

中华人民共和国国家标准

GB/T XXXX-XXXX

电子文件存储与交换格式 文书类版式文档

Specification for electronic files storage and exchange formats Fixed layout documents

在提交反馈意见时，请将您知道的相关专利连同支持性文档一并附上

(征求意见稿)

(本稿完成日期：2011-10-26)

200X-XX-XX 发布

200X-XX-XX 实施

中华人民共和国国家质量监督检验检疫总局 发布
中国国家标准化管理委员会

目 次

前 言	V
引 言	I
1 范围	1
2 规范性引用文件	1
3 缩略语	2
4 术语和定义	2
5 概述	3
5.1 文档特性.....	3
5.2 技术架构.....	4
5.3 成像模型.....	4
5.4 扩展名.....	5
6 文件结构	5
6.1 打包.....	5
6.1.1 XDA 包组织结构方案	5
6.1.2 ZIP 包组织结构方案	5
6.2 文件组织.....	5
7 基本结构	6
7.1 命名空间.....	6
7.2 字符编码.....	6
7.3 基础数据类型.....	7
7.4 主入口.....	7
7.5 文档根节点.....	10
7.6 页树.....	14
7.7 页结构.....	15
7.8 大纲.....	19
7.9 资源.....	20
8 页面描述	22
8.1 坐标系统.....	22
8.1.1 设备空间.....	22
8.1.2 页面空间.....	22
8.1.3 对象空间.....	22
8.1.4 变换矩阵.....	22
8.2 绘制参数.....	24
8.2.1 线条连接样式	26
8.2.2 线条连接点截断值	27
8.2.3 线条的虚线样式	27
8.2.4 线条的端点样式	28
8.3 颜色.....	28
8.3.1 颜色空间	28

8.3.2	基本颜色	29
8.3.3	底纹	31
8.3.4	渐变	34
8.4	裁剪区.....	44
8.5	图元对象.....	45
9	图形	49
9.1	图形对象.....	49
9.2	图形绘制参数.....	51
9.3	填充规则.....	52
9.4	图形的非紧缩描述.....	52
9.4.1	线段	53
9.4.2	贝塞尔曲线	53
9.4.3	圆弧	55
10	图像	56
11	文字	57
11.1	字体.....	57
11.2	文字对象.....	58
11.3	字形定位.....	61
11.4	字形变换.....	64
11.4.1	一对一	65
11.4.2	多对一	65
11.4.3	一对多	65
11.4.4	多对多	66
12	视频	67
13	SVG 对象.....	68
14	复合对象	68
15	动作	69
15.1	动作序列.....	69
15.2	动作类型.....	70
15.2.1	Goto 动作.....	71
15.2.2	URI 动作.....	72
15.2.3	Sound 动作.....	72
15.2.4	Movie 动作.....	73
16	注释	74
17	自定义标引	76
18	扩展	77
附录 A (规范性附录)	XDA 打包方案.....	79
附录 B (规范性附录)	使用 SVG 描述页面	85
附录 C (资料性附录)	SVG 描述版式页面示例.....	117
附录 D (规范性附录)	电子签名	121

前 言

本标准按照 GB/T 1.1-2009 的规则起草。

本标准由电子文件存储与交换格式标准工作组提出, 由全国信息技术标准化技术委员会归口。

本标准起草单位:

本标准主要起草人:

引 言

制定本标准的目的是为了定义统一的版式电子文件格式，保证版式电子文件的可读、可解析、可理解，降低电子文件管理和维护成本。

本标准的发布机构提请注意，声明符合本标准时，可能涉及到下表中的相关专利的使用。

本标准涉及专利

专利申请号	专利名称

本标准的发布机构对于所涉及的相关专利的真实性、有效性和范围无任何立场。

相关专利持有人已向本标准的发布机构保证，他愿意同任何申请人在合理且无歧视的条款和条件下，进行免费专利授权许可。相关专利持有人的声明已在本标准的发布机构备案。具体信息可以与标准归口单位联系。

请注意除上述已经识别出的专利外，本标准的某些内容仍有可能涉及其它专利。本标准的发布机构不承担识别这些专利的责任。

电子文件存储与交换格式 文书类版式文档

1 范围

本标准描述了一个适合交换和浏览的版式电子文件格式，主要对版式文件的文件结构、基本结构、页面描述、图形、图像、文字、注释等进行规定。

本标准适用于固定版式的电子文件存储与交换。

本标准规定的文档格式简称 OFD(Open Fixed-layout Document)是一种独立于软件、硬件、操作系统、呈现/打印设备的文档格式。

2 规范性引用文件

下列档中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用档，其随后所有的修改单(不包括勘误的内容)或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些档的最新版本。凡是不注日期的引用档，其最新版本适用于本标准。

GB/T 18793:2002 信息技术 可扩展置标语言(XML)1.0(GB/T 18793-2002, neq W3C RFC-xml-19980210: 1998)

GB 13000:2010 信息技术 通用多八位编码字符集(UCS)

GB 18030:2005 信息技术 中文编码字符集

ISO/IEC 14496-22:2007 Information technology -- Coding of audio-visual objects -- Part 22: Open Font Format

ISO/IEC 15948-2003 (E) Information technology — Computer graphics and image processing — Portable Network Graphics (PNG): Functional specification. (W3C Recommendation 10 November 2003)

[SVG] Scalable Vector Graphics(SVG)Tiny 1.2 Specification(W3C Recommendation 22 December 2008)

[ZIP] PKWARE Inc. Zip APPNOTE Version 6.2.0, April 26, 2004. Available on the Internet at <http://www.pkware.com/support/application-note-archives>.

[JPEG] ISO/IEC 10918, Information technology – Digital Compression and Coding of Continuous-Tone Still Images.

[PNG] ISO/IEC 15948-2004, Information technology – Computer Graphics and Image Processing – Portable Network Graphics (PNG).

[TIFF] TIFF Revision 6.0, Final, (June 1992), Adobe Systems Incorporated

[XML Signature Syntax and Processing] XML Signature Syntax and Processing (Second Edition)W3C Recommendation 10 June 2008

[ICC Profile]ISO 15076-1-2005, Image technology colour management – Architecture, profile format and data structure – Part 1:Based on ICC.1:2004-10

3 缩略语

下列缩略语适用于本文件。

OFD 开放版式文档 (Open Fixed-layout Document)

4 术语和定义

下列术语和定义适用于本档。

4.1

开放的版式文档 Open Fixed-layout Document

一种独立于软件、硬件、操作系统、呈现/打印设备的文档格式。

4.2

版式 Fixed-layout

版式是可以将文字、图形、图像等多种数字内容对象按照一定规则进行版面固化呈现。

4.3

版式文件 Fixed-layout Document

版式档指的是排版后生成的档，包含了版面呈现所需要的所有数据。

4.4

数字签名 digital signature

以电子形式存在于数据信息之中的，或作为其附件的或逻辑上与之有联系的数据，可用于辨别资料签署人的身份，并表明签署人对数据信息中包含的信息的认可。

4.5

成像模型 Imaging Model

一种设备无关的页面描述方法，采用抽象的图形元素描述页面中出现的文字、图形、图像等。

4.6

字符 Character

供组织、控制或表示数据用的元素集合中的一个元素。

4.7

字符集 Charset

一组无歧义的规则，用于建立一个字符和该字符集中的字符及其编码表示之间的一一对应关系。

【源自 GB/T 5271.1】

4.8

编码 Encoding

一组无歧义的规则，用于建立一个字符和该字符集中的字符及其编码表示之间的一一对应关系。

【源自 GB/T 5271.1】

4.9

字形 Glyph

一个可以辨认的抽象的图形符号，它不依赖于任何特定的设计，字形通常由字体内的点阵或者曲线轮廓描述。

4.10

像素 **Graphic Unit**

即图形元素，页面上任何对象都属于某种图元对象或者是某些图元对象的组合。

4.11

资源 **Resource**

是对一组像素绘制用参数或其它数据(如字体、图像等)描述的集合，在文件中按照不同层级可分为文档资源和页资源。

4.12

路径 **Path**

一系列点、线和曲线的集合，路径可以是开放的，也可以是封闭的。

4.13

裁剪区 **Clip Region**

指定裁剪的区域，通常由一组路径构成。

4.14

线性化 **Linearize**

文件的数据内容按照某种逻辑顺序(比如阅读顺序)在字节流上排列，这样的档在 Web 浏览等方面更有效率。

4.15

颜色空间 **Color Space**

描述使用一组值(通常使用三个、四个值或者颜色成分)表示颜色方法的抽象数学模型，在颜色模型和一个特定的参照颜色空间之间加入一个特定的映像函数就在参照颜色空间中出现了明确的色域，并且与颜色模型一起定义为一个新的颜色空间。

4.16

可扩展置标语言 **Extensible Markup Language XML**

定义语义标记的规则，这些标记将文件分成许多部件并对部件加以标识。作为元标记语言，用于定义与特定领域有关的结构化标记语言的句法语言。

4.17

出血框 **Bleed Box**

定义了在生产环境中输出时页面内容应当被剪切的区域。它包括任何需要适合切割、折叠和裁切设备物理限制的额外区域。

5 概述

5.1 文档特性

OFD 采用 XML 技术描述文件数据。

OFD 应真实地保持原有文件中文字、图表、公式、色彩等版式信息，且这种特性不应随着平台变化而发生改变。

5.2 技术架构

OFD 采用“容器 + 档”的方式描述和存储数据。容器是一个虚拟存储系统(Virtual Storage System)，应将各类数据描述文件聚合起来，并提供相应的访问接口和数据压缩方法。

OFD 的基本技术架构示意图如图 1 所示：

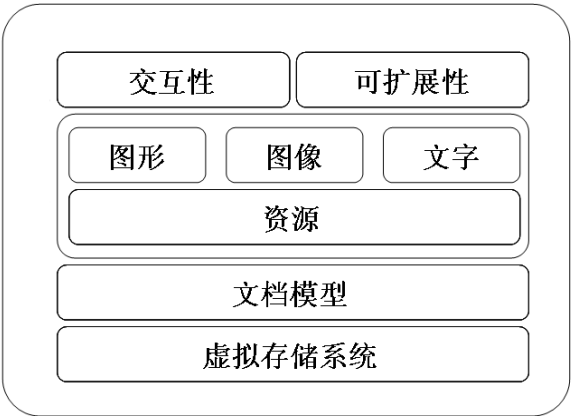


图 1 OFD 基本技术架构

OFD 文档格式的基本技术架构分为四层：

- a) 虚拟存储系统：包括包组织结构及包内目录组织结构；
- b) 文件模型：包括文档、页面、大纲、文件级资源等逻辑组织结构；
- c) 页面内容描述：包括页面级资源、图形、图像和文字等；
- d) 扩展特性:包括文档的交互性和可扩展性等。

5.3 成像模型

OFD 采用二维向量图像模型，可以描述任何经过精密排版的像素，包括文字、图形、图像等页面元素。成像模型应是高效和设备无关的，并可满足打印、显示等各种输出需求。

OFD 成像模型首先根据页面描述生成一个设备无关的输出结果描述，然后，执行程序可将它展示到任何的输出设备上。

页面绘制对象存在以下几种情况：

- a) 绘制对象可以是文字轮廓、用直线和曲线定义的区域、图像等。
- b) 像素可以使用任何颜色绘制。单色、渐变和底纹都属于颜色的一种。
- c) 所有像素都可以被裁剪。

页面内容包含一系列的图层、页面块和图元对象。在输出页面时，应以空白页开始绘制，依次执行页面内容的像素绘制操作。

图元对象应由其自有数据描述及其修饰参数构成，修饰参数使用“属性+绘制参数”的模式。页面中三种最基本的图元对象如下：

- a) 图形对象：由一系列的贝塞尔曲线(最高 3 阶)和圆弧组成，最终形成一个区域。图形对象可以被填充或者描边。
- b) 文字对象：由一系列的字符及其定位信息组成。每个字符的字形由其指定的字体所确定。文字对象可以被填充或者描边。
- c) 图像对象：由一个矩形区域的像素值组成，每个像素值确定矩形区域一个指定点的颜色值。

绘制参数是指精确修饰图元对象绘制渲染效果所需的特性，包括填充颜色(FillColor)、描边颜色(StrokeColor)、线宽(LineWidth)、虚线样式(重复样式 DashPattern 和偏移值 DashOffset)、结合点样式(Join)、端点样式(Cap)、结合点限值(MiterLimit)。这些特性可以作为像素的直接修饰属性，也可以作为被多个像素共同引用的绘制参数资源。

像素绘制时除需要绘制参数所包含的参数外，还需要一些其它参数：

- a) 图元对象包含一个可选的坐标变换参数。坐标变换综合描述了平移、缩放、旋转、切变等特性，将影响图元对象的最终绘制结果；
- b) 图元对象包含一个可选的裁剪区对象。裁剪区确定了图元对象的哪些部分将被绘制到页面上。图元对象落在裁剪区以外的部分将不被绘制。

5.4 扩展名

本标准规定打包后的档扩展名为 ofd。

6 文件结构

6.1 打包

6.1.1 XDA 包组织结构方案

容器及线性化等功能由一个 XDA 文件来执行，XDA 打包方案见附录 A。

6.1.2 ZIP 包组织结构方案

容器及线性化等功能由一个 ZIP 文件来执行。本标准仅可使用 ZIP 规范中默认的 Deflate 压缩方法；多文件的数据组织方式采用 [ZIP](#) 规范 6.2.0。

6.2 文件组织

OFD 文件的层次组织结构如图 2 所示，表 1 对该图进行了说明。

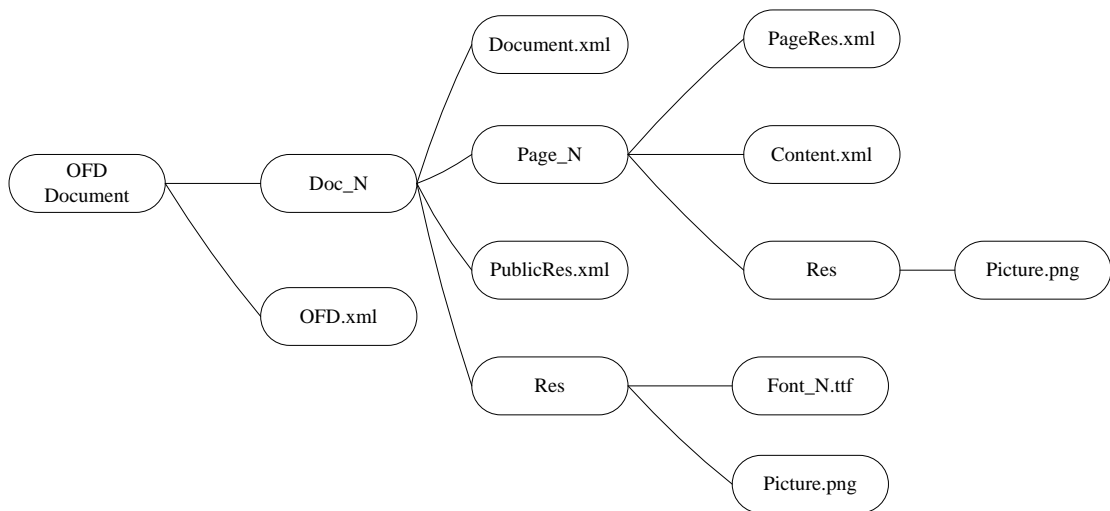


图 2 OFD 目录层次结构

表 1 OFD 目录层次结构说明

名称	说明
OFD. xml	文件主入口文件，一个包内存在且只存在一个 OFD. xml 文件，此文件名不应修改
Doc_N	第 N 个文档的目录
Document. xml	文档的根节点
PageN	第 N 页目录
Content. xml	第 N 页的内容描述
Res. xml	第 N 页的资源描述
Res	资源目录
PublicRes. xml	文档公共资源索引
Picture. png/Font_N. ttf	资源文件

7 基本结构

7.1 命名空间

本标准中 XML 档使用的命名空间为 <http://www.ofdspec.org>，其标识符宜为 ofd；宜在包内各 XML 档的根节点中声明 defaults:ofd。元素节点宜使用命名空间标识，元素属性不使用命名空间标识。

7.2 字符编码

本标准中 xml 文件编码方式采用 UTF-8，见 GB13000。

7.3 基础数据类型

本标准中定义了 6 种基本数据类型，见表 2。

表 2 基本数据类型

类型	说明	示例
ST_Loc	包结构内文件的路径，“.”表示当前路径，“..”表示父路径。 【约定】 1、“/”代表根节点 2、未显式指定时代表当前路径 3、路径区分大小写	“/Pages/P1/Content.xml” “./Res/Book1.jpg” 当前路径 “../Pages/P1/Res.xml” “Pages/P1/Res.xml” 当前路径
ST_Array	数组，以空格来分割元素。元素可以是除 Loc、Array 外的数据类型，不可嵌套，在描述时可能会有元素数量和范围的限定	“1 2.0 5.0”
ST_ID	标识，无符号整数，必须在文档内全局唯一。0 表示无效标识	“1000”
ST_RefID	标识引用，无符号整数，应文档内定义了此标识	“1000”
ST_Pos	点坐标，以空格分割，前者为 x 值，后者为 y 值，可以是整数或者浮点数	“0 0”
ST_Box	矩形区域，以空格分割，前两个值代表了该矩形的左上角的坐标，后两个值依次表示该矩形的宽和高，可以是整数或者浮点数，后两个值必须大于 0	“10 10 50 50”

7.4 主入口

OFD.xml 文件的结构描述如图 3 所示，OFD 元素的属性描述见表 3。

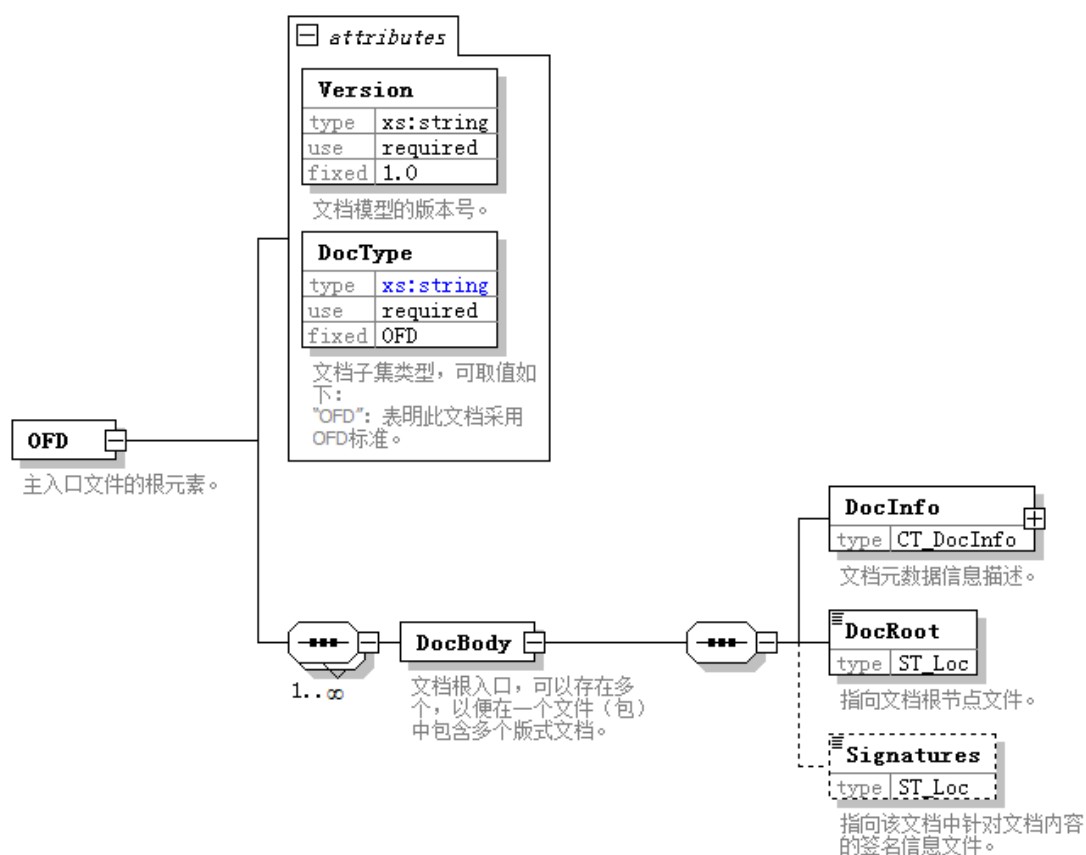


图 3 OFD 元素结构

表 3 OFD 元素属性

名称	类型	说明	备注
Version	xs:string	文件模型的版本号，该版本取值“1.0”	必需
DocType	xs:string	文档子集类型，取值为“OFD”，表明此文档采用 OFD 标准	必需
DocBody		文件对象入口，可以存在多个，以便在一个档中包含多个版式文档	必需
DocInfo	CT_DocInfo	文件元数据信息描述，具体见图 4	可选
DocRoot	ST_Loc	指向文档根节点档，有关文档根节点详细描述见 7.5	可选
Signatures	ST_Loc	指向该文档中针对文件内容的签名信息文件，见附录 C	可选

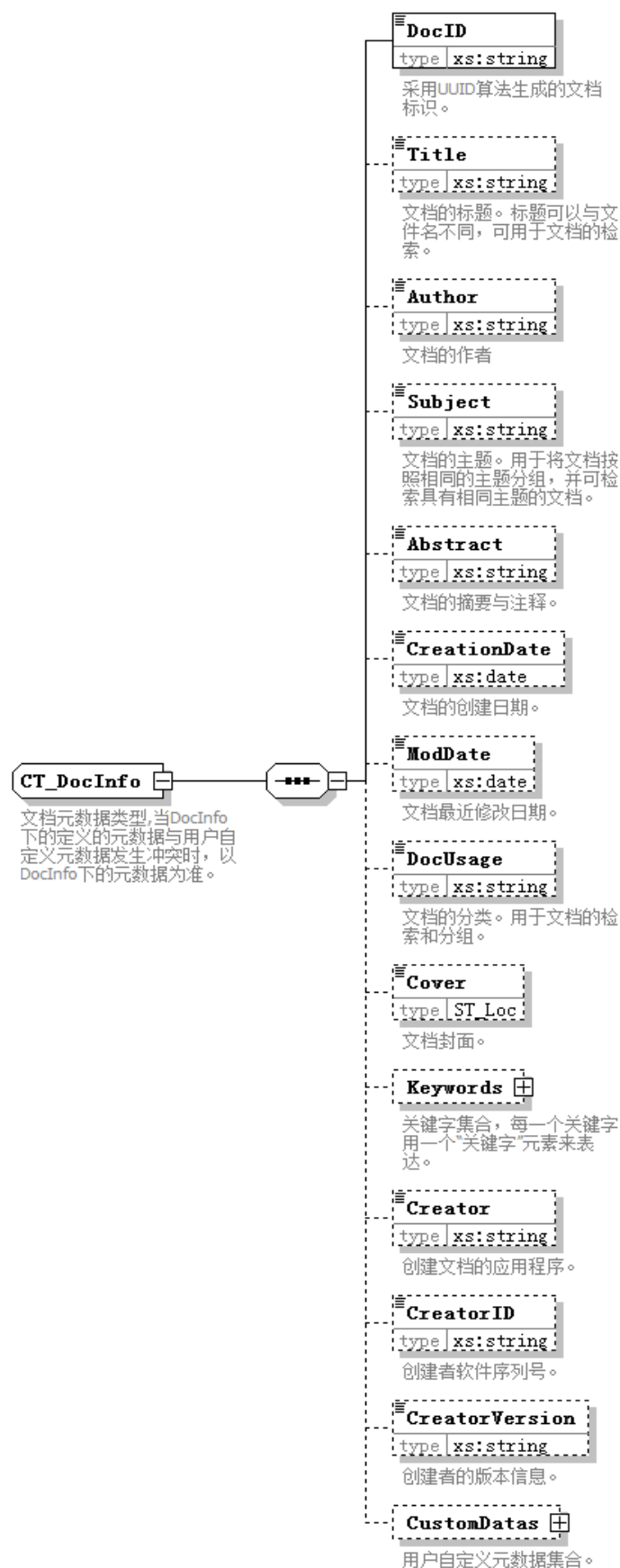


图 4 文件元数据结构

表 4 文件元数据属性

名称	类型	说明	备注
DocID	xs:string	采用 UUID 算法生成的由 32 个字符组成的文件标识。每个 DocID 在文档创建或生成的时候进行分配，取值全球唯一	可选
Title	xs:string	文档的标题。标题可以与文件名不同	可选
Author	xs:string	文档的作者	可选
Subject	xs:string	文档的主题	可选
Abstract	xs:string	文档的摘要与注释	可选
CreationDate	xs:date	文档的创建日期	可选
ModDate	xs:date	文档的最近修改日期	可选
DocUsage	xs:string	文档的分类，可取值如下： Normal — 普通文档 Book — 电子图书 NewsPaper — 电子报纸 Magzine — 电子期刊杂志	可选
Cover	ST_Loc	文档的封面，此路径指向一个图片文件	可选
Keywords		关键词集合，每一个关键词用一个“Keyword”子节点来表达	可选
Keyword	xs:string	关键词	必需
Creator	xs:string	创建文档的应用程序	可选
CreatorVersion	xs:string	创建文档的应用程序的版本信息	可选
CustomDatas		用户自定义元数据集合。其子节点为 CustomData	可选
CustomData	xs:string	用户自定义元数据，可以指定一个名称及其对应的值	必需

7.5 文档根节点

文档根节点的结构如图 5 所示：

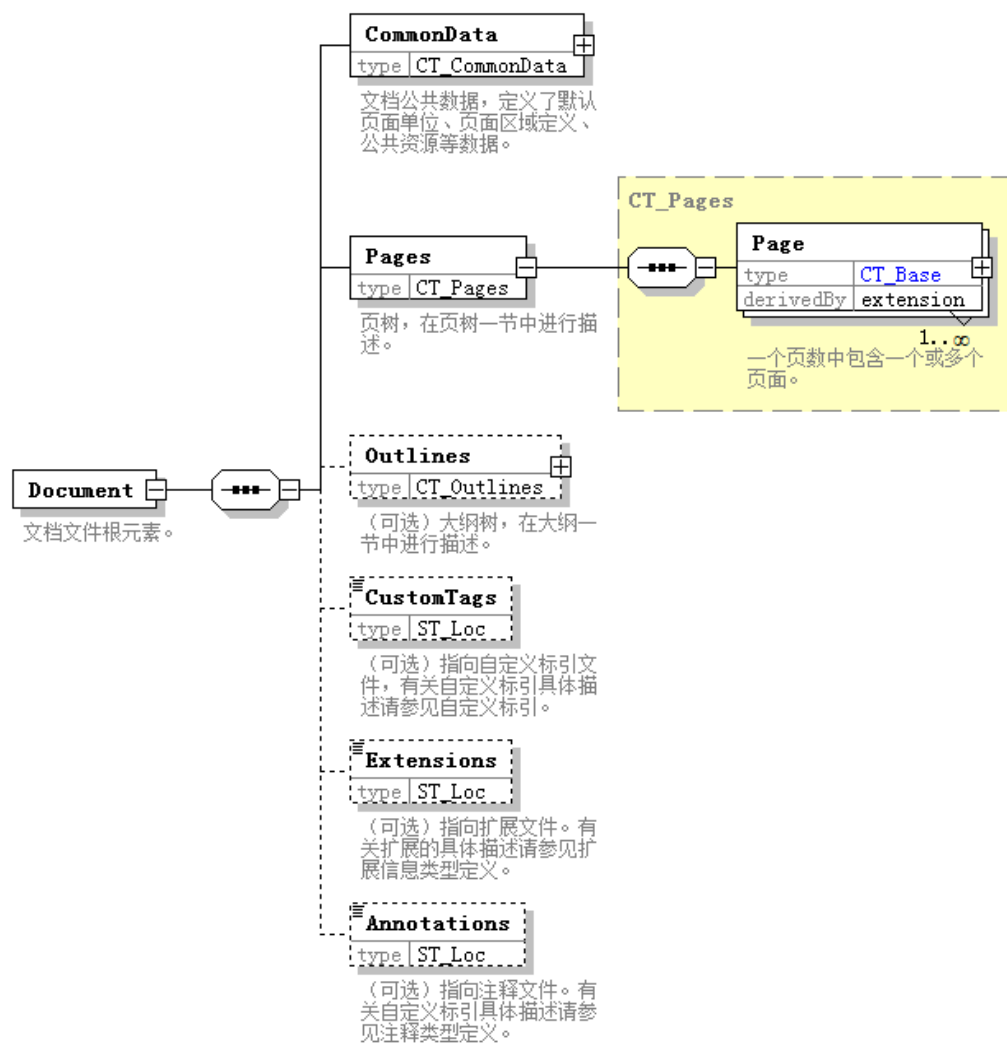


图 5 文件元素结构

表 5 文件元素属性

名称	类型	说明	备注
CommonData		文档公共数据，定义了页面区域、公共资源等数据	必需
Pages		页树，有关页树的具体描述见 7.6	必需
Outlines		大纲，有关大纲的具体描述见 7.8	可选
CustomTags	ST_Loc	指向自定义标引文件，有关自定义标引具体描述请见 17	可选
Extensions	ST_Loc	指向扩展文件。有关扩展的具体描述见 18	可选
Annotations	ST_Loc	指向注释文件。有关自定义标引具体描述见 16	可选

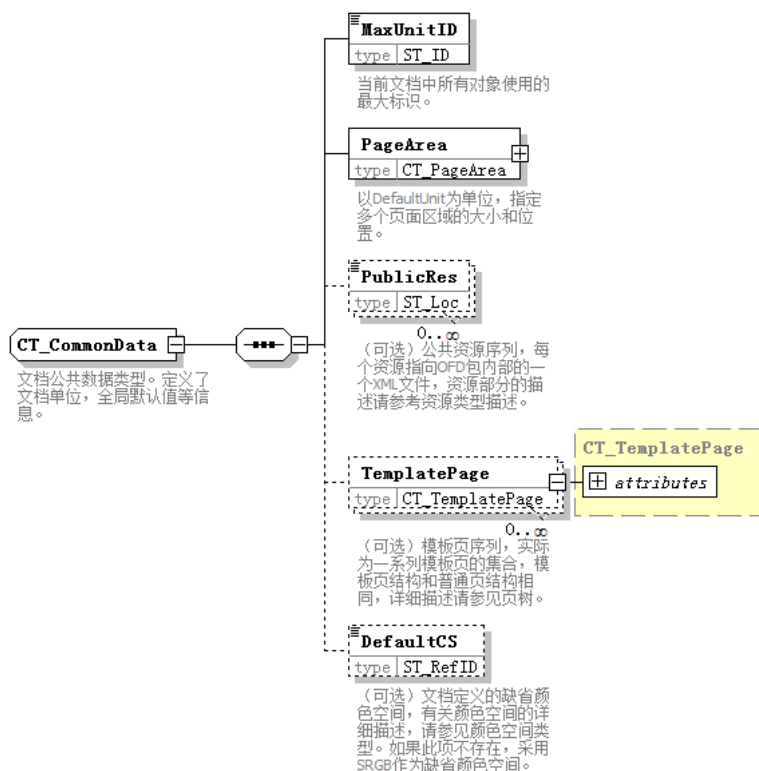


图 6 文档公共数据结构

表 6 文档公共数据属性

名称	类型	说明	备注
MaxUnitID	ST_ID	当前文档中所有对象使用标识的最大值。MaxUnitID 主要用于文档编辑，在向文档中新增加一个对象时，需要分配一个新的标识，新标识取值宜为 MaxUnitID + 1，同时需要修改此 MaxUnitID 值	必需
PageArea		指定多个页面区域的大小和位置。页面区域结构见图 7，区域描述见 9.4	必需
PublicRes	ST_Loc	公共资源序列，每个资源指向 OFD 包内部的一个 XML 档，资源部分的描述见 7.9	可选
TemplatePage		模板页序列，为一系列模板页的集合，模板页结构和普通页结构相同，详细描述见 7.7	可选
DefaultCS	ST_RefID	文档定义的默认颜色空间，有关颜色空间的详细描述，见 8.3.1。如果此项不存在，采用 sRGB 作为默认颜色空间	可选

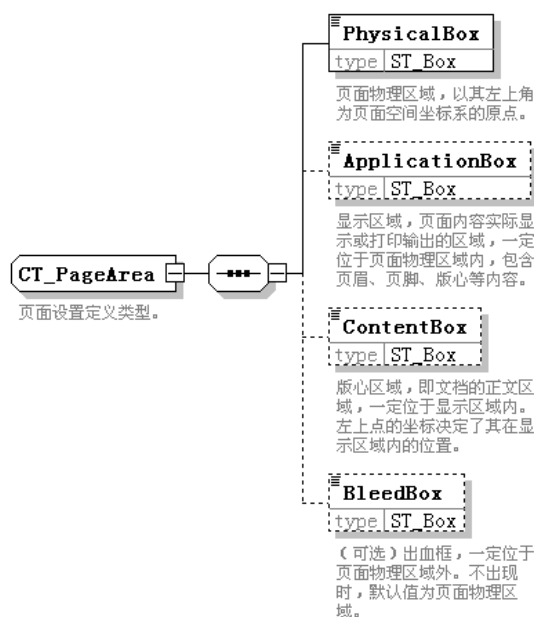


图 7 页面区域结构

表 7 页面区域属性

名称	类型	说明	备注
PhysicalBox	ST_Box	页面物理区域，左上角的坐标为页面空间坐标系的原点	必需
ApplicationBox	ST_Box	显示区域，页面内容实际显示或打印输出的区域，位于页面物理区域内，包含页眉、页脚、版心等内容 [例外处理]如果显示区域不完全位于页面物理区域内，页面物理区域外的部分则被忽略。如果显示区域完全位于页面物理区域外，则该页为空白页	可选
ContentBox	ST_Box	版心区域，即文件的正文区域，一定位于显示区域内。左上角的坐标决定了其在显示区域内的位置。 [例外处理]如果版心区域不完全位于显示区域内，显示区域外的部分则被忽略。如果版心区域完全位于显示区域外，则版心内容不被绘制	可选
BleedBox	ST_Box	出血框，即超出设备物理限制的额外出血区域，位于页面物理区域外。不出现时，默认值为页面物理区域。 [例外处理]如果出血框不完全位于页面物理区域外，页面物理区域内的部分则被忽略。如果出血框完全位于页面物理区域内，出血框无效	可选

