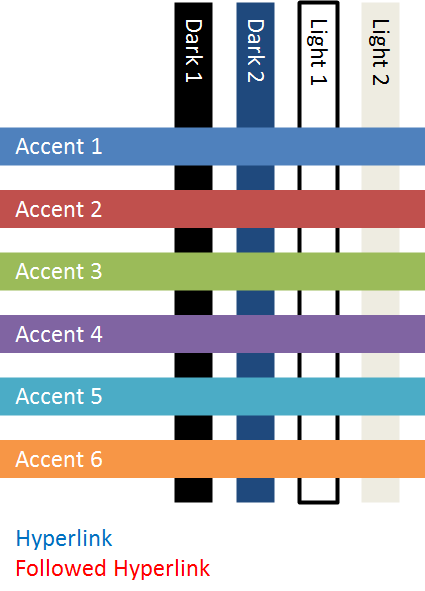
</sequence>

</complexType>

##### dk2 (Dark 2)

This element defines a color that happens to be the dark 2 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

</sequence>

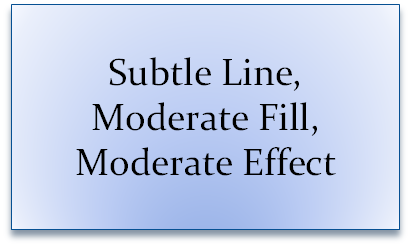
</complexType>

##### effectStyle (Effect Style)

This element defines a set of effects and [3D](3D.docx) properties that can be applied to an object.

[Example: Consider the following example of an [effect](effect.docx) style within DrawingML:

<effectStyle>  
 <[effectLst](effectLst.docx)>  
 <[outerShdw](outerShdw.docx) blurRad="57150" dist="38100" [dir](dir.docx)="5400000" algn="ctr"  
 rotWithShape="0">  
 <schemeClr [val](val.docx)="phClr">  
 <shade [val](val.docx)="9000"/>  
 <satMod [val](val.docx)="105000"/>  
 <alpha [val](val.docx)="48000"/>  
 </schemeClr>  
 </[outerShdw](outerShdw.docx)>  
 </[effectLst](effectLst.docx)>  
</effectStyle>



In this example, an outer shadow is being applied to a shape as the moderate effect. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| effectStyleLst (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [effectDag](effectDag.docx) (Effect Container) | § |
| [effectLst](effectLst.docx) (Effect Container) | § |
| [scene3d](scene3d.docx) (3D Scene Properties) | § |
| [sp3d](sp3d.docx) (Apply [3D](3D.docx) shape properties) | § |

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

<complexType [name](name.docx)="CT\_EffectStyleItem">

<sequence>

<group ref="EG\_EffectProperties" minOccurs="1" maxOccurs="1"/>

<element name="[scene3d](scene3d.docx)" type="CT\_Scene3D" minOccurs="0" maxOccurs="1"/>

<element name="[sp3d](sp3d.docx)" type="CT\_Shape3D" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>

##### effectStyleLst (Effect Style List)

This element defines a set of three [effect](effect.docx) styles that create the [effect](effect.docx) style list for a theme. The [effect](effect.docx) styles are arranged in [order](order.docx) of subtle to moderate to intense.

[Example: Consider the following example of an [effect](effect.docx) style list within DrawingML:

<effectStyleLst>  
 <effectStyle>  
 <[effectLst](effectLst.docx)>  
 <[outerShdw](outerShdw.docx) blurRad="57150" dist="38100" [dir](dir.docx)="5400000"   
 algn="ctr" rotWithShape="0">  
…  
 </[outerShdw](outerShdw.docx)>  
 </[effectLst](effectLst.docx)>  
 </effectStyle>

<effectStyle>  
 <[effectLst](effectLst.docx)>  
 <[outerShdw](outerShdw.docx) blurRad="57150" dist="38100" [dir](dir.docx)="5400000"   
 algn="ctr" rotWithShape="0">  
…  
 </[outerShdw](outerShdw.docx)>  
 </[effectLst](effectLst.docx)>  
 </effectStyle>

