ICS 35.040

L 71



中华人民共和国国家标准

GB/T XXXXX—XXXX

|  |
| --- |
|       |

信息技术 可关闭字幕

Information technology - Closed Captioning

|  |
| --- |
| （征求意见稿） |
| 在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。 |

XXXX - XX - XX发布

XXXX - XX - XX实施



目  次

[前言 II](#_Toc454788357)

[引言 III](#_Toc454788358)

[1　范围 1](#_Toc454788359)

[2　规范性引用文件 1](#_Toc454788360)

[3　术语和定义 1](#_Toc454788361)

[4　缩略语 2](#_Toc454788362)

[5　约定 2](#_Toc454788363)

[6　架构 2](#_Toc454788364)

[7　可关闭字幕编码 8](#_Toc454788366)

[8　可关闭字幕存储 26](#_Toc454788367)

[9　可关闭字幕传输 31](#_Toc454788367)

前  言

本文件为GB/T XXXXX。

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

本文件由全国信息技术标准化技术委员会（SAC/TC 28）提出并归口。

本文件起草单位： 北京工业大学、上海工程技术大学、上海国茂数字技术有限公司、帕科视讯科技（杭州）股份有限公司、北京大学、中国国际电视总公司、上海交通大学、腾讯科技（深圳）有限公司、央视网、中兴通讯股份有限公司、中国电子技术标准化研究院、科大讯飞股份有限公司、咪咕文化科技有限公司、中国互联网协会信息无障碍工作委员会、清华大学无障碍发展研究院、北京牡丹电子集团有限责任公司、北京数字电视国家工程实验室有限公司、贵州多彩新媒体股份有限公司、河北无线传媒股份有限公司、广东博华超高清创新中心有限公司、深圳龙岗智能视听研究院。

本文件主要起草人：牟伦田、赵海武、付强、罗威、徐异凌、胡颖、盛尧、黄成、李婧欣、王玮、刘聪、何永春、李琳、徐嵩、黄畅、邵磊、毛珂、边祥国、蒋运雄、杨彦军、王宇、范涛、侯朴玥、姜志乾、白雅贤、李秋婷、邱溥业、柳建龙、宋国栋、曲文雍、原源、隋康尼、徐晖、李思远、毛健、李杰、焦磊、卢金禹、龙仕强、肖铁军、赵海英、许晓中、刘杉、尹宝才、王国中、张伟民、黄铁军、高文。

引  言

本文件可能涉及专利列表

本文件的发布机构对于该专利的真实性、有效性和范围无任何立场。

该专利持有人已向本文件的发布机构保证，他愿意同任何申请人在合理且无歧视的条款和条件下，就专利授权许可进行谈判。该专利持有人的声明已在本文件的发布机构备案，相关信息可以通过以下联系方式获得：

联 系 人：黄铁军（数字音视频编解码技术标准工作组秘书长）

通讯地址：北京大学理科2号楼2641室

邮政编码：100871

电子邮件：tjhuang@pku.edu.cn

电 话：+8610-62756172

传 真：+8610-62751638

网 址：http://www.avs.org.cn

请注意除上述专利外，本文件的某些内容仍可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

信息技术 可关闭字幕

1. 范围

GB/T XXXXX规定了可关闭字幕（Closed Captioning）的编码、存储和传输格式。

本文件适用于电视广播、应急广播、网络流媒体、网络电视和视频点播等应用，为听障人士提供信息无障碍服务，或为不同语种地区观众提供当地所使用主语言的相应翻译文本。

1. 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 1988-1998 信息技术 信息交换用七位编码字符集

GB 2312-1980 信息交换用汉字编码字符集 基本集

GB/T 4880.3-2009 语种名称代码 第3部分：所有语种的3字母代码

GB 13000-2010 信息技术 通用多八位编码字符集（UCS）

GB/T 17975.1 信息技术 运动图象及其伴音信息的通用编码 第1部分:系统

GB 18030-2005 信息技术 中文编码字符集

GB/T 18793-2002 信息技术 可扩展置标语言（XML）1.0

GB/T 20090.11-2015 信息技术 先进音视频编码 第11部分：同步文本

GB/T 33475.1-2019 信息技术 高效多媒体编码 第1部分：系统

GB/T 33475.6-XXXX 信息技术 高效多媒体编码 第6部分：智能媒体传输

W3C XML 可扩展置标语言（Extensible Markup Language (XML)）

IETF RFC 3550 RTP：实时应用传输协议（RTP： A transport protocol for real-time applications）

ISO/IEC 13818-1 信息技术 运动图像及其伴音信息的通用编码 第1部分：系统（ISO/IEC 13818-

1:2018 Information technology — Generic coding of moving pictures and associated audio

information — Part 1: Systems）

ISO/IEC 14496-12 信息技术 音视频对象的编码 第12部分：ISO基媒体文件格式（Information

technology – Coding of audio-visual objects – Part 12: ISO base media file format）

1. 术语和定义

下列术语和定义适用于本文件。

* 1.

可关闭字幕 closed captioning

与音视频内容具有关联关系的可以打开或关闭其呈现的文本。

注1：通常用于在电视、视频屏幕或其他视觉显示设备上为听力障碍人群提供语音的文本转录或非语音内容的文本描述；

注2：还可为不同语种地区观众提供当地所使用主语言的相应翻译文本。

* 1.

保留值 reserved

用于扩展目的保留的二进制位的值。不做特别说明情况下，其所有比特值为1。

* 1.

字幕文本 caption text

终端屏幕用于字幕显示的文字。

* 1.