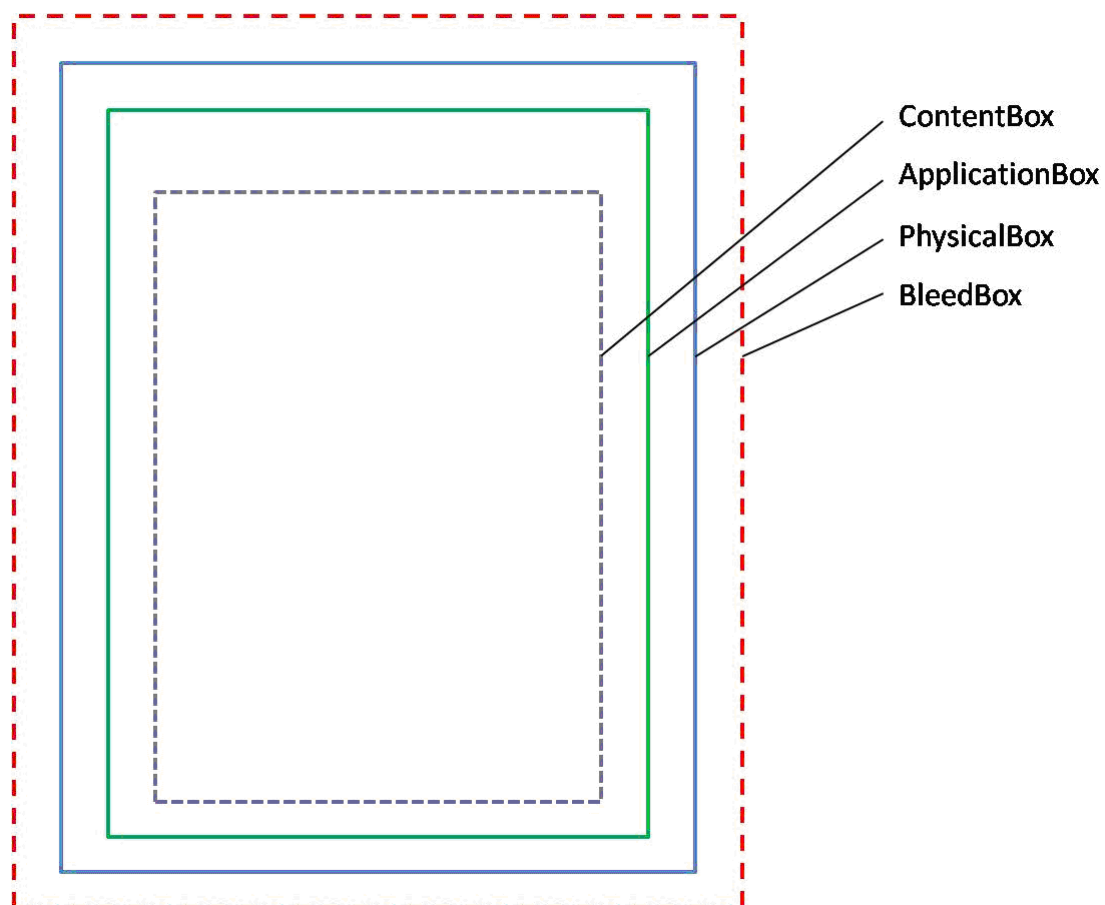


图 8 页边界示意图

7.6 页树

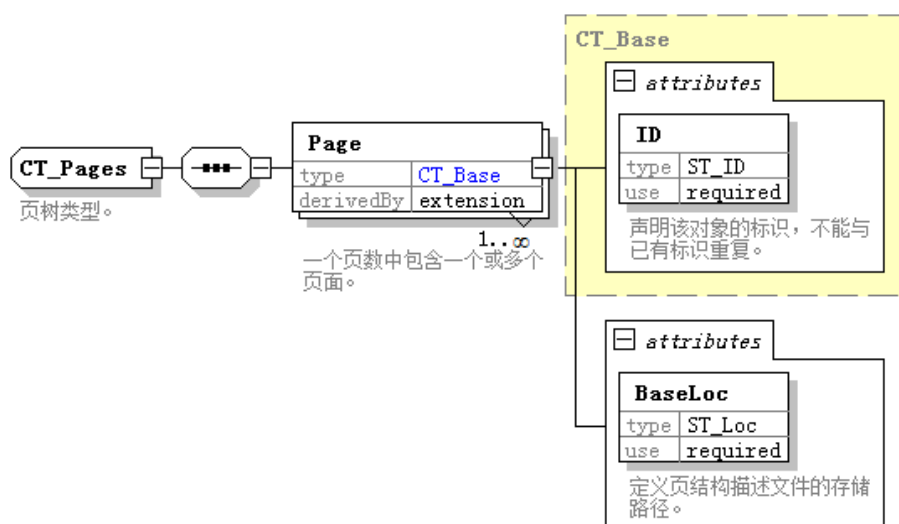


图 9 页树结构

表 8 页树属性

名称	类型	说明	备注
Page		页节点。一个页树中可以包含一个或多个页节点，页顺序是根据页树进行前序遍历时叶节点的访问顺序	必需
ID	ST_ID	声明该页的标识，不能与已有标识重复	必需
BaseLoc	ST_Loc	指向页结构描述文件	必需

7.7 页结构

页结构支持模板页描述，模板页表示每一页经常需要重复显示的内容，用户可以定义多个模板页，通过使用模板页可以使重复显示的内容不必出现在描述每一页的页结构中，每一页通过 Template 节点指定需要使用的模板页，可节省存储空间并减少发生错误的可能。

页对象结构如图 10 所示，各节点的说明见表 9。

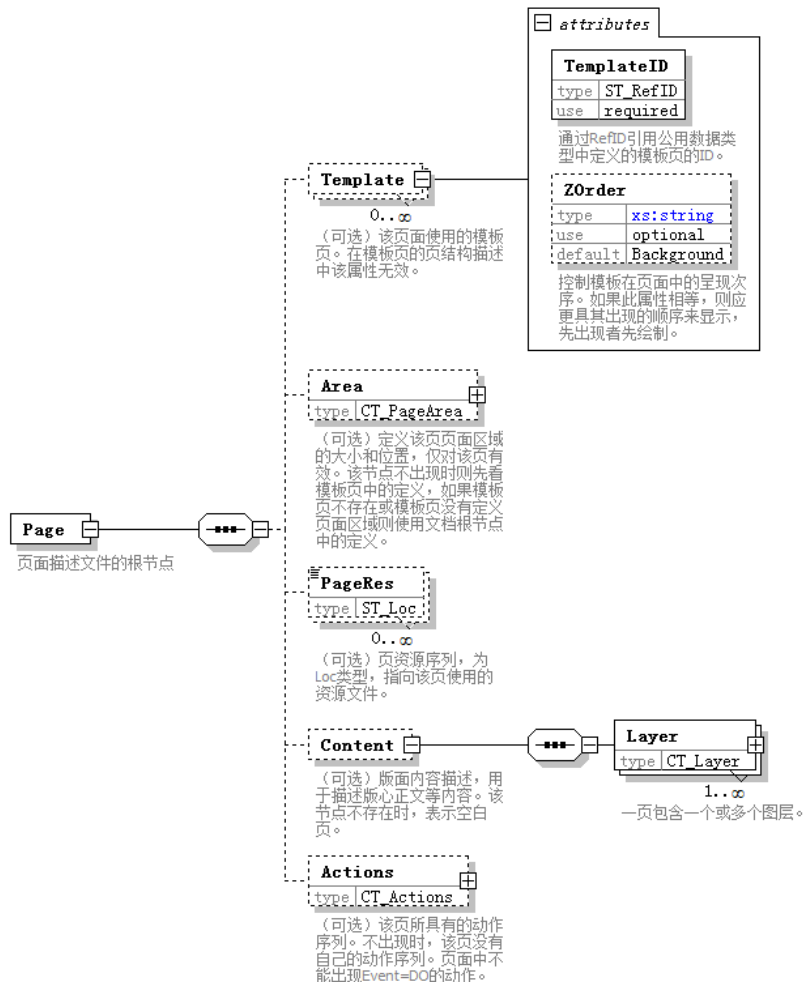


图 10 页结构

表 9 页属性列

名称	类型	说明	备注
----	----	----	----

Template		该页所使用的模板页。模板页的结构和页结构相同，定义在 CommonData 指定的档中，一个页可以使用多个模板页，使用时通过 TemplateID 来指定具体的模板，并通过 ZOrder 属性来控制模板在页面中的呈现顺序。 在模板页的页结构描述中该属性无效	可选
TemplateID	ST_RefID	通过 ST_RefID 引用公用数据类型中定义的模板页	必需
ZOrder	xs:string	控制模板在页面中的呈现顺序，其类型描述和呈现顺序与 Layer 中 Type 的描述和处理一致，见表 10。如果此属性相等，则应根据其出现的顺序来显示，先出现者先绘制。默认值为 Background	可选
Area	CT_PageArea	定义该页页面区域的大小和位置，仅对该页有效。该节点不出现时则使用模板页中的定义，如果模板页不存在或模板页中没有定义页面区域则使用文件 CommonData 中的定义	可选
PageRes	ST_Loc	页资源，指向该页使用的资源文件	可选
Content		页面内容描述。该节点不存在时，表示空白页	可选
Layer	CT_Layer	层节点。一页包含一个或多个层	可选
Actions	CT_Actions	该页所具有的动作序列。页面中不能出现 DO 的动作	可选

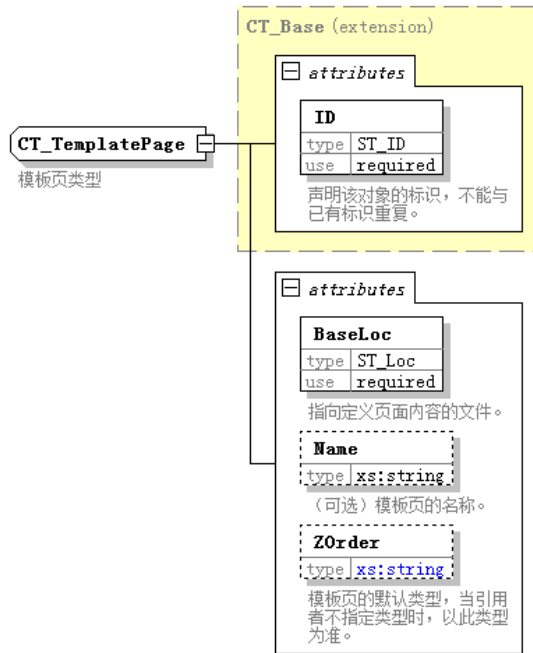


图 11 模板页结构

表 10 模板页属性

名称	类型	说明	备注
ID	ST_ID	模板页的标识，不能与已有标识重复	必需

BaseLoc	ST_Loc	指向模板页结构描述文件	必需
Name	xs:string	模板页名称	可选
ZOrder	xs:string	模板页的默认类型，其类型描述和呈现顺序与 Layer 中 Type 的描述和处理一致，见表 11。如果此属性相等，则应根据其出现的顺序来显示，先出现者先绘制。默认值为 Background	可选

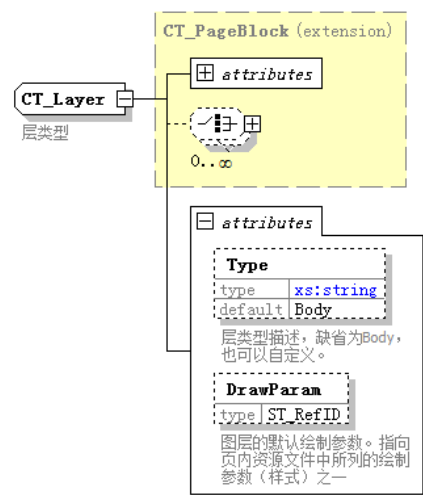


图 12 层结构

表 11 层属性

名称	类型	说明	备注
Type	xs:string	层类型描述，预定义的值见表 12，默认为 Body	可选
DrawParam	ST_RefID	内容流的默认绘制参数	可选

表 12 Type 取值范围

值	说明
Body	正文层
Foreground	前景层
Background	背景层

Background、Body、Foreground 形成了多层内容，这些层按照出现的先后顺序依次进行渲染，每一层的默认颜色采用全透明。层的渲染顺序如图 13。

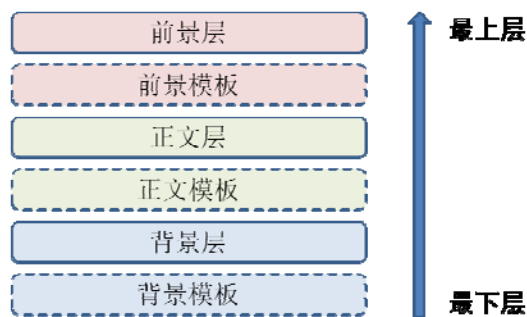


图 13 层顺序示意图

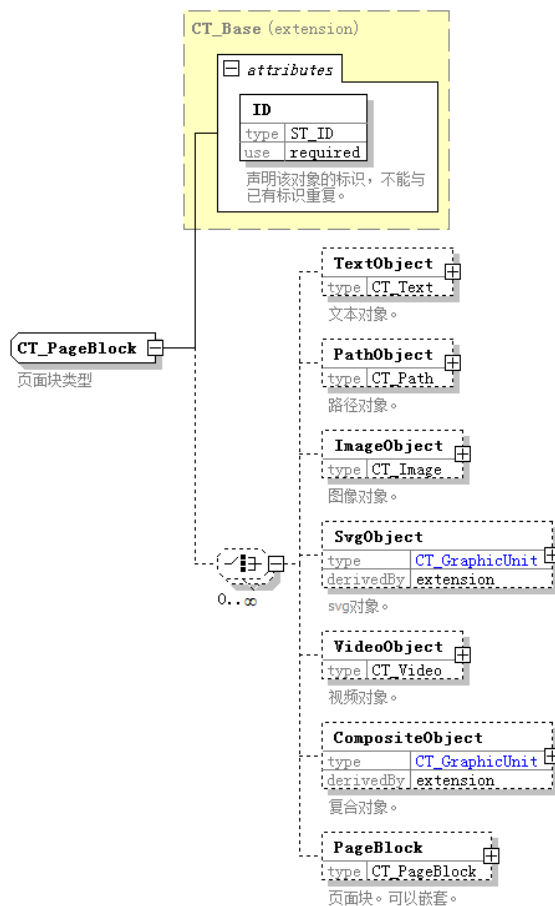


图 14 页面块结构

表 13 页面块属性

名称	类型	说明	备注
ID	ST_ID	声明该页面块的标识，不能与已有标识重复	必需
PageBlock	PageBlock	子页面块。可以嵌套	可选
TextObject	CT_Text	文本对象，见 11.2	可选
PathObject	CT_Path	路径对象，见 9.1	可选
ImageObject	CT_Image	图像对象，见 10	可选
VideoObject	CT_Video	视频对象，见 12	可选

SvgObject		Svg 对象，见 13	可选
CompositeObject		复合对象。该对象自身通过使用符合像素 (CompositeGraphicUnit) 资源描述，见 14	可选

7.8 大纲

大纲按照树状的结构进行组织，每个节点的结构类型为 OutlineElem。大纲根节点结构如图 15 所示：

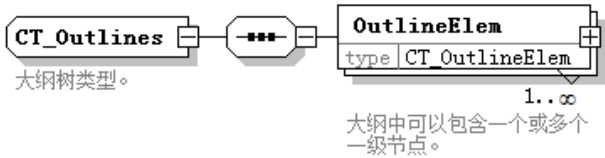


图 15 大纲根结点结构

OutlineElem 节点的结构如图 16 所示，其说明见表 14。

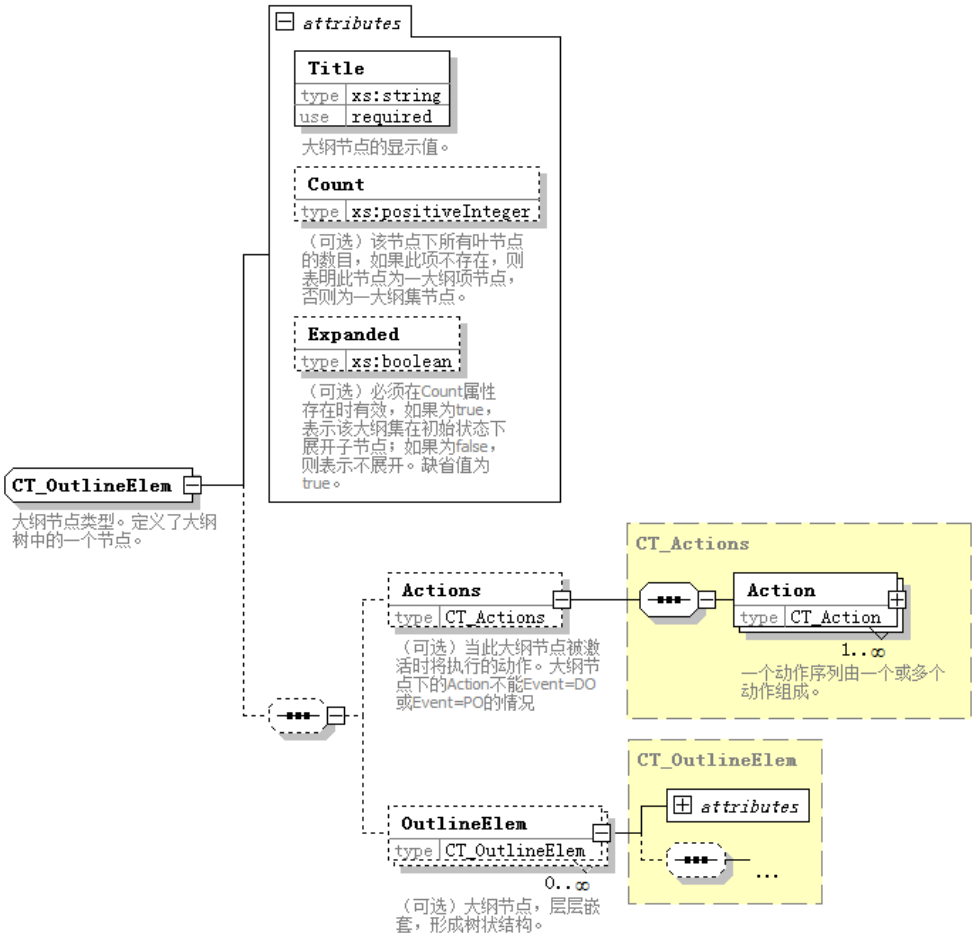


图 16 大纲结构

表 14 大纲结构表

名称	类型	说明	备注
Title	xs:string	大纲节点标题	必需

Count	xs:int	该节点下所有叶节点的数目参考值，应根据该节点下实际出现的子节点数为准。默认值为 0	可选
Expanded	xs:boolean	在有子节点存在时有效，如果为 true，表示该大纲在初始状态下展开子节点；如果为 false，则表示不展开。默认值为 true	可选
Actions	CT_Actions	当此大纲节点被启动时将执行的动作，关于动作的描述见 15	可选
OutlineElem	CT_OutlineElem	大纲节点，层层嵌套，形成树状结构	可选

7.9 资源

资源是对像素绘制时所需数据(如绘制参数、颜色空间、字体、图像、音视频等)的集合。在页面中出现的资源数据内容都保存在容器的特定文件中，通常要求组织在相应的文件夹内，但其索引信息保存在资源文件中。一个文档可能包含一个或多个资源文件。资源根据其作用范围分为公共资源和页资源。公共资源文件在文档根档中进行指定。页资源文件在页结构中进行指定。

资源文件的结构见图 17，各节点对应的内容说明见图 15。

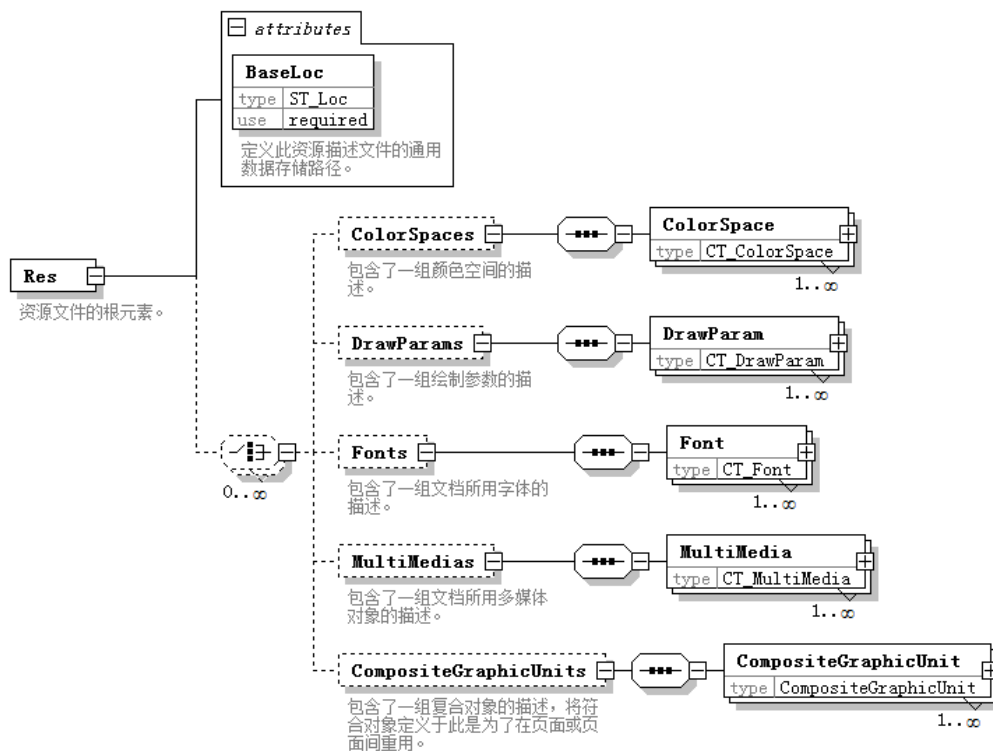


图 17 资源结构

表 15 资源属性

名称	类型	说明	备注
----	----	----	----

BaseLoc	ST_Loc	定义此资源描述文件的通用数据存储路径，BaseLoc 属性的意义在于明确资源数据文件存储的位置，比如 R1.xml 中可以指定 BaseLoc 为“./Res”，表明该资源文件中所有数据文件的默认存储位置在当前路径的 Res 目录下	必需
ColorSpaces		包含了一组颜色空间的描述	可选
DrawParams		包含了一组绘制参数的描述	可选
Fonts		包含了一组文件所用字体的描述	可选
MultiMedias		包含了一组文件所用多媒体对象的描述	可选
CompositeGraphicUnits		包含了一组复合图元的描述	可选

该标准中的资源包含以下类型：

- a) 字体见 11.1；
- b) 颜色空间见 8.3.1；
- c) 绘制参数见 8.2；
- d) 复合对象资源见 14；

复合对象可以使用之前定义的资源；也可以使用公共资源；

- e) 多媒体

多媒体结构的描述见图 18 错误！未找到引用源。所示，多媒体结构的描述见表 16。

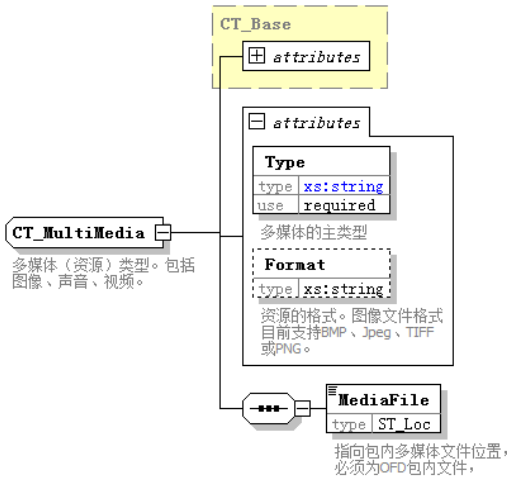


图 18 多媒体结构

表 16 多媒体属性

名称	类型	说明	备注
Type	xs:string	多媒体的主类型。支持 Image, Video, Audio 三种多媒体类型	必需
Format	xs:string	资源的格式。图像主类型的文件格式支持 BMP、Jpeg、TIFF 及 PNG，其中 TIFF 格式不支持多页	可选
MediaFile	Loc	指向多媒体文件。必须为 OFD 包内的档	可选

8 页面描述

8.1 坐标系统

页面中的所有呈现元素都在坐标空间内进行描述。一个坐标空间包括坐标原点、轴的方向、坐标单位的实际长度三个要素。坐标空间根据用途不同分为设备空间、页面空间、对象空间三类。不同的坐标空间之间通过平移、缩放、旋转、切变进行变换。

8.1.1 设备空间

页面中的内容最终需要呈现在某一设备上。而每个设备都会拥有自己的坐标空间以便在自己的绘制区域内能够正确绘制每个像素。设备本身的坐标空间就称之为设备空间。

设备空间的原点、轴方向与坐标单位的实际长度都会由于设备不同而有很大的差异，所以为了屏蔽设备差异，页面的内容不应直接在设备空间上描述。

8.1.2 页面空间

页面空间是一个设备无关的坐标空间系统，用来承载呈现的像素和其它页面要素。

页面空间规定页面的左上角为原点，X轴向右增长，Y轴向下增长，以mm为单位。整个页面空间的大小由PageArea节点(见7.5文档根节点[错误!未找到引用源。](#))中的PhysicalBox确定。页面空间根据原点平移、轴方向变换、坐标数值变换等来完成到设备空间的变换。其中坐标数值变换就是将像素的长度数据通过设备的分辨率(像素每英寸，Dot Per Inch，简称DPI)和其它信息换算成设备空间中的像素长度。

计算机屏幕的显示分辨率一般为96DPI，则页面空间的12.7mm换算到此屏幕上的过程如下：

$$12.7\text{mm} = 0.127\text{cm} = 0.5 \text{ 英寸}$$

$$12.7\text{mm} = 0.5 \text{ 英寸} \times 96 \text{ 像素/英寸} = 48 \text{ 像素}$$

8.1.3 对象空间

图元对象使用其必需的Boundary属性确定在页面或其它容器中的绘制位置。图元对象的内部数据，包括路径数据和裁剪区数据，都以Boundary的左上角为坐标原点，X轴向右增长，Y轴向下增长，并采用mm为单位元，这样的局部坐标空间就称为对象空间。

像素执行绘制时，应首先通过Boundary参数平移到对象空间内，在对象空间内根据变换矩阵和剪切设置进行相应绘制。

8.1.4 变换矩阵

变换矩阵提供了两个坐标空间之间的变换规则，用一个长度为 6 的一维数组描述，形

如“a b c d e f”。变换矩阵是一个 3×3 的矩阵，其格式是 $\begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$ 。假设变换前的坐标是(x, y)，

变换后的坐标是(x',y')，那么满足公式：

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \cdot \begin{bmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{bmatrix}$$

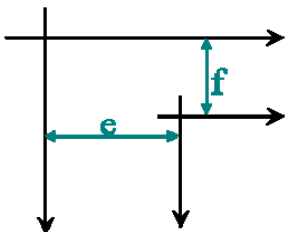
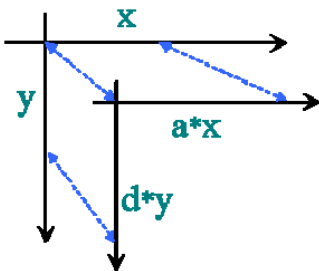
变换矩阵可以实现表 17 中的几种变换效果，这些效果可以相互迭加，迭加的方式通过矩阵乘法实现，但是必须严格按照变换的顺序进行迭加。例如，先将 x 轴放大为原来的两倍，

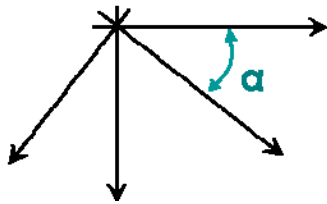
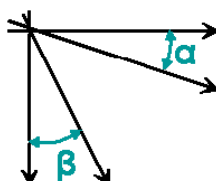
然后旋转是 $\pi/6$ ，那么最终的变换矩阵是 $\begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\pi/6 & \sin\pi/6 & 0 \\ -\sin\pi/6 & \cos\pi/6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ ，与先旋转 $\pi/6$

再将 x 轴放大为原来的两倍获得是变换矩阵 $\begin{bmatrix} \cos\pi/6 & \sin\pi/6 & 0 \\ -\sin\pi/6 & \cos\pi/6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ 是不一样

的。矩阵乘法的结果将会作为最终的变换矩阵进行保存。

表 17 矩阵变换说明表

矩阵	作用
$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ e & f & 1 \end{bmatrix}$	<p>平移。沿 X 轴平移 e 个单位，沿 Y 轴平移 f 个单位。</p> 
$\begin{bmatrix} a & 0 & 0 \\ 0 & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$	<p>缩放。将 X 轴上的一个单位缩放为 a 倍，将 Y 轴上的一个单位缩放为 d 倍。</p> 
$\begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}$	<p>旋转。将点(x, y) 顺着从 X 轴正半部分向 Y 轴正部分的方向旋转α角。</p>

矩阵	作用
	
$\begin{bmatrix} 1 & \tan \alpha & 0 \\ \tan \beta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	<p>切变。将变换后的 Y 轴向 X 轴正半部分歪斜 β 角，将变换后的 X 轴向 Y 轴的正半部分歪斜 α 角。</p> 

图元对象数据经过以下步骤完成向设备坐标系统的变换：

- 图元对象的数据通过像素的变化矩阵，变换到对象空间。
- 对象空间数据通过外接矩形(Boundary)，变换到外部的页面空间。
- 页面空间根据页面区域的大小、坐标单位的实际长度、设备信息变换到设备空间。

8.2 绘制参数

图元对象具有一系列绘制属性(如线宽、线型、颜色、结合点等)用于精确控制绘制渲染效果，被多个图元对象共享的绘制特性应抽取出来统一管理。绘制参数是一组绘制特性参数的集合，可以被不同的图元对象所共享，相当于是图元对象的绘制“样式”。绘制参数的定义可以继承已有的绘制参数，被继承的绘制参数称为该参数的“基础绘制参数”。绘制参数的结构见图 19。

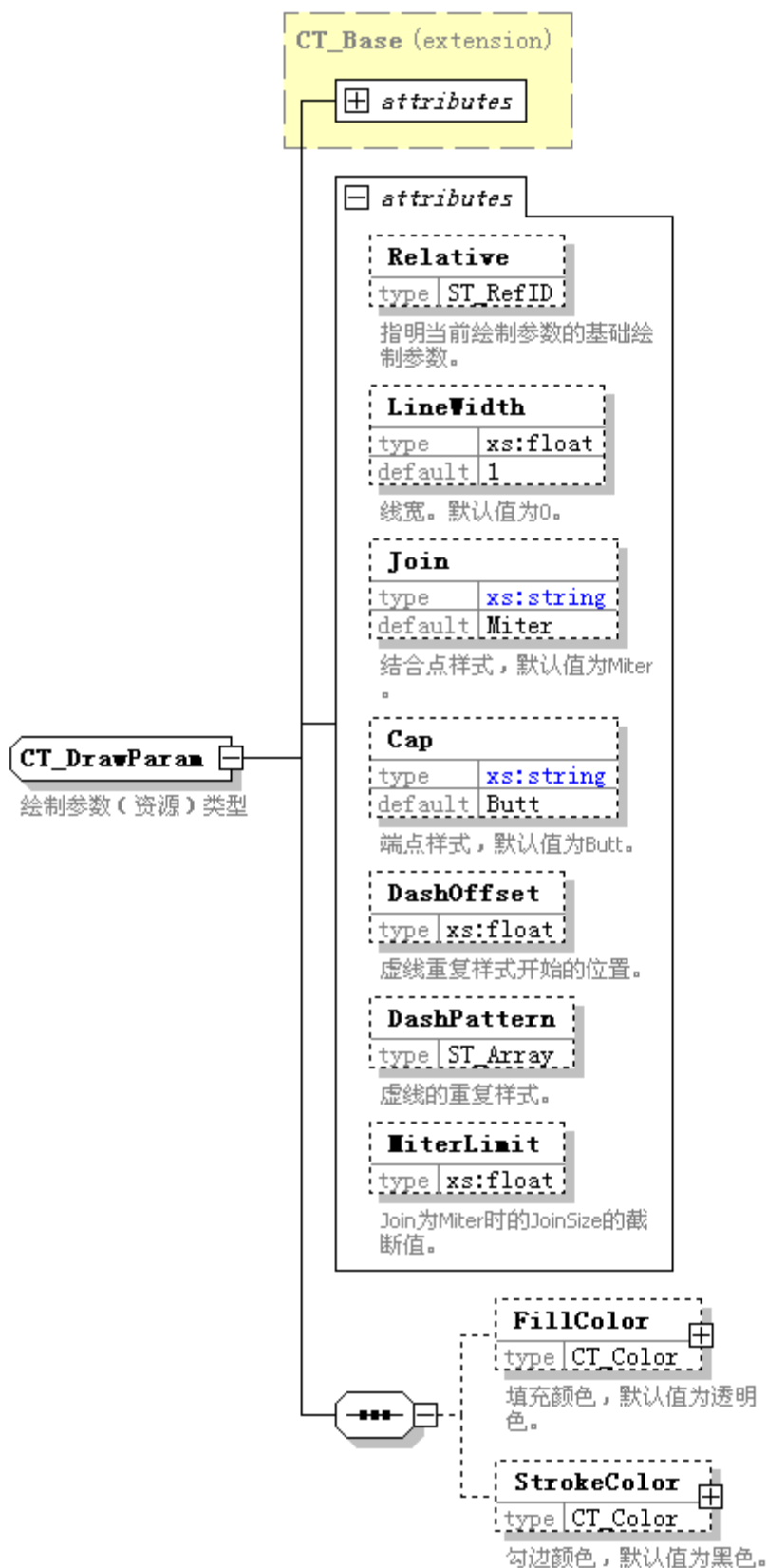


图 19 绘制参数结构

绘制参数有自己的标识，图元对象通过这个标识来引用绘制参数。图元对象在引用绘制参数的同时，还可以有自己的绘制特性属性，像素自有的绘制特性属性将覆盖其引用的绘制参数中的同名特性。