<effectStyle>  
 <[effectLst](effectLst.docx)>  
 <[outerShdw](outerShdw.docx) blurRad="57150" dist="38100" [dir](dir.docx)="5400000"   
 algn="ctr" rotWithShape="0">  
…  
 </[outerShdw](outerShdw.docx)>  
 </[effectLst](effectLst.docx)>

<[scene3d](scene3d.docx)>  
…  
 </[scene3d](scene3d.docx)>  
 <[sp3d](sp3d.docx) prstMaterial="powder">  
…  
 </[sp3d](sp3d.docx)>  
 </effectStyle>  
</effectStyleLst>

In this example, we see three [effect](effect.docx) styles defined. The first two (subtle and moderate) define an outer shadow as the [effect](effect.docx), while the third [effect](effect.docx) style (intense) defines an outer shadow along with [3D](3D.docx) properties which are to be applied to the object as well. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| fmtScheme (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| effectStyle (Effect Style) | § |

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

<complexType [name](name.docx)="CT\_EffectStyleList">

<sequence>

<element name="effectStyle" type="CT\_EffectStyleItem" minOccurs="3" maxOccurs="unbounded"/>

</sequence>

</complexType>

##### fillStyleLst (Fill Style List)

This element defines a set of three [fill](fill.docx) styles that are used within a theme. The three [fill](fill.docx) styles are arranged in [order](order.docx) from subtle to moderate to intense.

[Example: Consider the following example of a [fill](fill.docx) style list within DrawingML:

<fillStyleLst>  
 <[solidFill](solidFill.docx)>  
…  
 </[solidFill](solidFill.docx)>

<[gradFill](gradFill.docx) rotWithShape="1">  
…  
 </[gradFill](gradFill.docx)>

<[gradFill](gradFill.docx) rotWithShape="1">  
…  
 </[gradFill](gradFill.docx)>  
</fillStyleLst>

In this example, we see three [fill](fill.docx) styles being defined within the [fill](fill.docx) style list. The first style is the subtle style and defines simply a solid fill. The second and third styles (moderate and intense fills respectively) define gradient fills. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| fmtScheme (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [blipFill](blipFill.docx) (Picture Fill) | § |
| [gradFill](gradFill.docx) (Gradient Fill) | § |
| [grpFill](grpFill.docx) (Group Fill) | § |
| [noFill](noFill.docx) (No Fill) | § |
| [pattFill](pattFill.docx) (Pattern Fill) | § |
| [solidFill](solidFill.docx) (Solid Fill) | § |

