

DIRECTSMILE DOCUMENT AND COMPOSITION MARKUP LANGUAGE (VERSION 1.0)

<u>DirectSmile Document and Composition Markup language (Version 1.0)</u>	<u>1</u>
<u>About XML (taken from Wikipedia and slightly modified with DirectSmile examples)</u>	<u>2</u>
Key terminology	2
(Unicode) character	2
Processor and application	2
Markup and content	2
Tag	2
Element	2
Attribute	2
XML declaration – do not use in DirectSmile Document ML!	3
<u>Overview about the structure of the document ML and child elements</u>	<u>3</u>
Container	3
Child elements	3
Single	3
<u>Measurement units for the XML elements (DTP point "pt" is the default unit)</u>	<u>3</u>
pt	3
pica	3
inch	3
cicero	3
mm	3
cm	3
<u>Color definitions for the composition child element</u>	<u>4</u>
<u>Document ML elements and their attributes (do not use in XML composition editor)</u>	<u>4</u>
Documents	4
Document	4
Page	4
Frame	4
<u>Attributes for the composition element (for document ML and XML composition editor)</u>	<u>5</u>
Composition	5
<u>Common attributes for composition child elements (Barcode, Line, Picture, Set, Shape, Text)</u>	<u>5</u>
Positioning	5
Appearance	5
<u>Edit options for web edit (do not use in composition an document MLframes)</u>	<u>5</u>
<u>Additional attributes for particular items in the composition</u>	<u>6</u>
Barcode	6
Line	6
Picture	6
Set	6
Shape	6
Text	6
<u>Text container child element</u>	<u>6</u>
Paragraph	6
<u>Paragraph container child element</u>	<u>7</u>
Run	7
LineBreak	7
<u>An example for a simple text output in the XML composition editor</u>	<u>7</u>

The XML Composition container allows you to personalize everything you want in printable documents. And it's easy to use: Just insert a <Text> or a <Image> tag to xml. Variables can be placed where ever it's needed. Even the entire composition can be filled by a variable which you create via a flash or javascript client. If you generate your XML via a DOM please switch off any formatting option during serialisation as unencoded Whitespace and linebreaks are displayed within runs!

About XML

(taken from Wikipedia and slightly modified with DirectSmile examples. Please see http://wikimediafoundation.org/wiki/Terms_of_Use)

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all free open standards.

The design goals of XML emphasize simplicity, generality, and usability over the Internet. It is a textual data format with strong support via Unicode for the languages of the world. Although the design of XML focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services.

Many application programming interfaces (APIs) have been developed to aid software developers with processing XML data, and several schema systems exist to aid in the definition of XML-based languages.

As of 2009, hundreds of document formats using XML syntax have been developed, including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for many office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org and LibreOffice (OpenDocument), and Apple's iWork. XML has also been employed as the base language for communication protocols, such as XMPP.

Key terminology

The material in this section is based on the XML Specification. This is not an exhaustive list of all the constructs that appear in XML; it provides an introduction to the key constructs most often encountered in day-to-day use.

(Unicode) character

By definition, an XML document is a string of characters. Almost every legal Unicode character may appear in an XML document.

Processor and application

The processor analyzes the markup and passes structured information to an application. The specification places requirements on what an XML processor must do and not do, but the application is outside its scope. The processor (as the specification calls it) is often referred to colloquially as an XML parser.

Markup and content

The characters making up an XML document are divided into markup and content, which may be distinguished by the application of simple syntactic rules. Generally, strings that constitute markup either begin with the character < and end with a >. Strings of characters that are not markup are content.

Tag

A markup construct that begins with < and ends with >. Tags come in three flavors:

- start-tags; for example: <Document>
- end-tags; for example: </Document>
- empty-element tags; for example: <LineBreak/>

Element

A logical document component which either begins with a start-tag and ends with a matching end-tag or consists only of an empty-element tag. The characters between the start- and end-tags, if any, are the element's content, and may contain markup, including other elements, which are called child elements.

An example of an element is <Run>Hello, world.</Run>. Another is <LineBreak/>.

Attribute

A markup construct consisting of a name/value pair that exists within a start-tag or empty-element tag. In the example (below) the element Picture has two attributes, Source and CornerRadius:

```
<Picture Source="madonna.jpg" CornerRadius="18pt"/>
```

An XML attribute can only have a single value and each attribute can appear at most once on each element. In the common situation where a list of multiple values is desired, this must be done by encoding the list into a well-formed XML attribute with some format beyond what XML defines itself. Usually this is either a comma or semi-colon delimited list.

```
<Shape Background="rgb, 100,0,0"/>
```

where the attribute "Background" has the a RGB value of 100% Red , 0% Green, 0%Blue light.