绘制参数可通过基础绘制参数的方式形成嵌套，对单个参数而言，它继承了其基础绘制参数中的所有特性，并且可以重定义其中的基础绘制参数中的特性。




表 18 说明了绘制参数中各节点含义。

表 18 绘制参数属性

属性	类型	说明	备注
ID	ST_ID	声明该绘制参数的标识，不能与已有标识重复	必需
Relative	ST_RefID	指明当前绘制参数的基础绘制参数	可选
Join	xs:string	结合点，指定了两个线的端点结合时采用的样式。默认值为 Miter。 Join 属性的控制效果见表 26	可选
LineWidth	xs:double	线宽，非负浮点数，指定了路径绘制时线的宽度。由于某些设备不能输出一个像素宽度的线，因此强制规定当线宽大于 0 时，无论多小都最少要绘制两个像素的宽度；当线宽为 0 时，绘制一个像素的宽度。由于线宽 0 的定义与设备相关，所以不推荐使用线宽 0。默认值为 1.0	可选
DashOffset	xs:double	虚线重复样式开始的位置，默认值为 0。当 DashPattern 不出现时，该参数无效	可选
DashPattern	ST_Array	虚线的重复样式。默认值为空。虚线样式的控制效果见 XXX,	可选
Cap	xs:string	线端点样式，枚举值，指定了一条线的端点样式。默认值为 Butt。 Cap 属性的控制效果见 XXX	可选
MiterLimit	xs:double	Join 为 Miter 时小角度 JoinSize 的截断值，默认值为 12.0。当 Join 不等于 Miter 时该参数无效	可选
FillColor	CT_Color	填充颜色，用以填充路径形成的区域以及文字轮廓内的区域，默认值为透明色。关于颜色的定义见颜色	可选
StrokeColor	CT_Color	描边颜色，指定路径绘制的颜色以及文字轮廓的颜色，默认值为黑色。关于颜色的定义见颜色	可选

8.2.1 线条连接样式

表 19 线条连接样式

属性	取值	样式
Join	Miter	
	Round	
	Bevel	

8.2.2 线条连接点截断值

MiterLimit 属性是为了限制线条相交时产生的结合点长度，如图 20 所示：

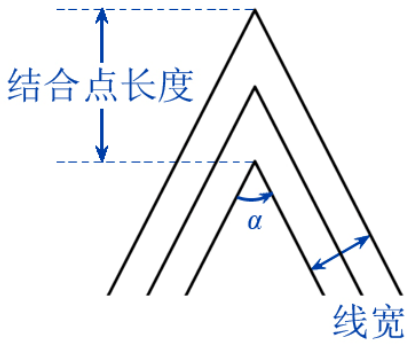



图 20 结合点长度




其中， $\sin\left(\frac{\alpha}{2}\right) = \frac{W}{L}$ ，W 为结合点长度，L 为线宽。

8.2.3 线条的虚线样式

线条的虚线样式通过 DashPattern 和 DashOffset 两个属性进行控制。各种组合如表 20 所示：

表 20 线条的虚线样式




Dash 设置	呈现效果
DashPattern=null	

DashPattern="30 30"	
DashOffset=10 DashPattern="30 30"	
DashPattern="30 15"	
DashOffset=25 DashPattern="15 30"	
DashOffset=50 DashPattern="30 15"	

8.2.4 线条的端点样式

线段样式属性 **Cap** 的取值范围及其呈现效果如表 21 所示。

表 21 线条的端点样式

属性	取值	呈现效果
Cap	Butt	
	Round	
	Square	

8.3 颜色

8.3.1 颜色空间

本标准支持 **Gray**、**RGB**、**CMYK** 颜色空间。除通过通道值使用色域内的任意颜色之外，还可在颜色空间内定义调色板或指定相应的颜色配置文件，供颜色通过索引值引用。颜色空间的结构见图 21，说明见表 22。

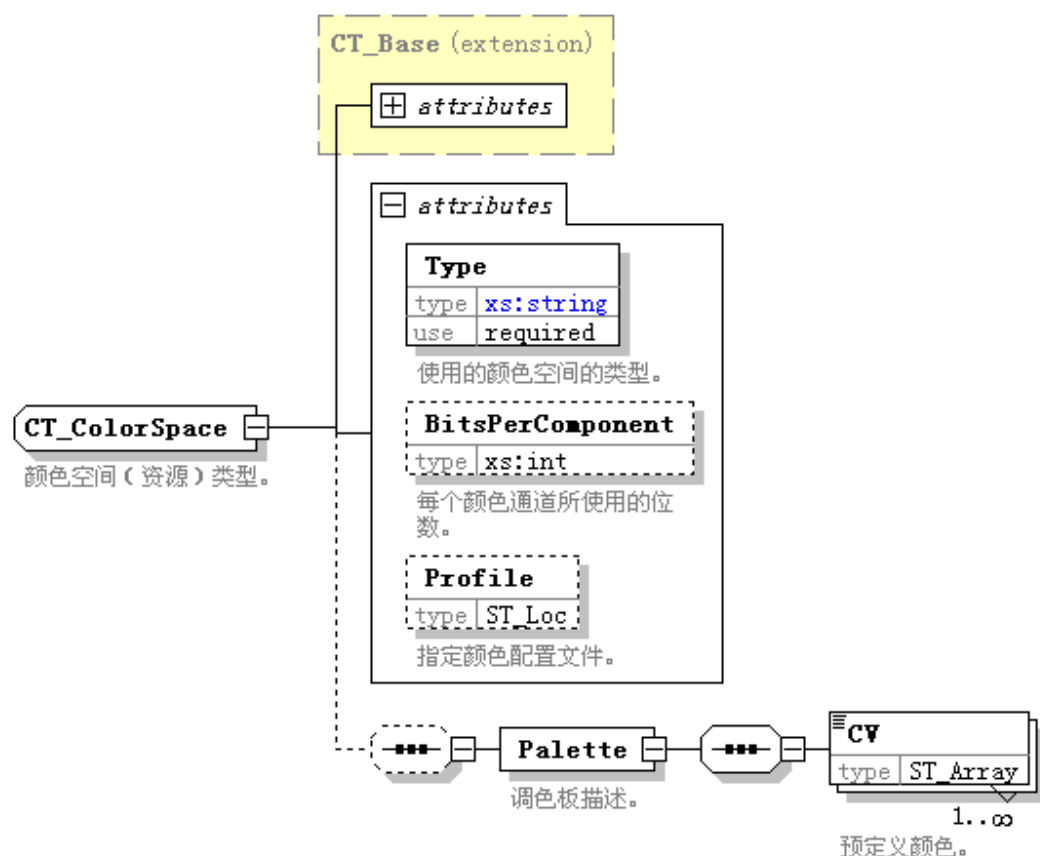


图 21 颜色空间结构

表 22 颜色空间说明

属性	类型	说明	备注
ID	ST_ID	声明该颜色空间的标识，不能与已有标识重复	必需
Type	xs:string	颜色空间的类型。可取值如下：Gray、RGB、CMYK	必需
BitsPerComponent	xs:int	每个颜色通道所使用的位数，默认值为 8	可选
Profile	ST_Loc	指向包内颜色描述档，见规范性引用 ICC Profile	可选
Palette		调色板描述	可选
CV	ST_Array	调色板中预定义颜色，预定义颜色的各通道值被显式指定。调色板中颜色的编号从 0 开始	必需

8.3.2 基本颜色

本规范中定义的颜色是一个广义的概念，包括基本颜色、底纹和渐变，结构见图 22 错误！未找到引用源。，各属性和节点的说明见表 23。

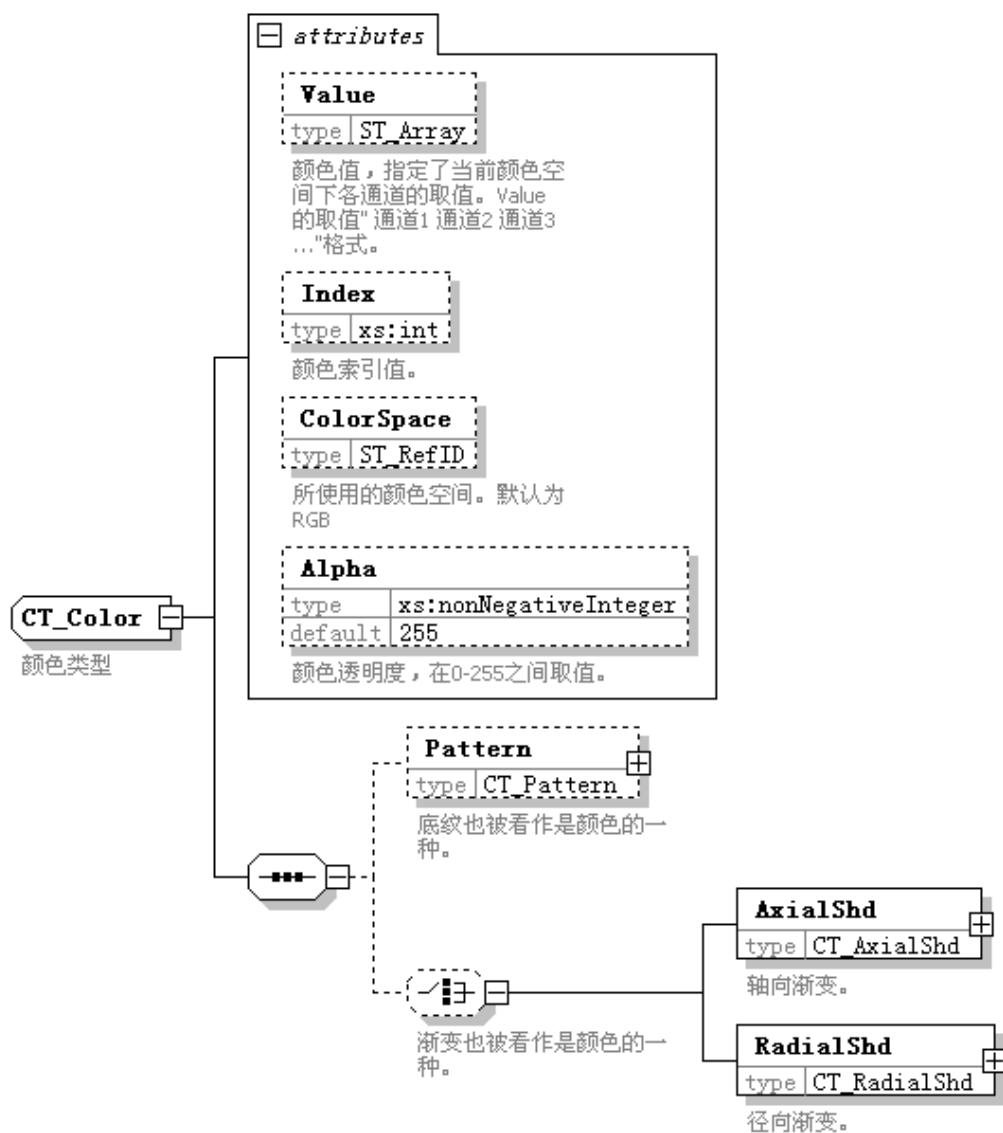


图 22 颜色结构

表 23 颜色说明

属性	类型	说明	备注
Value	ST_Array	颜色值，指定了当前颜色空间下各通道的取值。Value 的取值应符合"通道 1 通道 2 通道 3 ..."格式。此属性不出现时，应参考 Index 属性从颜色空间的调色版中取值。当二者都不出现时，该颜色各通道的值全部为 0。	可选
Index	xs:int	调色板中颜色的编号，非负整数，将从当前颜色空间的调色板中取出相应索引的预定义颜色用来绘制。索引从 0 开始。	可选
Color_Space	ST_RefID	指向所使用的颜色空间。	可选
Alpha	xs:int	颜色透明度，在 0-255 之间取值。默认为 255，表示完全不透明。	可选

节点	类型	说明	备注
Pattern	CT_Pattern	底纹填充，复杂颜色的一种。具体描述见 8.3.3	可选
AxialShd	CT_AxialShd	轴向渐变，复杂颜色的一种。具体描述见 8.3.4.1	可选
RadialShd	CT_RadialShd	径向渐变，复杂颜色的一种。具体描述见 8.3.4.2	可选

简单颜色支持两种指定方式：一种是通过设定颜色各通道值指定色域中的某个颜色，另一种是通过索引值取得颜色空间中的一个预定义颜色。

由于不同颜色空间下，颜色通道的含义、数目各不相同，所以在表 24 中对颜色空间的类型、颜色值的描述格式等做出了详细的说明。BitsPerComponent 有效时，颜色通道值的取值下限是 0，上限由 BitsPerComponent 决定，即取区间 $[0, 2^{\text{BPC}}-1]$ 内的整数，采用 16 进制的形式来表示。当颜色通道的取值超出了相应的区间，则按照这个值最接近的区间的边界进行处理。

表 24 Type 和 BPC 关系

Type	BitsPerComponent	说明
Gray	有效	只包含一个通道来表明灰度值。 例如"#FF"
RGB	有效	包含三个通道，依次是红、绿、蓝。 例如"#11 #22 #33"
CMYK	有效	包含四个通道，依次是 Cyan(青)、Yellow(黄色)、Magenta(品红)、Black(黑色)。 例如"#11 #22 #33 #44"

8.3.3 底纹

底纹是复杂颜色的一种，用于图形和文字的填充以及描边处理。底纹对象属性的结构见图 23，具体说明见表 25：

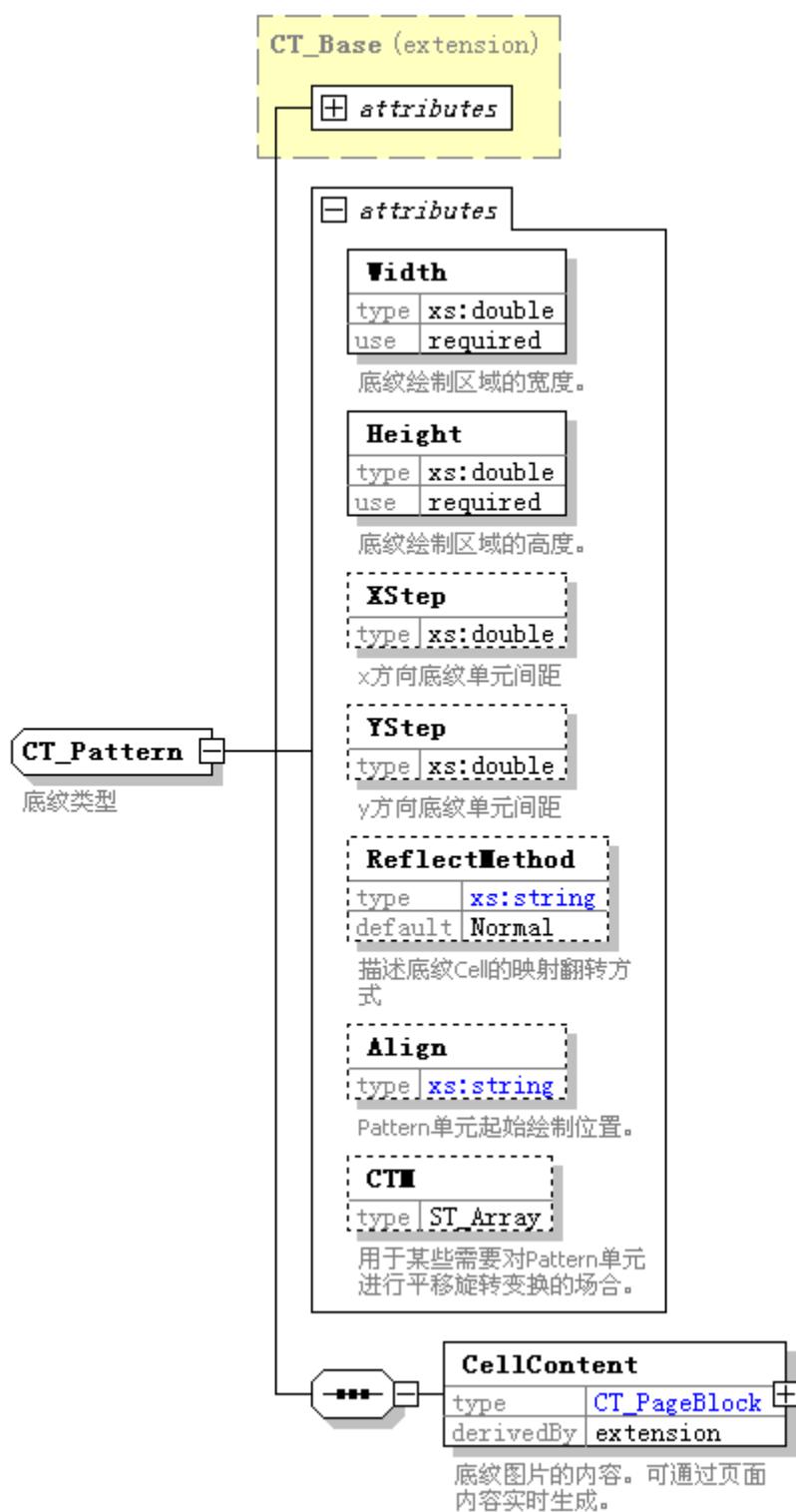


图 23 底纹结构

表 25 底纹说明

名称	类型	说明	备注
Width	xs:double	底纹绘制区域的宽度	必需
Height	xs:double	底纹绘制区域的高度	必需
XStep	xs:double	X 方向底纹单元间距，默认值为底纹单元的宽度	可选
YStep	xs:double	Y 方向底纹单元间距，默认值底纹单元的高度	可选
ReflectMethod	xs:string	描述底纹 Cell 的映像翻转方式，枚举值，默认值为 Normal。	可选
RelativeTo	xs:string	Pattern 单元起始绘制位置。可取值如下 Page: 相对于页面坐标系的原点 Object: 相对于对象坐标系的原点 默认值是 Object	可选
CTM	ST_Array	Pattern 单元的变换矩阵，用于某些需要对 Pattern 单元进行平移旋转变换的场合，默认为单位矩阵；Pattern 呈现时先做 XStep,YStep 排布，然后一起做 CTM 处理	可选
CellContent	CT_PageBlock	底纹单元，用底纹画刷填充目标区域时，所使用的单元对象	必需

翻转示例

底纹以 CellContent 指定的一个向量对象为一个画刷单元，在底纹绘制区域中以此画刷上的内容平铺填满。图 24 为一个图像，底纹绘制将以其为画刷单元对目标区域进行填充。以此为例对 ReflectMethod 属性进行详细的说明：



图 24 ReflectMethod 说明

图 25 分别为 ReflectMethod 取值为 Normal、Column、Row、RowAndColumn 的绘制效果。

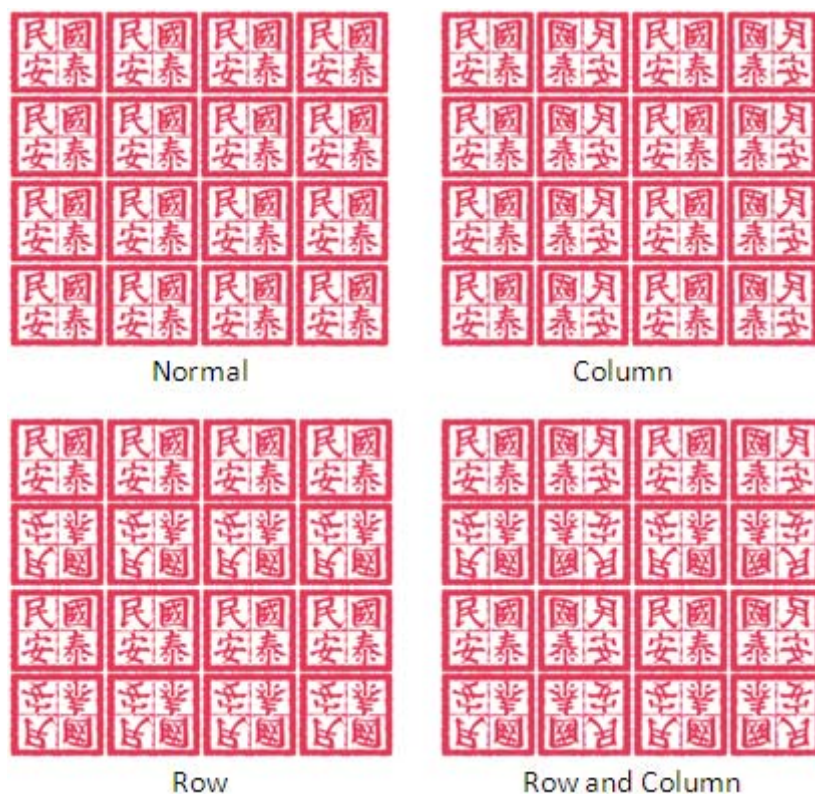


图 25 翻转绘制效果

CellContent 作为底纹对象的绘制单元，使用一种和外界没有任何关联的独立的坐标空间：坐标以左上角为原点，X 轴向右增长，Y 轴向下增长，使用 mm 为单位。

8.3.4 渐变

渐变提供了一种预定义的渲染模式，描述在指定区域内的颜色过渡过程，与具体的输出设备、处理方式和处理过程无关。渐变用于图形和文字的填充以及描边处理。推荐在使用渐变对象的同时使用裁剪区与之相配合，以便用较小的代价描绘出复杂的渲染效果。

本标准支持轴向和径向两种渐变类型。本标准中，渐变区间定义为由起点位置到终点位置的一次颜色渐变。

8.3.4.1 轴向渐变

图 26 所示是一个典型的轴向渐变示例。



图 26 轴向渐变

在轴向渐变中，颜色渐变沿着一条指定的轴线方向进行，轴线由起点和终点决定，与这条轴线垂直的直线上的点颜色相同。

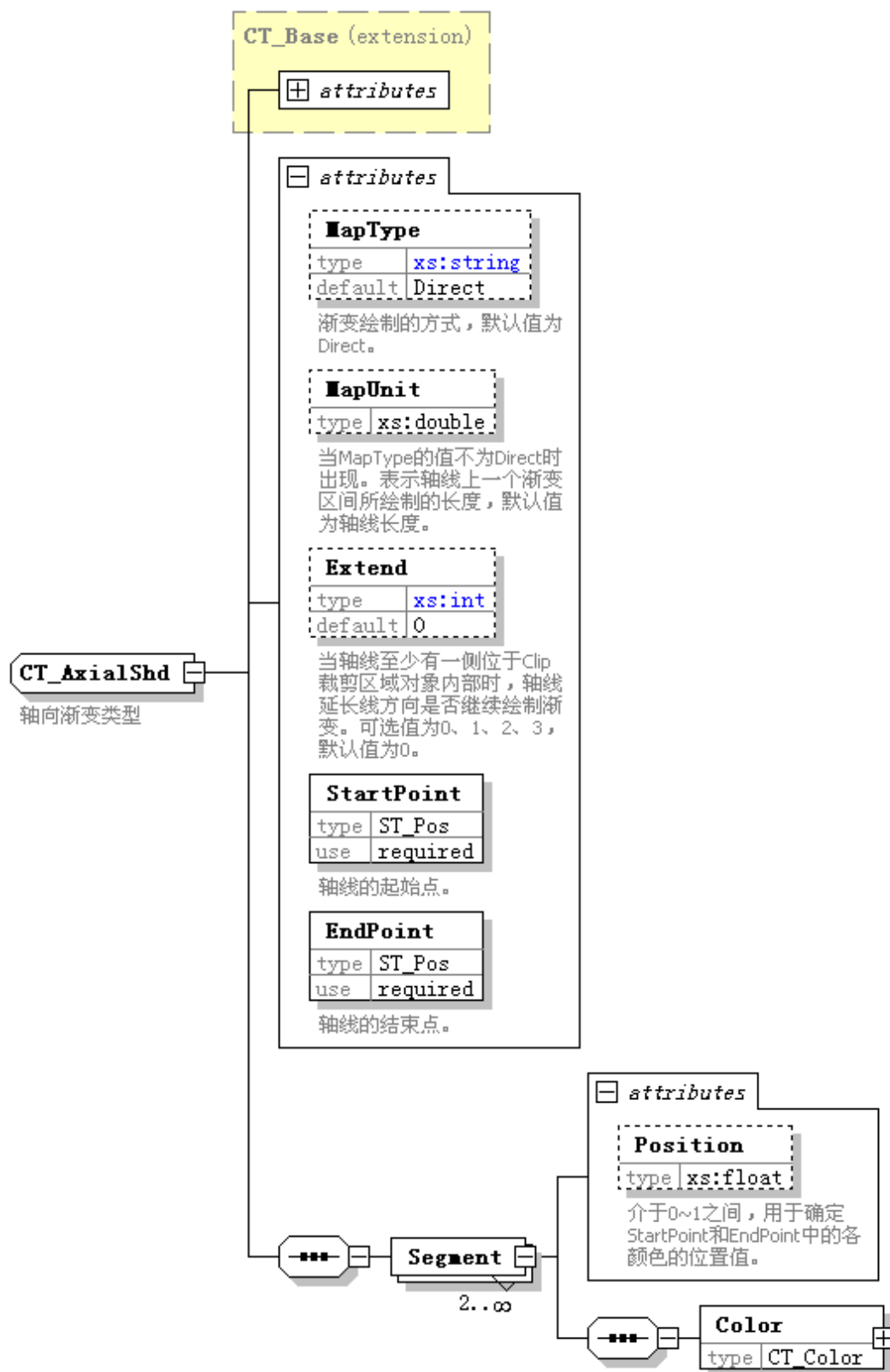


图 27 轴向渐变结构

表 26 轴向渐变说明

名称	类型	说明	备注
----	----	----	----

MapType	xs:string	渐变绘制的方式，可选值为 Direct，Repeat，Reflect，默认值为 Direct	可选
MapUnit	xs:double	轴线一个渐变区间的长度，当 MapType 的值不等于 Direct 时出现，默认值为轴线长度	可选
Extend	xs:int	轴线延长线方向是否继续绘制渐变。可选值为 0、1、2、3。 0：不向两侧继续绘制渐变。 1：在终点至起点延长线方向绘制渐变。 2：在起点至终点延长线方向绘制渐变。 3：向两侧延长线方向绘制渐变。 默认值为 0	可选
StartPoint	ST_Pos	轴线的起始点	必需
EndPoint	ST_Pos	轴线的结束点	必需
Segment		颜色的段，至少出现两个	必需
Position	xs:double	用于确定 StartPoint 和 EndPoint 中的各颜色的位置值，取值范围是[0, 1.0]，各段颜色的 Position 值应根据颜色出现的顺序递增。 第一个 Segment 的 Position 属性默认值为 0，最后一个 Segment 的 Position 属性默认值为 1.0，当不存在时，在空缺区间内平均分配。例如 Segment 个数等于 2 且不出现 Position 属性时，按照“0 1.0”处理；Segment 个数等于 3 且不出现 Position 属性时，按照“0 0.5 1.0”处理；Segment 个数等于 5 且不出现 Position 属性时，按照“0 0.25 0.5 0.75 1.0”处理	可选
Color	CT_Color	该段的颜色，必须是简单颜色	必需

当轴向渐变某个方向设定为延伸时(Extend 不等于 0)，渐变应沿轴在该方向的延长线延伸到超出裁剪区在该轴在线的投影区域为止。延伸的区域的渲染颜色使用该方向轴点的颜色。

轴向渐变的 MapType 示例

以下给出 MapType 的值分别为 Direct，Repeat，Reflect 时的绘制结果。

```

<ofd:PathObject ID="10005" Boundary="10 10 140 40" Fill="true">
  <ofd:FillColor>
    <ofd:AxialShd ID="10007" StartPoint="0 0" EndPoint="140 0">
      <ofd:Segment>
        <ofd:ColorValue="255 255 0" />
      </ofd:Segment>
      <ofd:Segment>
        <ofd:ColorValue="0 0 255" />
      </ofd:Segment>
    </ofd:AxialShd>
  </ofd:FillColor>
  <ofd:AbbreviatedData>M 0 0 L 140 0 L 140 40 L 0 40 C</ofd:AbbreviatedData>
</ofd:PathObject>

```

```

<ofd:PathObject ID="10006" Boundary="10 60 140 40" Fill="true">
  <ofd:FillColor>
    <ofd:AxialShd ID="10007" StartPoint="0 0" EndPoint="140 0" MapType="Repeat" MapUnit="25">
      <ofd:Segment>
        <ofd:ColorValue="255 255 0" />
      </ofd:Segment>
      <ofd:Segment>
        <ofd:ColorValue="0 0 255" />
      </ofd:Segment>
    </ofd:AxialShd>
  </ofd:FillColor>
  <ofd:AbbreviatedData>M 0 0 L 140 0 L 140 40 L 0 40 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10007" Boundary="10 110 140 40" Fill="true">
  <ofd:FillColor>
    <ofd:AxialShd ID="10007" StartPoint="0 0" EndPoint="25 0" MapType="Reflect" MapUnit="25">
      <ofd:Segment>
        <ofd:ColorValue="255 255 0" />
      </ofd:Segment>
      <ofd:Segment>
        <ofd:ColorValue="0 0 255" />
      </ofd:Segment>
    </ofd:AxialShd>
  </ofd:FillColor>
  <ofd:AbbreviatedData>M 0 0 L 140 0 L 140 40 L 0 40 C</ofd:AbbreviatedData>
</ofd:PathObject>

```

上述示例的显示效果如图 28、图 29、图 30 所示：

Direct:



图 28 Direct 渐变效果

Repeat:

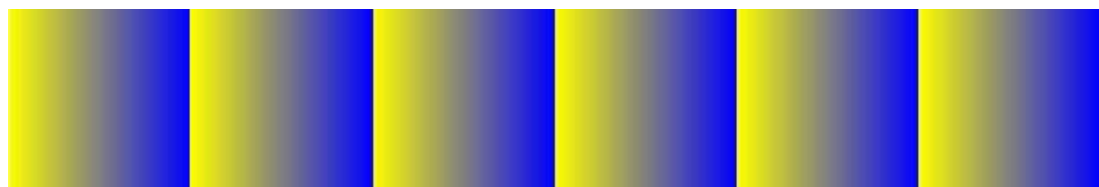


图 29 Repeat 渐变效果

Reflect:

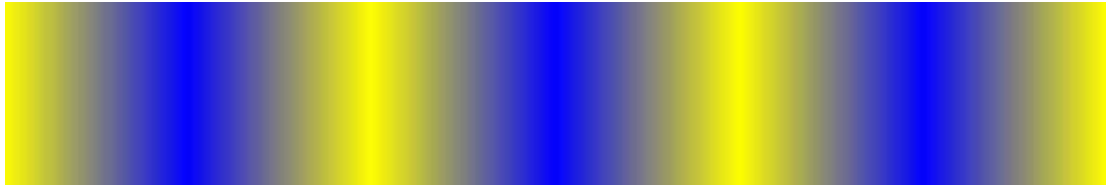


图 30 Reflect 渐变效果

8.3.4.2 径向渐变

图 31 所示是一个典型的径向渐变示例。

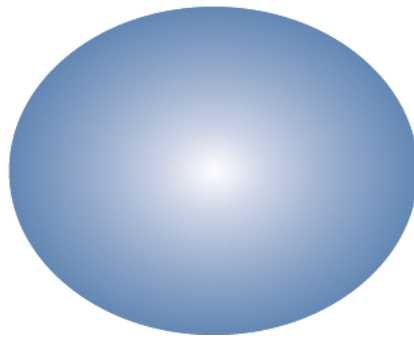


图 31 径向渐变

径向渐变定义了两个离心率和倾斜角度均相同的椭圆，并在其中心点连线上进行渐变绘制。具体方法是，先由起点椭圆中心点开始绘制一个起点颜色的空心椭圆，随后沿着中心点连线不断绘制离心率与倾斜角度相同的空心椭圆，颜色由起点颜色逐渐变为终点颜色，椭圆大小由起点椭圆逐渐变为终点椭圆。

