

中国电信集团公司技术标准

Q/CT 2370-2011

IPTV 业务系统 EPG 页面制作及显示技术要求 V3.0

Technical specification of EPG producting and display

2011-09-07 发布

2011-09-07 实施

中国电信集团公司 发布

目 次

目 次	I
前 言	VI
EPG 页面制作及显示技术要求.....	1
1 范围	1
2 规范性引用文件.....	1
3 缩略语、术语和定义.....	1
3.1 缩略语.....	1
3.2 术语和定义.....	2
4 概述	3
5 总体要求	3
5.1 EPG 相关.....	3
5.2 浏览器相关.....	4
6 EPG 制作要求	5
6.1 分辨率.....	5
6.2 语言文字编码.....	5
6.3 字体字号	5
6.4 图片格式.....	6
6.5 Cookie 的设置.....	6
6.6 浏览器脚本的使用.....	6
6.6.1 注意事项.....	6
6.6.2 Ajax.....	7
6.6.3 Browser 对象的支持	9
6.6.4 DOM 对象.....	12
Event.....	16
Handlers	16
6.6.5 JavaScript 对象.....	33
6.6.6 扩展 JavaScript 对象.....	41
6.7 HTML 标签的使用.....	67
6.7.1 基本属性.....	67
6.7.2 基本事件.....	67
6.7.3 HTML 4 标签列表	68
6.7.4 HTML 5 标签列表.....	83
6.8 CSS 的使用.....	84
6.8.1 尺寸单位.....	84
6.8.2 颜色单位.....	84
6.8.3 CSS 2 属性.....	85

6.8.4	CSS 3 属性.....	92
6.9	注意事项.....	97
7	浏览器要求.....	97
7.1	分辨率.....	97
7.2	语言文字编码.....	99
7.3	字体字号.....	99
7.4	图片格式.....	100
7.5	Cookie 的支持.....	100
7.6	浏览器脚本的支持.....	101
7.7	HTML 标签的支持.....	101
7.8	CSS 及 CSS 变换的支持.....	101
7.8.1	盒子模型的尺寸.....	102
7.8.2	盒子模型示例.....	103
7.8.3	外边距折叠.....	104
7.9	浏览器焦点处理原则.....	105
7.9.1	焦点显示规则.....	105
7.9.2	焦点移动规则.....	105
7.10	“多窗口”功能.....	107
附录 A	(规范性附录) 浏览器默认 CSS.....	109
附录 B	(资料性附录) 规范版本修订记录.....	110

表

表 1	XMLHttpRequest 对象.....	8
表 2	Window 对象说明.....	9
表 3	Navigator 对象说明.....	10
表 4	History 对象说明.....	11
表 5	Location 对象说明.....	11
表 6	Screen 对象说明.....	12
表 7	Anchor 对象说明.....	12
表 8	Area 对象说明.....	13
表 9	Base 对象说明.....	14
表 10	Body 对象说明.....	14
表 11	Canvas 对象说明.....	14
表 12	Document 对象说明.....	15
表 13	Event 对象说明.....	16
表 14	Form 对象说明.....	17
表 15	Frame 对象说明.....	17
表 16	Frameset 对象说明.....	18
表 17	IFrame 对象说明.....	18