CCF可关闭字幕文件 closed caption file

一种独立存储可关闭字幕的文本文件格式。

1. 缩略语

下列缩略语适用于本文件。

bslbf  左位在先比特串（bit string left bit first）

CC 可关闭字幕（Closed Captioning）

CCF 可关闭字幕文件（Closed Caption File）

CTS 组合时间戳（Composition Time Stamp）

DTS 解码时间戳（Decoding Time Stamp）

ISO BMFF ISO基媒体文件格式（ISO Base Media File Format）

MPEG-2 TS MPEG-2传输流（MPEG-2 Transport Stream）

MSB 最高位 (most significant bit)

PCR 节目时钟参考（Program Clock Reference）

PTS 显示时间戳（Presentation Time Stamp）

uimsbf  高位在前无符号整数（unsigned integer most significant bit first）

UTF 统一码转换格式（Unicode Tranformation Format）

1. 约定

下列约定适用于本文件。

描述符

b(8)：一个任意取值的字节；

f(n)：取特定值的连续n个二进制位；

r(n)：连续n个‘1’；

u(n)：n位无符号整数，高位在前；

utf8：用UTF-8格式编码的字符代码。

函数

以下函数用于语法描述。假定解码器中存在一个比特流指针，这个指针指向比特流中要读取的下一个比特的位置。函数由函数名及左右圆括号内的参数构成。函数也可没有参数。

next\_bits(n)

 返回比特流的随后n个比特，MSB在前，不改变比特流指针。如果剩余的比特少于n，则返回0。

round()

 四舍五入取整。

1. 架构

本标准的架构如图1所示，主要包含如下部分：

1）字幕编码

规定可关闭字幕基本流的语法和语义。

2）字幕存储

规定字幕文件格式，包括独立字幕文件格式CCF和基于ISO BMFF的扩展。

3）字幕传输

规定字幕基本流在MPEG-2 TS中传输的格式。在其他协议上的传输在附录A（规范性）中给出。

与字幕编码具有对应关系的可关闭字幕表示在附录B（资料性）中给出。



图 1. 可关闭字幕架构

1. 可关闭字幕编码

7.1字幕基本流语法

7.1.1 字幕序列定义

字幕序列定义见表1。

表1 字幕序列定义

|  |  |
| --- | --- |
| 字幕序列定义 | 描述符 |
| CC\_sequence() {  |  |
| do { |  |
| while (( next\_bits(32) == sample\_start\_code ) ) { |  |
| CC\_sample() |  |
| }  |  |
| } while ( next\_bits(32) != CC\_sequence\_end\_code) |  |
| **CC\_sequence\_end\_code** | f(32) |
| } |  |

7.1.2 字幕样本定义

字幕样本定义见表2：

表2 字幕样本定义

|  |  |
| --- | --- |
| 字幕样本定义 | 描述符 |
| CC\_sample() { |  |
| **sample\_start\_code** | f(32) |
| **CC\_type** //字幕类型：255=应急广播、1=普通文本字幕、2=图片、3=手语注释、4=直播字幕 | u(8) |
| **language** //三字符语言代码 | b(24) |
| **caption\_string\_offset** //本元素与字幕字符串之间的字节数 | u(8) |
| if(CC\_type!=4 && CC\_type!=255){//直播字幕和应急广播没有时间信息 |  |
| time\_infomation() |  |
| } |  |
| if(CC\_type!=255){//应急广播没有格式描述信息 |  |
| position\_description() |  |
| display\_description() |  |
| color\_description() |  |
| font\_description() |  |
| style\_description() |  |
| } |  |
| for(i=0;i<N;i++) |  |
| **user\_data\_byte** | b(8) |
| if(CC\_type!=2){ |  |
| caption\_string() |  |
| } |  |
| else{ |  |
|  **picture\_data\_byte** | b(8) |
| } |  |
| } |  |

7.1.3 时间信息定义

时间信息定义见表3：

表3 时间信息定义

|  |  |
| --- | --- |
| 时间信息定义 | 描述符 |
| time\_infomation() { |  |
| **time\_reference** //时间参考：1=PCR，2=节目起始时间 | u(2) |
| **time\_format** //时间格式：1=33位PTS，2=时分秒毫秒 | u(2) |
| **end\_type** //终止时间表达方式：0=指定终止时间，1=指定持续时间 | u(2) |
| **reserved** | r(2) |
| if(time\_format == 1){ |  |
| **reserved** | bslbf(4) |
| **PTS[32..30]** | bslbf(3) |
| **marker\_bit** | bslbf(1) |
| **PTS[29..15]** | bslbf(15) |
| **marker\_bit** | bslbf(1) |
| **PTS[14..0]** | bslbf(15) |
| **marker\_bit** | bslbf(1) |
| if(end\_type == 0){ |  |
| **reserved** | bslbf(4) |
| **ETS[32..30]** | bslbf(3) |
| **marker\_bit** | bslbf(1) |
| **ETS[29..15]** | bslbf(15) |
| **marker\_bit** | bslbf(1) |
| **ETS[14..0]** | bslbf(15) |
| **marker\_bit** | bslbf(1) |
| } |  |
| if(end\_type == 1){ |  |
| **reserved** | bslbf(4) |
| **duration[32..30]** | bslbf(3) |
| **marker\_bit** | bslbf(1) |
| **duration[29..15]** | bslbf(15) |
| **marker\_bit** | bslbf(1) |
| **duration[14..0]** | bslbf(15) |
| **marker\_bit** | bslbf(1) |
| } |  |
| } |  |
| if(time\_format == 2){ |  |
| **start\_hour\_add\_1** | u(8) |
| **start\_minute\_add\_1** | u(8) |
| **start\_second\_add\_1** | u(8) |
| **start\_millisecond\_add\_1** | u(10) |
| **reserved** | r(6) |
| if(end\_type == 0){ |  |
| **end\_hour\_add\_1** | u(8) |
| **end\_minute\_add\_1** | u(8) |
| **end\_second\_add\_1** | u(8) |
| **end\_millisecond\_add\_1** | u(10) |
| **reserved** | r(6) |
| } |  |
| if(end\_type == 1){ |  |
| **duration\_hour\_add\_1** | u(8) |
| **duration\_minute\_add\_1** | u(8) |
| **duration\_second\_add\_1** | u(8) |
| **duration\_millisecond\_add\_1** | u(10) |
| **reserved** | r(6) |
| } |  |
| } |  |
| } |  |

7.1.4 位置描述定义

位置描述定义见表4：

表4 位置描述定义

|  |  |
| --- | --- |
| 位置描述定义 | 描述符 |
| position\_description() {//指定字幕窗口的位置 |  |
| **origin** //原点：1=屏幕左上角，2=视频显示窗口左上角 | u(2) |
| **abs\_or\_relative** //绝对的还是相对的：1=绝对的，单位是像素；2=相对的，单位是1‰ | u(2) |
| **position\_format** //指定字幕窗口位置的方式：1=指定中心点，2=指定左上角和右下角坐标 | u(4) |
| if(position\_format == 1) { |  |
| **center\_x** //中心点横坐标 | u(15) |
| **marker\_bit** | f(1) |
| **center\_y** //中心点纵坐标 | u(15) |
| **marker\_bit** | f(1) |
| **reserved** | r(32) |
| } |  |
| if(position\_format == 2) { |  |
| **left** //左上角横坐标  | u(15) |
| **marker\_bit** | f(1) |
| **top** //左上角纵坐标 | u(15) |
| **marker\_bit** | f(1) |
| **right** //右下角横坐标 | u(15) |
| **marker\_bit** | f(1) |
| **bottom** //右下角纵坐标 | u(15) |
| **marker\_bit** | f(1) |
| } |  |
| } |  |