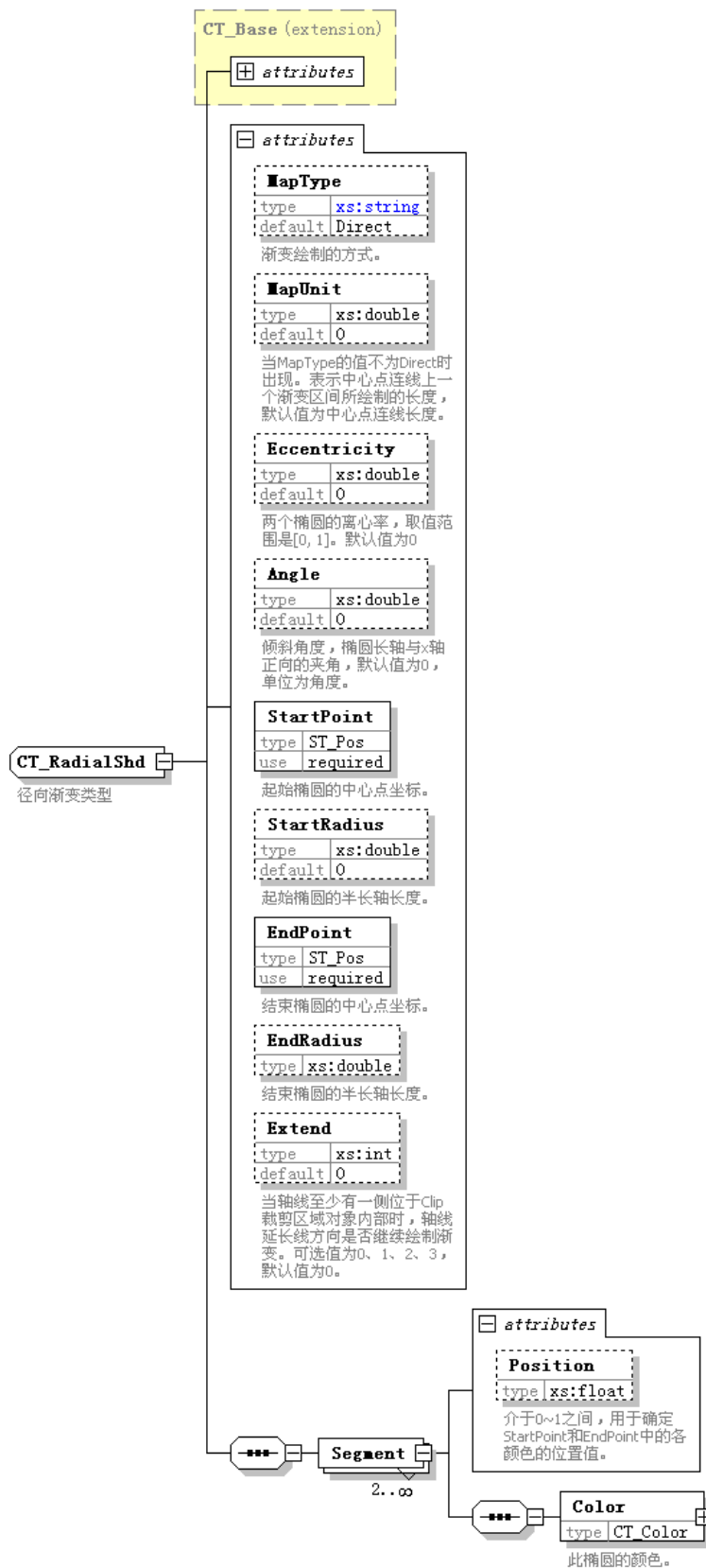


图 32 径向渐变结构图

表 27 径向渐变说明

名称	类型	说明	备注
MapType	xs:string	渐变绘制的方式，可选值为 Direct, Repeat, Reflect，默认值为 Direct	可选
MapUnit	xs:double	中心点连线上一个渐变区间所绘制的长度，当 MapType 的值不为 Direct 时出现。默认值为中心点连线长度	可选
Eccentricity	xs:double	两个椭圆的离心率，即椭圆焦距与长轴的比值，取值范围是[0, 1.0]。默认值为 0，在这种情况下椭圆退化为圆	可选
Angle	xs:double	两个椭圆的倾斜角度，椭圆长轴与 x 轴正向的夹角，单位为度。默认值为 0	可选
StartPoint	ST_Pos	起始圆的的中心点	必需
EndPoint	ST_Pos	结束圆的的中心点	必需
StartRadius	xs:double	起始圆的长半轴，默认值为 0	可选
EndRadius	xs:double	结束圆的长半轴	必需
Extend	xs:int	径线延长线方向是否继续绘制渐变。可选值为 0、1、2、3。 0: 不向圆心联机两侧继续绘制渐变 1: 在终点圆至起点圆延长线方向绘制渐变 2: 在起点圆至终点圆延长线方向绘制渐变 3: 向两侧延长线方向绘制渐变 默认值为 0	可选
Segment		颜色段，至少出现两个	必需
Position	xs:double	用于确定 StartPoint 和 EndPoint 中的各颜色的位置值，取值范围是[0, 1.0]，各颜色的 Position 应根据颜色出现的顺序递增。第一个 Segment 的 Position 属性默认值为 0，最后一个 Segment 的 Position 属性默认值为 1.0，当不存在时，在空缺区间内平均分配。例如 Segment 个数等于 2 且不出现 Position 属性时，按照“0 1.0”处理；Segment 个数等于 3 且不出现 Position 属性时，按照“0 0.5 1.0”处理；Segment 个数等于 5 且不出现 Position 属性时，按照“0 0.25 0.5 0.75 1.0”处理	可选
Color	CT_Color	此段的颜色，必须使用简单颜色	必需

径向渐变的 Extend 示例

```

<ofd:PathObject ID="10010" Boundary="50 50 200 150" Fill="true">
  <ofd:FillColor>
    <ofd:RadialShd StartPoint="40 70" StartRadius="10" EndPoint="140 70" EndRadius="50" >
      <ofd:Segment>
        <ofd:ColorValue="255 255 0" />
      </ofd:Segment>
      <ofd:Segment>

```

```

        <ofd:ColorValue="0 0 255" />
    </ofd:Segment>
</ofd:RadialShd>
</ofd:FillColor>
    <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10011" Boundary="350 50 200 150" Fill="true">
    <ofd:FillColor>
        <ofd:RadialShd StartPoint="40 70" StartRadius="10" EndPoint="140 70" EndRadius="50"
Extend="1">
            <ofd:Segment Position="0">
                <ofd:ColorValue="255 255 0" />
            </ofd:Segment>
            <ofd:Segment Position="1">
                <ofd:ColorValue="0 0 255" />
            </ofd:Segment>
        </ofd:RadialShd>
    </ofd:FillColor>
    <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10012" Boundary="650 50 200 150" Fill="true">
    <ofd:FillColor>
        <ofd:RadialShd StartPoint="40 70" StartRadius="10" EndPoint="140 70" EndRadius="50"
Extend="2">
            <ofd:Segment>
                <ofd:ColorValue="255 255 0" />
            </ofd:Segment>
            <ofd:Segment>
                <ofd:ColorValue="0 0 255" />
            </ofd:Segment>
        </ofd:RadialShd>
    </ofd:FillColor>
    <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10013" Boundary="50 250 200 150" Fill="true">
    <ofd:FillColor>
        <ofd:RadialShd StartPoint="40 70" StartRadius="50" EndPoint="140 70" EndRadius="10">
            <ofd:Segment>
                <ofd:Color Value="255 255 0" />
            </ofd:Segment>
            <ofd:Segment>
                <ofd:Color Value="0 0 255" />
            </ofd:Segment>
        </ofd:RadialShd>
    </ofd:FillColor>
    <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>

```

```

        </ofd:FillColor>
        <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10014" Boundary="350 250 200 150" Fill="true">
    <ofd:FillColor>
        <ofd:RadialShd StartPoint="40 70" StartRadius="50.0" EndPoint="140 70" EndRadius="10.0"
Extend="1">
            <ofd:Segment>
                <ofd:Color Value="255 255 0" />
            </ofd:Segment>
            <ofd:Segment>
                <ofd:Color Value="0 0 255" />
            </ofd:Segment>
        </ofd:RadialShd>
    </ofd:FillColor>
    <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10015" Boundary="650 250 200 150" Fill="true">
    <ofd:FillColor>
        <ofd:RadialShd StartPoint="40 70" StartRadius="50" EndPoint="140 70" EndRadius="10"
Extend="2">
            <ofd:Segment>
                <ofd:Color Value="255 255 0" />
            </ofd:Segment>
            <ofd:Segment>
                <ofd:Color Value="0 0 255" />
            </ofd:Segment>
        </ofd:RadialShd>
    </ofd:FillColor>
    <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>

```

上述示例的显示效果如图 33 所示：

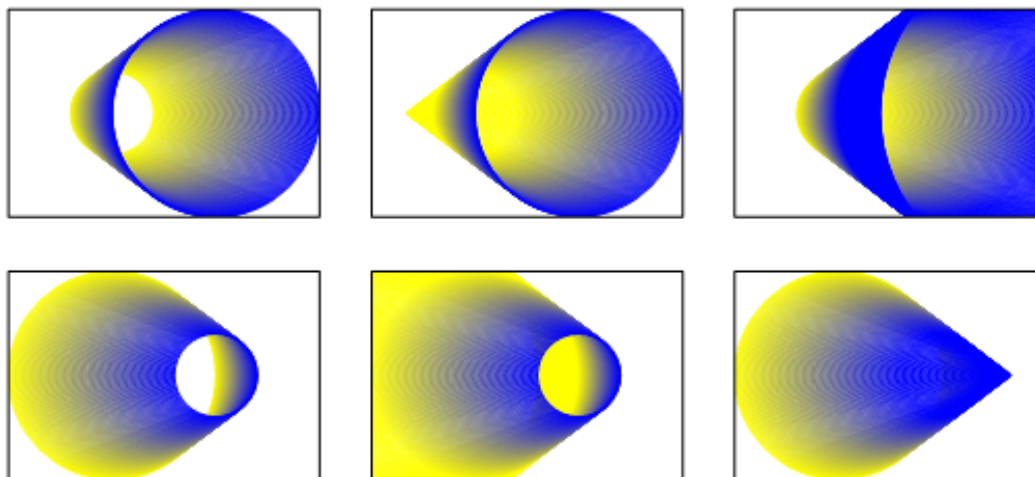


图 33 径向渐变的 Extend 属性

径向渐变的 MapType 示例

```
<ofd:PathObject ID="10010" Boundary="50 250 200 150" Fill="true">
  <ofd:FillColor>
    <ofd:RadialShd StartPoint="40 70" StartRadius="10" EndPoint="50 90" EndRadius="70">
      <ofd:Segment>
        <ofd:ColorValue="255 255 0" />
      </ofd:Segment>
      <ofd:Segment>
        <ofd:ColorValue="0 0 255" />
      </ofd:Segment>
    </ofd:RadialShd>
  </ofd:FillColor>
  <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10011" Boundary="350 250 200 150" Fill="true">
  <ofd:FillColor>
    <ofd:RadialShd StartPoint="40 70" StartRadius="10" EndPoint="50 90" EndRadius="70"
MapType="Repeat">
      <ofd:Segment>
        <ofd:ColorValue="255 255 0" />
      </ofd:Segment>
      <ofd:Segment>
        <ofd:ColorValue="0 0 255" />
      </ofd:Segment>
    </ofd:RadialShd>
  </ofd:FillColor>
  <ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>
<ofd:PathObject ID="10012" Boundary="650 250 200 150" Fill="true">
```

```

<ofd:FillColor>
  <ofd:RadialShd StartPoint="40 70" StartRadius="10" EndPoint="50 90" EndRadius="70"
MapType=" Reflect">
    <ofd:Segment>
      <ofd:ColorValue="255 255 0" />
    </ofd:Segment>
    <ofd:Segment>
      <ofd:ColorValue="0 0 255" />
    </ofd:Segment>
  </ofd:RadialShd>
</ofd:FillColor>
<ofd:AbbreviatedData>M 0 0 L200 0 L 200 150 L 0 150 C</ofd:AbbreviatedData>
</ofd:PathObject>

```

上述示例的显示效果如图所示：

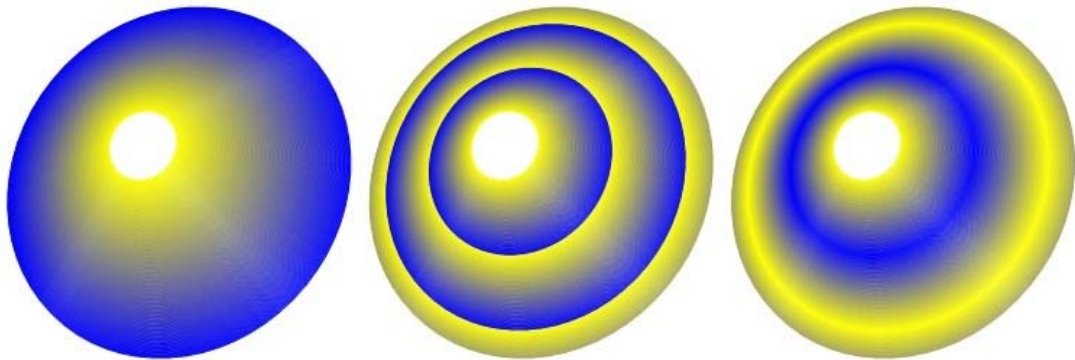


图 34 径向渐变的 MapType 属性

8.4 裁剪区

裁剪区(Clip)由一组路径或文字构成，用以指定页面上的一个有效绘制区域，落在裁剪区以外的部分不受绘制指令的影响。

一个裁剪区可由多个分路径组成，最终的裁剪范围是各个分路径的并。裁剪区中的数据均相对于该图元对象的 Boundary。

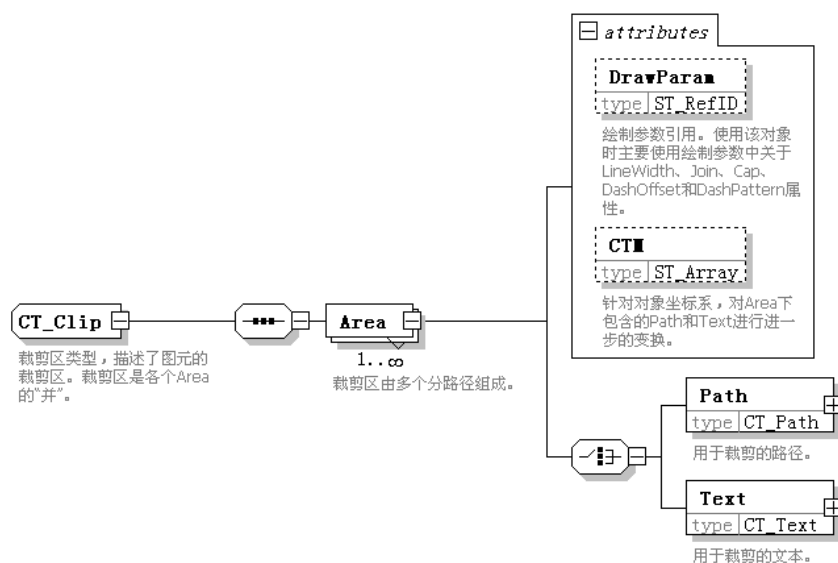


图 35 裁剪区结构

表 28 裁剪区说明

名称	类型	说明	备注
Area		裁剪区域，用一个路径对象或文字对象来描述裁剪区的一个组成部分，最终裁剪区是这些区域的并集	必需
DrawParam	ST_RefID	裁剪区所使用的绘制参数，线宽、结合点和端点样式等绘制特性对裁剪效果会产生影响，有关绘制参数的详细描述见绘制参数	可选
CTM	ST_Array	针对对象坐标系，对 Area 下包含的 Path 和 Text 进行进一步的变换。	可选
Path	CT_Path	用于裁剪的路径	可选
Text	CT_Text	用于裁剪的文本。文本描述的裁剪区域是由文本的轮廓组成的一个封闭路径。	可选

8.5 图元对象

图元对象是版式文件中页面上呈现内容的最基本单元，所有页面显示内容，包括文字、图形、图像等，都属于图元对象，或是图元对象的组合。

页面描述中对图元的使用如图 36 所示。说明如表 29 所示

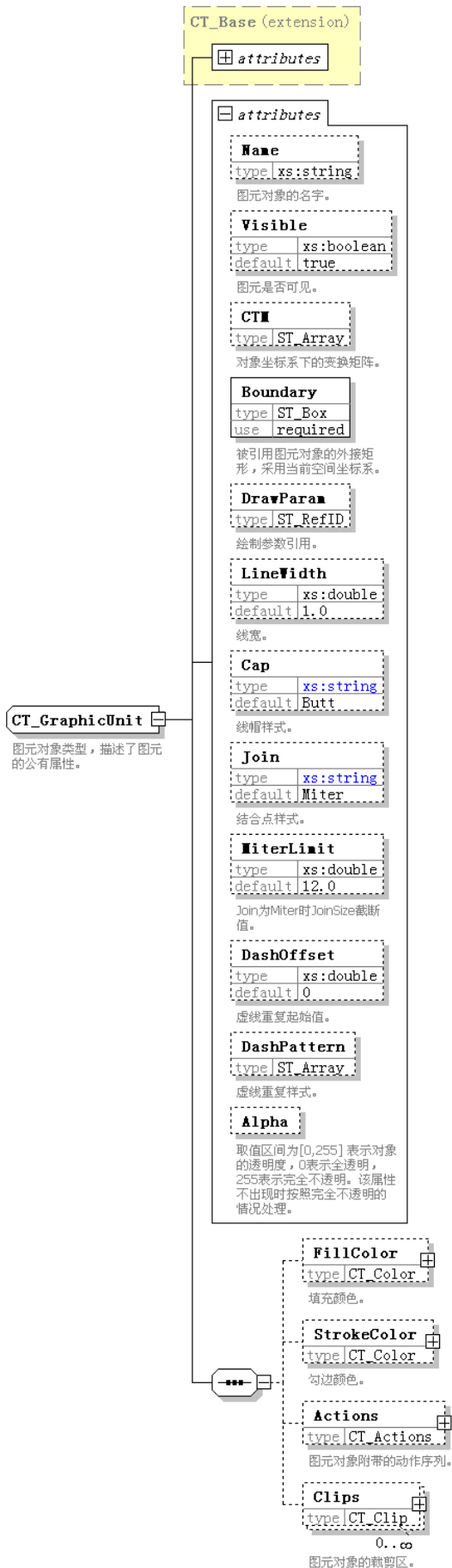
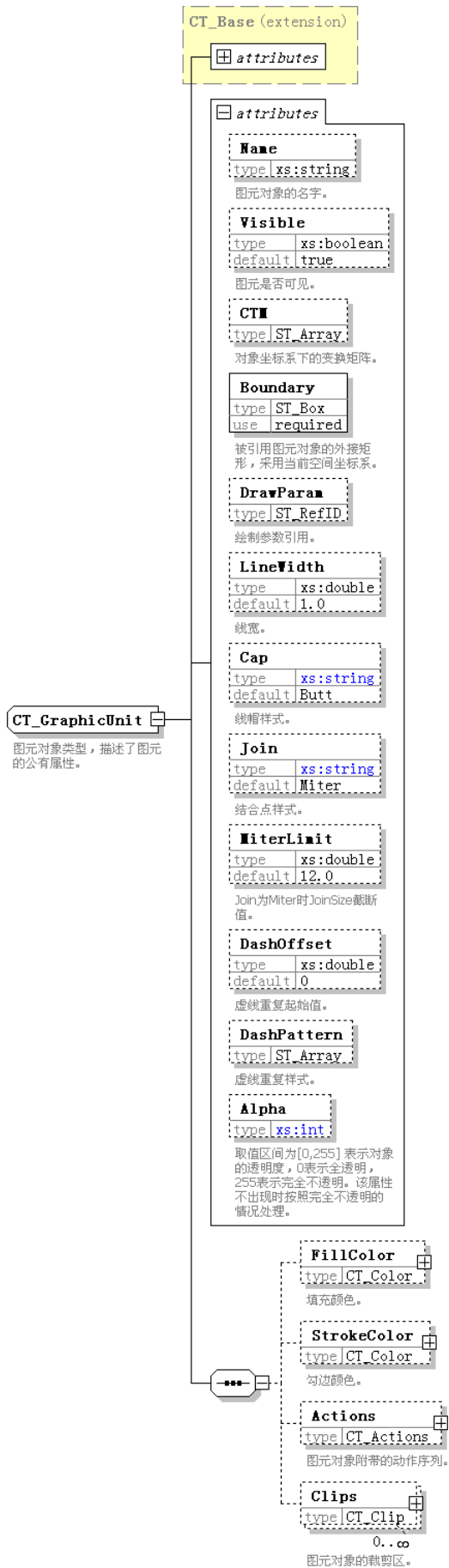


图 36 页面对象

表 29 页面对象说明

名称	说明
TextObject	文字对象，具体内容见 11
PathObject	路径对象，具体内容见 9
ImageObject	图像对象，具体内容见 10
VideoObject	视频对象，具体内容见 12
SvgObject	Svg 对象，具体内容见 13
CompositeObject	复合对象，具体内容见 14
PageBlock	页面块可形成嵌套

图元对象的共有特性如图 37 所示，具体说明见表 30。



Join

type

xs:string

default

Miter

结合点样式。

MiterLimit

type

xs:double

default

12.0

Join为Miter时JoinSize截断值。

DashOffset

type

xs:double

default

0

虚线重复起始值。

DashPattern

type

ST_Array

虚线重复样式。

Alpha

type

xs:int

取值区间为[0,255] 表示对象的透明度，0表示全透明，255表示完全不透明。该属性不出现时按照完全不透明的情况处理。

FillColor

type

CT_Color

填充颜色。

StrokeColor

type

CT_Color

勾边颜色。

Actions

type

CT_Actions

图元对象附带的动作序列。

Clips

type

CT_Clip

0..∞

图元对象的裁剪区。

CT_GraphicUnit

图元对象类型，描述了图元的公有属性。

图 37 图元对象结构图

表 30 图元对象说明

名称	类型	说明	备注
ID	ST_ID	声明该对象的标识，不能与已有标识重复	必需
Name	xs:string	图元对象的名字。默认值为空	可选
Visible	xs:boolean	像素是否可见，默认值为 true	可选
CTM	ST_Array	对象空间内的像素变换矩阵	可选
Boundary	ST_Box	外接矩形，采用当前空间坐标系(页面坐标或其它容器坐标)，当像素绘制超出此矩形区域时进行裁剪	必需
DrawParam	ST_RefID	指向所引用的绘制参数。相应的绘制参数的内容在当前可见的资源中查找	可选
LineWidth	xs:double	绘制路径时使用的线宽。如果有 DrawParam，将覆盖 DrawParam 中的值	可选
Cap	xs:string	绘制线条端使用的线帽。如果有 DrawParam，将覆盖 DrawParam 中的值	可选
Join	xs:string	绘制路径时使用的结合点方式。如果有 DrawParam，将覆盖 DrawParam 中的值	可选
MiterLimit	xs:double	Join 为 Miter 时，MiterSize 的截断值。如果有 DrawParam，将覆盖 DrawParam 中的值	可选
DashOffset	xs:double	绘制路径时虚线的起始偏移。如果有 DrawParam，将覆盖 DrawParam 中的值	可选
DashPattern	ST_Array	绘制路径时虚线的重复样式。如果有 DrawParam，将覆盖 DrawParam 中的值	可选
Alpha	Xs:int	图元对象的透明度,取值区间为[0,255] 表示对象的透明度，0 表示全透明，255 表示完全不透明	可选
Actions	Actions	该图元对象附带的动作序列，详见 15 动作。不出现时，该图元对象没有动作序列。像素的动作触发类型应为“CLICK”	可选
FillColor	CT_Color	填充颜色	可选
StrokeColor	CT_Color	描边颜色	可选
Clips	CT_Clip	被引用图元对象的裁剪区域，采用对象空间坐标系，当存在多个 Clip 对象时，最终裁剪区为所有 Clip 区域的并集	可选

9 图形

9.1 图形对象

图形对象具有一般图元对象的一切属性和行为特征。本标准中的图形对象结构图如图 38 错误！未找到引用源。，对象结构具体说明见表 31。

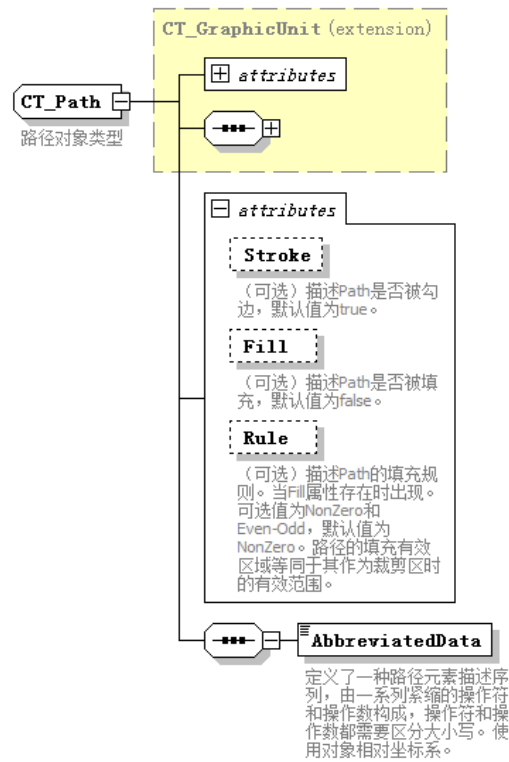


图 38 图形对象结构

表 31 图形对象属性

名称	类型	说明	备注
Stroke	xs:boolean	描述 Path 是否被描边, 默认值为 true	可选
Fill	xs:boolean	描述 Path 是否被填充, 默认值为 false	可选
Rule	xs:string	描述 Path 的填充规则。当 Fill 属性存在时出现。可选值为 NonZero 和 Even-Odd, 默认值为 NonZero。路径的填充有效区域等同于其作为裁剪区时的有效范围	可选

图形对象的数据描述方法可分为两种, 一种是紧缩命令的方式, 以字符串的形式出现; 另一种是非紧缩描述方式, 即采用路径元素的方式。

图形对象由一系列子路径组成, 每条子路径从开始节点指定的起始点坐标逐步构建最终的路径, 在本标准中路径元素包括起始点、当前点、线段、贝塞尔曲线、圆弧等, 具体说明见表 32。

表 32 图形对象描述方法

名称	类型	说明
AbbreviatedData	xs:string	定义了一种路径元素描述序列, 由一系列紧缩的操作符和操作数构成。详细说明请见 9.4 错误！未找到引用源。

Start	ST_Pos	定义 SubPath 的起始点坐标
Line	ST_Pos	从当前点连接一条到指定点的线段, 并将当前点移动到指定点。具体描述见 9.4.1
QuadraticBezier	CT_QuadraticBezier	从当前点连接一条到 Point2 点的二次贝塞尔曲线, 并将当前点移动到 Point2 点, 此贝塞尔曲线使用 Point1 点作为其控制点。具体描述见 9.4.2
CubicBezier	CT_Bezier	从当前点连接一条到 Point3 点的三次贝塞尔曲线, 并将当前点移动到 Point3 点, 此贝塞尔曲线使用 Point1 和 Point2 点作为其控制点。具体描述见 9.4.2
Arc	CT_Arc	从当前点连接一条到 EndPoint 点的圆弧, 并将当前点移动到 EndPoint 点。具体描述见 9.4.3

表 33 图形对象紧缩描述方式运算符

操作符	操作数	说明
S	x y	定义 SubPath 的起始点坐标 (x, y)
M	x y	将当前点移动到指定点 (x, y)
L	x y	从当前点连接一条到指定点 (x, y) 的线段, 并将当前点移动到指定点
Q	x ₁ y ₁ x ₂ y ₂	从当前点链接一条到点 (x ₂ , y ₂) 的二次贝塞尔曲线, 并将当前点移动到点 (x ₂ , y ₂), 此贝塞尔曲线使用点 (x ₁ , y ₁) 作为其控制点
B	x ₁ y ₁ x ₂ y ₂ x ₃ y ₃	从当前点连接一条到点 (x ₃ , y ₃) 的三次贝塞尔曲线, 并将当前点移动到点 (x ₃ , y ₃), 此贝塞尔曲线使用点 (x ₁ , y ₁) 和点 (x ₂ , y ₂) 作为其控制点
A	r _x r _y angle large sweep x y	从当前点连接一条到点 (x, y) 的圆弧, 并将当前点移动到点 (x, y)。r _x 表示椭圆的长轴长度, r _y 表示椭圆的短轴长度。angle 表示椭圆在当前坐标系下旋转的角度, 正值为顺时针, 负值为逆时针, large 为 1 时表示对应度数大于 180 度的弧, 为 0 时表示对应度数小于 180 度的弧。sweep 为 1 时表示由圆弧起点到终点是顺时针旋转, 为 0 时表示由圆弧起点到终点是逆时针旋转
C		SubPath 自动闭合, 表示将当前点和 SubPath 的起始点用线段直接连接

9.2 图形绘制参数

绘制参数包括 FillColor 和 StrokeColor, 对于图形对象, 如果绘制参数中的 FillColor 和 StrokeColor 不存在, 则默认为当前页绘制参数中定义的 FillColor 和 StrokeColor 的值。此外, 绘制参数还有 Join、JoinLimit、DashPattern、DashOffset、Cap、LineWidth 等几个属性可供设置。具体描述见 7.2。

9.3 填充规则

图形对象的填充采用两种规则：非零绕数规则(NonZero Winding Number Rule)和奇偶规则(Even-Odd Rule)。

当值为 NonZero 时，填充遵循如下原则：从所需判断的点处向任意方向无穷远处引一条射线，同时引入一个初始值为 0 的计数。射线每经过一条由左至右方向的线型时计数加 1，射线每经过一条由右至左方向的线型时则计数减 1。如果每条射线总计数均为 0，则判断该点在路径外部，反之，则该点在路径内部，见图 39。

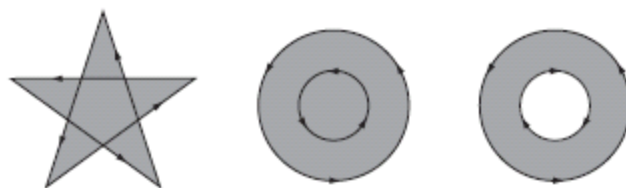


图 39 非零绕数规则示意

当值为 Even-Odd 时，填充遵循如下原则：从所需判断的点处向任意方向无穷远处引一条射线，同时引入一个初始值为 0 的计数，射线每经过任意线型时计数加 1。如果每条射线总计数均为奇数，则判断该点在路径内部，反之，则该点在路径外部，见图 40。



图 40 奇偶规则示意

在路径内部的点作为填充时的有效区域，或作为裁剪区时的有效范围。

9.4 图形的非紧缩描述

图形非紧缩描述方式，采用路径元素的方式。这种方式主要用于 Region。Region 定义了一个复杂区域，由一系列的 Area 组成，每个 Area 是一个封闭的路径，结构如图 41 所示。

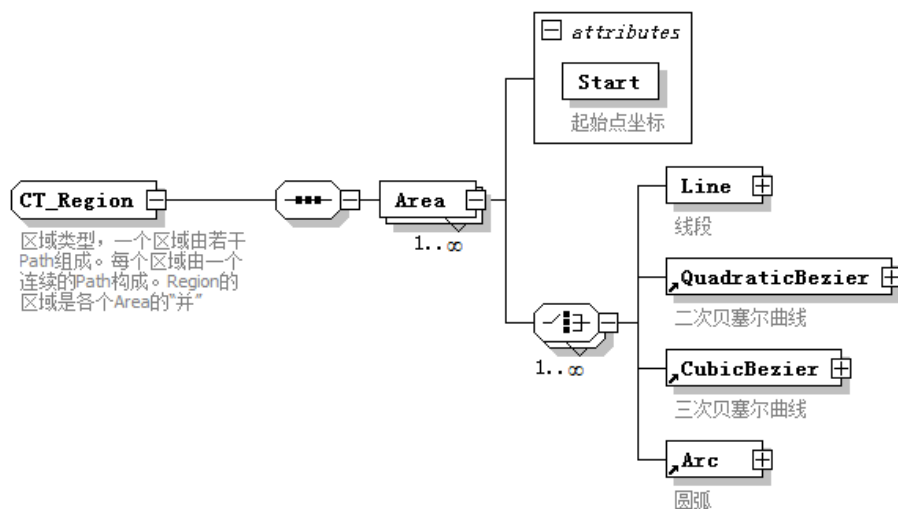


图 41 Region 结构图

9.4.1 线段

线段结构如图 42，属性表见表 34

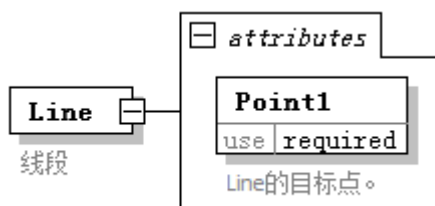


图 42 Line 结构图

表 34 线段参数表

属性名称	类型	说明	备注
Point1	ST_Pos	线段的目标点	必需

9.4.2 贝塞尔曲线

本标准中支持二次贝塞尔曲线以及三次贝塞尔曲线。

二次贝塞尔曲线，结构见图 43，属性表见表 35。

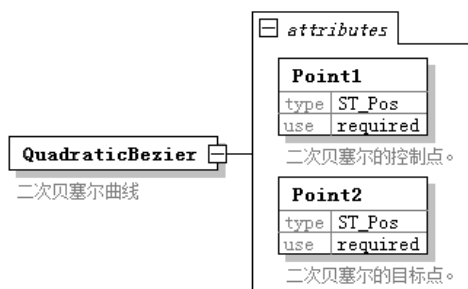


图 43 二次贝塞尔曲线结构

表 35 二次贝塞尔曲线属性

名称	类型	说明	备注
Point1	ST_Pos	二次贝塞尔曲线的控制点	必需
Point2	ST_Pos	二次贝塞尔曲线的终点，下一路径的起点	必需

二次贝塞尔曲线公式：

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \quad t \in [0, 1]$$

三次贝塞尔曲线，结构见图 44，属性表见表 36。

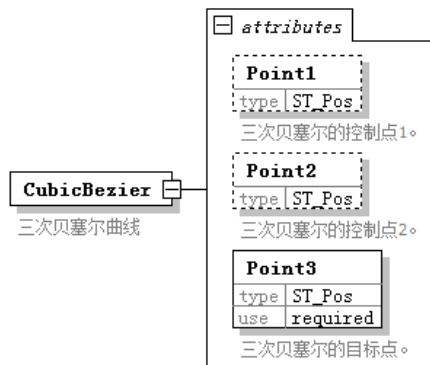


图 44 三次贝塞尔曲线结构

表 36 三次贝塞尔曲线属性

名称	类型	说明	备注
Point1	ST_Pos	三次贝塞尔曲线的第一个控制点	可选
Point2	ST_Pos	三次贝塞尔曲线的第二个控制点	可选
Point3	ST_Pos	三次贝塞尔曲线的终点，下一路径的起点	必需