表 18	Image 对象说明	19
表 19	Button 对象说明	20
表 20	Checkbox 对象说明	20
表 21	Hidden 对象说明	21
表 22	Input Password 对象说明	22
表 23	Radio 对象说明	23
表 24	Reset 对象说明	24
表 25	Submit 对象说明	24
表 26	Text 对象说明	25
表 27	Link 对象说明	26
表 28	Meta 对象说明	26
表 29	Object 对象说明	27
表 30	Option 对象说明	27
表 31	Select 对象说明	28
表 32	Style 对象说明	29
表 33	display 属性说明	31
表 34	Textarea 对象说明	32
表 35	Array 对象说明	33
表 36	Boolean 说明	34
表 37	Date 对象说明	34
表 38	Math 对象说明	36
表 39	Number 对象说明	37
表 40	Object 对象说明	38
表 41	RegExp 说明	38
表 42	String 说明	39
表 43	Kernel Property and Method	40
表 44	Authentication 对象说明	41
表 45	CTCSetConfig/ CTCGetConfig 方法支持参数列表	42
表 46	MediaPlayer 属性列表	43
表 47	MediaPlayer 方法列表	46
表 48	媒体对象相关参数列表	51
表 49	Navigation 对象说明	56
表 50	Utility 对象说明	57
表 51	EVENT 对象的结构	57
表 52	对象的类型	58
表 53	EVENT_GO_CHANNEL 事件	58
表 54	EVENT_MEDIA_END 事件	58
表 55	EVENT_MEDIA_BEGINING 事件	59
表 56	EVENT_MEDIA_ERROR 事件	59
表 57	EVENT_PLAYMODE_CHANGE 事件	59
表 58	PLAY_MODE 播放状态	59
表 59	PLAY_RATE 播放速率	60
表 60	EVENT_REMINDER 事件	60
表 61	EVENT_JVM_CLIENT 事件	60

表 62	event_code 事件类型.....	61
表 63	event_result 事件操作结果.....	61
表 64	EVENT 详细信息示例.....	61
表 65	HTML 标准属性.....	67
表 66	窗口事件.....	67
表 67	表单元素事件.....	68
表 68	键盘事件.....	68
表 69	<A>标签说明.....	69
表 70	<AREA>标签说明.....	70
表 71	<base>标签说明.....	71
表 72	<BODY>标签说明.....	71
表 73	<FORM>标签说明.....	72
表 74	<FRAME>标签说明.....	73
表 75	<FRAMESET>标签说明.....	74
表 76	<IFRAME>标签说明.....	74
表 77	标签说明.....	75
表 78	<INPUT>标签说明.....	75
表 79	<LINK> 标签说明.....	76
表 80	<MARQUEE>标签说明.....	77
表 81	<META>标签说明.....	78
表 82	<OBJECT>标签说明.....	79
表 83	<OPTGROUP>标签说明.....	79
表 84	<OPTION>标签说明.....	80
表 85	<PARAM>标签说明.....	80
表 86	<SCRIPT>标签说明.....	80
表 87	<SELECT>标签说明.....	81
表 88	<STYLE>标签说明.....	82
表 89	<TD>标签说明.....	82
表 90	<TEXTAREA>标签说明.....	83
表 91	<CANVAS>标签说明.....	83
表 92	<KEYGEN>标签说明.....	84
表 93	CSS 颜色单位说明.....	85
表 94	Background 元素说明.....	85
表 95	Border 元素说明.....	86
表 96	Dimension 元素说明.....	87
表 97	Font 元素说明.....	87
表 98	List 元素说明.....	88
表 99	Margin 元素说明.....	89
表 100	Padding 元素说明.....	89
表 101	Position 元素说明.....	90
表 102	Pseudo-Class 元素说明.....	91
表 103	Text 元素说明.....	91
表 104	Background 元素说明.....	92
表 105	Border 元素说明.....	92

表 106	Text 元素说明.....	93
表 107	支持变换的 CSS 属性列表.....	93
表 108	CSS 变换的属性.....	94
表 109	transition-timing-function 属性值.....	95
表 110	HTML 字体对照表.....	99
表 111	ASCII 字符显示对照表.....	99

前 言

本标准是中国电信 IPTV 系列技术标准之一，截止本标准发布之日，该系列规范的名称如下：

- (1) 《IPTV 业务系统 业务技术体制 (V3.0)》
- (2) 《IPTV 业务系统 内容提供商与内容合成管理平台接口技术要求 (V3.0)》
- (3) 《IPTV 业务系统 内容合成管理平台技术要求 (V3.0)》
- (4) 《IPTV 业务系统 内容合成管理平台与业务运营平台接口技术要求 (V3.0)》
- (5) 《IPTV 业务系统 终端管理平台技术要求 (V3.0)》
- (6) 《IPTV 业务系统 业务管理平台技术要求 (V3.0)》
- (7) 《IPTV 业务系统 EPG 页面制作及显示技术要求 (V3.0)》
- (8) 《IPTV 业务系统 业务能力平台接口技术要求 (V3.0)》
- (9) 《IPTV 业务系统 业务管理平台与业务能力平台接口技术要求 (V3.0)》
- (10) 《IPTV 业务系统 终端与终端管理平台接口技术要求 (V3.0)》
- (11) 《IPTV 业务系统 终端与业务运营平台接口技术要求 (V3.0)》
- (12) 《IPTV 业务系统 标清机顶盒设备技术要求 (V3.0)》
- (13) 《IPTV 业务系统 高清机顶盒设备技术要求 (V3.0)》