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

<complexType [name](name.docx)="CT\_FillStyleList">

<sequence>

<group ref="EG\_FillProperties" minOccurs="3" maxOccurs="unbounded"/>

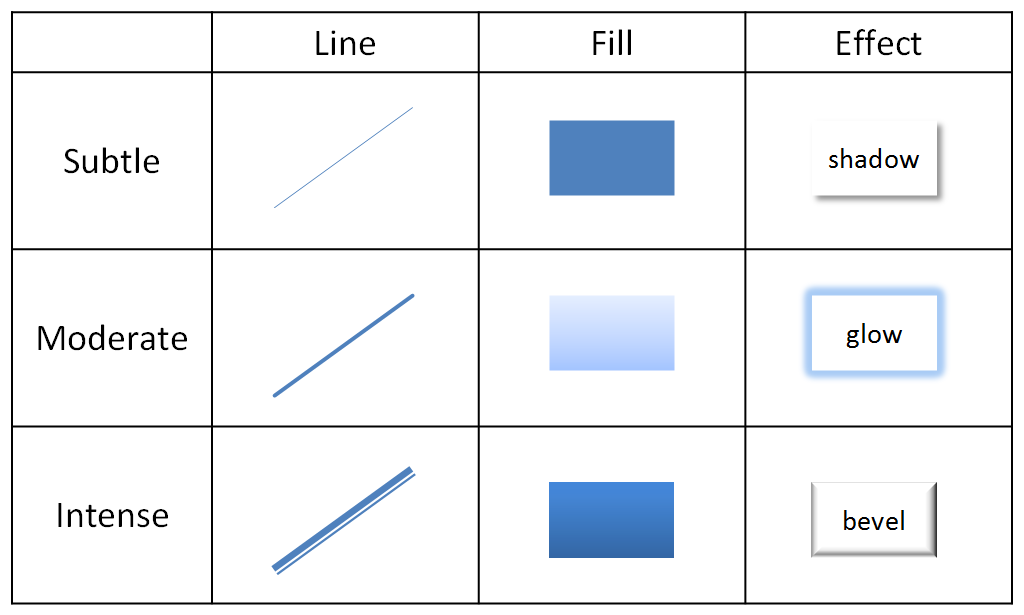
</sequence>

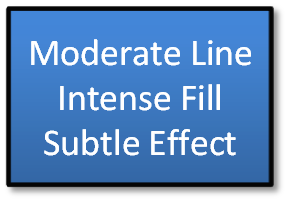
</complexType>

##### fmtScheme (Format Scheme)

This element contains the background [fill](fill.docx) styles, [effect](effect.docx) styles, [fill](fill.docx) styles, and line styles which define the style matrix for a theme. The style matrix consists of subtle, moderate, and intense fills, lines, and effects. The background fills are not generally thought of to directly be associated with the matrix, but do play a role in the style of the overall document. Usually, a given object will [choose](choose.docx) a single line style, a single [fill](fill.docx) style, and a single [effect](effect.docx) style in [order](order.docx) to define the overall final look of the object.

[Example: Consider the following example of the style matrix in use within DrawingML:





In this example, we see a shape styled which utilizes different aspects from the above defined style matrix. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [themeElements](themeElements.docx) (§); [themeOverride](themeOverride.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| bgFillStyleLst (Background Fill Style List) | § |
| effectStyleLst (Effect Style List) | § |
| fillStyleLst (Fill Style List) | § |
| lnStyleLst (Line Style List) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [name](name.docx) (Name) | Defines the name for the format scheme. The name is simply a human readable string which identifies the format scheme in the user interface.  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](name.docx)="CT\_StyleMatrix">

<sequence>

<element name="fillStyleLst" type="CT\_FillStyleList" minOccurs="1" maxOccurs="1"/>

<element name="lnStyleLst" type="CT\_LineStyleList" minOccurs="1" maxOccurs="1"/>

<element name="effectStyleLst" type="CT\_EffectStyleList" minOccurs="1" maxOccurs="1"/>

<element name="bgFillStyleLst" type="CT\_BackgroundFillStyleList" minOccurs="1" maxOccurs="1"/>

</sequence>

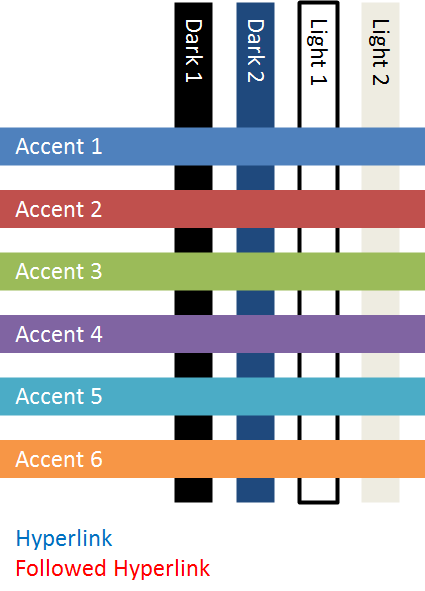
<attribute name="name" type="xsd:string" use="optional" default=""/>

</complexType>

##### folHlink (Followed Hyperlink)

This element defines a color that happens to be the followed hyperlink color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

</sequence>

</complexType>

##### font (Font)

This element defines a font within the styles area of DrawingML. A font is defined by a script along with a typeface.