三次贝塞尔曲线公式：

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 \quad t \in [0, 1]$$

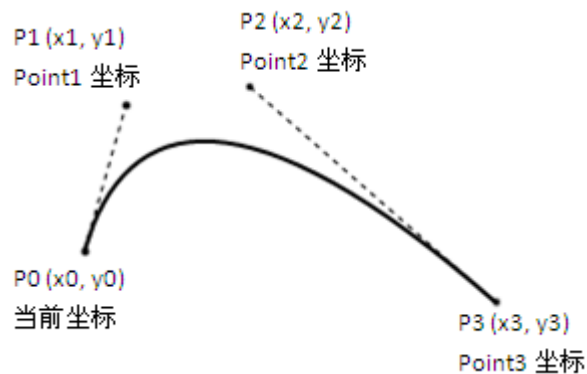


图 45 三次贝塞尔曲线示意

其中 P_0 、 P_1 、 P_2 、 P_3 分别为上个路径的终点以及本卷标中属性 Point1, Point2, Point3 对应的坐标。

当 Point1 不存在时，第一控制点取当前点的值，当 Point2 不存在时，第二控制点取 Point3 的值。

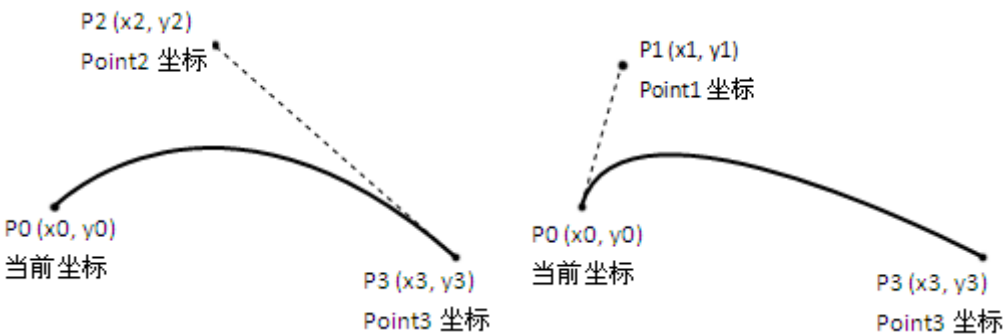


图 46 三次贝塞尔曲线示意

9.4.3 圆弧

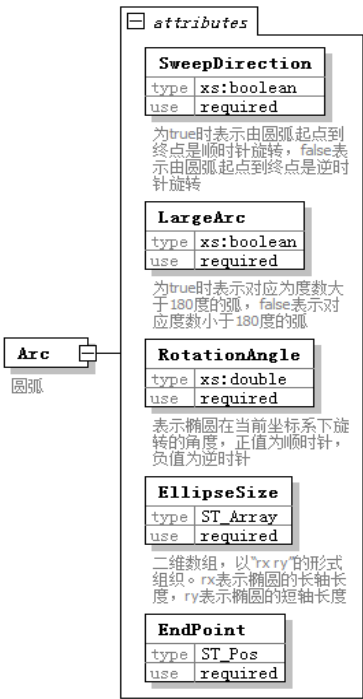


图 47 圆弧结构

表 37 圆弧属性

名称	类型	说明	备注
EndPoint	ST_Pos	圆弧的终点，也是下个路径的起点	必需
EllipseSize	ST_Array	形如[200 100]的数组，2 个正浮点数值依次对应椭圆的长、短轴长度，较大的一个为长轴 [异常处理]如果数组长度超过 2，则只取前两个数值	必需

名称	类型	说明	备注
		<p>[异常处理]如果数组长度为 1，则认为这是一个圆，该数值为圆半径</p> <p>[异常处理]如果数组前两个数值中有一个为 0，或者数组为空，则圆弧退化为一从当前点到 EndPoint 的线段</p> <p>[异常处理]如果数组数值为负值，则取其绝对值</p>	
RotationAngle	xs:double	<p>表示按 EllipseSize 绘制的椭圆在当前坐标系下旋转的角度。正值为顺时针，负值为逆时针。</p> <p>[异常处理]如果角度大于 360 度，则以 360 取模</p>	必需
LargeArc	xs:boolean	<p>对于一个给定长、短轴的椭圆以及起点和终点，有一大一小两条圆弧。true 表示此线型对应的为度数大于 180 度的弧，false 表示对应度数小于 180 度的弧。如果所描述线型恰好为 180 度的弧，则此属性的值不被参考，可由 SweepDirection 属性确定圆弧的形状</p>	必需
SweepDirection	xs:boolean	<p>对于经过坐标系上指定两点，给定旋转角度和长短轴长度的椭圆，满足条件的可能有 2 个，由起点和终点所构成的对应圆弧共有 4 条。通过 LargeArc 属性可以排除 2 条，由此属性从余下的 2 条圆弧中确定一条。true 表示由圆弧起点到终点是顺时针旋转，false 表示由圆弧起点到终点是逆时针旋转</p>	必需

10 图像

图像对象的基本结构如图 48 所示，具体属性描述见表 38。

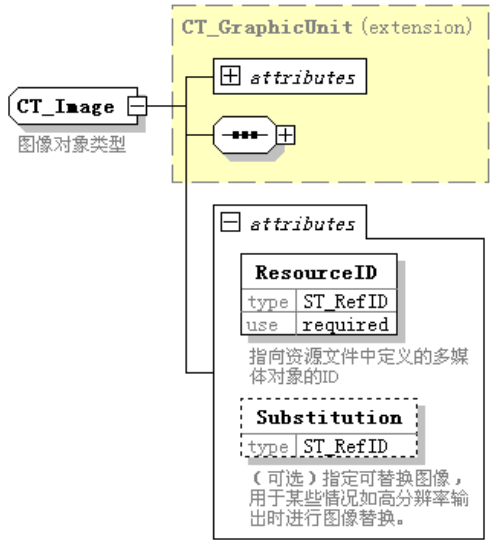


图 48 图像对象的基本结构

表 38 图像对象的属性描述

名称	类型	说明	备注
----	----	----	----

ResourceID	ST_RefID	指向资源文件中定义的多媒体对象的 ID	必需
Substitution	ST_RefID	指定可替换图像，需要高分辨率图像の場合进行图像替换	可选

11 文字

11.1 字体

字体结构的描述见图 49，各节点和属性的说明见表 39。

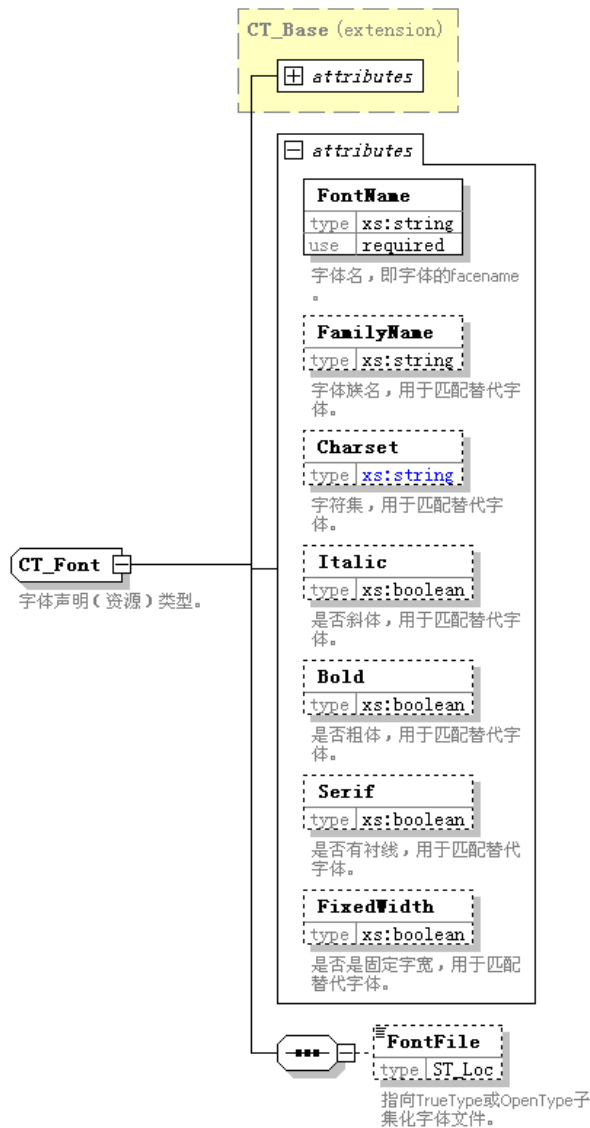


图 49 字体结构

表 39 字体说明

属性	类型	说明	备注
----	----	----	----

ID	ST_ID	声明该字体的标识，不能与已有标识重复	必需
FontName	xs:string	字体名	必需
FamilyName	xs:string	字体族名，用于匹配替代字体	可选
CharSet	xs:string	字体所使用的字符集，用于匹配替代字体 枚举值如下： unicode symbol 符号 prc 中文 big5 繁体中文 shift-jis 日文 wansung 韩文 johab 朝鲜文 默认值是 unicode	可选
Serif	xs:boolean	是否是带衬线字体，用于匹配替代字体 默认值是 false	可选
Bold	xs:boolean	是否是粗体字体，用于匹配替代字体 默认值是 false	可选
Italic	xs:boolean	是否是斜体字体，用于匹配替代字体 默认值是 false	可选
FixedWidth	xs:boolean	是否是等宽字体，用于匹配替代字体 默认值是 false	可选
FontFile	ST_Loc	指向内嵌字体文件，嵌入的字体文件应使用 OpenType 格式	可选

11.2 文字对象

文字具有图元对象的一切基本属性和行为特征，除此以外还具有某些文字对象特有的绘制特性。文字对象的结构见图 50，各节点和属性的说明见表 40。

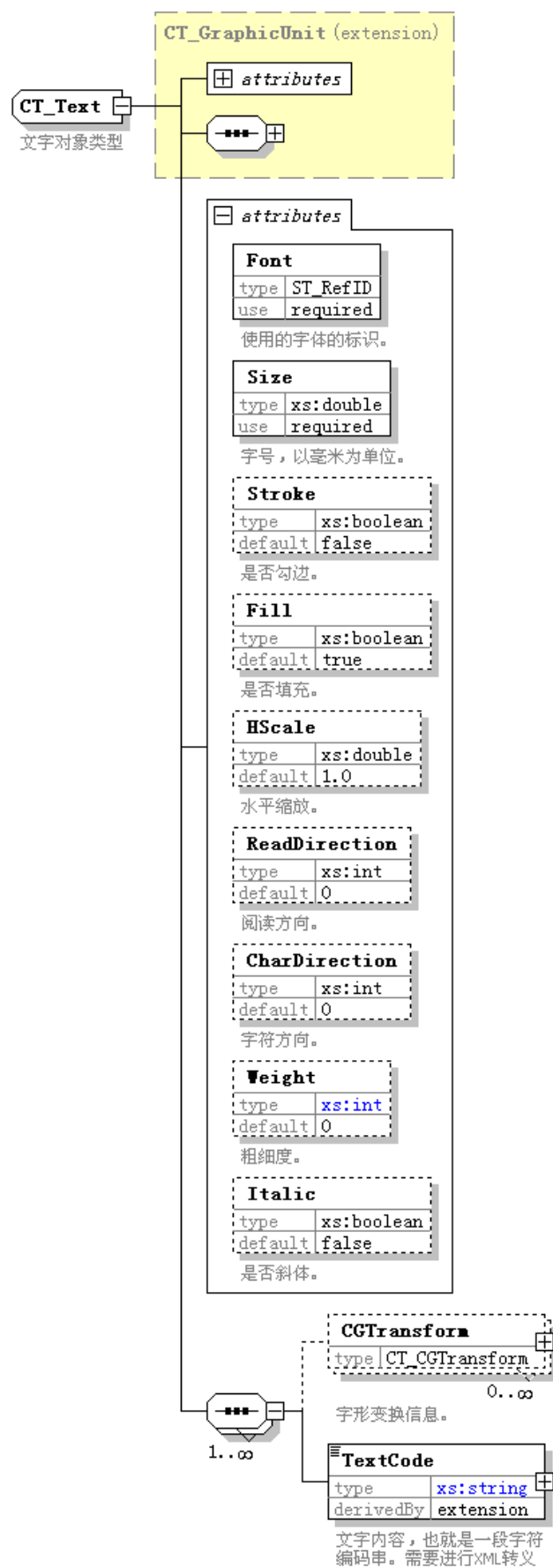


图 50 文字对象结构

表 40 文字对象说明

属性	类型	说明	备注
Font	ST_RefID	指向使用的字体	必需
Size	xs:double	字号，使用 mm 为单位	必需
Stroke	xs:boolean	是否描边。默认值为 false 当文字对象被裁剪区引用时此属性被忽略	可选
Fill	xs:boolean	是否填充。默认值为 true 当文字对象被裁剪区引用时此属性被忽略	可选
HScale	xt:double	字形在水平方向的放缩比，取值为[0 1.0]，默认值为 1.0 例如：当 HScale 值为 0.5 时表示实际显示的字宽为原来字宽的一半	可选
ReadDirection	xs:int	阅读方向，指定了文字排列的方向，默认值为 0 具体内容见字形定位	可选
CharDirection	xs:int	字符方向，指定了文字放置的方式，默认值为 0 具体内容见字形定位	可选
Weight	xs:int	文字对象的粗细值；可选取值为 0,100,200,300,400,500,600,700,800,900,1000 默认值为 400 当取值为 1000 时按照 900 处理，取值为 0 或其它值时按照默认值处理	可选
Italic	xs:boolean	是否是斜体样式，默认值为 false	可选
CGTransform	CT_CGTransform	指定字符编码到字形索引之间的变换关系，具体描述见 11.4 字形变换	可选
TextCode	xs:string	文字内容，也就是一段字符编码串，如果字符编码不在 XML 标准允许的字符范围之内，必须采用“\”加四位十六进制 数的格式转义；文字内容中出现的空格也需要转义	必需

文本物件示例

```

<ofd:TextObject ID="6" Font="2" Size="25.4" Boundary="50 20 112 26">
  <ofd:TextCode X="0" Y="25" DeltaX="14 14 14">Font</ofd:TextCode>
  <ofd:TextCode X="60" Y="25" DeltaX="25">字体</ofd:TextCode>
</ofd:TextObject>

```

当 Font 为“宋体”时，上述示例的显示效果如图 51 所示：

Font 字体

图 51 字体

11.3 字形定位

文字对象使用了严格的字形定位信息，如图 52 所示，说明如表 41 所示

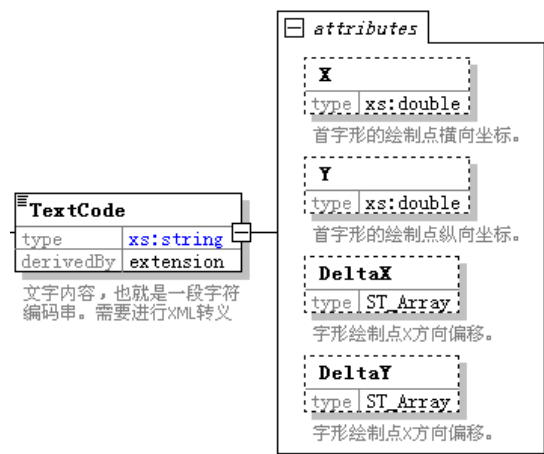
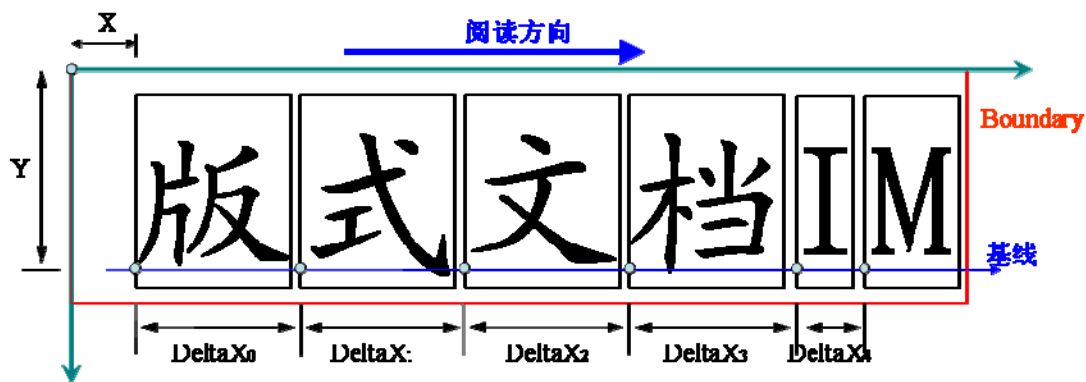


图 52 字形定位结构

表 41 字形定位说明

X	xs:double	第一个字形绘制点在对象坐标系下的 X 坐标。当 X 不出现，则采用上一个 TextCode 的 X 值。文字对象中的第一个 TextCode 的 X 属性必需	可选
Y	xs:double	第一个字形绘制点在对象坐标系下的 y 坐标。当 Y 不出现，则采用上一个 TextCode 的 Y 值。文字对象中的第一个 TextCode 的 Y 属性必需	可选
DeltaX	ST_Array	double 型数值队列，队列中的每个值代表后一个字形与前一个字形之间在 X 方向的偏移值。DeltaX 不出现时，表示字形的绘制点在 X 方向不做偏移。 具体描述见 11.4 字形变换位	可选
DeltaY	ST_Array	double 型数值队列，队列中的每个值代表后一个字形与前一个字形之间在 Y 方向的偏移值。DeltaY 不出现时，表示字形的绘制点在 Y 方向不做偏移。 具体描述见 11.4 字形变换位	可选

X、Y、DeltaX 和 DeltaY 相结合确定了 TextCode 中对应的每个字形原点的精确位置，上述属性的定位机制如图 53 所示：



文本的Delta定位模型

图 53 字形定位机制

CharDirection 与 ReadDirection 规定了文字显示时的排列方向。CharDirection 指定了单个文字绘制方向，也就是文字的基线方向，用从 x 轴正方向顺时针到字形基线的角度表示。ReadDirection 指定了阅读方向，用从 x 轴正方向顺时针到文字排列方向的角度表示。这二者的数值规定见表 42 错误！未找到引用源。。

表 42 文字排列方向、阅读方向说明

CharDirection 值	定义
0	默认值，以'A'为例子，显示效果为 A
90	字形顺时针旋转 90 度，以'A'为例子，显示效果为 A
180	字形顺时针旋转 180 度，以'A'为例子，显示效果为 A
270	字形顺时针旋转 270 度，以'A'为例子，显示效果为 A
ReadDirection 值	定义
0	默认值，从左往右阅读，以字符串"ABC"为例，CharDirection 为 0，显示效果为 ABC
90	从上往下阅读，以字符串"ABC"为例，CharDirection 为 0，显示效果为 ABC
180	从右往左阅读，以字符串"ABC"为例，CharDirection 为 0，显示效果为 CBA
270	从下往上阅读，以字符串"ABC"为例，CharDirection 为 0，显示效果为 CBA

ReadDirection 等于 90 时的字形定位如图 54 所示：

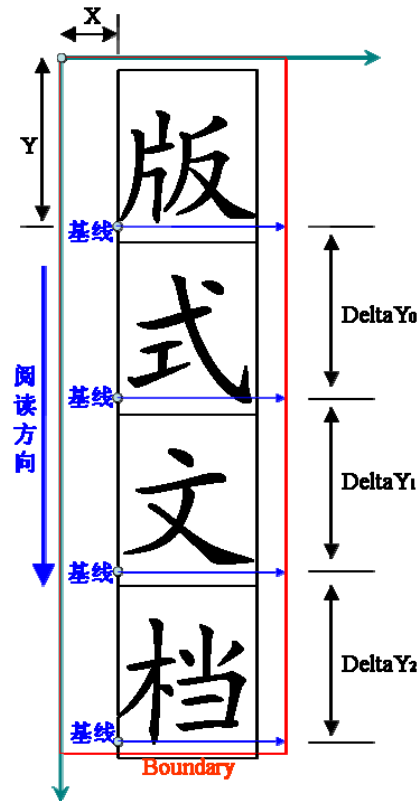


图 54 字体阅读方向机制(a)

等宽字体进行竖排时，起绘点在 X 方向上无变化的，可省略 DeltaX 属性。

ReadDirection 等于 180 时的字形定位如图 55 所示：

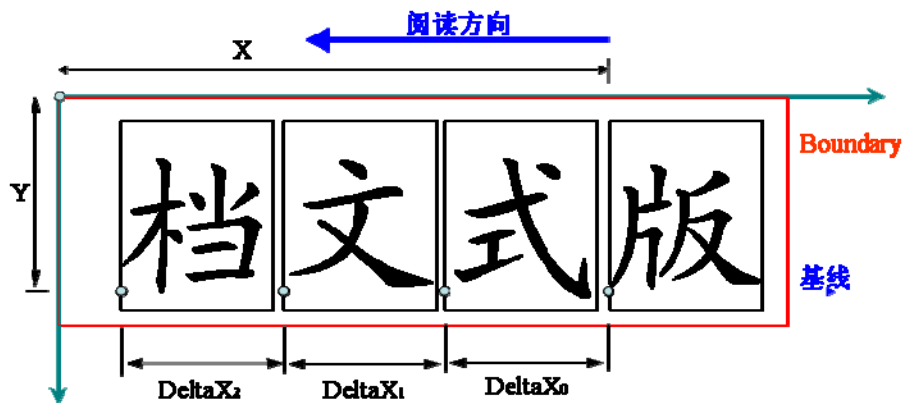


图 55 字体阅读方向机制(b)

不同字符方向下的绘制点如图 56 所示：

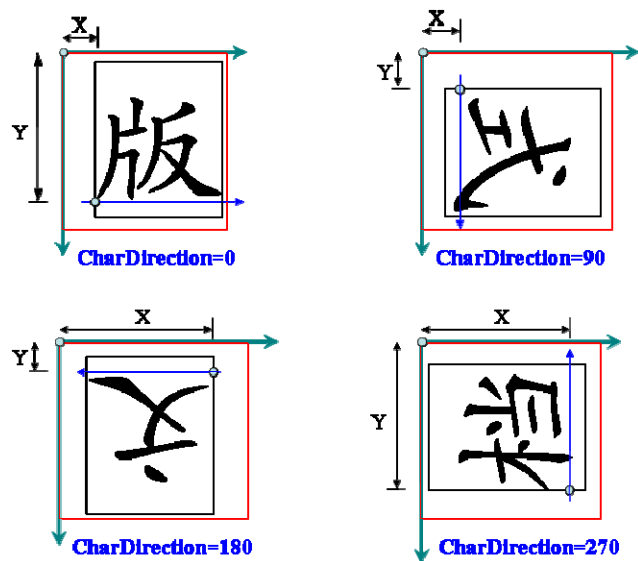


图 56 绘制点

11.4 字形变换

文字内容应以 11.4 中规定的编码在 TextCode 中表示, 大部分语言(东亚文字和拉丁文字)的编码值及其字形都是一对一的关系, 即一个字符对应一个字形。而一些非拉丁语言中会出现一些复杂的对应关系: 例如一个字符对应两个或两个以上字形; 多个字符对应一个字形, 多个字符对应多个字形。

在 TextCode 对象中我们使用字形映像(CGTransform)节点来详细描述字符编码和字形索引之间的关系, 该节点结构如图 57 所示:

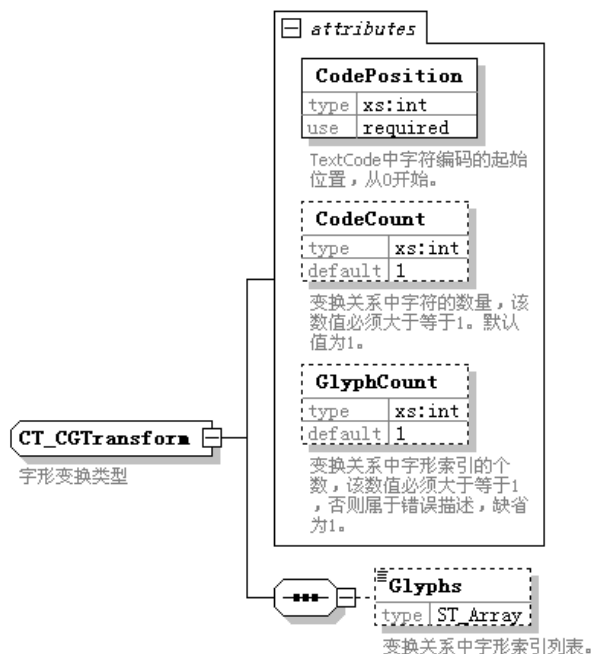


图 57 字形映像结构

表 43 字形映像说明

名称	类型	说明	备注
CodePosition	xs:int	TextCode 中字符编码的起始位置，从 0 开始	必需
CodeCount	xs:int	变换关系中字符的数量，该数值必须大于等于 1，否则属于错误描述，默认为 1	可选
GlyphCount	xs:int	变换关系中字形索引的个数，该数值必须大于等于 1，否则属于错误描述，默认为 1	可选
Glyphs	ST_Array	变换关系中字形索引列表	可选

字形的索引跟具体的字体文件紧密相关，同一个字形在不同的字体文件中的索引值并不一样，因此当使用到字形变换时，宜将对应的字体文件嵌入到版式档中。

字符编码到字形之间的变换关系主要可分为四种，分别是一对一、多对一、一对多以及多对多。

11.4.1 一对一

当一个字符对应一个字形时，如果文字使用非内嵌字体的时候，则根据文字使用的字体，使用字体的内置 CMap 表计算并取得相应的字形。如果文字使用的是内嵌字体，则使用该内嵌字体数据中的 CMap 表来取得字形索引。

11.4.2 多对一

多个字符对应一个字形的情况描述如图 58 所示，这是一个常用的英文连写示例，在例子中，f 和 l 字符在显示的时候被一个 fl 的连字符所代替：

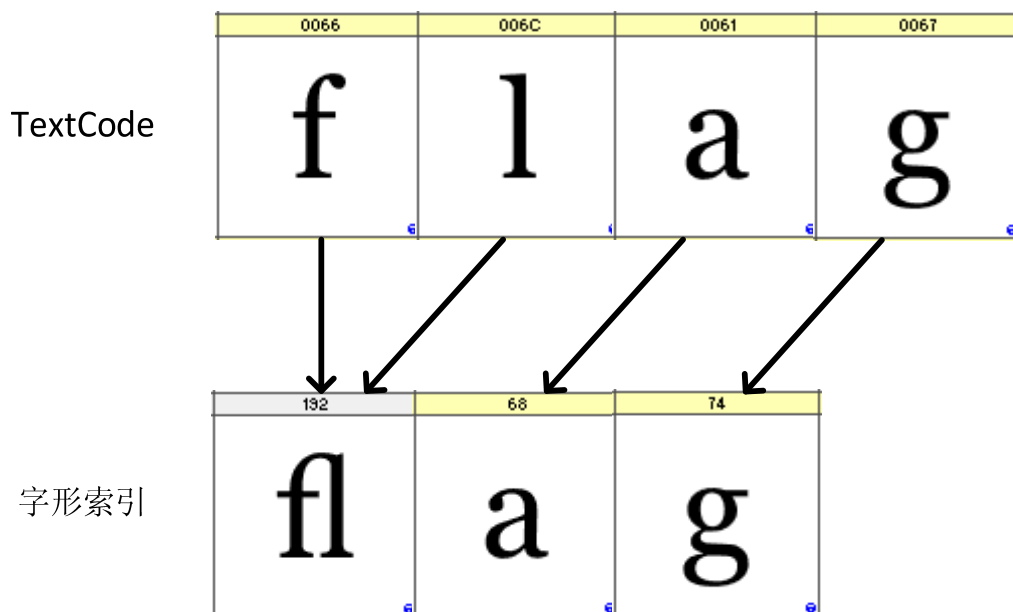


图 58 多对一

11.4.3 一对多

一个字符对应多个字形的情况描述如图 59 所示，这是一个泰国文字中的例子。

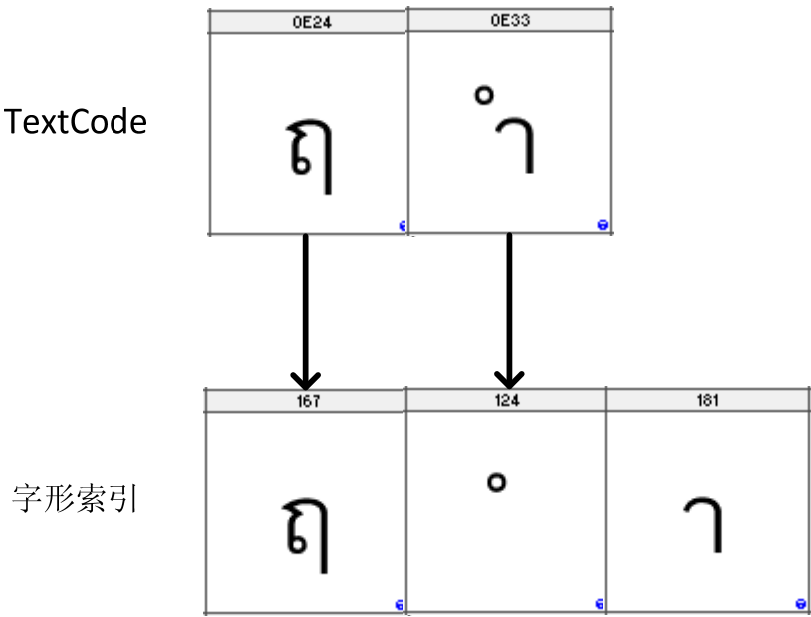


图 59 一对多

11.4.4 多对多

多个字符对应多个字形的情况描述如图 60 所示，这是一个泰米尔语文字的例子，例子中两个泰米尔语字符在有些字体中对应三个字形。

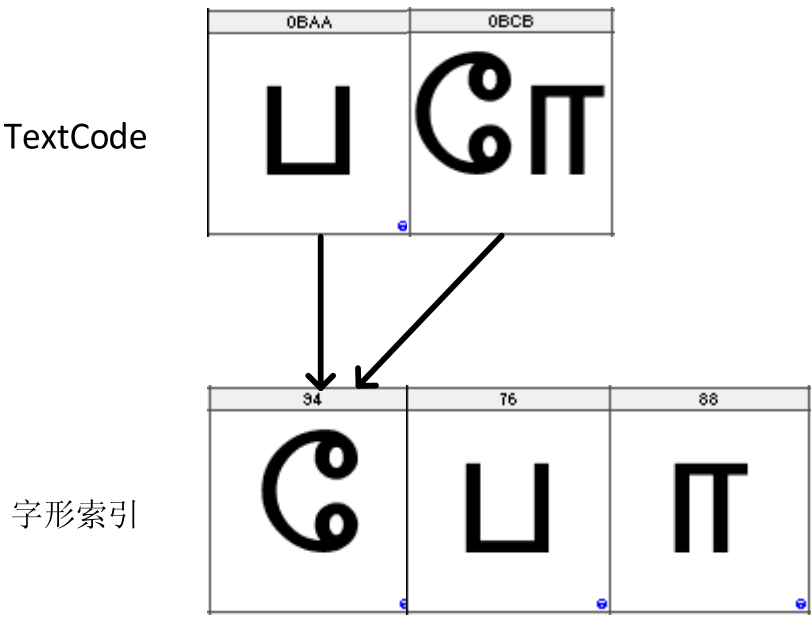


图 60 多对多

渲染结果如下：



图 61 多对多渲染效果

12 视频

视频对象基本结构如图 64 所示。

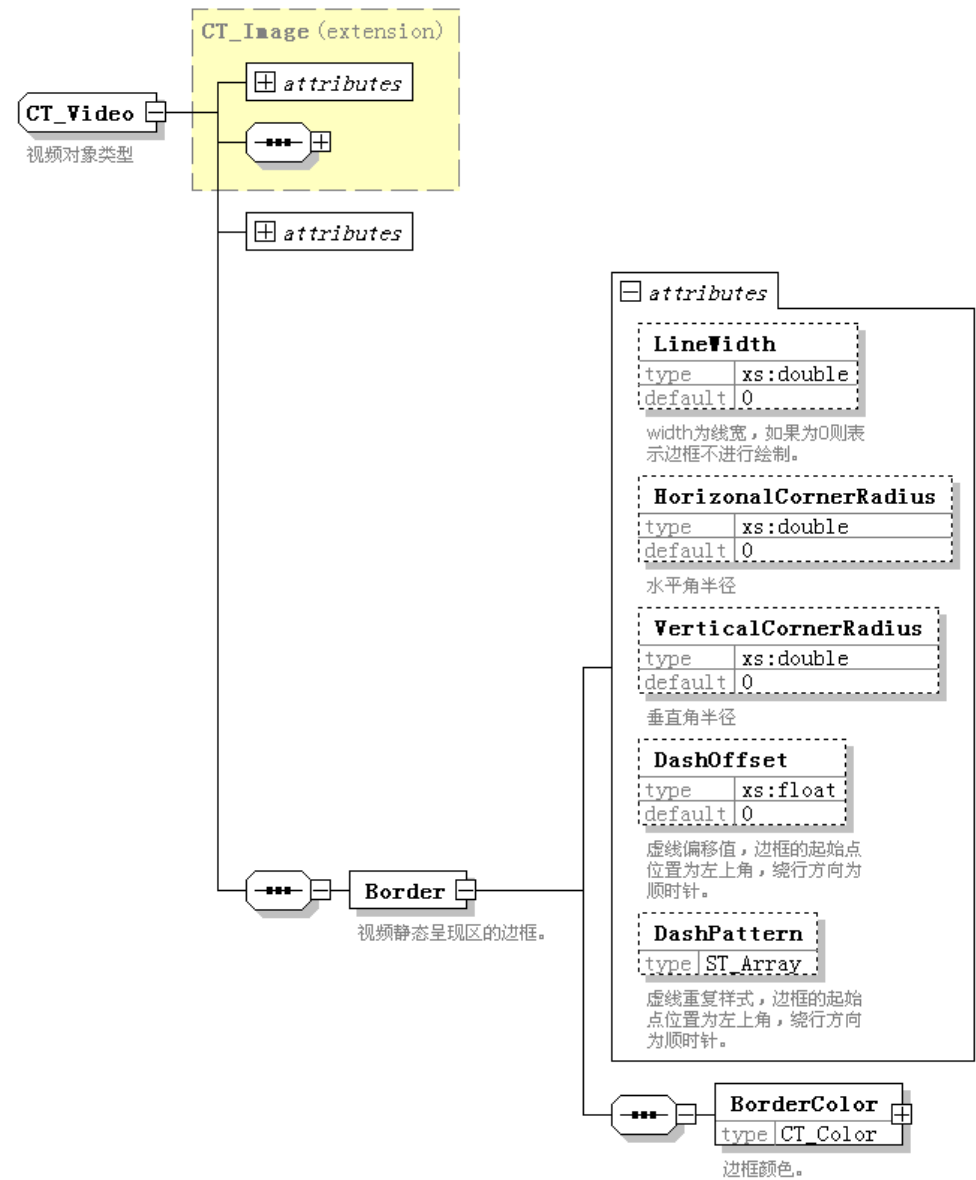


图 62 视频对象基本结构

表 44 视频对象基本属性

名称	类型	说明	备注
Title	xs:string	视频物件的标题	可选

名称	类型	说明	备注
LineWidth	xs:double	width 为线宽，如果为 0 则表示边框不进行绘制，默认值为 1.0	可选
HorizontalCornerRadius	xs:double	表示水平角半径，默认值为 0	可选
VerticalCornerRadius	xs:double	表示垂直角半径，默认值为 0	可选
DashOffset	xs:double	虚线重复样式开始的位置，边框的起始点位置为左上角，绕行方向为顺时针，默认值为 0	可选
DashPattern	ST_Array	表示虚线重复样式，边框的起始点位置为左上角，绕行方向为顺时针	可选
BorderColor	CT_Color	表示边框颜色，有关边框颜色详细描述请见 8.3.2 基本颜色	必需