XML declaration – do not use in DirectSmile Document ML!

Regular XML documents may begin by declaring some information about themselves, as in the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Please do not use this declaring in DirectSmile's ML, neither in the XML composition editor nor using JavaScript because we will add this add runtime Simply begin with the "<Composition>" (XML frame on a printing page) or "<Documents>" (complete documents for print).

DocML always have to be encoded UTF-8.

(end of edited Wikipedia Content)

Overview about the structure of the document ML and child elements

Container	Child elements	Single
Documents:	Document	
	Resources	X
Document:	Page	
Resources:		
Page:	Frame	
Frame:	Composition	X
Composition:	Barcode	
	Line	
	Picture	
	Set	
	Shape	
	Text	
Text:	Paragraph	
Paragraph:	Run	
	LineBreak	

Measurement units for the XML elements (DTP point "pt" is the default unit)

Unit	Point	Inch	Milimetres
pt	1	1/72	25.4/72
pica	12	1/6	≈4.233
inch	72 pt	1	25.4
cicero	≈12.79	1/5.629	≈4.512
mm	≈0.353	1/25.4	1
cm	≈3.527	1/2.54	10

Color definitions for the composition child element

Web color names can be used by their names:

Black, Blue, Brown, Cyan, Darkgray, Gray, Green, LightGray, Magenta, Orange, Red, White, Red, Yellow

Other colors need the color definition name as the first argument (rgb, cmyk, or gray) followed by the percent values of the components in the named order.

RGB definitions: `rgb, 100,0,0`

CMYK color: `cmyk, 0,0,0,100`

Gray scale: `gray, 50`

Spot colors need the color definition "spot", the spot color name, the percent value of the tint, the preview color definition, and the percent values of the components in the named order.

Spot colors: `spot, SampleSpotName, 100, cmyk, 0,0,50.5,100`

Document ML elements and their attributes (do not use in XML composition editor)

Documents	Documents container for a document collection		
Attributes	Values	Format	Note
Version	1 (default)	Number	

Document	Document definition, the container for a page collection		
Attributes	Values	Format	Note
InsertPageWidth	Floating point value and mesurement (default pt)	String	
InsertPageHeight	Floating point value and mesurement (default pt)	String	

Page	Page definition, the container for a frame collection		
Attributes	Values	Format	Note
width	Floating point value and mesurement (default pt)	String	
Height	Floating point value and mesurement (default pt)	String	

Frame	Page definition, the container for one composition element		
Attributes	Values	Format	Note
Caption	Display name	String	
Left	Floating point value and mesurement unit (pt is default)	String	
Top	Floating point value and mesurement unit (pt is default)	String	
width	Floating point value and mesurement unit (pt is default)	String	
Height	Floating point value and mesurement unit (pt is default)	String	
Rotation	Floating point value in degrees	Float	
VAlign	Top (default), Center, Bottom (Vertical alignment for the content)	String	
HAling	Left (default), Center, Right (Horizontal alignment for the content)	String	
Stretch	None (default), Uniform, UnifomToFill, Fill	String	
HasLeftBleed	False (default) if content should get clipped on the left side or True if the content should extend beyond the frames left Border	Boolean	
HasTopBleed	False (default) if content should get clipped on the Top edge or True if the content should extend beyond the frames top edge.	Boolean	
HasRightBleed	False (default) if content should get clipped on the right side or True if the content should extend beyond the frames right Border	Boolean	
HasBottomBleed	False (default) if content should get clipped on the bottom edge or True if the content should extend beyond the frames bottom edge.	Boolean	

Attributes for the composition element (for document ML and XML composition editor)

Composition	Composition definition, the container for different child (design) elements		
Attributes	Values	Format	Note
width	Floating point value and measurement unit (default pt)	String	
Height	Floating point value and measurement unit (default pt)	String	
Version	1 (default)	Number	

Common attributes for composition child elements (Barcode, Line, Picture, Set, Shape, Text)

Positioning	Common positioning attributes for child elements of a composition container		
Attributes	Values	Format	Note
Left	Floating point value and measurement unit (pt is default)	String	
Top	Floating point value and measurement unit (pt is default)	String	
Width	Floating point value and measurement unit (pt is default)	String	
Height	Floating point value and measurement unit (pt is default)	String	
Rotation	Floating point value in degrees	Float	
PadLeft	Floating point value and measurement unit (pt is default)	String	
PadTop	Floating point value and measurement unit (pt is default)	String	
PadRight	Floating point value and measurement unit (pt is default)	String	
PadBottom	Floating point value and measurement unit (pt is default)	String	