[Example: Consider the following example of a font in DrawingML:

<font script="Thai" typeface="Cordia New"/>

In this example, we see that the script 'Thai' is supposed to use the font face 'Cordia New'. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| font (§); majorFont (§); minorFont (§) |

|  |  |
| --- | --- |
| Attributes | Description |
| script (Script) | Specifies the script, or language, in which the typeface is supposed to be used.  The possible values for this attribute are defined by the XML Schema string datatype. |
| typeface (Typeface) | Specifies the font face to use.  The possible values for this attribute are defined by the [ST\_TextTypeface](ST_TextTypeface.docx) simple type (§). |

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

<complexType [name](name.docx)="CT\_SupplementalFont">

<attribute [name](name.docx)="script" type="xsd:string" use="required"/>

<attribute [name](name.docx)="typeface" type="[ST\_TextTypeface](ST_TextTypeface.docx)" use="required"/>

</complexType>

##### fontRef (Font Reference)

This element represents a reference to a themed font. When used it specifies which themed font to use along with a choice of color.

[Example: Consider the following example of a font reference within DrawingML:

<fontRef [idx](idx.docx)="minor">  
 <schemeClr [val](val.docx)="tx1"/>  
</fontRef>

In this example, we see a font referencing the minor font defined within the theme. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| style (§); style (§); style (§); style (§); style (§); tcTxStyle (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [idx](idx.docx) (Identifier) | Specifies the identifier of the font to reference.  The possible values for this attribute are defined by the [ST\_FontCollectionIndex](ST_FontCollectionIndex.docx) simple type (§). |

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

<complexType [name](name.docx)="CT\_FontReference">

<sequence>

<group ref="EG\_ColorChoice" minOccurs="0" maxOccurs="1"/>

</sequence>

<attribute [name](name.docx)="[idx](idx.docx)" type="[ST\_FontCollectionIndex](ST_FontCollectionIndex.docx)" use="required"/>

</complexType>

##### fontScheme (Font Scheme)

This element defines the font scheme within the theme. The font scheme consists of a pair of major and minor fonts for which to use in a document. The major font corresponds well with the heading areas of a document, and the minor font corresponds well with the normal text or paragraph areas.

[Example: Consider the following example of a font scheme within DrawingML:

<fontScheme [name](name.docx)="sample">  
 <majorFont>  
…  
 </majorFont>

<minorFont>  
…  
 </minorFont>  
</fontScheme>

In this example, we see the major and minor font lists within the font scheme that is named 'sample'. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [themeElements](themeElements.docx) (§); [themeOverride](themeOverride.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [extLst](extLst.docx) (Extension List) | § |
| majorFont (Major Font) | § |
| minorFont (Minor fonts) | § |

|  |  |
| --- | --- |
| Attributes | Description |
| [name](name.docx) (Name) | The name of the font scheme shown in the user interface.  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](name.docx)="CT\_FontScheme">

<sequence>

<element name="majorFont" type="CT\_FontCollection" minOccurs="1" maxOccurs="1"/>

<element name="minorFont" type="CT\_FontCollection" minOccurs="1" maxOccurs="1"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

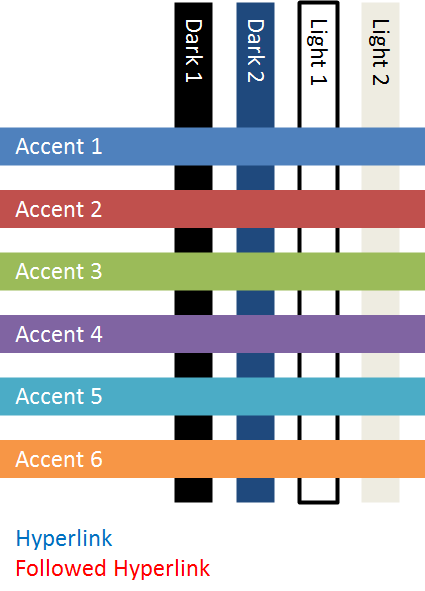
<attribute [name](name.docx)="[name](name.docx)" type="xsd:string" use="required"/>

</complexType>

##### hlink (Hyperlink)

This element defines a color that happens to be the hyperlink color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

</sequence>

</complexType>

##### lnDef (Line Default)

This element defines a default line that is used within a document.