本标准作为中国电信 IPTV 3.0 标准化系列规范中的新增规范，主要是依据 W3C 关于 HTML、CSS、JavaScript 等规范的相关内容，结合中国电信在 IPTV 业务运营过程中与内容提供商、服务商提供商、终端提供商在 EPG 页面制作及 IPTV 终端浏览器显示等方面合作所积累的经验，结合最新的 Web 技术发展，以基于 Web 展现的 IPTV 页面的制作及显示的规范化为目标编写完成。随着中国电信 IPTV 技术标准体系的发展完善，还将制定后续的相关标准。

本标准由中国电信集团公司提出并归口。

本标准由中国电信集团公司提出并归口。

本标准起草单位：中国电信股份有限公司上海研究院

本标准主要起草人：沈昕、贾立鼎、包盛、祝谷乔

本标准于 2011 首次发布。

EPG 页面制作及显示技术要求

1 范围

本标准通过对 IPTV EPG 页面的制作以及在终端浏览器上的显示效果进行标准化，以达到在不同的类型的 IPTV 终端浏览器上 EPG 页面显示效果一致的目的。在本标准中主要规定了中国电信 IPTV 业务发展过程中 EPG 页面制作时使用的包括图片、文字、分辨率、HTML 标签、CSS 以及脚本语言等所需要遵循的标准，以及 IPTV 终端浏览器在对于采用以上标准制作的 EPG 页面的支持以及显示。

本标准适用于中国电信 IPTV 业务中所有基本业务及扩展业务 EPG 页面的制作以及 IPTV 终端浏览器在对于相关页面标准的支持及显示。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，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文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

RFC2616	Hypertext Transfer Protocol
RFC3275	Extensible Markup Language
RFC 2109	HTTP State Management Mechanism
ECMA-262	ECMAScript Language Specification
W3C	Web Services Description Language

3 缩略语、术语和定义

3.1 缩略语

下列缩略语适用于本标准：

CSS	Cascading Style Sheet	层叠样式表
DOM	Document Object Model	文档对象模型
EPG	Electronic Program Guide	电子节目单

HTML	Hypertext Markup Language	超文本标记语言
HTTP	HyperText Transfer Protocol	超文本传输协议
SOAP	Simple Object Access Protocol	简单对象访问协议
URL	Uniform Resource Locator	统一资源定位
XML	eXtensible Markup Language	可扩展标记语言

3.2 术语和定义

电子节目菜单(Electronic Program Guide) [YD/T 1823-2008]: 将所有数字电视节目按不同的分类规则组合在一起, 用户通过遥控器就可以进行查看, 也可以从电子节目单 (EPG) 中直接切换到正在播放的节目中去。

IPTV终端(IPTV Terminal)[YD/T 2016.3-2009]: 同时支持IPTV网络层和IPTV业务层接入的终端设备。IPTV终端通过与应用功能交互, 获取EPG、内容版权许可证、密钥等业务信息; 与内容分发平台交互, 接受IPTV业务, 完成解密和解码功能; 并负责接收终端用户的控制指令。IPTV终端或者直接与电信网链接, 或者通过家庭网关与电信网链接。

EPG服务系统(EPG Service System) [YD/T 1920-2009]: EPG服务系统分为EPG系统和应用服务系统。EPG系统为用户提供业务服务的入口界面, 通过此系统, 用户可以完成节目的浏览、节目点播和收看并可以完成一些与用户相关的自服务项目 (如查询账户余额、修改服务类型等)。应用服务系统负责区域内用户认证、机顶盒的非法性检查并负责为区域内用户提供上层服务器的服务接口 (如查询节目价格、生成使用话单等)。