7.1.5 显示描述定义

显示描述定义见表5：

表5 显示描述定义

|  |  |
| --- | --- |
| 显示描述定义 | 描述符 |
| display\_description () { |  |
| **write\_direction** //书写方向，值为0表示从左到右,值为1表示从上到下 | u(1) |
| **horizontal\_justification** //水平对齐方式，0表示左对齐，1表示居中，2表示右对齐 | u(2) |
| **vertical\_justification** //垂直对齐方式，0表示顶端对齐，1表示居中，2表示底端对齐 | u(2) |
| **reserved** //保留 | u(11) |
| } |  |

7.1.6 颜色描述定义

颜色描述定义见表6：

表6 颜色描述定义

|  |  |
| --- | --- |
| 颜色描述定义 | 描述符 |
| color\_description() { |  |
| **background\_color\_red** //背景色的红色分量 | u(8) |
| **background\_color\_green** //背景色的绿色分量 | u(8) |
| **marker\_bit** | f(1) |
| **background\_color\_transparency** //背景色的透明度分量 | u(7) |
| **background\_color\_blue** //背景色的蓝色分量 | u(8) |
| **background\_width** //背景宽度，255=填充整个字幕窗口，1~15=从文字向外扩展指定个像素 | u(8) |
| **foreground\_color\_red** //前景色的红色分量 | u(8) |
| **foreground\_color\_green** //前景色的绿色分量 | u(8) |
| **marker\_bit** | f(1) |
| **foreground\_color\_transparency** //前景色的透明度分量 | u(7) |
| **foreground\_color\_blue** //前景色的蓝色分量 | u(8) |
| **reserved** | b(32) |
| } |  |

7.1.7 字体描述定义

字体描述定义见表7：

表7 字体描述定义

|  |  |
| --- | --- |
| 字体描述定义 | 描述符 |
| font\_description() { |  |
| **font\_id** //字体序号 | u(8) |
| **font\_size** //像素高度，abs\_or\_relative=1时单位是像素，abs\_or\_relative=2时单位是1‰ | u(8) |
| **reserved** | r(8) |
| } |  |

7.1.8 风格描述定义

风格描述定义见表8：

表8 风格描述定义

|  |  |
| --- | --- |
| 风格描述定义 | 描述符 |
| style\_description() { |  |
| if(CC\_type != 2) {  |  |
| **bold\_flag** //加粗标志，0=正常，1=加粗  | u(1) |
| **italic\_flag** //斜体标志，0=正常，1=斜体  | u(1) |
| **underline\_flag** //下划线标志，0=正常，1=加下划线  | u(1) |
| **reserved** | r(13) |
| **}** |  |
| if(CC\_type == 2) {  |  |
| **picture\_format** //图片格式，0=JPG，1=PNG，2=TIFF，3=GIF | u(8) |
| **reserved** | r(8) |
| **}** |  |
| } |  |

7.1.9 字幕字符串定义

字幕字符串定义见表9：

表9 字幕字符串定义

|  |  |
| --- | --- |
| 字幕字符串定义 | 描述符 |
| caption\_string() { |  |
| do { |  |
| while(next\_bits(8) != 0) { |  |
| **caption\_char** | utf8 |
| } |  |
| **zero\_byte** | f(8) |
| }while(next\_bits(32) != CC\_sequence\_end\_code && next\_bits(32) != sample\_start\_code) |  |
| } |  |

7.2字幕基本流语义

7.2.1 字幕序列

字幕序列由若干字幕样本组成，以字幕样本起始码开始，以序列结束码结束。

**字幕序列结束码 CC\_sequence\_end\_code**

位串‘0x000001C1’。标识字幕序列的结束。

起始码是一组特定的位串。在符合本文件的位流中，除起始码外的任何情况下都不应出现这些位串。

起始码由起始码前缀和起始码值构成。起始码前缀是位串“0000 0000 0000 0000 0000 0001”。所有的起始码都应字节对齐。

起始码值是一个8位无符号整数，用来表示起始码的类型，见表10。

表10 起始码值

|  |  |
| --- | --- |
| 起始码类型 | 起始码值（十六进制） |
| 字幕样本起始码（sample\_start\_code） | C0 |
| 字幕序列结束码（CC\_sequence\_end\_code） | C1 |
| 保留 | C2-C7 |
| 用户自定义起始码 | C8-FF，00-BF |

**掩码位 marker\_bit**

1位，取固定值1。设置掩码位的目的是避免出现伪起始码。

**保留位 reserved**

保留的若干位，每一位都取固定值1。

7.2.2 字幕样本

**字幕样本起始码 sample\_start\_code**

位串‘0x000001C0’。标识字幕样本的开始。

**字幕类型 CC\_type**

8位无符号整数，值为1表示字幕样本中包含的字幕是普通文本；值为2表示字幕样本中包含的字幕是图片；值为3表示字幕样本中包含的字幕是手语注释（带注释的文本，用于说明手语动作）；值为4表示字幕样本是直播字幕；值为255表示字幕样本是应急广播。其他取值保留。CC\_type取值分配见表11。

表11 CC\_type取值

|  |  |
| --- | --- |
| CC\_type | 含义 |
| 0 | 禁止 |
| 1 | 普通文本 |
| 2 | 图片 |
| 3 | 手语注释 |
| 4 | 直播字幕 |
| 5..254 | 保留 |
| 255 | 应急广播 |

当CC\_type的值为2时，字幕类型为图片。终端应忽略display\_description()、color\_description()、font\_description()中的格式信息。如果图片的宽度和高度与position\_description()中指定的字幕窗口的宽度和高度不一致,终端应调整图片使其适应字幕窗口。