13 SVG 对象

SVG 对象(SvgObject)是一种扩展的图元对象，其呈现的区域在 SvgObject 所指定的 Boundary 中。其内容描述应使用与其它像素相同的绘制修饰属性。SvgObject 的基本结构如图 63 所示，具体属性描述见表 45。

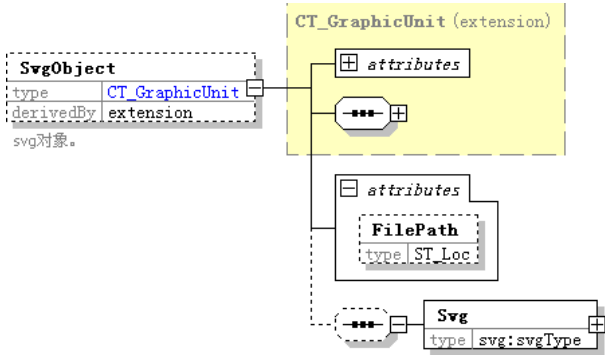


图 63 SvgObject 的基本结构

表 45 SvgObject 的属性描述

名称	类型	说明	备注
FilePath	ST_Loc	指向包内的一个 Svg 档	可选
Svg	svg:SvgType	SvgObject 内容。使用 Svg 规范进行描述	必需

14 复合对象

复合对象是一种复杂的图元对象，拥有图元对象的一切属性和节点，其绘制参数拥有基础绘制参数的特性。图元对象的描述见图元对象章节索引。

复合对象可视为宽为 Width、高为 Height 的图像对象，基本结构如图 64 所示。

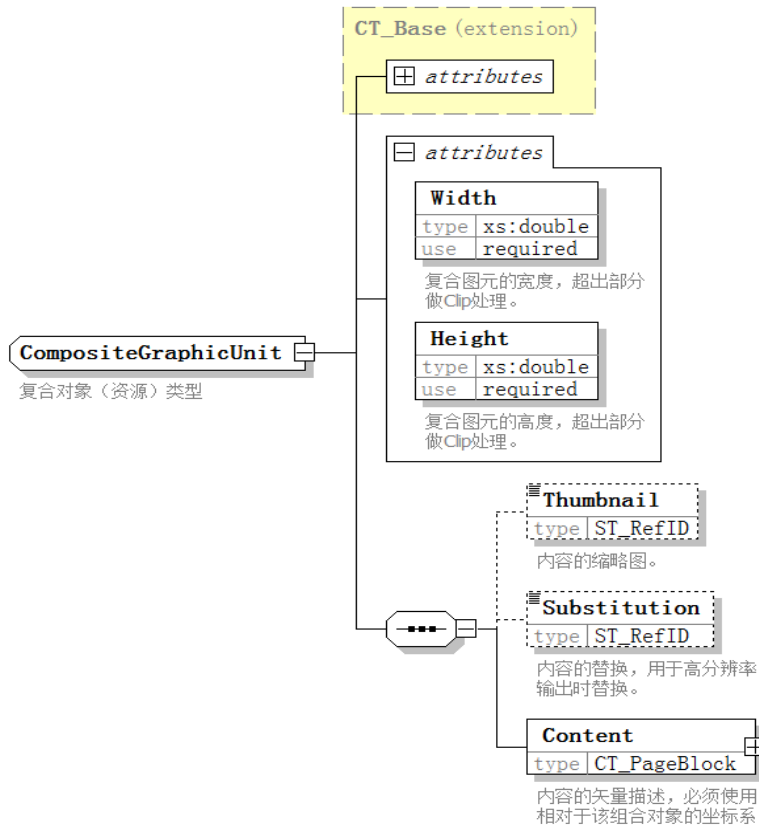


图 64 复合对象类型结构

表 46 复合对象类型属性 47

名称	类型	说明	备注
Width	xs:double	复合图元的宽度，超出部分做剪切处理	必需
Height	xs:double	复合图元的高度，超出部分做剪切处理	必需
Thumbnail	ST_Loc	该复合物件的缩略图	可选
Substitution	ST_Loc	指定可替换图像，用于高分辨率处理时进行图像替换	可选
Content	Content	内容的向量描述，必须使用相对于该组合对象的坐标系	必需

15 动作

15.1 动作序列

动作序列是一系列动作的集合，可以定义对象中各种 Event 事件中的行为。动作由事件 (Event) 来触发，包含若干子动作，如图 65 所示：

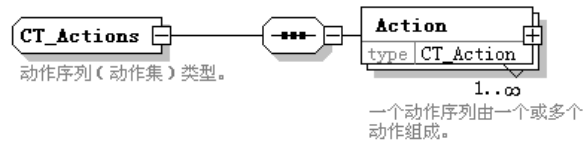


图 65 动作序列结构

表 48 动作序列属性

名称	类型	说明	备注
Action	CT_Action	动作节点，一个动作序列由一个或多个动作节点组成	必需

15.2 动作类型

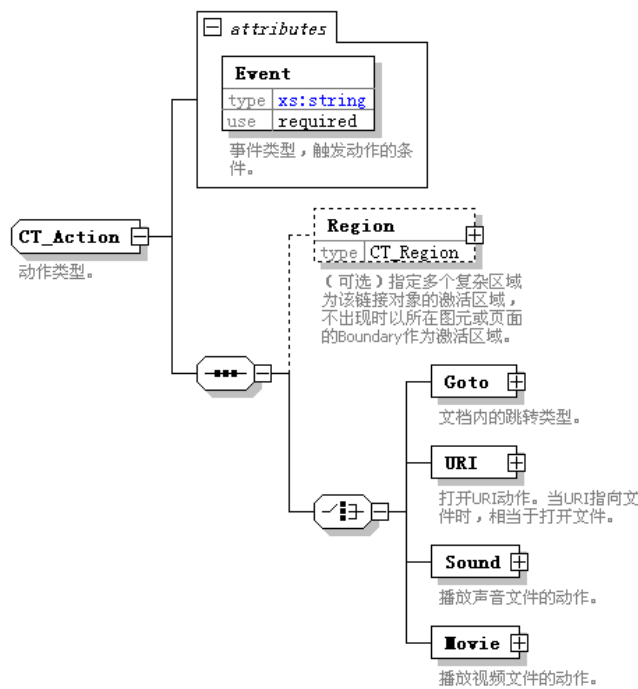


图 66 动作类型结构

表 49 动作类型属性

名称	类型	说明	备注
Event	xs:string	事件类型，触发动作的条件，事件的具体类型见表 50	必需
Region	CT_Region	指定多个复杂区域为该链接对象的启动区域，不出现时以所在像素或页面的 Boundary 作为启动区域，Region 具体描述见 9.4	可选
Goto		Goto 动作表明的是一个文档内的跳转，包含一个目标区域描述	可选
URI		URI 动作表明的是指向一个 URI 位置，URI (Uniform Resource Identifier) 描述参考 RFC 2396	可选
Sound		Sound 动作表明播放一段音频	可选
Movie		Movie 动作表明播放一段视频	可选

事件类型限定于 D0、P0、CLICK 三种，分别对应于文档打开动作、页面打开动作和区域内单击动作。

表 50 事件类型

Event 事件	说明
D0	文档打开
P0	页面打开
CLICK	单击区域

15.2.1 GOTO 动作

Goto 动作表明的是一个文档内的跳转，包含一个目标区域描述(Dest)，其中的 Left、Right、Top、Bottom 均采用页面空间坐标。

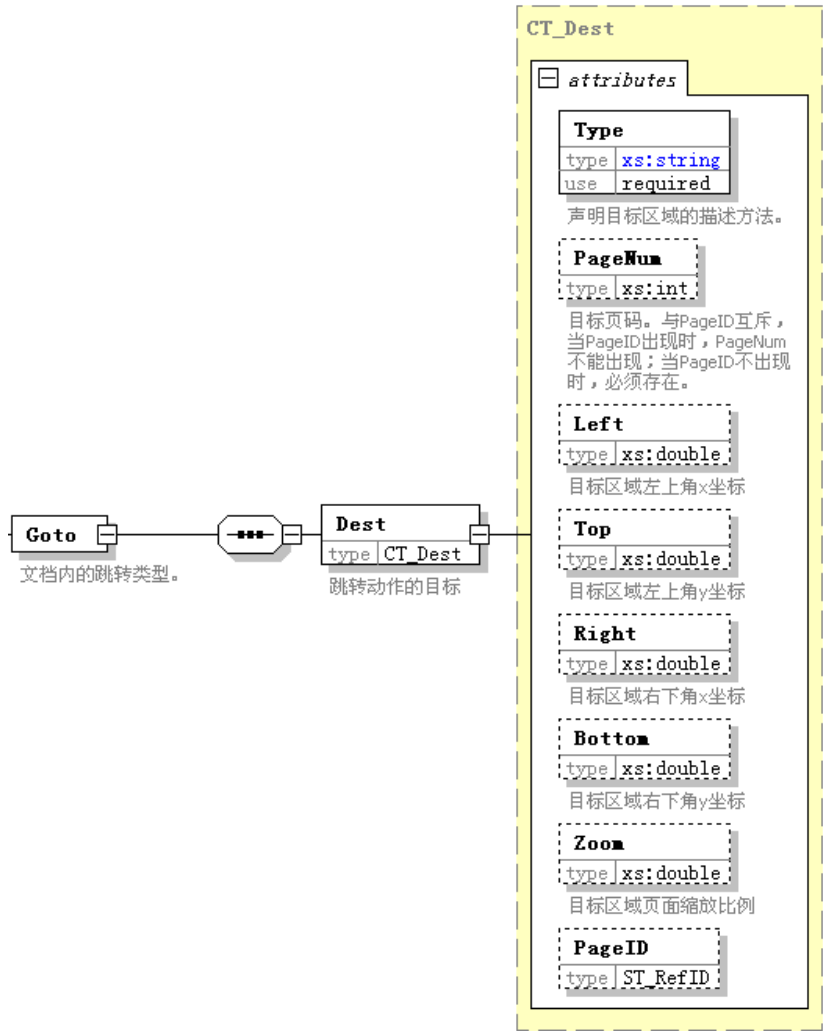


图 67 Goto 动作结构

表 51 Goto 动作属性

名称	类型	说明	备注
Type	xs:string	声明目标区域的描述方法，可取值列举如下： XYZ — 目标区域由左上角位置(Left, Top)以及页面缩放比例	必需

名称	类型	说明	备注
		(Zoom) 确定; Fit — 适合整个窗口区域; FitH — 适合窗口宽度, 目标区域位置仅由 Top 坐标确定; FitV — 适合窗口高度, 目标区域位置仅由 Left 坐标确定; FitR — 适合窗口内的目标区域, 目标区域为 (Left, Top, Right, Bottom) 所确定的矩形区域	
Left	xs:double	目标区域左上角 x 坐标	可选
Right	xs:double	目标区域右下角 x 坐标	可选
Top	xs:double	目标区域左上角 y 坐标	可选
Bottom	xs:double	目标区域右下角 y 坐标	可选
Zoom	xs:double	目标区域页面缩放比例, 如果为 0 或者不出现则按照当前缩放比例进行跳转, 可取值范围[0 正无穷], 推荐取值范围[0 64.0]	可选
PageID	ST_RefID	跳转页面的 ID	可选

15.2.2 URI 动作

URI 动作表明的是指向一个 URI 位置, URI (Uniform Resource Identifier) 描述参考 RFC 2396



图 68 URI 动作结构

表 52 URI 动作属性

名称	类型	说明	备注
URI	xs:string	目标 URI 的位置	必需
Base	xs:string	Base URI, 用于相对地址	可选

15.2.3 SOUND 动作

Sound 动作表明播放一段音频。

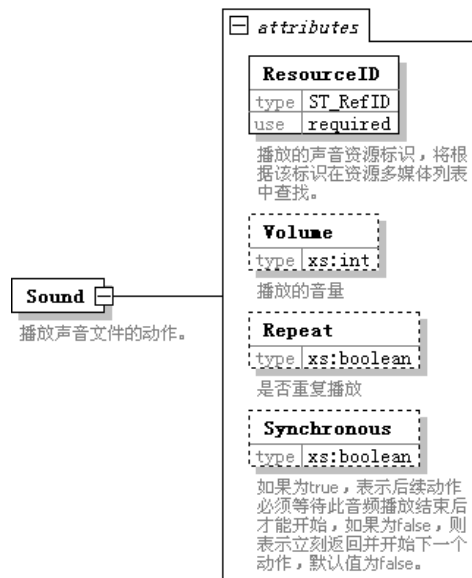


图 69 Sound 动作结构

表 53 Sound 动作属性

名称	类型	说明	备注
ResourceID	ST_RefID	播放的声音资源标识, 将根据该标识在资源多媒体列表中查找	必需
Volume	xs:int	播放的音量, 取值范围[0 100], 默认值为 100	可选
Repeat	xs:boolean	此音频是否需要循环播放, 如果此属性为 true, 则 Synchronous 值无效, 默认为 false	可选
Synchronous	xs:boolean	如果为 true, 表示后续动作必须等待此音频播放结束后才能开始, 如果为 false, 则表示立刻返回并开始下一个动作, 默认值为 false	可选

15.2.4 MOVIE 动作

Movie 动作用于播放视频。

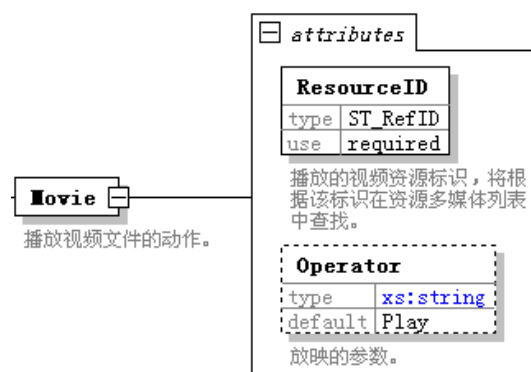


图 70 Movie 动作结构

表 54 Movie 动作属性

名称	类型	说明	备注
ResourceID	ST_RefID	播放的视频资源标识, 将根据该标识在资源多媒体列表中查找	必需
Operator	xs:string	放映参数, 默认值为 play, 具体定义详见放映参数表	可选

表 61 放映参数表

名称	类型	说明
Play	xs:string	播放
Stop	xs:string	停止
Pause	xs:string	暂停
Resume	xs:string	重新开始 sss 播放

16 注释

图元对象都可作为注释对象定义，用户可通过鼠标或键盘与其进行交互。在图元对象中注释节点结构如图所示：

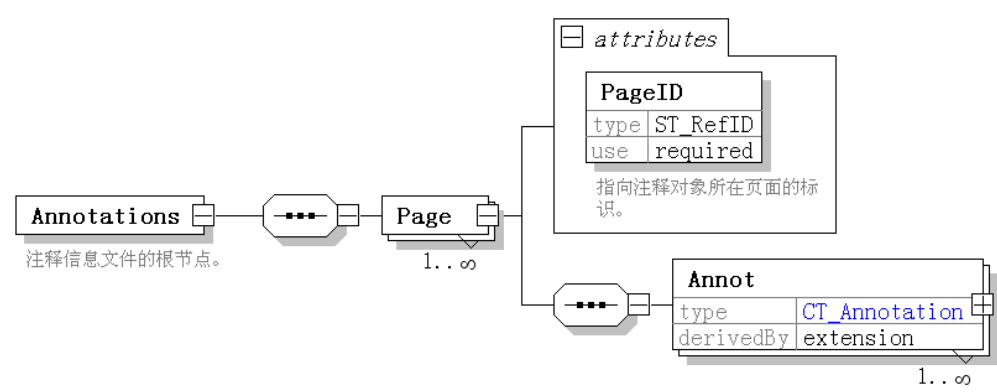


图 71 Annotations 结构

表 55 Annotations 属性

名称	类型	说明	备注
PageID	ST_RefID	注释所在页的 ID	必需
Annot	CT_Annotation	注释内容	必需

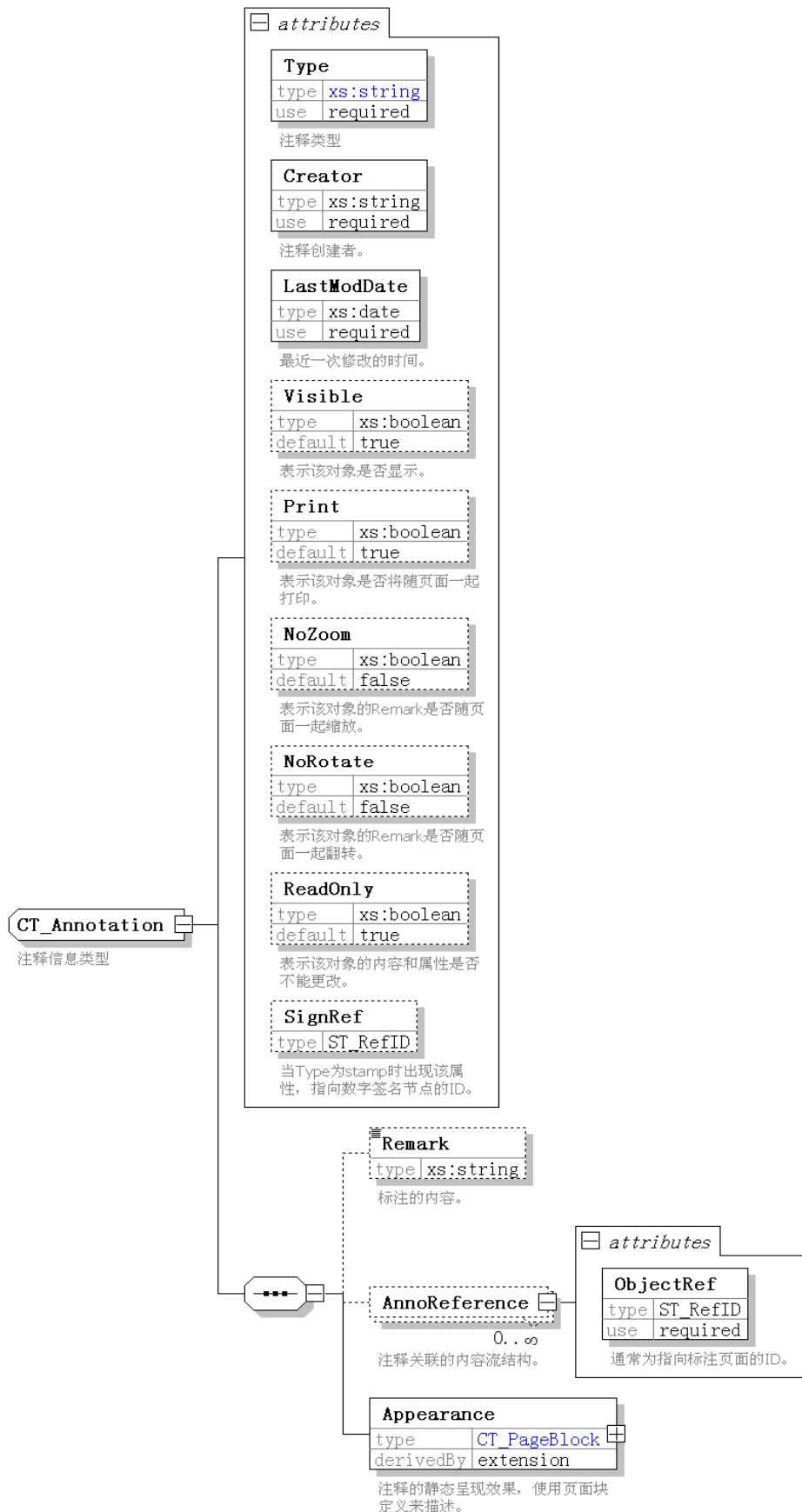


图 72 Annotation 结构

表 56 Annotation 属性表

名称	类型	说明	备注
Type	xs:string	注释类型，具体取值请见表 57	必需
Creator	xs:string	注释创建者	必需
LastModDate	xs:Date	最近一次修改的时间	必需
Visible	xs:boolean	表示该注释对象是否显示，默认值为 true	可选
Print	xs:boolean	表示对象的 Remark 信息是否随页面一起打印	可选
NoZoom	xs:boolean	表示对象的 Remark 信息是否随页面缩放而同步缩放，默认值为 false	可选
NoRotate	xs:boolean	表示对象的 Remark 信息是否随页面旋转而同步旋转，默认值为 true	可选
ReadOnly	xs:boolean	表示对象的 Remark 信息是否不能被用户更改，默认值为 true	可选
SignRef	ST_RefID	当 type 等于 Stamp 的时候，指向签名的引用	可选
Remark	xs:string	注释说明内容	可选
AnnoReference		被注释实体集	可选
ObjectRef	ST_RefID	通常为指向标注页面的 ID	必需
Appearance	CT_PageBlock	注释的静态呈现效果，使用页面块定义来描述	必需

注释类型由 Type 指定，Type 可取值见下表：

表 57 注释类型取值

类型	说明
Link	链接注释
Path	路径注释，此对象一般为图形对象，比如矩形、多边形、贝塞尔曲线等
Highlight	高亮注释
Stamp	签章注释，此对象一般为图像对象，Galleries 域中的 Signature 节点指明了其引用的数字签名
Watermark	水印注释

17 自定义标引

外部系统或用户可以添加自定义的标记和信息，从而达到与其它系统、数据进行交互的目的并扩展应用。

自定义标引架构的入口点在 7.5 文档根节点中定义，其结构如图 73 所示。

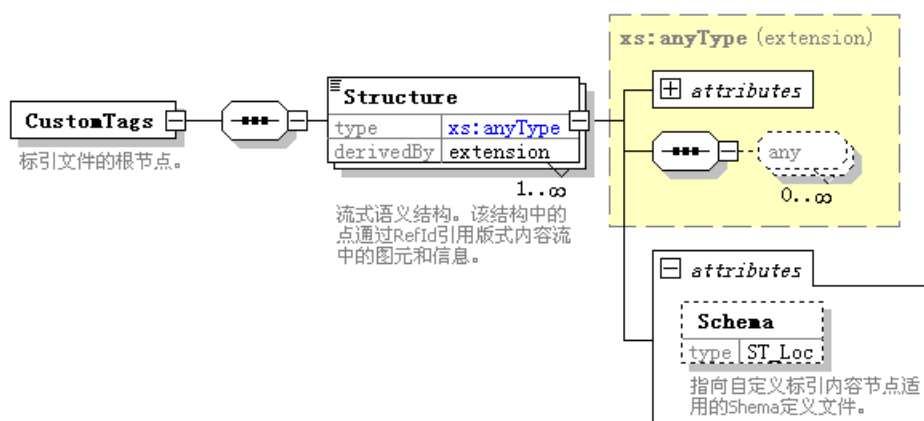


图 73 CustomTags 结构

表 58 CustomTags 属性

名称	类型	说明	备注
Structure	xs:anyType	流式语义结构。该结构中的点通过 RefId 引用版式内容流中的像素和信息	必需
Schema	ST_Loc	指向自定义标引内容节点适用的 Schema 定义档	可选

18 扩展

扩展信息的节点 Extensions 下由 1 到多个 Extension 节点组成。扩展的结构见图 75。

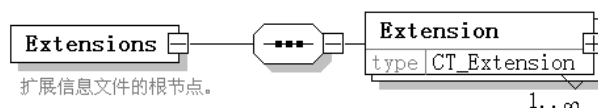


图 74 Extensions 结构

表 59 Extensions 属性

属性	类型	说明	备注
Extensions		扩展信息文件的根节点	
Extension	CT_Extension	扩展信息节点	

Extension 结构定义如图 75 所示。

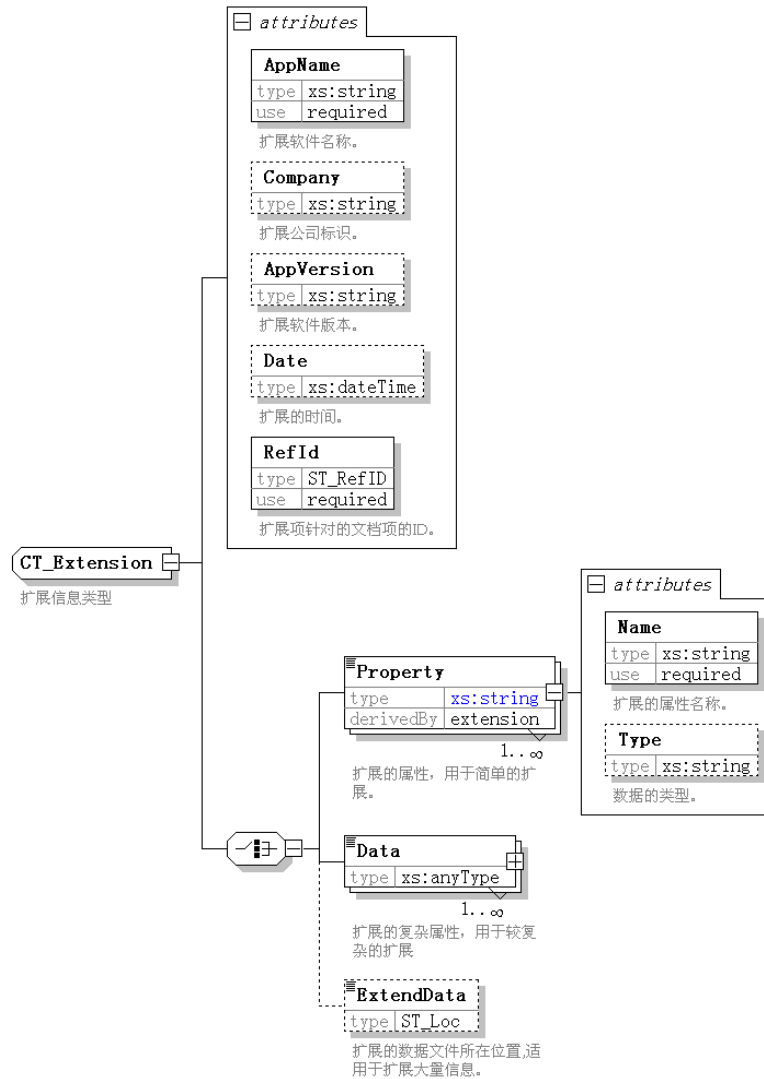


图 75 Extension 结构

表 60 Extension 属性

属性	类型	说明	备注
AppName	xs:string	用于生成或解释该自定义对象数据的应用程序名称	必需
Company	xs:string	扩展公司标识	可选
AppVersion	xs:string	扩展软件版本	可选
Date	xs:dateTime	扩展的时间	可选
RefId	ST_RefID	扩展项针对的文档项的 ID	必需
Property	xs:string	扩展的属性，是一个简单的“Name Type Value”的数值组。用于简单的扩展	
Name	Xs:string	扩展的属性名称	必需
Type	Xs:string	扩展的属性类型	可选
Data	xs:anyType	扩展的复杂属性，使用 xs:anyType，用于较复杂的扩展	
ExtendData	ST_Loc	扩展的数据文件所在位置，用于扩展大量信息	

附录 A

(规范性附录)

XDA 打包格式方案

A.1 包的组织

物理包结构应由文件头描述(Header), 档流入口描述(Entry), 内容流(BitStream)组成。其中, Entry 和 BitStream 在一个包中可出现多个, 形成一组修改的历史记录, 如图 2 所示。

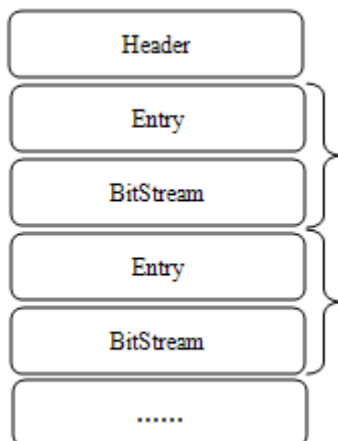


图 A1 包组织结构图

包结构的说明见表 A1。

表A1 包组织结构说明

包元素	说明
Header	档头, 描述版权版本、包含的历史版本数量等等一些最基本的信息。
Entry	文件流入口, 描述包内文件流的位置及相关信息。
BitStream	文件的内容流, 包含安全性信息描述。

文件头(Header)的结构如图 A2 所示:

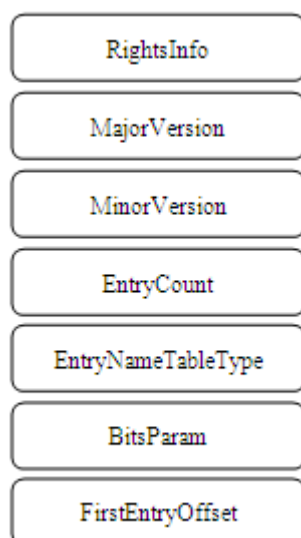


图 A2 文件头结构

Header 结构的说明见表 A2。

表A2 文件头结构说明

名称	长度(字节)	说明
RightsInfo	14	版权信息，指出文件遵循的包结构规范，版权信息采用固定的字符串“@XDA”，不足部分补零。
MajorVersion	1	包结构的主版本号，目前取值为 0x01。
MinorVersion	1	包结构的次版本号，目前取值为 0x00。
EntryCount	4	Entry 部分的数目。
EntryNameTableType	1	Entry 中 NameTable 类型，取值如下： 0x00 — 保留。
BitsParam	1	位参数，用以确定包结构中所有文件偏移量以及 BitStream 部分文件长度等数据占用的字节数，当 BitsParam 为 0 的时候表示默认值为 4 字节，有效值为 0x02、0x04、0x08。
FirstEntryOffset	由 BitsParam 决定	第一个 Entry 相对于档头部的偏移位置，以字节为单位。

Entry 部分的结构见图 A3 所示：

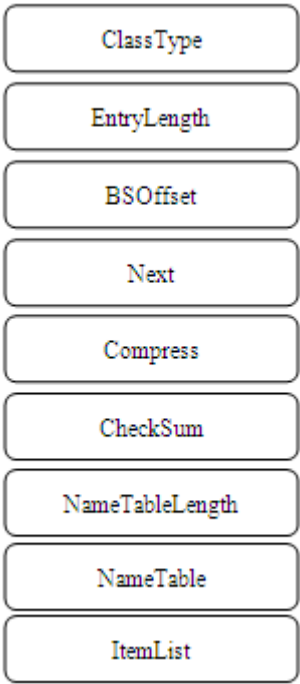


图 A3 文件流入口结构

文件流入口结构的说明见表 A3。

表A3 文件流入口结构说明

名称	长度(字节)	说明
ClassType	4	Entry 块类型标识，固定值为“C.En”。
EntryLength	4	此 Entry 块的长度，以字节为单位。
BSOffset	由 BitsParam 决定	BitStream 部分的偏移位置，以 Header 头部为起始位置。
Next	由 BitsParam 决定	下个 Entry 的偏移位置，如果没有则应为 0。

Compress	1	此 Entry 块的压缩方法,从低位 1 开始每一标记位定义如下: 1 — 如果为 1 表示 NameTable 采用 Deflate 算法压缩。 2 — 如果为 1 表示 ItemList 采用 Deflate 算法压缩。 其余未定义位应取值为 0。
CheckSum	16	此 Entry 块内 NameTable 和 ItemList 部分的校验码,采用 MD5 摘要算法。
NameTableLength	4	路径名映像表长度,以字节为单位元,表示 NameTable 部分数据的总长度,如果该部分数据使用压缩模式,则为压缩后的长度。
NameTable		项路径名映像表,应包括此 Entry 块所有项所使用的项路径映像,详细描述见图 5 和表 5。
ItemList		包内项入口描述列表,详细描述见图 6 和表 6。

路径名映像表 NameTable 结构见图 A4 所示:

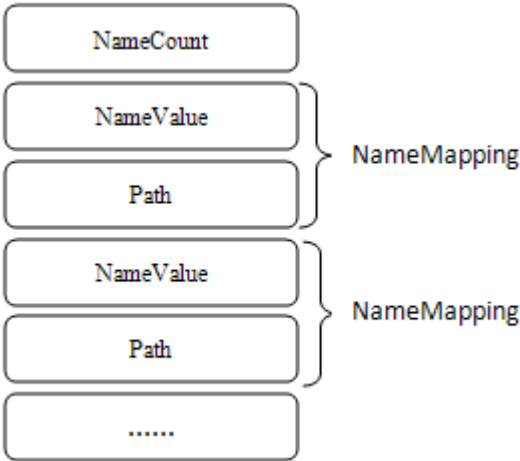


图 A4 路径名映像表结构

NameTable 结构的说明见表 A4。

表A4 路径名映像表结构说明

名称	长度(字节)	说明
NameCount	4	路径名个数。
NameMapping		路径名映像对,每一映像对表示一个路径名到映像值之间的对应关系。
NameValue	16	映射值。
Path	不定长	项路径,以 0 结尾,统一采用 UTF-8 编码。

包内项入口描述列表 ItemList 结构见图所示:

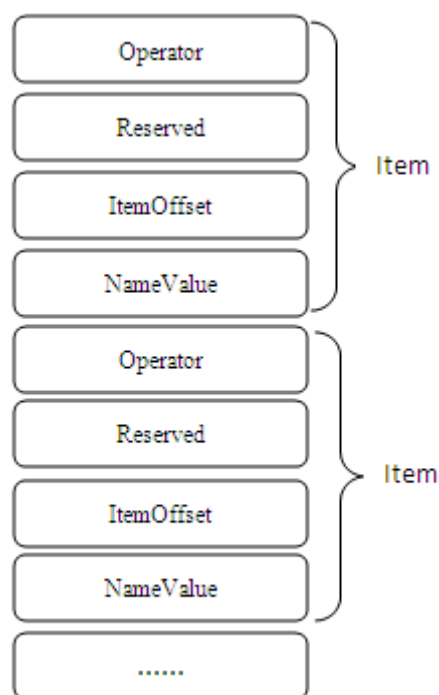


图 A5 包内项入口描述列表结构

包内项入口描述列表 ItemList 结构说明见表 A5。

表A5 包内项入口描述列表结构说明

名称	长度(字节)	说明
Item		包内项入口描述，需按照 NameValue 升序排序。
Operator	0.5	项操作符，占据低 4 位。各操作符的含义如下： 0001 New Operator：根据 NameValue 新建档，如果创建时档已经存在则报错。 0010 Append Operator：根据 NameValue 追加档，如果操作时档无效则报错。 0011 Replace Operator：根据 NameValue 替换档，如果操作时档无效则报错。 0100 Delete Operator：根据 NameValue 删除档，如果操作时档无效则报错。 1111 End Operator：当前 Entry 的档操作结束，以后的操作均无效。参数 ItemOffset 和 NameValue 部分以 0 补齐。每个 Entry 都应有一个 End 操作符，否则报错。
Reserved	0.5	保留，用于以后扩展。
ItemOffset	由 BitsParam 决定	项偏移位置，以该 Entry 的 BitStream 为起始位置，仅新建、追加或替换档时有效，根据 Operator 和 Type 进行判断。
NameValue	16	项路径映像值。

档内容流(BitStream)是多个文件流的顺序组合，其结构见图 A6 所示：

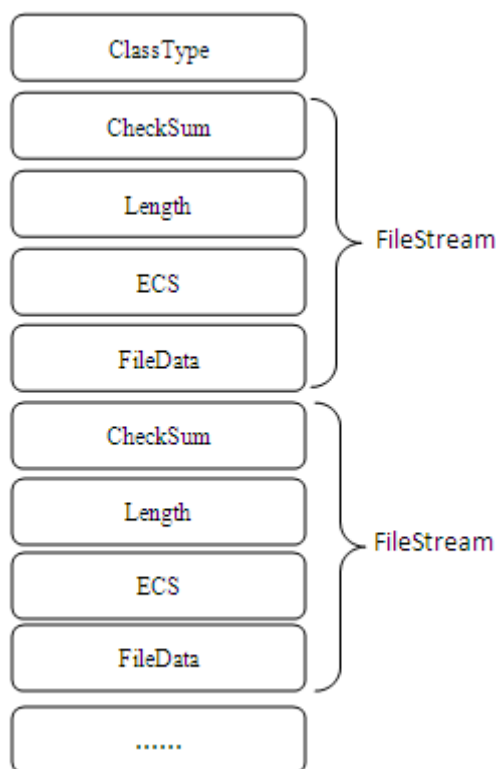


图 A6 文件内容流结构

文件内容流(BitStream)结构说明见表A6。

A6 文件内容流结构说明

名称	长度(字节)	说明
ClassType	4	BitStream 块类型标识，固定值为“C. BS”。
FileStream		文件流。
CheckSum	1	每个档流压缩后二进制数据内容的简单校验码，采用逐字节异或的算法。
Length	由 BitsParam 决定	文件流长度。
ECS	可变长	<p>说明每个档流生成时采用的压缩流程，每个算法占用一个字节，有先后顺序，最后以 FF 结尾，如果没有压缩，则仅保留 FF。</p> <p>0x01—0x0F：预定义压缩算法，取值如下：</p> <p>0x01 — 表明采用 SXC 算法压缩</p> <p>0x02 — 表明采用 Deflate 算法压缩</p> <p>0x10 —0x7F：自定义压缩算法的保留区间。</p> <p>[例外处理]ECS 序列不允许出现 0x00，如果出现则报错。</p> <p>[例外处理] ECS 序列最大长度为 8，包括结束符 0xFF，如果超出则报错。</p>
FileData		文件流资料。

A. 2 线性化

本打包结构应对包结构内部的数据块做线性化处理，以达到适应流式传输的需要。如图9所示，线性化的操作过程如下：

- 将所有Entry块合并后放在文件最开始的Header之后；
- 将所有BitStream顺序组织成一个BitStream；
- 修改 BSOffset 以及 ItemOffset 等指针，使其指向正确的位置。

在 Entry 块内部，应可以按页的顺序把每页所需的文件项顺序排列，可达到更好的线性化效果。

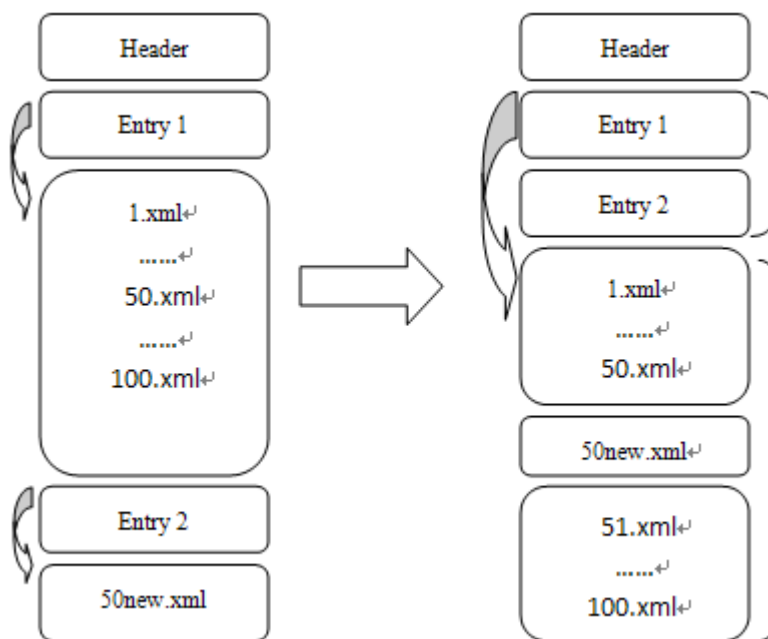


图 A7 线性化结构示意图

附录 B

（规范性附录）

使用 SVG 描述页面

B.1 字体

为了提供一个通用的字体格式来保证 SVG 可以支持所有符合其标准的用户，SVG 提供了可以在其上定义字体的功能，该功能被称为 SVG 字体。

SVG 字体由它的 **font** 元素定义。一个 SVG 字体可以被嵌入在使用该字体的相同文件中，也可以保存在外部资源中。

B.1.1 font 元素

font 属性：

名称	类型	动画属性	说明
horiz-origin-x	number	无	X 坐标用来绘制水平方向的文本，如果该属性没有指定具体的值，默认情况下为 0。.
horiz-origin-y	number	无	Y 坐标同样用来绘制水平方向的文本，如果该属性没有指定具体的值，默认情况下为 0。.
horiz-adv-x	number	无	在渲染一个字形之后，默认在水平方向的增长。
vert-origin-x	number	无	X 坐标用来绘制垂直方向的文本，如果该属性没有指定具体的值，默认情况下为 horiz-adv-x 的一半。
vert-origin-y	number	无	Y 坐标同样用来绘制垂直方向的文本，如果该属性没有指定具体的值，由字体的 ascent 属性进行设置。
vert-adv-y	number	无	在渲染一个字形之后，默认在垂直方向的增长。

B.1.2 font-face 元素

在描述字体的特征时，每一个 font 元素都有一个 font-face 子元素。该元素可以描述任何一种字体的特征。

font-face 属性：

名称	类型	说明
font-family	<string>	字体族
font-style	all [normal italic oblique] [, [normal italic oblique]]*	字体的风格。默认值为“all”。
font-variant	[normal small-caps] [, [normal	font-face 规则中的描述符，默认

	small-caps]]*	值为 normal。
font-weight	all [normal bold 100 200 300 400 500 600 700 800 900] [, [normal bold 100 200 300 400 500 600 700 800 900]]*	磅值
font-stretch	all [normal ultra-condensed extra-condensed condensed semi-condensed semi-expanded expanded extra-expanded ultra-expanded] [, [normal ultra-condensed extra-condensed condensed semicondensed semi-expanded expanded extra-expanded ultra-expanded]]*	在同一个 font family 中，一个 face 相对其他 face 的压缩或扩展属性。
font-size	<string>	字号
unicode-range	<urange> [, <urange>]*	ISO10646 字符范围，该范围覆盖了所有可能的字体中的字形。
units-per-em	<number>	在 em 正方形空间中坐标单元的数量。
panose-1	[<integer>]{10}	由 10 个十进制整数组成
stemv	<number>	在 font-face 规则中，该描述符具有相同的语法和语义。
stemh	<number>	在 font-face 规则中，该描述符具有相同的语法和语义。
slope	<number>	字体的竖向笔触角度。
cap-height	<number>	大写字母符号的高度。
x-height	<number>	小写字母符号的高度。
accent-height	<number>	accent 字符底部到顶部的距离。
ascent	<number>	字体最大 unaccented 高度
descent	<number>	字体最大 unaccented 深度
widths	<string>	在 font-face 规则中，该描述符具有相同的语法和语义。
bbox	<string>	在 font-face 规则中，该描述符具有相同的语法和语义。
ideographic	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现表意基线对齐。
alphabetic	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现字母基线对齐。
mathematical	<number>	针对水平方向的字形布局，该属性

		表示字形的对齐坐标，目的实现数学基线对齐。
hanging	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现挂基线对齐。
v-ideographic	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现表意基线对齐。
v-alphabetic	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现字母基线对齐。
v-mathematical	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现数学基线对齐。
v-hanging	<number>	针对水平方向的字形布局，该属性表示字形的对齐坐标，目的实现挂基线对齐。
underline-position	<number>	下划线在字体坐标系中的理想位置。
underline-thickness	<number>	下划线的理想厚度，在字体坐标系中用长度表示。
strikethrough-position	<number>	在字体坐标系中，strike-through的理想位置。
strikethrough-thickness	<number>	strike-through的理想厚度，在字体坐标系中用长度表示。
overline-position	<number>	字体坐标系中上划线的理想位置。
overline-thickness	<number>	上划线的理想厚度，在字体坐标系中用长度表示。

B. 1.3 glyph

字形元素用来定义给定字形的图像。字体元素中的各种属性定义了字形的坐标系。

glyph 属性:

名称	类型	说明
unicode	string	一个或者多个 unicode 字符表示对应字形的 unicode 字符序列。
d	path data	定义字形的轮廓。
orientation	h v	表示给定的字形只能是特定的 inline-progression-direction (例如水平或是垂直)
horiz-adv-x	<number>	渲染字形后在水平方向上的增长。

B. 1.4 missing-glyph

missing-glyph 属性:

名称	类型	说明
d	path data	定义字形的轮廓。
horiz-adv-x	<number>	渲染字形后在水平方向上的增长。

B. 2 裁剪

B. 2.1 裁剪路径 (Clipping paths)

裁剪路径用来约束可以绘画的区域。

clip 属性:

值	<shape> auto inherit
初始值	Auto
应用	建立视图的元素, ‘pattern’ 和 ‘maker’
继承	no
百分比	N/A
媒介	visual
动画效果	yes

B. 2.2 建立新的裁剪路径

clipPath 元素定义裁剪路径, 裁剪路径引用 clip-path 属性。

clipPath 属性:

名称	类型	说明
clipPathUnits	<i>userSpaceOnUse</i> <i>objectBoundingBox</i>	为 clip-path 内容定义坐标系。

clip-path 属性:

值	<uri> none inherit
初始值	none
应用	容器元素和图像元素
继承	no
百分比	N/A
媒介	visual
动画效果	yes

clip-rule 属性:

值	nonzero evenodd inherit
初始值	nonzero
应用	在 ‘clipPath’ 元素里面的图像元素
继承	yes

百分比	N/A
媒介	visual
动画效果	yes

B.3 渐变

B.3.1 轴向渐变

使用 `linearGradient` 元素定义轴向渐变。

`linearGradient` 属性:

名称	类型	说明
<code>gradientUnits</code>	<code>userSpaceOnUse</code> <code>objectBoundingBox</code>	定义 <code>x1</code> , <code>y1</code> , <code>x2</code> , <code>y2</code> 坐标的长度单位。 <code>userSpaceOnUse</code> 时, 长度单位取决于引用本 <code><linearGradient></code> 元素所在用户坐标系。 <code>objectBoundingBox</code> 时, 表明长度单位元以引用本 <code><linearGradient></code> 元素的图像对象的矩形外框为参考坐标系。
<code>gradientTransform</code>	<code><transform-list></code>	该属性表示渐变元素所定义的效果在施加于具体元素时的坐标变换方式。
<code>x1</code>	<code><coordinate></code>	渐变向量起点的 <code>x</code> 坐标。默认值 0%
<code>y1</code>	<code><coordinate></code>	渐变向量起点的 <code>y</code> 坐标。默认值 0%
<code>x2</code>	<code><coordinate></code>	渐变向量终点的 <code>x</code> 坐标。默认值 100%
<code>y2</code>	<code><coordinate></code>	渐变向量终点的 <code>y</code> 坐标。默认值 0%
<code>spreadMethod</code>	<code>pad</code> <code>reflect</code> <code>repeat</code>	说明了当渐变向量的定义区域小于其效果所施加元素的区域时所超出区域部分的处理方法。值为 <code>pad</code> 时表示仅在渐变向量起点和终点之间的区域实行渐变, 渐变效果只一次。值为 <code>reflect</code> 时表示使用镜像重复的方法。 <code>Repeat</code> 则表示简单地重复渐变过程。缺省值为 <code>pad</code>
<code>xlink:href</code>	<code><uri></code>	通过该属性可以引用另外一个已经定义好的渐变元素

示例：

shows how to fill a rectangle by referencing a linear gradient paint server
<pre><?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="8cm" height="4cm" viewBox="0 0 800 400" xmlns="http://www.w3.org/2000/svg" version="1.1"> <desc>Example linrad01 - fill a rectangle using a linear gradient paint server</desc> <g> <defs> <linearGradient id="MyGradient"> <stop offset="5%" stop-color="#F60" /> <stop offset="95%" stop-color="#FF6" /> </linearGradient> </defs> <!-- Outline the drawing area in blue --> <rect fill="none" stroke="blue" x="1" y="1" width="798" height="398"/> <!-- The rectangle is filled using a linear gradient paint server --> <rect fill="url(#MyGradient)" stroke="black" stroke-width="5" x="100" y="100" width="600" height="200"/> </g> </svg></pre>

效果如下图：



B. 3. 2 径向渐变

径向渐变由 radialGradient 元素定义。

radialGradient 属性：

名称	类型	说明
gradientUnits	userSpaceOnUse objectBoundingBox	定义 cx, cy, r, fx, fy 坐标的长度单位。userSpaceOnUse 时，长度单位取决于引用本 <

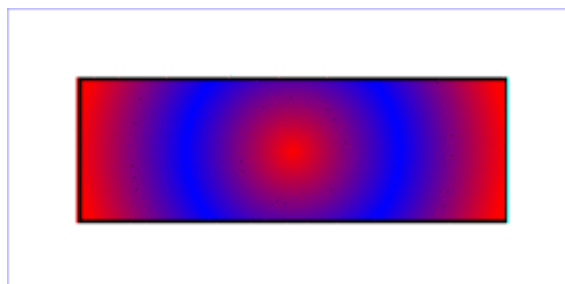
		linearGradient>元素所在用户坐标系。objectBoundingBox 时，表明长度单位元以引用本<linearGradient>元素的图像对象的矩形外框为参考坐标系。
gradientTransform	<transform-list>	该属性表示渐变元素所定义的效果在施加于具体元素时的坐标变换方式。
cx	<coordinate>	径向渐变的方法是定义一个圆，cx 是圆心的 x 坐标。缺省值是 50%
cy	<coordinate>	cy 是圆心的 y 坐标。缺省值是 50%
r	<coordinate>	圆的半径，不能为负值。缺省值 50%
fx	<coordinate>	Fx, fy 为径向渐变定义焦点。
fy	<coordinate>	Fx, fy 为径向渐变定义焦点。
spreadMethod	pad reflect repeat	描述渐变开始或结束时目标矩形范围内发生的变化。
xlink:href	<uri>	通过该属性可以引用另外一个已经定义好的渐变元素

示例：

shows how to fill a rectangle by referencing a radialgradient paint server
<pre> <?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="8cm" height="4cm" viewBox="0 0 800 400" xmlns="http://www.w3.org/2000/svg" version="1.1"> <desc>Example radgrad01 - fill a rectangle by referencing a radial gradient paint server</desc> <g> <defs> <radialGradient id="MyGradient" gradientUnits="userSpaceOnUse" cx="400" cy="200" r="300" fx="400" fy="200"> <stop offset="0%" stop-color="red" /> <stop offset="50%" stop-color="blue" /> <stop offset="100%" stop-color="red" /> </radialGradient> </defs> <!-- Outline the drawing area in blue --> <rect fill="none" stroke="blue" x="1" y="1" width="798" height="398"/> <!-- The rectangle is filled using a radial gradient paint server --> </pre>

```
<rect fill="url(#MyGradient)" stroke="black" stroke-width="5"
x="100" y="100" width="600" height="200"/>
</g>
</svg>
```

效果如下图：



B. 4 纹理

纹理用来对一个图形对象进行填充或描边。纹理定义一个 ‘pattern’ 元素，在目标图形元素上使用 ‘fill’ 或 ‘stroke’ 属性对图形对象进行填充或描边。

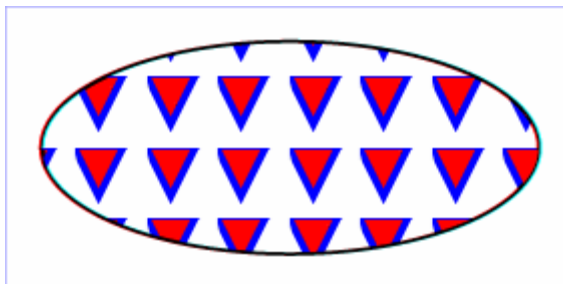
pattern 属性：

名称	类型	说明
patternUnits	userSpaceOnUse objectBoundingBox	定义 x，y， width， height 的长度单位。
patternContentUnits	userSpaceOnUse objectBoundingBox	定义模板内元素在矩形内分布时所使用的内部坐标系
patternTransform	<transform-list>	坐标系变换
x	<coordinate>	Pattern 元素需要定义一个矩形来容纳模板的内容，x 属性为矩形左上角的坐标。缺省值为 0%
y	<coordinate>	矩形左上角的 y 坐标。缺省值为 0%
width	<length>	矩形的宽度，不能为负值
height	<length>	矩形的高度，不能为负值
xlink:href	<uri>	引用另外一个已经引用好的 pattern 元素

示例:

shows how to fill a rectangle by referencing a pattern paint server
<pre><?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="8cm" height="4cm" viewBox="0 0 800 400" xmlns="http://www.w3.org/2000/svg" version="1.1"> <defs> <pattern id="TrianglePattern" patternUnits="userSpaceOnUse" x="0" y="0" width="100" height="100" viewBox="0 0 10 10"> <path d="M 0 0 L 7 0 L 3.5 7 z" fill="red" stroke="blue" /> </pattern> </defs> <!-- Outline the drawing area in blue --> <rect fill="none" stroke="blue" x="1" y="1" width="798" height="398"/> <!-- The ellipse is filled using a triangle pattern paint server and stroked with black --> <ellipse fill="url(#TrianglePattern)" stroke="black" stroke-width="5" cx="400" cy="200" rx="350" ry="150" /> </svg></pre>

效果如下图:



B. 5 文字

B. 5. 1 text 元素

‘text’ 元素定义一个图形元素来构成文字。‘text’ 元素里面包含 XML 的字符数据，文字本身包含一些相关的属性以及字符-字形的映像表。

text 属性:

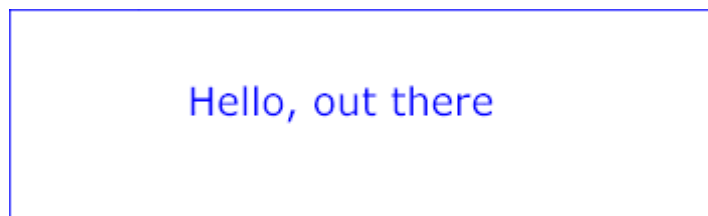
名称	类型	说明
x	<coordinate>+	默认值为 0。
y	<coordinate>+	默认值为 0。

dx	<length>+	该元素或是它的子元素中字符在当前位置沿着 X 轴发生的位移。
rotate	<number>+	当前文字位置的补充旋转将被应用于对应每个字符的所有字形中。

示例：

Render the text string "Hello, out there"
<pre><?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="10cm" height="3cm" viewBox="0 0 1000 300" xmlns="http://www.w3.org/2000/svg" version="1.1"> <desc>Example text01 - 'Hello, out there' in blue</desc> <text x="250" y="150" font-family="Verdana" font-size="55" fill="blue" > Hello, out there </text> <!-- Show outline of canvas using 'rect' element --> <rect x="1" y="1" width="998" height="298" fill="none" stroke="blue" stroke-width="2" /> </svg></pre>

效果如下图：



B. 5. 2 tspan 元素

属性：

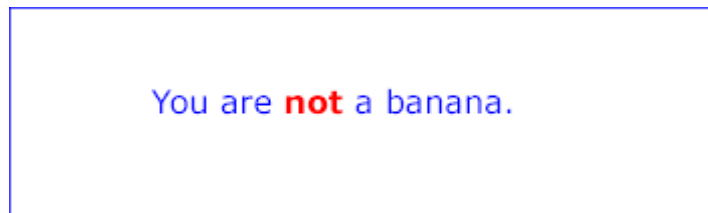
名称	类型	说明
x	<coordinate>+	如果存在一个 coordinate 该值表示当前文字位置的新的绝对 X 坐标。
dx	<length>+	在第一个字符字形被渲染之前，当前文字位置沿着用户坐标系的 X 轴发生的移动。
rotate	<number>+	当前文字位置的补充旋转将被应用于对应每个字符的所有字形中。

示例 1:

uses a 'tspan' element to indicate that the word "not" is to use a bold font and have red fill

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="10cm" height="3cm" viewBox="0 0 1000 300"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example tspan01 - using tspan to change visual attributes</desc>
<g font-family="Verdana" font-size="45" >
<text x="200" y="150" fill="blue" >
You are
<tspan font-weight="bold" fill="red" >not</tspan>
a banana.
</text>
</g>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="998" height="298"
fill="none" stroke="blue" stroke-width="2" />
</svg>
```

效果如下图:



示例 2:

uses the dx and dy attributes on the 'tspan' element to adjust the current text position horizontally and vertically for particular text strings within a 'text' element.

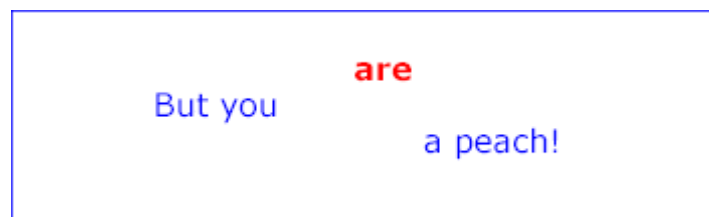
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="10cm" height="3cm" viewBox="0 0 1000 300"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example tspan02 - using tspan's dx and dy attributes
for incremental positioning adjustments</desc>
<g font-family="Verdana" font-size="45" >
<text x="200" y="150" fill="blue" >
But you
<tspan dx="2em" dy="-50" font-weight="bold" fill="red" >
are
```

```

</tspan>
<tspan dy="100">
a peach!
</tspan>
</text>
</g>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="998" height="298"
fill="none" stroke="blue" stroke-width="2" />
</svg>

```

效果如下图：



示例 3：

Example tspan03 uses the x and y attributes on the 'tspan' element to establish a new absolute current text position for each glyph to be rendered

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="10cm" height="3cm" viewBox="0 0 1000 300"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example tspan03 - using tspan's x and y attributes
for multiline text and precise glyph positioning</desc>
<g font-family="Verdana" font-size="45" >
<text fill="rgb(255,164,0)" >
<tspan x="300 350 400 450 500 550 600 650" y="100">
Cute and
</tspan>
<tspan x="375 425 475 525 575" y="200">
fuzzy
</tspan>
</text>
</g>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="998" height="298"
fill="none" stroke="blue" stroke-width="2" />
</svg>

```

效果如下图:



B.5.3 tref 元素

Text 元素中可以使用 tref 元素进行文本内容的引用, 该元素只能作为 text 元素的子元素而存在, 它包含一个属性。

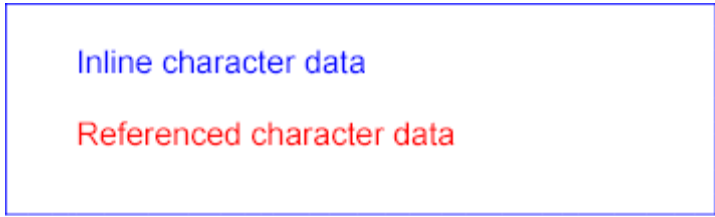
xlink:href = "<uri>"

下面是 tref 元素的一个例子:

示例 4:

shows how to use character data from a different element as the character data for a given 'tspan' element. The first 'text' element (with id="ReferencedText") will not draw because it is part of a 'defs' element.
<pre><?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="10cm" height="3cm" viewBox="0 0 1000 300" xmlns="http://www.w3.org/2000/svg" version="1.1" xmlns:xlink="http://www.w3.org/1999/xlink"> <defs> <text id="ReferencedText"> Referenced character data </text> </defs> <desc>Example tref01 - inline vs reference text content</desc> <text x="100" y="100" font-size="45" fill="blue" > Inline character data </text> <text x="100" y="200" font-size="45" fill="red" > <tref xlink:href="#ReferencedText"/> </text> <!-- Show outline of canvas using 'rect' element --> <rect x="1" y="1" width="998" height="298" fill="none" stroke="blue" stroke-width="2" /> </svg></pre>

效果如下图：



B. 5. 4 textPath 元素

textPath 属性：

名称	类型	动画属性	说明
startOffset	<length>	Yes	当前文本的起始位置，默认值为 0 并且不能为负值。
method	align stretch	Yes	表示文本沿着路径被渲染的方法。默认值为 align。
spacing	auto exact	Yes	空格的处理方式。默认值为 exact。
xlink:href	<uri>	Yes	指明所引用的 path 元素。

示例 5：

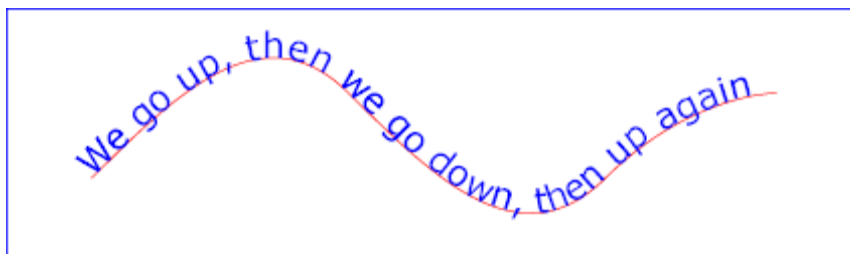
a simple example of text on a path

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="3.6cm" viewBox="0 0 1000 300" version="1.1"
xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
<defs>
<path id="MyPath"
d="M 100 200
C 200 100 300 0 400 100
C 500 200 600 300 700 200
C 800 100 900 100 900 100" />
</defs>
<desc>Example toap01 - simple text on a path</desc>
<use xlink:href="#MyPath" fill="none" stroke="red" />
<text font-family="Verdana" font-size="42.5" fill="blue" >
<textPath xlink:href="#MyPath">
We go up, then we go down, then up again
</textPath>
</text>
<!-- Show outline of canvas using 'rect' element -->
```



```
<rect x="1" y="1" width="998" height="298"
fill="none" stroke="blue" stroke-width="2" />
</svg>
```

效果如下图：



设置 inline-progression-direction

Writing-mode 属性：

值	lr-tb rl-tb tb-rl lr rl tb inherit
初始值	lr-tb
应用	Text 元素
继承	yes
百分比	N/A
媒介	visual
动画效果	no

B. 5. 5 字形方向

glyph-orientation-horizontal 属性：

值	<angle> inherit
初始值	0deg
应用	Text 包含的元素
继承	yes
百分比	N/A
媒介	visual
动画效果	no

direction 属性：

值	ltr rtl inherit
初始值	ltr
应用	Text 包含的元素
继承	yes
百分比	N/A
媒介	visual

动画效果	no
------	----

unicode-bidi 属性:

值	normal embed bidi-override inherit
初始值	normal
应用	Text 包含的元素
继承	no
百分比	N/A
媒介	visual
动画效果	no

B. 5. 6 基线对齐属性

‘alignment-baseline’ 属性:

值	auto baseline before-edge text-before-edge middle central after-edge text-after-edge ideographic alphabetic hanging mathematical inherit
初始值	auto
应用	‘tspan’, ‘tref’, ‘altGlyph’, ‘textPath’ 元素
继承	no
百分比	N/A
媒介	visual
动画效果	yes

B. 5. 7 字体选择属性

‘font-family’ 属性:

值	[[<family-name> <generic-family>],]*[<family-name> <generic-family>] inherit
初始值	depends on user agent
应用	Text 包含的元素
继承	yes
百分比	与 ‘text’ 元素的 ‘line-height’ 相关
媒介	visual
动画效果	yes

‘font-style’ 属性:

值	normal italic oblique inherit
初始值	normal
应用	Text 包含的元素
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

‘font-weight’ 属性:

值	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit
初始值	normal
应用	Text 包含的元素
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

‘font-size’ 属性:

值	<absolute-size> <relative-size> <length> <percentage> inherit
初始值	medium
应用	Text 包含的元素
继承	yes
百分比	refer to parent element's font size
媒介	visual
动画效果	yes

‘font’ 属性:

值	[[<'font-style'> <'font-variant'> <'font-weight'>]? <'font-size'>[/<'line-height'>]? <'font-family'>] caption icon menu message-box small-caption status-bar inherit
初始值	独立属性
应用	allowed on
继承	yes
百分比	允许的 ‘font-size’ 和 ‘line-height’
媒介	visual
动画效果	yes

B.6 图形

B.6.1 直线

line 元素用来定义两点间的一条直线。

line 属性:

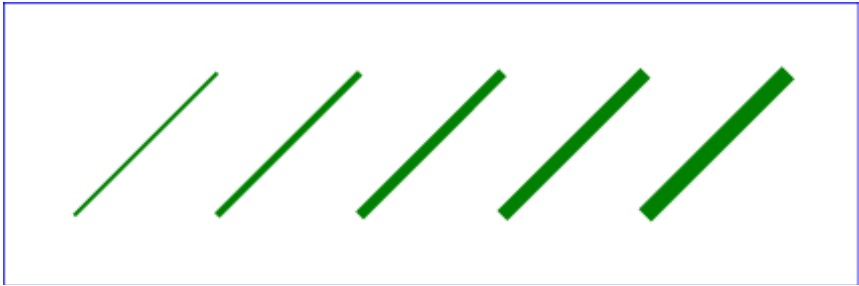
名称	类型	说明
x1	<coordinate>	直线起点的 X 轴坐标，默认值为 0。
y1	<coordinate>	直线起点的 Y 轴坐标，默认值为 0。
x2	<coordinate>	直线终点的 X 坐标，默认值

		为 0。
y2	<coordinate>	直线终点的 Y 坐标，默认值为 0。

示例：

The lines have different thicknesses
<pre> <?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="12cm" height="4cm" viewBox="0 0 1200 400" xmlns="http://www.w3.org/2000/svg" version="1.1"> <desc>Example line01 - lines expressed in user coordinates</desc> <!-- Show outline of canvas using 'rect' element --> <rect x="1" y="1" width="1198" height="398" fill="none" stroke="blue" stroke-width="2" /> <g stroke="green" > <line x1="100" y1="300" x2="300" y2="100" stroke-width="5" /> <line x1="300" y1="300" x2="500" y2="100" stroke-width="10" /> <line x1="500" y1="300" x2="700" y2="100" stroke-width="15" /> <line x1="700" y1="300" x2="900" y2="100" stroke-width="20" /> <line x1="900" y1="300" x2="1100" y2="100" stroke-width="25" /> </g> </svg> </pre>

效果如下图：



B. 6. 2 圆弧

本规范中，使用 path 来获取一段圆弧，具体属性如下：

名称	类型	说明
x1	<coordinate>	圆弧起点的 X 轴坐标，默认值为 0。

y1	<coordinate>	圆弧起点的 Y 轴坐标，默认值为 0。
x2	<coordinate>	圆弧终点的 X 坐标，默认值为 0。
y2	<coordinate>	圆弧终点的 Y 坐标，默认值为 0。
x-axis-rotation	double	旋转角度
rx	double	X 半轴长度
ry	double	Y 半轴长度

B. 6. 3 贝赛尔曲线

属性：

名称	类型	说明
x1	<coordinate>	贝塞尔曲线起点的 X 轴坐标，默认值为 0。
y1	<coordinate>	贝塞尔曲线起点的 Y 轴坐标，默认值为 0。
x2	<coordinate>	贝塞尔曲线终点的 X 坐标，默认值为 0。
y2	<coordinate>	贝塞尔曲线终点的 Y 坐标，默认值为 0。
cx1	<coordinate>	第一个控制点的 x 轴坐标
cy1	<coordinate>	第一个控制点的 y 轴坐标
cx2	<coordinate>	第二个控制点的 x 轴坐标
cy2	<coordinate>	第二个控制点的 y 轴坐标