4 概述

在现有《IPTV 业务系统 业务技术体制》中定义了 IPTV 业务运营平台通过以 Web 技术制作的 EPG 页面向用户提供业务导航，并由 IPTV 终端通过内嵌浏览器完成对 EPG 页面显示及相关操作的流程。

在 IPTV 业务运营平台中使用的 EPG 页面是基于 HTML、CSS、JavaScript、DOM 对象及中国电信定义的扩展 JavaScript 等技术实现的，对于 IPTV EPG 页面制作及终端浏览器显示应支持相关技术总体要求如下：

- HTML：应至少支持 W3C 定义的 HTML 4.01 规范、部分微软扩展 HTML 标签以及 HTML 5 的部分标签；
- CSS：应至少支持 CSS 2.1 及 CSS 3 的部分属性；
- JavaScript（也称为 ECMAScript）：至少支持 JavaScript 1.1 以上版本以及 DOM Level1 及 DOM Level 2；

5 总体要求

5.1 EPG 相关

由于 IPTV 是基于电视完成 EPG 页面向用户的展现，因此在 EPG 制作上也应与传统 Internet 上的网页有所区别。对于 IPTV EPG 的制作应遵循以下规则：

● 页面设计

根据观看电视的习惯，EPG 页面应尽量在屏幕上完整的显示，避免横向滚动。页面的主体和背景的对比度高，整体视觉效果好。从业务首页到最后一级页面不要超过四级，当单层结构无法表达页面焦点逻辑时也可以有分层的结构，但要求所有的界面的结构要统一，且应该一致地在类似的业务上使用相同的操作方式，以保证在业务展现及业务体验上的一致性，便于用户从局部来推断出全局的操作方式。

页面设计应该允许使用者因过失操作错误而具有恢复的功能，且对各种出错或异常状态给予用户一个友好的提示和帮助，并提示用户出错或异常的原因。

能够在屏幕上清晰的显示文字，字间距和行间距适中。对于标清及高清 IPTV 业务来说，使用的字体和字号应至少能在 2 米左右距离的情况下，在 4:3（标清）及 16:9（高清）比率的电视上清晰显示。

EPG 页面应使用【附录 A】中给出的 CSS 作为页面默认页面显示效果。而当需要控制浏览器进行页面跳转时，指向页面的 URL 地址长度最长不超过 2048 字节。

● 用户操作

应在每个页面中设置默认焦点，以使用户能够在页面加载完毕后立即进行业务操作。页面设计中应尽量避免用户进行遥控器输入，而采用选择的方式完成操作。页面中如果需要用到特殊的遥控器按键，必须在该页面中进行操作说明的提示，且给出帮助文件并能让用户很容易的找到，务必避免选单太杂，或是使用难懂的设计隐喻或是太多的隐喻；

在 EPG 页面中对于用户的操作应具有回馈性，包括视觉回馈、声音回馈等方面。

- 视觉回馈：一般常见的手法则是瞬间改变色彩变化，使得视觉产生落差。比如，采用 JavaScript 或 CSS 变换控制页面元素按下后的图片切换、反色等操作，但也应避免误用过多颜色产生而炫乱或扰乱使用者操作的显示；
- 声音回馈：能够在遥控器按下按键时，通过声音进行提示。比如，采用 JavaScript 控制在遥控器按下上、下、左、右、确定等按键后播放提示音等。

5.2 浏览器相关

IPTV 终端浏览器在显示 EPG 页面时也应与传统 PC 浏览器有所区别，需注意以下事项：

● 页面显示

IPTV 终端浏览器在 EPG 显示时的默认底色为黑色，即当页面被设置为透明时的显示效果。当页面超出一屏的情况下不会出现上下或左右的滚动条。

若 EPG 页面未采用 CSS 或其他方法对焦点框进行设置，则 IPTV 终端浏览器应使用【附录 A】中给出的 CSS 作为页面默认页面显示效果。IPTV 终端浏览器焦点框应采用醒目颜色，可以使用户可以在 3M 外观察到焦点框的去向，在 EPG 页面未设置焦点框显示效果的情况下，IPTV 终端浏览器默认的焦点框显示效果为：