当CC\_type的值为4时，字幕类型为直播字幕。当终端接收到直播字幕时，应按照以下方法处理：

——如果直播字幕的字幕字符串不为空，则按照以下方式处理：

1. 如果终端正在显示直播字幕，则立即停止正在显示的直播字幕，并立即显示本条直播字幕；
2. 如果终端没有正在显示的直播字幕，则立即显示本条直播字幕。

——如果直播字幕的字幕字符串为空，即没有出现caption\_char字段，则按照以下方式处理；

1. 如果终端正在显示直播字幕，则立即停止正在显示的直播字幕；
2. 如果终端没有正在显示直播字幕，则不做任何动作。

当CC\_type的值为255时，字幕类型为应急广播。当终端接收到应急广播时，应按照以下方法处理：

——如果应急广播的字幕字符串不为空，则立即开始按照以下规定的方式开始播放应急广播的内容,直到接收到下一条应急广播为止；

——如果应急广播的字幕字符串为空，即没有出现caption\_char字段，则按照以下方式处理；

1. 如果终端正在播放应急广播，则应立即停止正在播放的应急广播；
2. 如果终端没有正在播放应急广播，则不做任何动作。

应急广播的格式信息取以下值，其中有些字段的取值在指定的范围之内即可,终端可根据自身情况决定具体取值：

origin = 1

abs\_or\_relative = 2

position\_format = 2

left = 0

top应大于等于800小于等于900

right = 1000

bottom = 1000

write\_direction = 0

horizontal\_justification = 3（此字段在应急广播的格式描述中应被忽略）

vertical\_justification = 1

background\_color\_red = 16

background\_color\_green = 16

background\_color\_transparency = 100

background\_color\_blue = 240

background\_width = 255

foreground\_color\_red = 240

foreground\_color\_green = 240

foreground\_color\_transparency = 100

foreground\_color\_blue = 240

font\_id = 0

font\_size的值大于等于（bottom - top）\*0.7小于等于（bottom - top）\*0.9

bold\_flag = 0

italic\_flag = 0

underline\_flag = 0

应急广播应采用走马灯的方式播放。文字从屏幕右侧向左侧匀速运动，运动速度为每秒5个汉字。走马灯文字头尾间隔大于等于10个汉字，小于等于屏幕宽度。

如果应急广播的字幕字符串中出现回车、换行符，则终端应忽略这些回车、换行符。

**语言 language**

24位比特串， GB/T 4880.3-2009规定的三字母语言代码。符合本标准的终端至少应支持中文（zho）和英文（eng）。

**字幕字符串偏移 caption\_string\_offset**

8位无符号整数，表示本字段与字幕字符串之间的字节数。

**用户数据字节 user\_data\_byte**

用户数据字节，其数量由字幕字符串偏移和字幕格式信息、时间信息的字节数共同确定。用户数据也可以起到填充的作用。用户数据的含义本标准不做规定，但用户数据应保证不出现伪起始码。解码器可忽略这些数据。

**图片数据字节 picture\_data\_byte**

当CC\_type的值为2时，此字段中包含图片文件数据字节。

7.2.3 时间信息

**时间参考 time\_reference**

2位无符号整数。值为1表示字幕样本中的时间相对于节目时钟参考（PCR）；值为2表示字幕样本中的时间相对于节目的开始时刻，节目的开始时刻定义为0时刻。

**时间格式 time\_format**

2位无符号整数。值为1表示字幕样本中的时间格式是33位时间戳，单位是1/90000秒；值为2表示字幕样本中的时间格式是时分秒毫秒，分别用1/1/1/2个字节表示。如果时间参考取值1，则时间格式应取值1；如果时间参考取值2，则时间格式应取值2。

**终止时间表达方式 end\_type**

2位无符号整数。值为0表示字幕样本中给出字幕显示的终止时刻，值为1表示字幕样本中给出字幕显示的持续时间。

**显示时间戳 PTS**

33位无符号整数，表示字幕开始显示的时刻。为了防止出现伪起始码，显示时间戳被分为3个部分，中间用掩码位分割开。显示时间戳的编码格式与ISO/IEC 13818-1相同。

**结束时间戳 ETS**

33位无符号整数，表示字幕结束显示的时刻。为了防止出现伪起始码，结束时间戳被分为3个部分，中间用掩码位分割开。结束时间戳的编码格式与显示时间戳相同。

**字幕持续显示时间 duration**

33位无符号整数，表示字幕持续显示的时间长度，单位为1/90000秒。为了防止出现伪起始码，字幕持续显示时间被分为3个部分，中间用掩码位分割开。字幕持续显示时间的编码格式与显示时间戳相同。