[Example: Consider the following example of a default line defined in DrawingML:

<lnDef>  
 <[spPr](spPr.docx)/>  
 <bodyPr/>  
 <lstStyle/>

<style>  
 <lnRef [idx](idx.docx)="1">  
 <schemeClr [val](val.docx)="accent2"/>  
 </lnRef>

<fillRef [idx](idx.docx)="0">  
 <schemeClr [val](val.docx)="accent2"/>  
 </fillRef>

<effectRef [idx](idx.docx)="0">  
 <schemeClr [val](val.docx)="accent2"/>  
 </effectRef>

<fontRef [idx](idx.docx)="minor">  
 <schemeClr [val](val.docx)="tx1"/>  
 </fontRef>  
 </style>  
</lnDef>

In this example, we see that the default line for the document is being defined as a themed line which references the subtle line style with [idx](idx.docx) equal to 1. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [objectDefaults](objectDefaults.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| bodyPr (Body Properties) | § |
| [extLst](extLst.docx) (Extension List) | § |
| lstStyle (Text List Styles) | § |
| [spPr](spPr.docx) (Shape Properties) | § |
| [style](style.docx) (Shape Style) | § |

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

<complexType [name](name.docx)="CT\_DefaultShapeDefinition">

<sequence>

<element name="[spPr](spPr.docx)" type="CT\_ShapeProperties" minOccurs="1" maxOccurs="1"/>

<element name="bodyPr" type="CT\_TextBodyProperties" minOccurs="1" maxOccurs="1"/>

<element name="lstStyle" type="CT\_TextListStyle" minOccurs="1" maxOccurs="1"/>

<element name="style" type="CT\_ShapeStyle" minOccurs="0" maxOccurs="1"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>

##### lnStyleLst (Line Style List)

This element defines a list of three line styles for use within a theme. The three line styles are arranged in [order](order.docx) from subtle to moderate to intense versions of lines. This list makes [up](up.docx) part of the style matrix.

[Example: Consider the following example of a line style list within DrawingML:

<lnStyleLst>  
 <ln [w](w.docx)="9525" cap="flat" cmpd="sng" algn="ctr">  
 <[solidFill](solidFill.docx)>  
 <schemeClr [val](val.docx)="phClr">  
 <shade [val](val.docx)="50000"/>  
 <satMod [val](val.docx)="103000"/>  
 </schemeClr>  
 </[solidFill](solidFill.docx)>  
 <[prstDash](prstDash.docx) [val](val.docx)="solid"/>  
 </ln>

<ln [w](w.docx)="25400" cap="flat" cmpd="sng" algn="ctr">  
 <[solidFill](solidFill.docx)>  
 <schemeClr [val](val.docx)="phClr"/>  
 </[solidFill](solidFill.docx)>  
 <[prstDash](prstDash.docx) [val](val.docx)="solid"/>  
 </ln>

<ln [w](w.docx)="38100" cap="flat" cmpd="sng" algn="ctr">  
 <[solidFill](solidFill.docx)>  
 <schemeClr [val](val.docx)="phClr"/>  
 </[solidFill](solidFill.docx)>  
 <[prstDash](prstDash.docx) [val](val.docx)="solid"/>  
 </ln>  
</lnStyleLst>

In this example, we see three lines defined within a line style list. The first line corresponds to the subtle line, the second to the moderate, and the third corresponds to the intense line defined in the theme. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| fmtScheme (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| ln (Outline) | § |

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

<complexType [name](name.docx)="CT\_LineStyleList">

<sequence>

<element name="ln" type="CT\_LineProperties" minOccurs="3" maxOccurs="unbounded"/>