- 颜色：RGB - #FF8F1F
- 粗细：3 像素
- 下拉列表元素选中底色：无需设置，保持原有颜色
- 下拉列表元素选中文字：无需设置，保持原有颜色

● 用户操作

当 EPG 页面未设置默认焦点时，IPTV 终端浏览器会将默认焦点放置在屏幕左上角第一个能够被聚焦的元素上。当页面链接不可用时，IPTV 终端浏览器会在 10 秒后通过友好

界面提示用户，即 HTTP 连接的超时时间被定义为 10 秒。

当页面加载或用户操作导致 IPTV 终端浏览器需要执行 Javascript 脚本，且该脚本存在错误的情况下，为保证 IPTV 终端浏览器显示的一致性，终端浏览器应终止脚本的继续执行。

6 EPG 制作要求

6.1 分辨率

EPG 页面在制作过程中，要求在页面<body>元素中至少定义宽度与高度（如标清业务页面定义为 640×530，高清业务页面定义为 1280×720），且在其“style”属性中必须添加“background-Repeat:no-repeat”值的设置，以标清业务为例，示例代码如下（以标清为例）：

```
<body style="background-Repeat:no-repeat" width="640" height="530">
```

因为单像素宽的垂直线和点在旧式频率回应糟糕的电视机中可能会显示为逐渐消失或根本不会显示，并且单像素宽水平线只能在两个隔行区中的一行中画出，会导致在隔行显示时的画面闪烁，因此在 EPG 制作时为防止单像素宽成为线或点，线条的宽度至少为 2 个像素。

6.2 语言文字编码

在 EPG 页面制作过程中页面的字符集可以使用以下语言文字：

- 英文字符（ISO-8859-1）
- 中文 GBK 简体汉字

编码要求支持：

- GBK
- UTF-8

【注】如果字符集为中文集，则需要在<html>标签中指定编码格式，外部 css 及链接文件采用同样编码格式。

6.3 字体字号

在 EPG 页面制作时推荐使用 px 单位，因为 px 以像素为单位时，表达最准确，并且能够保证不同 IPTV 终端浏览器在同一显示设备上显示差异最小。

6.4 图片格式

在 IPTV EPG 制作中，可以使用 BMP、GIF（包括动态 GIF）、JPG、PNG 、APNG 格式的图片以增强 EPG 的展现效果。各种图片的分辨率可包括 720×576、352×288、640×480、1280×720、1920×1080 等（其中 1280×720、1920×1080 等分辨率仅在高清业务所使用）。

同时，在 EPG 制作过程中由于考虑 IPTV 终端浏览器资源有限，因此在页面设计时应注意以下事项：

- 一个页面设置的图片总大小不要超过 1M，且应尽量减少同一页面中动态 GIF 格式的文件个数，过多的动态 GIF 图片会影响浏览器性能。为提高页面显示性能，建议同一页面动态 GIF 文件个数不要超过 3 个；
- 在放置图片时，图片实际大小尽量跟页面设置的图片排版区域大小保持一致，否则会因图片缩放导致终端浏览器显示性能的降低；
- 设置标签的属性时，不应将高度、宽度设置为 0，因为这样做不仅没有任何意义，而且还会浪费浏览器的资源；

6.5 Cookie 的设置

由于终端浏览器资源有限，因此在 IPTV EPG 页面制作过程中应注意设置 Cookie 的大小以及总量。通常来说，在普通的标清终端中浏览器支持的 Cookie 总数不超过 30 个，因此建议在同一个域下尽量避免使用超过 5 个 Cookie，且单个 cookie 或一次设置的多个 cookie 的最大尺寸为 4K（4096 字节），否则 IPTV 终端浏览器将自动删除最早的 cookie。

6.6 浏览器脚本的使用

6.6.1 注意事项

在本章节中列出了在 IPTV EPG 制作过程中允许使用的浏览器脚本，使用所有超出此集合范围内的浏览器脚本可能会由于 IPTV 终端浏览器不支持导致无法得到预期的展现效果。同时，在使用本章节中列出的浏览器脚本进行 EPG 制作过程中，仍需注意以下问题：