**字幕开始时加1 start\_hour\_add\_1**

8位无符号整数，表示字幕开始显示时间的小时加1。start\_hour\_add\_1的取值范围是1..24。

**字幕开始分加1 start\_minute\_add\_1**

8位无符号整数，表示字幕开始显示时间的分钟加1。start\_minute\_add\_1的取值范围是1..60。

**字幕开始秒加1 start\_second\_add\_1**

8位无符号整数，表示字幕开始显示时间的秒加1。start\_second\_add\_1的取值范围是1..60。

**字幕开始毫秒加1 start\_millisecond\_add\_1**

10位无符号整数，表示字幕开始显示时间的毫秒加1。start\_millisecond\_add\_1的取值范围是1..1000。

字幕开始时间记为SH:SM:SS:SX，计算方式如下：

SH = start\_hour\_add\_1 - 1

SM = start\_minute\_add\_1 - 1

SS = start\_second\_add\_1 – 1

SX = start\_millisecond\_add\_1 – 1

**字幕结束时加1 end\_hour\_add\_1**

8位无符号整数，表示字幕结束显示时间的小时加1。end\_hour\_add\_1的取值范围是1..24。

**字幕结束分加1 end\_minute\_add\_1**

8位无符号整数，表示字幕结束显示时间的分钟加1。end\_minute\_add\_1的取值范围是1..60。

**字幕结束秒加1 end\_second\_add\_1**

8位无符号整数，表示字幕结束显示时间的秒加1。end\_second\_add\_1的取值范围是1..60。

**字幕结束毫秒加1 end\_millisecond\_add\_1**

10位无符号整数，表示字幕结束显示时间的毫秒加1。end\_millisecond\_add\_1的取值范围是1..1000。

字幕结束时间记为EH:EM:ES:EX，计算方式如下：

EH = end\_hour\_add\_1 - 1

EM = end\_minute\_add\_1 - 1

ES = end\_second\_add\_1 – 1

EX = end\_millisecond\_add\_1 – 1

**字幕持续时加1 duration\_hour\_add\_1**

8位无符号整数，表示字幕持续显示时间的小时加1。duration\_hour\_add\_1的取值范围是1..24。

**字幕持续分加1 duration\_minute\_add\_1**

8位无符号整数，表示字幕持续显示时间的分钟加1。duration\_minute\_add\_1的取值范围是1..60。

**字幕持续秒加1 duration\_second\_add\_1**

8位无符号整数，表示字幕持续显示时间的秒加1。duration\_second\_add\_1的取值范围是1..60。

**字幕持续毫秒加1 duration\_millisecond\_add\_1**

10位无符号整数，表示字幕持续显示时间的毫秒加1。duration\_millisecond\_add\_1的取值范围是1..1000。

字幕持续时间记为DH:DM:DS:DX，计算方式如下：

DH = duration\_hour\_add\_1 - 1

DM = duration\_minute\_add\_1 - 1

DS = duration\_second\_add\_1 – 1

DX = duration\_millisecond\_add\_1 – 1

7.2.4 位置描述

位置描述中包含样本显示位置的信息。字幕窗口示意图见图2。规定屏幕左上角为坐标原点，横坐标从左向右增加，纵坐标从上到下增加。屏幕水平分辨率记为ScreenWidth，屏幕垂直分辨率记为ScreenHeight，视频窗口左上角坐标记为（VideoWindowX0，VideoWindowY0），视频窗口的宽度和高度记为VideoWindowWidth和VideoWindowHeight。



图 2. 字幕窗口示意图

**原点 origin**

2位无符号整数，值为1表示坐标原点在屏幕左上角，值为2表示坐标原点在视频显示窗口左上角。

**绝对或相对 abs\_or\_relative**

2位无符号整数，值为1表示给出的位置坐标是绝对坐标，单位是像素；值为2表示给出的位置坐标是相对坐标，单位是1‰。

**指定字幕窗口位置的方式 position\_format**

4位无符号整数，值为1表示指定字幕窗口的中心坐标，字幕窗口的大小根据所需显示的字幕确定。值为2表示指定字幕窗口的左上角和右下角坐标。

**中心横坐标 center\_x**

**中心纵坐标 center\_y**

15位无符号整数。当abs\_or\_relative值为1时表示字幕窗口中心的绝对横坐标和纵坐标，以像素为单位；当abs\_or\_relative的值为2时表示字幕窗口中心的相对横坐标和纵坐标，单位是1‰。字幕窗口的中心坐标（CCWCenterX,CCWCenterY）按照以下方式计算：

if(origin==1){

 if(abs\_or\_relative==1){

CCWCenterX = center\_x

CCWCenterY = center\_y

}

 if(abs\_or\_relative==2){

CCWCenterX = round(ScreenWidth \* center\_x / 1000)

CCWCenterY = round(ScreenHeight \* center\_y / 1000)

}

 }

 if(origin==2){

 if(abs\_or\_relative==1){

CCWCenterX = VideoWindowX0 + center\_x

CCWCenterY = VideoWindowY0 + center\_y

}

 if(abs\_or\_relative==2){

CCWCenterX = VideoWindowX0 + round(VideoWindowWidth \* center\_x / 1000)

CCWCenterY = VideoWindowY0 + round(VideoWindowHeight \* center\_y / 1000)

}

 }

**字幕窗口左上角横坐标 left**

**字幕窗口左上角纵坐标 top**

**字幕窗口右下角横坐标 right**

**字幕窗口右下角纵坐标 bottom**

15位无符号整数。当abs\_or\_relative值为1时表示字幕窗口左上角和右下角的绝对横坐标和纵坐标，以像素为单位；当abs\_or\_relative的值为2时表示字幕窗口左上角和右下角的相对横坐标和纵坐标，单位是1‰。字幕窗口左上角和右下角的相对于屏幕左上角的绝对坐标按照以下方式计算：

if(origin==1){

 if(abs\_or\_relative==1){

CCW\_X0 = left

CCW\_Y0 = top

CCW\_X1 = right

CCW\_Y1 = bottom

}

 if(abs\_or\_relative==2){

CCW\_X0 = round(ScreenWidth \* left / 1000)

CCW\_Y0 = round(ScreenHeight \* top / 1000)

CCW\_X1 = round(ScreenWidth \* right / 1000)

CCW\_Y1 = round(ScreenHeight \* bottom / 1000)

}

 }

 if(origin==2){

 if(abs\_or\_relative==1){

CCW\_X0 = VideoWindowX0 + left

CCW\_Y0 = VideoWindowY0 + top

CCW\_X1 = VideoWindowX0 + right

CCW\_Y1 = VideoWindowY0 + bottom

}

 if(abs\_or\_relative==2){

CCW\_X0 = VideoWindowX0 + round(VideoWindowWidth \* left / 1000)

CCW\_Y0 = VideoWindowY0 + round(VideoWindowHeight \* top / 1000)

CCW\_X1 = VideoWindowX0 + round(VideoWindowWidth \* right / 1000)

CCW\_Y1 = VideoWindowY0 + round(VideoWindowHeight \* bottom / 1000)