示例：

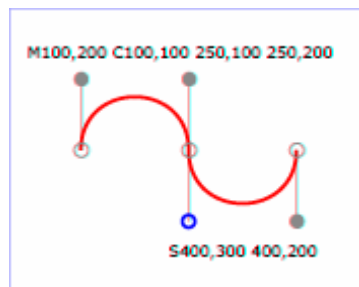
shows some simple uses of cubic Bézier commands within a path
<pre> <?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="5cm" height="4cm" viewBox="0 0 500 400" xmlns="http://www.w3.org/2000/svg" version="1.1"> <title>Example cubic01- cubic Bézier commands in path data</title> <desc>Picture showing a simple example of path data using both a "C" and an "S" command, along with annotations showing the control points and end points</desc> <style type="text/css"><![CDATA[.Border { fill:none; stroke:blue; stroke-width:1 } .Connect { fill:none; stroke:#888888; stroke-width:2 } .SamplePath { fill:none; stroke:red; stroke-width:5 } .EndPoint { fill:none; stroke:#888888; stroke-width:2 } </pre>

```

.CtlPoint { fill:#888888; stroke:none }
.AutoCtlPoint { fill:none; stroke:blue; stroke-width:4 }
.Label { font-size:22; font-family:Verdana }
]]></style>
<rect class="Border" x="1" y="1" width="498" height="398" />
<polyline class="Connect" points="100,200 100,100" />
<polyline class="Connect" points="250,100 250,200" />
<polyline class="Connect" points="250,200 250,300" />
<polyline class="Connect" points="400,300 400,200" />
<path class="SamplePath" d="M100,200 C100,100 250,100 250,200
S400,300 400,200" />
<circle class="EndPoint" cx="100" cy="200" r="10" />
<circle class="EndPoint" cx="250" cy="200" r="10" />
<circle class="EndPoint" cx="400" cy="200" r="10" />
<circle class="CtlPoint" cx="100" cy="100" r="10" />
<circle class="CtlPoint" cx="250" cy="100" r="10" />
<circle class="CtlPoint" cx="400" cy="300" r="10" />
<circle class="AutoCtlPoint" cx="250" cy="300" r="9" />
<text class="Label" x="25" y="70">M100,200 C100,100 250,100 250,200</text>
<text class="Label" x="325" y="350"
style="text-anchor:middle">S400,300 400,200</text>
</svg>

```

效果如下图：



B. 6. 4 矩形

属性：

名称	类型	说明
x	<coordinate>	矩形一边的 x 轴坐标，它是当前用户坐标系中较小的 x 轴坐标值。默认值为 0。.
y	<coordinate>	矩形一边的 y 轴坐标，它是当前用户坐标系中较小的 y 轴坐标值。默认值为 0。.
width	<length>	矩形的宽。

height	<length>	矩形的高。
rx	<length>	指圆角矩形的 x 轴半径。该值不能为负值。
ry	<length>	指圆角矩形的 y 轴半径。该值不能为负值。

示例：

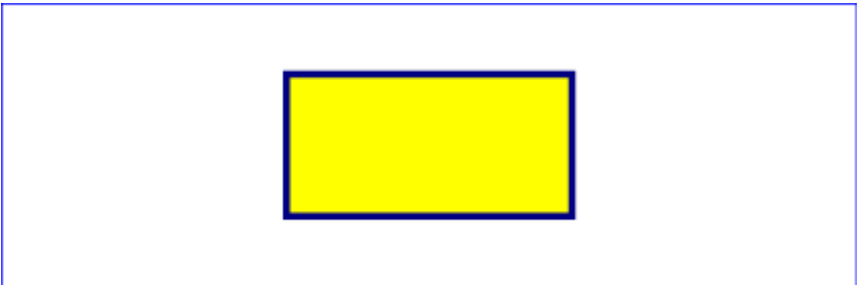
Example rect01
shows a rectangle with sharp corners

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example rect01 - rectangle with sharp corners</desc>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="1198" height="398"
fill="none" stroke="blue" stroke-width="2"/>
<rect x="400" y="100" width="400" height="200"
fill="yellow" stroke="navy" stroke-width="10" />
</svg>

```

效果如下图：



Example rect02
shows two rounded rectangles

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example rect02 - rounded rectangles</desc>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="1198" height="398"
fill="none" stroke="blue" stroke-width="2"/>
<rect x="100" y="100" width="400" height="200" rx="50"
fill="green" />
<g transform="translate(700 210) rotate(-30)">

```

```
<rect x="0" y="0" width="400" height="200" rx="50"
fill="none" stroke="purple" stroke-width="30" />
</g>
</svg>
```

效果如下图：



B. 6. 5 圆

‘circle’ 元素定义了一个基于一个中心点和一个半径的圆。

属性：

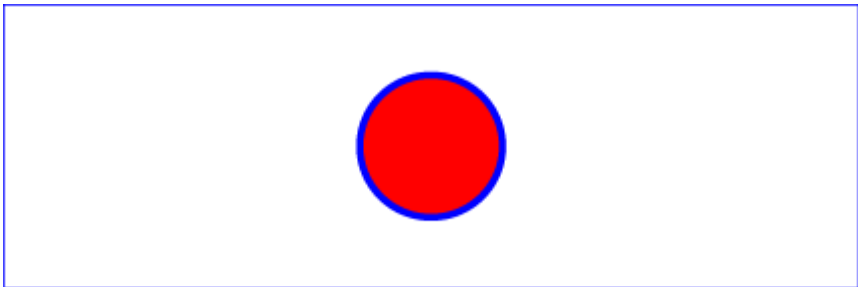
名称	类型	说明
cx	<coordinate>	圆中心的 x 轴坐标。默认值为 0。
cy	<coordinate>	圆中心的 y 轴坐标。默认值为 0。
r	<length>	圆的半径，不能为负值。

示例：

consists of a 'circle' element that is filled with red and stroked with blue

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example circle01 - circle filled with red and stroked with blue</desc>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="1198" height="398"
fill="none" stroke="blue" stroke-width="2"/>
<circle cx="600" cy="200" r="100"
fill="red" stroke="blue" stroke-width="10" />
</svg>
```

效果如下图：



B. 6. 6 椭圆

‘ellipse’ 元素定义了基于一个中心点和两个半径的椭圆。

属性:

名称	类型	说明
cx	<coordinate>	椭圆中心的 x 轴坐标。默认值为 0. .
cy	<coordinate>	椭圆中心的 y 轴坐标。默认值为 0.
rx	<length>	椭圆的 x 轴半径，不能为负值。0 值时，该元素不能被渲染。
ry	<length>	椭圆的 y 轴半径，不能为负值。0 值时，该元素不能被渲染。

示例:

below specifies the coordinates of the two ellipses in the user coordinate system established by the viewBox attribute on the 'svg' element and the transform attribute on the 'g' and 'ellipse' elements

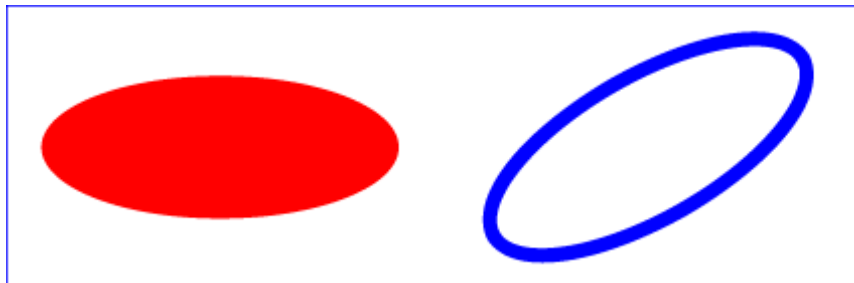
```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example ellipse01 - examples of ellipses</desc>
<!-- Show outline of canvas using 'rect' element -->
<rect x="1" y="1" width="1198" height="398"
fill="none" stroke="blue" stroke-width="2" />
<g transform="translate(300 200)">
<ellipse rx="250" ry="100"
fill="red" />
</g>
<ellipse transform="translate(900 200) rotate(-30)"
```

```

rx="250" ry="100"
fill="none" stroke="blue" stroke-width="20" />
</svg>

```

效果如下图：



B. 6. 7 路径

路径表现为一个图形的轮廓，它可以被填充、描边、或是作为裁剪区。一个路径使用当前点的概念进行描述，类似于在纸上绘画，当前点就是画笔的位置。画笔的位置是可以改变的，无论是直线还是曲线，图形的轮廓都可以通过拖动画笔来跟踪。

路径由 path 元素定义。

ath 属性：

名称	类型	说明
d	path data	图形轮廓定义。

B. 6. 8 裁剪区

- “overflow” 属性

Overflow= “visible|hidden|scroll|auto” 它的作用是当要显示元素的大小超出实际显示范围时，超出部分的处理方法。“visible” 表示可见，“hidden” 表示不可见，“scroll” 表示超出部分虽不可见，但可以加上滚动条使其有显示的余地。

- “Clip” 属性

Clip= “<shape>|auto” 该属性只能用于创建新视口的元素，它明确地说明了该新视口内图像的裁剪方式。

- <clipPath>元素

属性：

名称	类型	说明
----	----	----

clipPathUnits	userSpaceOnUse objectBoundingBox	子元素中各个坐标值的长度单位
---------------	------------------------------------	----------------

- Clip-path 属性:

Value	<uri> none inherit
Initial	none
Applies to	container elements and graphics elements
Inherited	no
Percentages	N/A
Media	visual
Animatable	yes

- Clip-rule 属性:

Value	nonzero evenodd inherit
Initial	nonzero
Applies to	graphics elements within a 'clipPath' element
Inherited	yes
Percentages	N/A
Media	visual
Animatable	yes

示例:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<svg height="10.905in" preserveAspectRatio="xMidYMin meet" viewBox="0 0 4488 6543"
width="10.905in"          xml:space="preserve"          xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
<defs>
  <clipPath id="cp1" style="clip-rule:nonzero">
    <path d="M 1000 300 L 1300 600 L 1200 350 Z"/>
  </clipPath>
</defs>
<rect style="fill:none" stroke="black" stroke-width="5"
x="1000" y="300" width="500" height="300"/>
<rect style="clip-path:url(#cp1)"
```

```
x="1000" y="300" width="500" height="300"/>
</svg>
```

B.7 图像

`image` 元素表示一个完整文件的内容将在当前用户坐标系中渲染成一个给定的矩形。

该元素和一些像素图文件相关联，例如 PNG、JPEG。

`ath` 属性：

名称	类型	说明
<code>x</code>	<code><coordinate></code>	文件矩形区域的一个角的 x 轴坐标。默认值为 0.
<code>y</code>	<code><coordinate></code>	文件矩形区域的一个角的 y 轴坐标。默认值为 0.
<code>width</code>	<code><length></code>	文件矩形区域的宽度。
<code>height</code>	<code><length></code>	文件矩形区域的高度。
<code>xlink:href</code>	<code><uri></code>	Uri 的一个引用。

B.8 绘制参数

‘`path`’ 元素，‘`text`’ 元素，以及一些基本图形都可以进行填充和描边。下面具体介绍这两种绘制参数。

B.6.1 fill 相关属性

	<code>fill</code>	<code>fill-rule</code>	<code>fill-opacity</code>
值	<code><paint></code>	<code>nonzero</code> <code>evenodd</code> <code>inherit</code>	<code><opacity-value></code> <code>inherit</code>
初始值	黑色	<code>nonzero</code>	1
应用	图形和文字	图形和文字	图形和文字
继承	yes	yes	yes
百分比	N/A	N/A	N/A
媒介	visual	visual	visual
动画效果	yes	yes	yes

B.6.2 stroke 相关属性

(1) `stroke`

值	<code><paint></code>
初始值	<code>none</code>
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual

动画效果	yes
------	-----

(2) stroke-width

值	<opacity-value> inherit
初始值	1
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

(3) stroke-linecap

值	butt round square inherit
初始值	butt
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

(4) stroke-linejoin

值	miter round bevel inherit
初始值	miter
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

(5) stroke-miterlimit

值	<miterlimit> inherit
初始值	4
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

(6) stroke-dasharray

值	none <dasharray> inherit
初始值	none
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

(7) stroke-dashoffset

值	<length> inherit
---	--------------------

初始值	0
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

(8) stroke-opacity

值	<opacity-value> inherit
初始值	1
应用	图形和文字
继承	yes
百分比	N/A
媒介	visual
动画效果	yes

B.9 xml:space 属性

名称	类型	说明
xml:space	{default preserve}	XML 的标准属性，用来表示空白空间是否保存在字符数据中。

B.10 transform 属性

名称	说明
matrix(<a> <c> <d> <e> <f>)	定义转换矩阵的格式，包括六个值
translate(<tx> [<ty>])	通过 tx 和 ty 定义一个转换，缺省情况下 ty 为 0
scale(<sx> [<sy>])	通过 sy 和 sx 定义一个 scale 操作，缺省情况下 sy 等于 sx
rotate(<rotate-angle> [<cx> <cy>])	定义给定点的旋转角度
skewX	沿着 x 轴倾斜角度的变化
skewY	沿着 y 轴倾斜角度的变化

B.11 viewBox 属性

该属性的作用是规定视口的坐标范围，范围重新定义后，缺省的坐标度量范围也会改变。

示例：

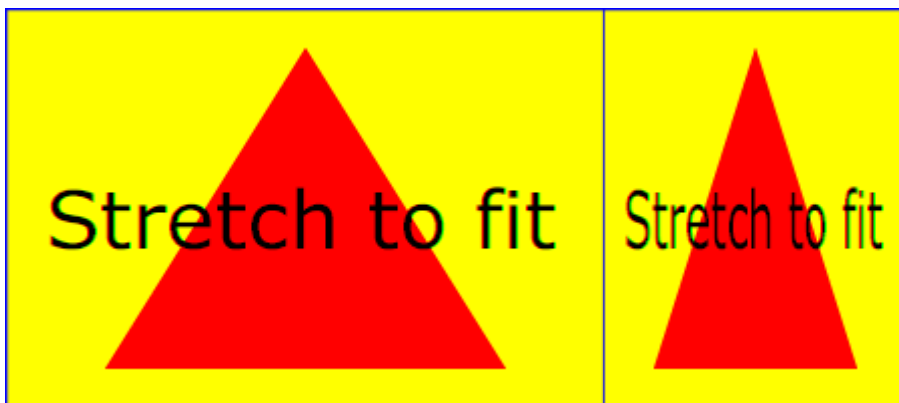
illustrates the use of the viewBox attribute on the outermost 'svg' element to specify that the SVG content should stretch to fit bounds of the viewport.
<pre><?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="300px" height="200px" version="1.1" viewBox="0 0 1500 1000" preserveAspectRatio="none" xmlns="http://www.w3.org/2000/svg"> <desc>Example ViewBox - uses the viewBox attribute to automatically create an initial user coordinate system which causes the graphic to scale to fit into the viewport no matter what size the viewport is.</desc></pre>

```

<!-- This rectangle goes from (0,0) to (1500,1000) in user space.
Because of the viewBox attribute above,
the rectangle will end up filling the entire area
reserved for the SVG content. -->
<rect x="0" y="0" width="1500" height="1000"
fill="yellow" stroke="blue" stroke-width="12" />
<!-- A large, red triangle -->
<path fill="red" d="M 750,100 L 250,900 L 1250,900 z"/>
<!-- A text string that spans most of the viewport -->
<text x="100" y="600" font-size="200" font-family="Verdana" >
Stretch to fit
</text>
</svg>

```

效果如下图：



B. 12 transform 属性

示例：translate 属性

```

<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="400px" height="150px" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<desc>Example NewCoordSys - New user coordinate system</desc>
<g fill="none" stroke="black" stroke-width="3" >
<!-- Draw the axes of the original coordinate system -->
<line x1="0" y1="1.5" x2="400" y2="1.5" />
<line x1="1.5" y1="0" x2="1.5" y2="150" />
</g>
<g>
<text x="30" y="30" font-size="20" font-family="Verdana" >
ABC (orig coord system)
</text>
</g>
<!-- Establish a new coordinate system, which is

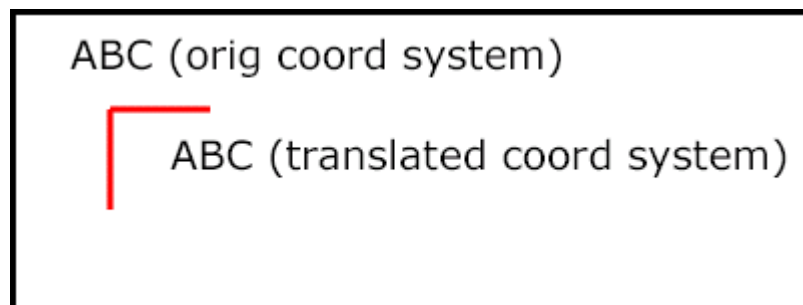
```

```

shifted (i.e., translated) from the initial coordinate
system by 50 user units along each axis. -->
<g transform="translate(50,50)">
<g fill="none" stroke="red" stroke-width="3" >
<!-- Draw lines of length 50 user units along
the axes of the new coordinate system -->
<line x1="0" y1="0" x2="50" y2="0" stroke="red" />
<line x1="0" y1="0" x2="0" y2="50" />
</g>
<text x="30" y="30" font-size="20" font-family="Verdana" >
ABC (translated coord system)
</text>
</g>
</svg>

```

效果如下：



示例：rotate 属性 和 scale 属性

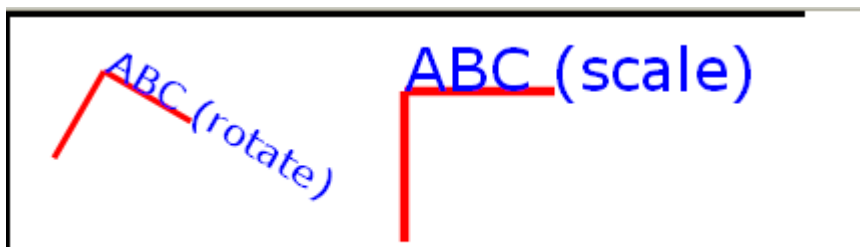
Rotate and scale
<pre> <?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="400px" height="120px" version="1.1" xmlns="http://www.w3.org/2000/svg"> <desc>Example RotateScale - Rotate and scale transforms</desc> <g fill="none" stroke="black" stroke-width="3" > <!-- Draw the axes of the original coordinate system --> <line x1="0" y1="1.5" x2="400" y2="1.5" /> <line x1="1.5" y1="0" x2="1.5" y2="120" /> </g> <!-- Establish a new coordinate system whose origin is at (50,30) in the initial coord. system and which is rotated by 30 degrees. --> <g transform="translate(50,30)"> <g transform="rotate(30)"> <g fill="none" stroke="red" stroke-width="3" > <line x1="0" y1="0" x2="50" y2="0" /> <line x1="0" y1="0" x2="0" y2="50" /> </pre>


```

</g>
<text x="0" y="0" font-size="20" font-family="Verdana" fill="blue" >
ABC (rotate)
</text>
</g>
</g>
<!-- Establish a new coordinate system whose origin is at (200,40)
in the initial coord. system and which is scaled by 1.5. -->
<g transform="translate(200,40)">
<g transform="scale(1.5)">
<g fill="none" stroke="red" stroke-width="3" >
<line x1="0" y1="0" x2="50" y2="0" />
<line x1="0" y1="0" x2="0" y2="50" />
</g>
<text x="0" y="0" font-size="20" font-family="Verdana" fill="blue" >
ABC (scale)
</text>
</g>
</g>
</svg>

```

效果如下：



示例：skew

defines two coordinate systems which are skewed relative to the origin coordinate system.

```

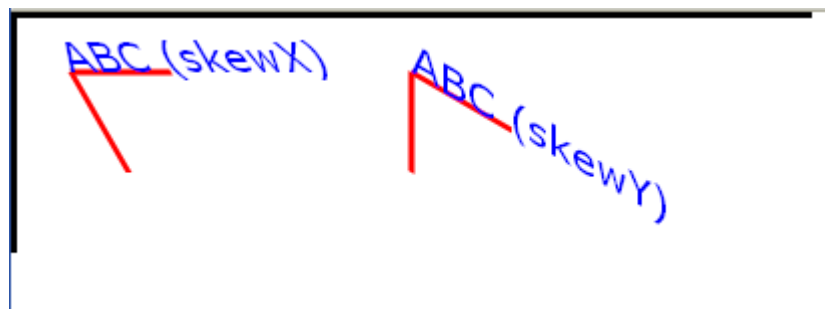
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="400px" height="120px" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<desc>Example Skew - Show effects of skewX and skewY</desc>
<g fill="none" stroke="black" stroke-width="3" >
<!-- Draw the axes of the original coordinate system -->
<line x1="0" y1="1.5" x2="400" y2="1.5" />
<line x1="1.5" y1="0" x2="1.5" y2="120" />
</g>
<!-- Establish a new coordinate system whose origin is at (30,30)

```

in the initial coord. system and which is skewed in X by 30 degrees. -->

```
<g transform="translate(30,30)">
<g transform="skewX(30)">
<g fill="none" stroke="red" stroke-width="3" >
<line x1="0" y1="0" x2="50" y2="0" />
<line x1="0" y1="0" x2="0" y2="50" />
</g>
<text x="0" y="0" font-size="20" font-family="Verdana" fill="blue" >
ABC (skewX)
</text>
</g>
</g>
<!-- Establish a new coordinate system whose origin is at (200,30)
in the initial coord. system and which is skewed in Y by 30 degrees. -->
<g transform="translate(200,30)">
<g transform="skewY(30)">
<g fill="none" stroke="red" stroke-width="3" >
<line x1="0" y1="0" x2="50" y2="0" />
<line x1="0" y1="0" x2="0" y2="50" />
</g>
<text x="0" y="0" font-size="20" font-family="Verdana" fill="blue" >
ABC (skewY)
</text>
</g>
</g>
</svg>
```

效果如下：



附录 C

（资料性附录）

SVG 描述版式页面示例

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<svg height="10.905in" preserveAspectRatio="xMidYMin meet" viewBox="0 0
4488 6543" width="10.905in" xml:space="preserve"
xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
  <rect fill="none" height="6542" stroke="lightgray" stroke-width="1"
width="4487" x="2" y="2"/>
```

```
  <text fill="#000000" fill-opacity="1" font-family="font1"
font-size="216"
transform="matrix(1, 0, 0, 1, 0, 0) translate(2027, 0) scale(1, 1)" x="0"
y="599">
    <tspan alignment-baseline="baseline" dx="0, 1"
xml:space="preserve">文字</tspan>
  </text>
```

```
  <image height="2597" preserveAspectRatio="none"
transform="matrix(1, 0, 0, 1, 0, 0)" width="3461" x="516"
xlink:href="image_00000001.png" y="2838"/>
```

```
  <g fill="#0000FF" fill-opacity="1" fill-rule="evenodd">
    <path
d="M2544, 981C1923, 981, 1419, 1200, 1419, 1469C1419, 1738, 1923, 1956, 2544, 19
56C3166, 1956, 3669, 1738, 3669, 1469C3669, 1200, 3166, 981, 2544, 981z"/>
  </g>
```

```
  <g fill="#FF0000" fill-opacity="1" fill-rule="nonzero">
    <path
d="M2544, 991L2548, 990L2551, 988L2553, 985L2553, 981L2553, 977L2551, 974L25
48, 972L2544, 972L2486, 972L2429, 974L2373, 977L2317, 981L2263, 987L2210, 994
L2157, 1001L2106, 1010L2056, 1020L2053, 1020L2004, 1031L1957, 1043L1912, 105
6L1868, 1069L1825, 1084L1784, 1099L1745, 1115L1708, 1132L1673, 1150L1639, 11
68L1608, 1187L1578, 1207L1575, 1210L1548, 1230L1524, 1251L1501, 1273L1481, 1
295L1463, 1318L1461, 1321L1445, 1344L1433, 1367L1423, 1391L1416, 1416L1415,
1419L1411, 1444L1410, 1469L1411, 1494L1415, 1519L1416, 1522L1423, 1547L1433
, 1571L1445, 1594L1461, 1617L1463, 1620L1480, 1643L1500, 1665L1523, 1687L154
8, 1708L1575, 1728L1604, 1748L1608, 1750L1639, 1769L1673, 1788L1708, 1805L17
45, 1822L1784, 1839L1825, 1854L1868, 1868L1912, 1882L1957, 1895L2004, 1906L2
```

053, 1917L2056, 1918L2106, 1927L2157, 1936L2210, 1944L2263, 1950L2317, 1956L2373, 1960L2429, 1963L2486, 1965L2544, 1966L2602, 1965L2659, 1963L2716, 1960L2771, 1956L2825, 1950L2879, 1944L2931, 1936L2982, 1927L3032, 1918L3036, 1917L3084, 1906L3131, 1895L3177, 1882L3221, 1868L3263, 1854L3304, 1839L3343, 1822L3381, 1805L3416, 1788L3449, 1769L3481, 1750L3510, 1731L3513, 1728L3540, 1708L3565, 1687L3587, 1665L3607, 1643L3625, 1620L3628, 1617L3643, 1594L3655, 1571L3665, 1547L3672, 1522L3673, 1519L3677, 1494L3679, 1469L3677, 1444L3673, 1419L3672, 1416L3665, 1391L3655, 1367L3643, 1344L3628, 1321L3626, 1318L3608, 1295L3588, 1273L3565, 1251L3540, 1230L3513, 1210L3484, 1190L3481, 1187L3449, 1168L3416, 1150L3381, 1132L3343, 1115L3304, 1099L3263, 1084L3221, 1069L3177, 1056L3131, 1043L3084, 1031L3036, 1020L3032, 1020L2982, 1010L2931, 1001L2879, 994L2825, 987L2771, 981L2716, 977L2659, 974L2602, 972L2544, 972L2540, 972L2537, 974L2535, 977L2535, 981L2535, 985L2537, 988L2540, 990L2544, 991L2602, 991L2659, 993L2716, 996L2771, 1000L2825, 1006L2879, 1013L2931, 1020L2982, 1029L3032, 1039L3032, 1029L3029, 1038L3077, 1049L3124, 1061L3170, 1074L3214, 1087L3256, 1102L3297, 1117L3336, 1133L3374, 1150L3409, 1168L3442, 1186L3474, 1205L3477, 1196L3470, 1203L3499, 1223L3526, 1243L3551, 1264L3574, 1286L3594, 1308L3612, 1331L3619, 1324L3610, 1328L3625, 1351L3637, 1374L3647, 1398L3654, 1423L3663, 1419L3654, 1419L3658, 1444L3660, 1469L3658, 1494L3654, 1519L3663, 1519L3654, 1515L3647, 1540L3637, 1564L3625, 1587L3610, 1610L3619, 1614L3612, 1607L3594, 1630L3574, 1652L3552, 1674L3527, 1695L3500, 1715L3506, 1722L3503, 1713L3474, 1732L3442, 1751L3409, 1770L3374, 1787L3336, 1804L3297, 1821L3256, 1836L3214, 1850L3170, 1864L3124, 1877L3077, 1888L3029, 1899L3032, 1908L3032, 1899L2982, 1908L2931, 1917L2879, 1925L2825, 1931L2771, 1937L2716, 1941L2659, 1944L2602, 1946L2544, 1947L2486, 1946L2429, 1944L2373, 1941L2317, 1937L2263, 1931L2210, 1925L2157, 1917L2106, 1908L2056, 1899L2056, 1908L2060, 1899L2011, 1888L1964, 1877L1919, 1864L1875, 1850L1832, 1836L1791, 1821L1752, 1804L1715, 1787L1680, 1770L1646, 1751L1615, 1732L1611, 1741L1618, 1735L1589, 1715L1562, 1695L1537, 1674L1514, 1652L1494, 1630L1477, 1607L1470, 1614L1479, 1610L1463, 1587L1451, 1564L1441, 1540L1434, 1515L1425, 1519L1434, 1519L1430, 1494L1429, 1469L1430, 1444L1434, 1419L1425, 1419L1434, 1423L1441, 1398L1451, 1374L1463, 1351L1479, 1328L1470, 1324L1476, 1331L1494, 1308L1514, 1286L1537, 1264L1561, 1243L1588, 1223L1582, 1216L1585, 1225L1615, 1205L1646, 1186L1680, 1168L1715, 1150L1752, 1133L1791, 1117L1832, 1102L1875, 1087L1919, 1074L1964, 1061L2011, 1049L2060, 1038L2056, 1029L2056, 1039L2106, 1029L2157, 1020L2210, 1013L2263, 1006L2317, 1000L2373, 996L2429, 993L2486, 991L2544, 991z"/>

</g>

<font-face font-family="font1" units-per-em="256"/>

<missing-glyph d="M36,36 H220V220H36z" horiz-adv-x="256"/>

<glyph horiz-adv-x="120" unicode=" " />

<glyph
d="M135, 171Q142, 172, 154, 172T179, 169T196, 157T201, 144T198, 137T175, 128T147, 120Q153, 114, 153, 107T149, 97Q140, 88, 132, 76Q140, 72, 144, 65Q147, 66, 150, 66Q161, 66, 168, 58T176, 45T173, 37T167, 35L162, 36Q159, 36, 153, 34Q153, 3, 147, -12T133, -35T120, -43T105, -35T89, -25T79, -16T66, 1T60, 19T68, 34T105, 54Q102, 62, 102, 67T104, 80T112, 87T124, 97T130, 108Q130, 109, 129, 109Q128, 109, 121, 105T99, 92T81, 83T78, 83T76, 87T78, 99T86, 109T128, 127T169, 149T175, 157T170, 160Q156, 160, 143, 154Q140, 141, 137, 135T128, 130T105, 140Q98, 136, 92, 136T81, 137T73, 139Q65, 139, 58, 132Q57, 125, 55, 115T47, 89T39, 70T36, 68Q33, 68, 29, 78T26, 93T30, 108Q42, 130, 42, 143Q42, 146, 41, 149T41, 153T42, 155T47, 150T54, 138Q74, 153, 103, 164Q96, 172, 92, 179T89, 188T91, 190Q114, 190, 135, 171L135, 171M135, 30Q130, 30, 126, 30T123, 31T126, 33Q133, 37, 133, 41Q133, 42, 132, 44Q128, 43, 121, 40T98, 27T79, 16T75, 12T74, 7T76, 1T84, -5T100, -12T114, -15Q124, -15, 129, -4T135, 30L135, 30" horiz-adv-x="256" unicode="字"/>

<glyph
d="M148, 52Q192, 30, 202, 27T220, 10T228, -15Q228, -21, 223, -21T200, -10T175, 6Q160, 20, 138, 33Q127, 21, 111, 14T79, 8T49, 12T29, 22T22, 32Q22, 34, 23, 34T28, 31Q37, 21, 60, 21Q76, 21, 92, 27T115, 44Q97, 51, 83, 60T57, 69T33, 61Q31, 59, 29, 59T28, 60Q28, 64, 36, 71T60, 78Q65, 78, 70, 77T92, 71T124, 61Q132, 83, 133, 104Q117, 93, 107, 88T94, 83T88, 85T80, 95T75, 106T79, 116Q92, 135, 96, 142T100, 151T98, 161T95, 172T93, 177T94, 178Q98, 178, 111, 171T130, 158T136, 145T129, 130Q145, 139, 150, 141T156, 145Q157, 147, 164, 147T175, 142T180, 132Q180, 123, 174, 123Q163, 123, 144, 111Q146, 102, 151, 94Q153, 91, 153, 81Q153, 64, 148, 52L148, 52" horiz-adv-x="256" unicode="文"/>

<defs>

<linearGradient id="lgStroke" gradientUnits="objectBoundingBox">

<stop offset="20%" stop-color="yellow"/>

<stop offset="80%" stop-color="blue"/>

</linearGradient>

<linearGradient id="lgFill" gradientUnits="objectBoundingBox">

<stop offset="20%" stop-color="blue"/>

<stop offset="80%" stop-color="yellow"/>

</linearGradient>

</defs>

<rect x="700" y="700" width="1100" height="400"
stroke="url(#lgStroke)" stroke-width="12" fill="url(#lgFill)"/>

<defs>

<radialGradient id="MyGradient" gradientUnits="userSpaceOnUse">

cx="400" cy="200" r="300" fx="400" fy="200">

<stop offset="10%" stop-color="white" />

```

        <stop offset="50%" stop-color="black" />
        <stop offset="100%" stop-color="gray" />
    </radialGradient>
</defs>
<rect fill="url(#MyGradient)" stroke="black" stroke-width="5"
x="100" y="100" width="600" height="200"/>

<defs>
    <clipPath id="cp1" style="clip-rule:nonezero">
        <path d="M 1000 300 L 1300 600 L 1200 350 Z"/>
    </clipPath>
</defs>
<rect style="fill:none" stroke="black" stroke-width="5"
x="1000" y="300" width="500" height="300"/>

<rect style="clip-path:url(#cp1)"
x="1000" y="300" width="500" height="300"/>

<defs>
    <pattern id="TrianglePattern" patternUnits="userSpaceOnUse"
x="0" y="0" width="100" height="100" viewBox="0 0 10 10" >
        <path d="M 0 0 L 7 0 L 3.5 7 z" fill="red" stroke="blue" />
    </pattern>
</defs>

<ellipse fill="url(#TrianglePattern)" stroke="black"
stroke-width="5"
cx="3000" cy="500" rx="350" ry="150" />

</svg>

```

附录 D

(规范性附录)

电子签名

D.1 电子签名结构

电子签名结构格式如图 D1，说明如表 D1 所示：

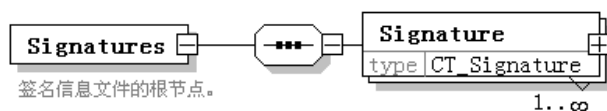


图 D1 Signatures 结构

表 D1 Signatures 说明

名称	类型	说明	备注
Signatures		签名节点	
Signature	CT_Signature	声明签名的标识	必需

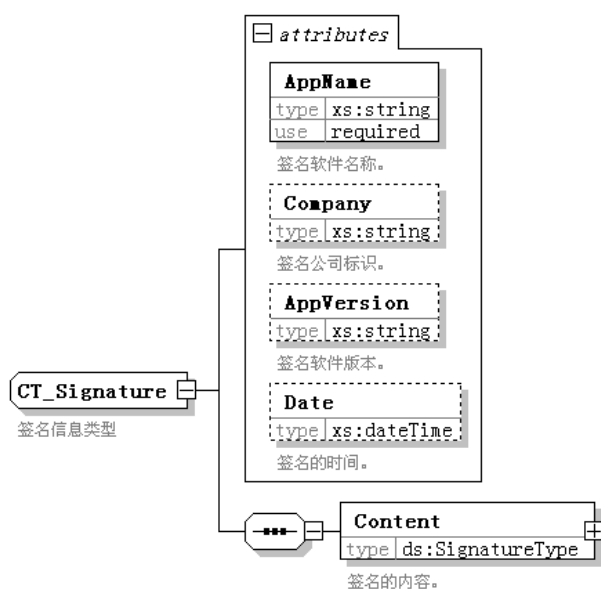


图 D2 Signature 结构

表 D2 Signature 说明

名称	类型	说明	备注
AppName	xs:string	签名软件名称	必需
Company	xs:string	签名公司名称	可选
AppVersion	xs:string	签名软件版本	可选
Date	xs:dateTime	签名的时间	可选
Content	ds:SignatureType	签名内容，见 XML Signature Syntax and Processing	必需