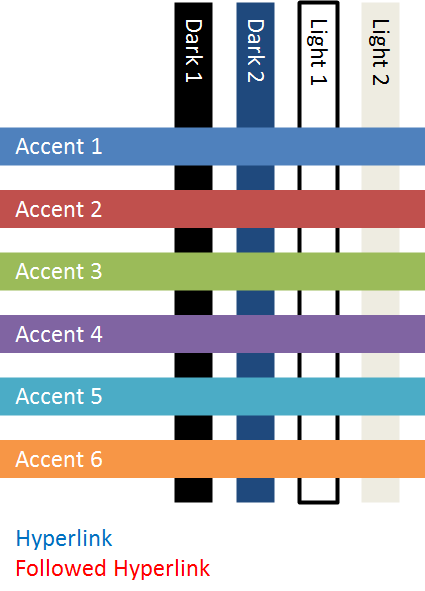
</sequence>

</complexType>

##### lt1 (Light 1)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

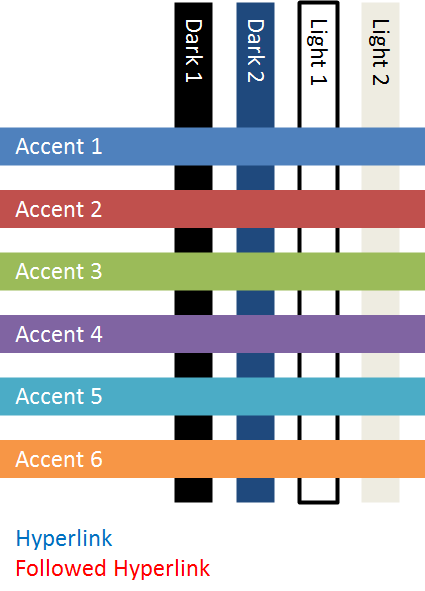
</sequence>

</complexType>

##### lt2 (Light 2)

This element defines a color that happens to be the accent 1 color. The set of twelve colors come together to form the color scheme for a theme.

[Example: Consider the following example of a set of colors that form a color scheme:



[end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [clrScheme](clrScheme.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| hslClr (Hue, Saturation, Luminance Color Model) | § |
| prstClr (Preset Color) | § |
| schemeClr (Scheme Color) | § |
| scrgbClr (RGB Color Model - Percentage Variant) | § |
| srgbClr (RGB Color Model - Hex Variant) | § |
| sysClr (System Color) | § |

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

<complexType [name](name.docx)="CT\_Color">

<sequence>

<group ref="EG\_ColorChoice"/>

</sequence>

</complexType>

##### majorFont (Major Font)

This element defines the set of major fonts which are to be used under different languages or locals.

[Example: Consider the following example of the major fonts being defined within DrawingML:

<majorFont>  
 <latin typeface="Calibri"/>  
 <ea typeface="Arial"/>  
 <cs typeface="Arial"/>  
 <font script="Jpan" typeface="ＭＳ Ｐゴシック"/>  
 <font script="Hang" typeface="HY중고딕"/>

<font script="Hans" typeface="隶书"/>  
 <font script="Hant" typeface="微軟正黑體"/>  
 <font script="Arab" typeface="Traditional Arabic"/>  
 <font script="Hebr" typeface="Arial"/>  
 <font script="Thai" typeface="Cordia New"/>  
 <font script="Ethi" typeface="Nyala"/>

<font script="Beng" typeface="Vrinda"/>  
 <font script="Gujr" typeface="Shruti"/>  
 <font script="Khmr" typeface="DaunPenh"/>  
 <font script="Knda" typeface="Tunga"/>  
</majorFont>

In this example, we see the latin, east asian, and complex script fonts defined along with many fonts for different locals. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| fontScheme (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| cs (Complex Script Font) | § |
| ea (East Asian Font) | § |
| [extLst](extLst.docx) (Extension List) | § |
| font (Font) | § |
| latin (Latin Font) | § |

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

<complexType [name](name.docx)="CT\_FontCollection">

<sequence>

<element name="latin" type="CT\_TextFont" minOccurs="1" maxOccurs="1"/>

<element name="ea" type="CT\_TextFont" minOccurs="1" maxOccurs="1"/>

<element name="cs" type="CT\_TextFont" minOccurs="1" maxOccurs="1"/>

<element name="font" type="CT\_SupplementalFont" minOccurs="0" maxOccurs="unbounded"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>

##### minorFont (Minor fonts)

This element defines the set of minor fonts that are to be used under different languages or locals.