}

 }

7.2.5 显示描述

**书写方向 write\_direction**

1位无符号整数，值为0表示文本从左到右显示，值为1表示文本从上到下显示。

**水平对齐 horizontal\_justification**

2位无符号整数，值为0表示左对齐，值为1表示居中，值为2表示右对齐，值为3表示两端对齐。

**垂直对齐 vertical\_justification**

2位无符号整数，值为0表示顶端对齐，值为1垂直居中，值为2表示底端对齐，值为3表示垂直两端对齐。

7.2.6 颜色描述

**背景色红色分量 background\_color\_red**

8位无符号整数，背景色的红色分量，取值范围是0到255。

**背景色绿色分量 background\_color\_green**

8位无符号整数，背景色的绿色分量，取值范围是0到255。

**背景色透明度 background\_color\_transparency**

7位无符号整数，背景色的透明度，取值范围是0到100，值为0表示完全透明，值为100表示完全不透明。

**背景色蓝色分量 background\_color\_blue**

8位无符号整数，背景色的蓝色分量，取值范围是0到255。

**背景宽度background\_width**

8位无符号整数，值为255表示背景色填满字幕窗口，值为0表示背景色无效，值为1到15表示用背景色给字幕字符镶边，镶边的宽度为指定的值，单位为像素。

**前景色红色分量 foreground\_color\_red**

8位无符号整数，前景色的红色分量，取值范围是0到255。

**前景色绿色分量 foreground\_color\_green**

8位无符号整数，前景色的绿色分量，取值范围是0到255。

**前景色透明度 foreground\_color\_transparency**

7位无符号整数，前景色的透明度，取值范围是0到100，值为0表示完全透明，值为100表示完全不透明。

**前景色蓝色分量 foreground\_color\_blue**

8位无符号整数，前景色的蓝色分量，取值范围是0到255。

7.2.7 字体描述

**字体序号 font\_id**

8位无符号整数，每一种语言最多支持256种字体。本字段的含义和language字段的取值有关，表12列出了语言为中文和英文时本字段的取值及其含义。

表12 font\_id取值及含义

|  |  |  |
| --- | --- | --- |
| **font\_id** | **language的值为zho时的含义** | **language的值为eng时的含义** |
| 0 | 黑体 | times new roman |
| 1 | 宋体 | arial |
| 2 | 楷体 | calibri |
| 3 | 仿宋 | helvetica |
| 4..255 | 保留 | 保留 |

**字体尺寸 font\_size**

8位无符号整数，字符的高度，取值范围是1..255，当abs\_or\_relative=1时单位是像素，当abs\_or\_relative=2且origin=1时单位是屏幕高度的1‰，当abs\_or\_relative=2且origin=2时单位是视频窗口高度的1‰。鉴于不同语言的字号规定不统一，本标准要求无论是哪种语言的哪种字体，解码器应将字符映射到指定高度，并保持纵横比。

7.2.8 风格描述

风格描述中包含样本显示风格的信息。

**加粗标志 bold\_flag**

1位无符号整数，值为0表示不加粗，值为1表示加粗。

**斜体标志 italic\_flag**

1位无符号整数，值为0表示不斜体，值为1表示斜体。

**下划线标志 underline\_flag**

1位无符号整数，值为0表示不加下划线，值为1表示加下划线。

**图片格式 picture\_format**

8位无符号整数，取值及含义见表13。

表13 picture\_format取值及含义

|  |  |
| --- | --- |
| picture\_format | 含义 |
| 0 | 禁止 |
| 1 | JPG |
| 2 | PNG |
| 3 | TIFF |
| 4 | GIF |
| 5..255 | 保留 |

7.2.9 字幕字符串

**字幕字符 caption\_char**

用UTF-8格式编码的一个字符。

**零字节 zero\_byte**

8位。取固定值0x00。

1. 可关闭字幕存储

8.1独立字幕文件

为了支持字幕的离线编辑，定义了可关闭字幕文件（CCF）格式。CCF文件是一个文本文件。

CCF文件定义见表14：

表14 CCF文件定义

|  |  |
| --- | --- |
| CCF文件定义 | 描述符 |
| CC\_file() { |  |
| for(i=0;i<n;i++){  |  |
| caption() //字幕条 |  |
| } |  |
| } |  |

字幕条定义见表15：

表15 字幕条定义

|  |  |
| --- | --- |
| 字幕条定义 | 描述符 |
| caption() { |  |
| for(i=0;i<n;i++){  |  |
| note\_line() //注释行 |  |
| } |  |
| for(i=0;i<n;i++){  |  |
| format\_line() //格式行 |  |
| } |  |
| counter\_line() //计数行 |  |
| time\_line() //时间行 |  |
| for(i=0;i<n;i++){  |  |
| caption\_line() //字幕行 |  |
| } |  |
| blank\_line() |  |
| } |  |

一个CCF文件由若干字幕条组成。一个字幕条由若干注释行、若干格式行、计数行、时间行、若干字幕行和一个空行组成。

**注释行note\_line()：**以‘#’开头的行，其他字符可以是任意可显示字符，直到行末。

**格式行format\_line()：**中间包含‘#’的行，‘#’后面是格式名称，‘#’前面是格式的值。

**计数行counter\_line()：**只包含一个整数的行，当前字幕文件第一个字幕条的计数行的数字应为0，计数行的数字宜每次加1.

**时间行time\_line()：**包含时间信息，有两种格式。当第二个时间表示结束时刻时，应与SRT文件保持一致，即

hh:mm:ss,ttt --> hh:mm:ss,ttt

当第二个时间表示持续显示时长时，采用以下格式：

hh:mm:ss,ttt dur hh:mm:ss,ttt

其中hh,mm,ss分别表示时分秒，ttt表示毫秒。

**字幕行caption\_line()：**在屏幕上显示的一行字幕，可包含任意可显示字符。一个字幕条可以包含多个字幕行。

**空行blank\_line()：**表示一个字幕条的结束。

一个字幕条中的注释行可以不出现。

一个CCF文件中的第一个字幕条应包含完整的格式行，对字幕格式进行完整的定义。其他字幕条可以不包含格式行，或只包含一部分格式行，对一部分格式进行重新设置。没有被重新设置的格式保持不变。

8.2基于ISOBMFF的文件封装

8.2.1 字幕媒体文件

 当可关闭字幕基本流基于ISOBMFF进行封装时，字幕基本流作为字幕媒体被封装为文件轨道，其轨道的HandlerBox 数据盒类型应为'subt' 且包含于MediaBox数据盒中。

当可关闭字幕基本流基于ISOBMFF进行封装时，可关闭字幕轨道的MediaInformationBox数据盒中应包含一个SubtitleMediaHeaderBox。

当可关闭字幕基本流基于ISOBMFF进行封装时，其DTS和CTS信息遵循ISOBMFF中TimeToSampleBox数据盒和CompositionOffsetBox 数据盒的使用规范。