Appearance	Common appearance attributes for child elements of a composition container		
Attributes	Values	Format	Note
Background	Color definition as described before	String	
GradientAngle	Floating point value in degrees	Float	
GradientStops	Pos, Col1%[, Col2%][, Col3%][, Col4%][[Pos, Col1% ...] (See below)	String	
Opacity	Floating point value: $0 \leq \text{value} \leq 1$ (default)	Float	
BorderColor	Color definition as described before	String	
BorderThickness	Floating point value and measurement unit (0 pt is default)	String	
CornerRadius	Floating point value and measurement unit (0 pt is default)	String	

GradientStops: Pos means a floating point value: $0 \leq \text{value} \leq 1$ (1 is default), Col1 to Col4 are the color components depending on the given background color GradientStops must be in the same color model (rgb, cmyk, gray or spot) than the Background color. Next stop follows after a vertical line sign "|".

Example:

```
<Shape Width="400pt" Height="200pt" Background="cmyk, 0,0,0,100" GradientAngle="0"
    GradientStops="0.5,100,10,30,10|0.9,20,100,30,0">
```

If the Background is a Spotcolor each Gradientstop must only name the tint value. Example:

```
<Shape Background="spot, SampleSpot, 100, cmyk, 0,0,50.5,0" GradientStops="0.5,10|0.8,100" />
```

Edit options for web edit (do not use in composition an document MLframes)

Id, lockedit, lockresize, lockposition, lockreplace, lockinfront, maxwidth, maxheight.

Additional attributes for particular items in the composition

Barcode			
Sets a QR code; in addition to the common attributes following specific one			
Attributes	Values	Format	Note
Content	The character string to encode	String	

Line			
Sets a line; in addition to the common attributes following specific ones			
Attributes	Values	Format	Note
LineType	Top, Bottom, Left, Right, Center, Diagonal1 (default) or Diagonal2	String	
LineThickness	Floating point value and measurement unit (pt is default)	String	
LineColor	Color definition as described before	String	

Picture			
Places an image; in addition to the common attributes following specific ones			
Attributes	Values	Format	Note
HAlign	Left, Center, Right	String	
VAlign	Top, Center, Bottom	String	
Stretch	None (default), Fill, Uniform, UniformToFill	String	
Source	available uploaded images (bmp, jpg, gif, png, pdf, eps, svg)	String	
Page	The number of the page from a multi page document	Number	

Set			
Places a DS-Set; in addition to the common attributes following specific ones			
Attributes	Values	Format	Note
HAlign	Left, Center, Right	String	
VAlign	Top, Center, Bottom	String	
Stretch	None (default), Fill, Uniform, UniformToFill	String	
SetAlias	System internal path (system, account, campaign) and file name	String	
Text	The number of the page from a document	Number	
Layer	Named Set layer only	String	
Transparency	False (default), True	Boolean	
Compression	(account default) or quality in percent ($0 \leq \text{value} \leq 100$)	Number	

Shape			
Places a shape; in addition to the common attributes following specific one			
Attributes	Values	Format	Note
ShapeType	Rectangle (default), Ellipse	String	

Text			
Places a text container; in addition to the common attributes following specific ones			
Attributes	Values	Format	Note
VerticalAlignment	Top, Center, Bottom	String	
RightToLeft	False (default), True (e.g. for Arabic or Hebrew)	Boolean	
FitText	False (default), True; Automatic resize for text overlength	Boolean	
HyphenationLanguage	A culture string like "en-US" or "EN"	String	

Text container child element

Paragraph			
Places a paragraph container as a child element in a Text container			
Attributes	Values	Format	Note
FontFamily	Any uploaded font available on the server (See below for more details on the correct font identifier)	String	

FontWeight	Available font weights	String	
FontSize	Floating point value and measurement unit (0 pt is default)	String	
LineHeight	Floating point value and measurement unit (default 0 means auto)	String	
Foreground	Color definition as described before	String	
TextAlignment	Left, Center, Right, Justify	String	

FontFamily: Because of uploaded fonts will not be installed on the server they need a particular identifier for addressing. Please use auto-completion in the Editor to find out the correct identifier in the XML Editor, here an example: "/SilverlightFontFuturaStdBold_otf;component/FuturaStd-Book.otf#Futura Std".

Paragraph container child element

Run	Child element in a Paragraph container		
Attributes	Values	Format	Note
FontFamily	Any uploaded font available on the server	String	See before
FontWeight	Available font weights	String	
FontSize	Floating point value and measurement unit (0 pt is default)	String	
Foreground	Color definition as described before	String	
Text	Full unicode support without encoding	String	

LineBreak	Child element in a Paragraph container; no attributes
------------------	--

An example for a simple text output in the XML composition editor

```
<Composition>
  <Text>
    <Paragraph>
      <Run>Hello World!</Run>
    </Paragraph>
  </Text>
</Composition>
```