[Example: Consider the following example of the minor fonts being defined within DrawingML:

<minorFont>  
 <latin typeface="Calibri"/>  
 <ea typeface="Arial"/>  
 <cs typeface="Arial"/>  
 <font script="Jpan" typeface="ＭＳ Ｐゴシック"/>  
 <font script="Hang" typeface="HY중고딕"/>  
 <font script="Hans" typeface="隶书"/>  
 <font script="Hant" typeface="微軟正黑體"/>  
 <font script="Arab" typeface="Traditional Arabic"/>  
 <font script="Hebr" typeface="Arial"/>  
 <font script="Thai" typeface="Cordia New"/>  
 <font script="Ethi" typeface="Nyala"/>  
 <font script="Beng" typeface="Vrinda"/>  
 <font script="Gujr" typeface="Shruti"/>  
 <font script="Khmr" typeface="DaunPenh"/>  
 <font script="Knda" typeface="Tunga"/>  
</minorFont>

In this example, we see the latin, east asian, and complex script fonts defined along with many fonts for different locals. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| fontScheme (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| cs (Complex Script Font) | § |
| ea (East Asian Font) | § |
| [extLst](extLst.docx) (Extension List) | § |
| font (Font) | § |
| latin (Latin Font) | § |

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

<complexType [name](name.docx)="CT\_FontCollection">

<sequence>

<element name="latin" type="CT\_TextFont" minOccurs="1" maxOccurs="1"/>

<element name="ea" type="CT\_TextFont" minOccurs="1" maxOccurs="1"/>

<element name="cs" type="CT\_TextFont" minOccurs="1" maxOccurs="1"/>

<element name="font" type="CT\_SupplementalFont" minOccurs="0" maxOccurs="unbounded"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>

##### [scene3d](scene3d.docx) (3D Scene Properties)

This element defines optional scene-level [3D](3D.docx) properties to apply to an object.

|  |
| --- |
| Parent Elements |
| bodyPr (§); effectStyle (§); [grpSpPr](grpSpPr.docx) (§); [grpSpPr](grpSpPr.docx) (§); [grpSpPr](grpSpPr.docx) (§); [grpSpPr](grpSpPr.docx) (§); [spPr](spPr.docx) (§); [spPr](spPr.docx) (§); [spPr](spPr.docx) (§); [spPr](spPr.docx) (§); [spPr](spPr.docx) (§); [spPr](spPr.docx) (§); [spPr](spPr.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| [backdrop](backdrop.docx) (Backdrop Plane) | § |
| [camera](camera.docx) (Camera) | § |
| [extLst](extLst.docx) (Extension List) | § |
| [lightRig](lightRig.docx) (Light Rig) | § |

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

<complexType [name](name.docx)="CT\_Scene3D">

<sequence>

<element name="camera" type="CT\_Camera" minOccurs="1" maxOccurs="1"/>

<element name="[lightRig](lightRig.docx)" type="CT\_LightRig" minOccurs="1" maxOccurs="1"/>

<element name="[backdrop](backdrop.docx)" type="CT\_Backdrop" minOccurs="0" maxOccurs="1"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>

##### spDef (Shape Default)

This element defines the formatting that is associated with the default shape. The default formatting can be applied to a shape when it is initially inserted into a document.

[Example: Consider the following example of a shape default being used within DrawingML:

<spDef>  
 <[spPr](spPr.docx)>  
 <[solidFill](solidFill.docx)>  
 <schemeClr [val](val.docx)="accent2">  
 <shade [val](val.docx)="75000"/>  
 </schemeClr>  
 </[solidFill](solidFill.docx)>  
 </[spPr](spPr.docx)>

<bodyPr rtlCol="0" [anchor](anchor.docx)="ctr"/>

<lstStyle>  
 <defPPr algn="ctr">  
 <defRPr/>  
 </defPPr>  
 </lstStyle>  
 <style>  
 <lnRef [idx](idx.docx)="1">  
 <schemeClr [val](val.docx)="accent1"/>  
 </lnRef>