8.2.2 可关闭字幕样本

可关闭字幕轨道中的一个样本应包含本标准第7章中规定的一个CC\_sample()，且不应包含CC\_sequence\_end\_code。

8.2.2可关闭字幕轨道样本入口

8.2.2.1 定义

**样本入口类型：**'avcc'
**容器：**Sample Table Box ('stbl')
**强制性：**可关闭字幕轨道中应至少包含一个'avcc'样本入口
**数量：**一个或多个

可关闭字幕轨道的每个样本入口应是一个'avcc'样本入口，其样本入口类型等于'avcc'。

8.2.2.2 语法

class AVSClosedCaption() extends SubtitleSampleEntry ('avcc') {
}

1. 可关闭字幕传输

9.1可关闭字幕基本流的PES语法

可关闭字幕基本流在PES中封装的语法见表16：

表16 可关闭字幕基本流的PES定义

|  |  |  |
| --- | --- | --- |
| 语 法 | 位数 | 助记符 |
| PES\_packet(){ |  |  |
| **packet\_start\_code\_prefix** //0x000001 | **24** | **bslbf** |
| **stream\_id** //0xFD |  **8** | **uimsbf** |
| **PES\_packet\_length** //根据实际情况设置 | **16** | **uimsbf** |
| **CC\_start\_code\_value** | **8** |  |
| if (stream\_id = = 0xFA){ |  |  |
| if (CC\_start\_code\_value == 0xC0){ |  |  |
| CC\_sample\_without\_startcode() //去掉起始码的CC\_sample() |  |  |
| } |  |  |
| for(i=0;i<N;i++){ |  |  |
| **stuffing\_byte** |  |  |
| } |  |  |
| } |  |  |
| } |  |  |

9.2可关闭字幕基本流的PES语义

**stream\_id**：流标识符字段，取值0xFD(extended\_stream\_id)；

**PES\_packet\_length**：PES分组长度字段，应等于字幕样本去掉起始码前缀后的字节数加填充字节数；

**CC\_start\_code\_value**：字幕起始码值，取值应为0xC0（字幕样本起始码）或0xC1（字幕序列结束码）。

1. IET Intelligent Transport Systems
2. （规范性附录）
可关闭字幕在MPEG TS之外的协议上传输
3. 1.字幕在RTP上传输
4. 1.1 RTP净载格式
5. 1.1.1 RTP头的使用

IETF RFC 3550中定义了RTP头的格式，见图A.1。

1. 0 1 2 3
2. 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
3. +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
4. |V=2|P|X| CC |M| PT | sequence number |
5. +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
6. | Timestamp |
7. +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
8. | synchronization source (SSRC) identifier |
9. +=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
10. | contributing source (CSRC) identifiers |
11. | .... |
12. +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
13. 图A.1 RTP头

本文件所述RTP净载格式对其中数据域的使用方法符合IETF RFC 3550的规定。

1. 1.1.2 净载的通用格式

本净载格式定义了两种不同的基本净载结构，每种又可分为不同的子类型：

• 单样本包

一个RTP包净载中仅包含一个本标准第7章中定义的CC\_sample()。

• 复合包

一个RTP包净载中包含多于一个单时间CC\_sample()。

不同的净载结构通过RTP净载中的第一个字节PSI（净载结构指示符）来进行区分。PSI格式定义见图A.2。

 +---------------+

 |0|1|2|3|4|5|6|7|

 +-+-+-+-+-+-+-+-+

 |F|NRI| Type |

 +---------------+

1. 图A.2 PSI（净载结构指示符）格式

PSI各字段语义如下：

F：1 bit

值0表示RTP净载中不宜包含比特错误或其它语法错误。值1表示RTP净载中可能包含比特错误或其它语法错误。如果检测到RTP净载中有比特错误，MANE宜将F设为1。

NRI：2 bit

NRI值表明RTP包的相对传输优先级。MANE可以利用这个信息更好地保护更重要的RTP包。优先级由高到低的NRI值依次为11b、10b、01b和00b。

Type：5 bit

用于指示RTP净载的结构，语义见表A.1。

表A.1 PSI中Type字段所对应的净载结构

 Type 净载结构 净载结构全称

 ---------------------------------------------------------

 0 未规定

 1-6 CC\_sample 单样本包

 7 STAP 单时间复合包

 8-31 未规定

1. 1.2.2.1 单样本包

单样本包必须包含且仅包含一个本标准第7章定义的CC\_sample()。复合包不能在单样本包中使用。通过按照RTP序列号顺序解封装单样本包所得到的字幕基本流必须符合字幕基本流的解码顺序。在单样本包净载结构中，RTP净载中的第一个字节为PSI信息，其后紧跟CC\_sample()，见图A.3。

 0 1 2 3

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 |F|NRI| Type |<---PSI |

 +-+-+-+-+-+-+-+-+ |

 | |

 | 第2..n字节（CC\_sample()） |

 | |

 | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | : ...可选的RTP填充数据 |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

1. 图A.3 单样本包的RTP净载格式
2. 1.2.2.2 复合包

本净载格式定义了单时间复合包（STAP）：包含具有相同时间信息的多个CC\_sample()，其中时间信息指一个CC\_sample()打包为一个RTP包时的RTP时间戳。

复合包的RTP净载格式见图A.4。

 0 1 2 3

 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 |F|NRI| Type |<---PSI |

 +-+-+-+-+-+-+-+-+ |

 | |

 | 一个或多个复合单元 |

 | |

 | +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 | :...可选的RTP填充数据 |

 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

1. 图A.4 复合包的RTP净载格式
2. 2 字幕在AS上传输

需要对GB/T 33475.1（AVS2-P1）做以下扩展以支持CC：

1）以下术语需要列举出CC

连续媒体 continuous media；

媒体内容 media content；

表示 representation。

2）MPD结构中需要列举出CC；

3）表 9 AdaptationSet、Representation 和 SubRepresentation 的公共属性和元素的语义需要补充：

CC字幕时间信息描述与7.2.3一致；

CC字幕窗口的位置描述与7.2.4一致；

CC字幕的显示信息描述与7.2.5-7.2.9一致。

4）同步插播需要支持CC字幕流与视频流、音频流的同步插播。

1. 3 字幕在SMT上传输
2. 3.1 SMTP包结构

SMTP包结构如图8所示。

1. 
2. 图A.5 SMTP包结构

字幕数据作为SMTP包负载封装在payload data字段中。同时，为支持字幕数据在SMTP的传输，增加了SMTP包对数据类型的定义。

type：标识负载数据类型。负载类型值见表A.2。

1. 表A.2 数据类型与数据单元定义

| **值** | **数据类型** | **数据单元定义** |
| --- | --- | --- |
| 0x00 | CEU | 媒体感知的CEU片段 |
| 0x01 | 信令消息 | 一条或多条信令消息或信令消息的片段 |
| 0x02 | 字幕数据 | 字幕数据片段或一个或多个字幕数据 |
| 0x03 | 为其它数据类型保留 |  |
| 0x04 ~ 0x3F | 私有用途保留 |  |

1. 3.2 字幕数据模式下SMTP负载头

字幕数据模式下SMTP负载头部结构见图A.6。

1. 
2. 图A.6 字幕数据模式下SMTP负载头部结构

Fragmentation\_Indicator（f\_i：2 bits）：指示SMTP负载中字幕数据的分片信息。有效值见表35。

1. 表A.3 片段指示值

|  |  |
| --- | --- |
| **值** | **描述** |
| 00 | 负载包含一或多个完整字幕数据。 |
| 01 | 负载包含字幕数据的第一个片段。 |
| 10 | 负载包含非第一个，也非最后一个部分的字幕数据片段。 |
| 11 | 负载包含字幕数据的最后一个片段。 |

aggregation\_flag（A：1bit）：该字段置‘1’表示负载中包含2个或以上字幕数据。

fragmentation\_counter（frag\_count：8bits）：该字段表示此字幕数据片段后包含同一字幕数据片段的SMTP负载个数。当aggregation\_flag字段置‘1’时该字段置‘0’。

RES（6bits）：该字段所有位置‘0’，保留以后使用。

CC\_data\_length（16bits）：该字段表示此字段后字幕数据长度。当aggregation\_flag置‘1’时，该字段出现次数与整合进负载中的字幕数据个数相同，并出现在每个字幕数据前面。

CC\_data\_playload：字幕数据负载。

1. 3.3 字幕业务描述符

字幕业务元数据封装在MP表中的媒体资源描述符中，以PA消息或MPT消息的形式进行传输。

1. 表A.4 CC\_service\_descriptor语法

|  |  |  |  |
| --- | --- | --- | --- |
| **语法** | **值** | **比特数** | **备注** |
| CC\_service\_descriptor() {descriptor\_tagdescriptor\_lengthnum\_of\_servicefor(i=0;i<N;i++){ CC\_service\_number}} | N | 161688 | uimsbfuimsbfuimsbfuimsbf |

descriptor\_tag：标识符，用于标志descriptor的类型。

descriptor\_length：标识符的长度，单位为字节。

num\_of\_service：标识字幕业务的数目。

CC\_service\_number：字幕业务的编号。

1. （资料性附录）
可关闭字幕schema

本附录是对可关闭字幕中元素的描述性说明，该描述性说明按照标准schema语法结构来组织。

该schema由以下URI来描述：cn:org:avs:schema:CC。

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

<schema xmlns="http://www.w3.org/2001/XMLSchema"

xmlns:ClosedCaption="cn:org:avs:schema:CC"

xmlns:ns1="cn:org:avs:schema:CC"

targetNamespace ="cn:org:avs:schema:CC"

elementFormDefault ="qualified" attributeFormDefault="unqualified">

 <element name="CCSeqHeader" type="CC:CCSeqHeaderType"/>

 <complexType name="CCSeqHeaderType">

 <element name="Language" type="CC:LanguageType"/>

 <complexType name="LanguageType">

 <attribute name="value\_of\_language" type="string"/>

 </complexType>

 </element>

 <element name="Timeinfo" type="CC:TimeinfoType"/>

 <complexType name="TimeinfoType"

 <attribute name="ctime" type="string"/>

 <attribute name="duration" type="integer"/>

 </complexType>

 </element>

 <attribute name="layer" type="integer"/>

 <element name="Position" type="CC:PositionType"/>

 <complexType name="PositionType"

 <attribute name="ctime" type="string"/>

 <attribute name="tx" type="integer"/>

 <attribute name="ty" type="integer"/>

 <attribute name="height" type="integer"/>

 <attribute name="width" type="integer"/>

 </complexType>

</element>

</complexType>

 <element name="SampleDescriptions" type="CC:SampleDescriptions"/>

 <complexType name="SampleDescriptionsType">

 <element name="SD" type="CC:SDType"/>

 <complexType name="SDType">

 <attribute name="id" type="integer"/>

 <element name="DispFlags" type="CC:DispFlagsType"/>

 <complexType name="DispFlagsType"

 <attribute name="write\_text\_vertically" type="integer"/>

 <attribute name="horizontal\_justification" type="integer"/>

 <attribute name="vertical\_justification" type="integer"/>

 </complexType>

 </element>

 <element name="FontTable" type="CC:FontTableType"/>

 <complexType name="FontTableType"

 <element name="FontName" type="CC:FontNameType"/>

 <complexType name="FontNameType"

 <attribute name="id" type="integer"/>

 <attribute name="font" type="string"/>

 </complexType>

 </element>

 </complexType>

 </element>

 <element name="StyleRecord" type="CC:StyleRecordType"/>

 <complexType name="StyleRecordType"

 <element name="FontInfo" type="CC:FontInfoType "/>

 <complexType name="FontInfoType"

 <attribute name="fontId" type="integer"/>

 <attribute name="fontSize" type="integer"/>

 </complexType>

 </element>

 <element name="FaceStyle" type="CC:FaceStyleType"/>

 <complexType name="FaceStyleType"

 <attribute name="bold" type="integer"/>

 <attribute name="italic" type="integer"/>

 <attribute name="underline" type="integer"/> </complexType>

 </element>

 <element name="Text\_color\_rgba" type="CC:Text\_color\_rgba

Type"/>

 <complexType name=" Text\_color\_rgbaType"

 <attribute name="red" type="integer"/>

 <attribute name="green" type="integer"/>

 <attribute name="blue" type="integer"/>

 <attribute name="transparency" type="integer"/>

 </complexType>

 </element>

 </complexType>

 </element>

 <element name="BoxRecord" type="CC:BoxRecordType"/>

 <complexType name="BoxRecordType"

 <attribute name="top" type="integer"/>

 <attribute name="left" type="integer"/>

 <attribute name="bottom" type="integer"/>

 <attribute name="right" type="integer"/>

 </complexType>

 </element>

 </complexType>

 </element>

</complexType>

</element>

<element name="CCSample" type="CC:CCSample"/>

 <complexType name="CCSampleType">

 <element name="Text" type="CC:Text"/>

 <complexType name="TextType">

 <attribute name="text" type="string"/>

 </complexType>

 </element>

 <element name="Dur" type="CC:Dur"/>

 <complexType name="DurType">

 <attribute name="duration" type="integer"/>

 </complexType>

 </element>

 <element name="SD\_id" type="CC:SD\_id"/>

 <complexType name="SD\_id">

 <attribute name="SDid" type="integer"/>

 </complexType>

 </element>

</complexType>

</element>

</schema>