<fillRef [idx](idx.docx)="2">  
 <schemeClr [val](val.docx)="accent1"/>  
 </fillRef>

<effectRef [idx](idx.docx)="1">  
 <schemeClr [val](val.docx)="accent1"/>  
 </effectRef>

<fontRef [idx](idx.docx)="minor">  
 <schemeClr [val](val.docx)="dk1"/>  
 </fontRef>  
 </style>  
</spDef>

In this example, we see a default shape which references a certain themed [fill](fill.docx), line, [effect](effect.docx), and font along with an override [fill](fill.docx) to these. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [objectDefaults](objectDefaults.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| bodyPr (Body Properties) | § |
| [extLst](extLst.docx) (Extension List) | § |
| lstStyle (Text List Styles) | § |
| [spPr](spPr.docx) (Shape Properties) | § |
| [style](style.docx) (Shape Style) | § |

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

<complexType [name](name.docx)="CT\_DefaultShapeDefinition">

<sequence>

<element name="[spPr](spPr.docx)" type="CT\_ShapeProperties" minOccurs="1" maxOccurs="1"/>

<element name="bodyPr" type="CT\_TextBodyProperties" minOccurs="1" maxOccurs="1"/>

<element name="lstStyle" type="CT\_TextListStyle" minOccurs="1" maxOccurs="1"/>

<element name="style" type="CT\_ShapeStyle" minOccurs="0" maxOccurs="1"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>

##### txDef (Text Default)

This element defines the default formatting which is applied to text in a document by default. The default formatting can and should be applied to the shape when it is initially inserted into a document.

[Example: Consider the following example of a text default being used within DrawingML:

<txDef>  
 <[spPr](spPr.docx)>  
 <[solidFill](solidFill.docx)>  
 <schemeClr [val](val.docx)="accent2">  
 <shade [val](val.docx)="75000"/>  
 </schemeClr>  
 </[solidFill](solidFill.docx)>  
 </[spPr](spPr.docx)>

<bodyPr rtlCol="0" [anchor](anchor.docx)="ctr"/>

<lstStyle>  
 <defPPr algn="ctr">  
 <defRPr/>  
 </defPPr>  
 </lstStyle>

<style>  
 <lnRef [idx](idx.docx)="1">  
 <schemeClr [val](val.docx)="accent1"/>  
 </lnRef>

<fillRef [idx](idx.docx)="2">  
 <schemeClr [val](val.docx)="accent1"/>  
 </fillRef>

<effectRef [idx](idx.docx)="1">  
 <schemeClr [val](val.docx)="accent1"/>  
 </effectRef>

<fontRef [idx](idx.docx)="minor">  
 <schemeClr [val](val.docx)="dk1"/>  
 </fontRef>  
 </style>  
</txDef>

In this example, we see a default text which references a certain themed [fill](fill.docx), line, [effect](effect.docx), and font along with an override [fill](fill.docx) to these. [end](end.docx) example]

|  |
| --- |
| Parent Elements |
| [objectDefaults](objectDefaults.docx) (§) |

|  |  |
| --- | --- |
| Child Elements | Subclause |
| bodyPr (Body Properties) | § |
| [extLst](extLst.docx) (Extension List) | § |
| lstStyle (Text List Styles) | § |
| [spPr](spPr.docx) (Shape Properties) | § |
| [style](style.docx) (Shape Style) | § |

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

<complexType [name](name.docx)="CT\_DefaultShapeDefinition">

<sequence>

<element name="[spPr](spPr.docx)" type="CT\_ShapeProperties" minOccurs="1" maxOccurs="1"/>

<element name="bodyPr" type="CT\_TextBodyProperties" minOccurs="1" maxOccurs="1"/>

<element name="lstStyle" type="CT\_TextListStyle" minOccurs="1" maxOccurs="1"/>

<element name="style" type="CT\_ShapeStyle" minOccurs="0" maxOccurs="1"/>

<element name="[extLst](extLst.docx)" type="CT\_OfficeArtExtensionList" minOccurs="0" maxOccurs="1"/>

</sequence>

</complexType>