- 由于终端浏览器资源有限，应注意页面中对象等所占用的内存大小，同一个页面中不建议使用过多变量或递归函数的多次调用；
- 对于键值捕捉的处理，给出示例代码（【注】本代码只做参考）：

```
// 响应 onkeypress 事件
window.document.onkeypress = function( keyEvent )
{
    keyEvent = keyEvent ? keyEvent : window.event;
    var keyvalue = keyEvent.which ? keyEvent.which : keyEvent.keyCode;

    // return true;
}
```

【注】：对于四色键的处理 IPTV 终端会先将键值上报至 EPG 页面，并由页面在键值处理函数中进行处理。若页面未有响应函数或有响应函数但响应返回为 **false/null** 时则机顶盒使用现有逻辑，直接跳转至功能键对应地址；若页面有响应函数且响应返回为 **true**，则此键值响应结束，机顶盒不做任何响应。

- 不应把 **event** 这样一个保留字做为函数名，以避免和 **event** 属性冲突。比如这样的写法不合法（**【注】**本代码只做参考）：

```
// 错误的用法
function event()
{ ..... }
```

- 双引号不能嵌套，单引号也不要嵌套，但是交叉嵌套可以。同时，在引号内的内容不能够换行，比如（**【注】**本代码只做参考）：

```
// 错误的用法
var errorExpression1 = "1"2"3";           //双引号不能嵌套
var errorExpression2 = '1'2'3';           //单引号不能嵌套
var errorExpression3 = "aaaa
bbbbbb";                                   //引号内的内容不能够换行

// 正确的用法
var rightExpression1 = '1"2"3';
var rightExpression2 = "1'2'3";
```

6.6.2 Ajax

AJAX 即“Asynchronous JavaScript and XML”（异步 JavaScript 和 XML），主要用于创建更好更快以及交互性更强的 Web 应用程序的技术。页面可以使用 JavaScript 的 XMLHttpRequest 对象来直接与服务器进行通信，在不重载页面的情况与 Web 服务器交换数据，有效减少页面展现时间。

表 1 XMLHttpRequest 对象

类别	方法/属性	说明
Property	responseText	获得字符串形式的响应数据。
	responseXML	获得 XML 形式的响应数据。
	readyState	存有 XMLHttpRequest 的状态。从 0 到 4 发生变化。 <ul style="list-style-type: none"> • 0: 请求未初始化 • 1: 服务器连接已建立 • 2: 请求已接收 • 3: 请求处理中 • 4: 请求已完成, 且响应已就绪
	status	200: "OK" 404: 未找到页面
Method	constructor	构造函数
	open(method,url,async)	规定请求的类型、URL 以及是否异步处理请求。 <ul style="list-style-type: none"> • <i>method</i>: 请求的类型; GET 或 POST • <i>url</i>: 文件在服务器上的位置 • <i>async</i>: true (异步) 或 false (同步)
	send(string)	将请求发送到服务器。 <ul style="list-style-type: none"> • <i>string</i>: 仅用于 POST 请求
	【备注】: 与 POST 相比, GET 更简单也更快, 并且在大部分情况下都能用。 然而, 在以下情况中, 请使用 POST 请求: <ul style="list-style-type: none"> • 无法使用缓存文件 (更新服务器上的文件或数据库) • 向服务器发送大量数据 (POST 没有数据量限制) 发送包含未知字符的用户输入时, POST 比 GET 更稳定也更可靠	
	setRequestHeader(header, value)	向请求添加 HTTP 头。 <ul style="list-style-type: none"> • <i>header</i>: 规定头的名称 • <i>value</i>: 规定头的值
	onreadystatechange	存储函数 (或函数名), 每当 readyState 属性改变时, 就会调用该函数。

6.6.3 Browser 对象的支持

6.6.3.1 Window

【说明】

Window 对象是 JavaScript 层级中的顶层对象，代表一个浏览器窗口或一个框架。Window 对象会在 `<body>` 或 `<frameset>` 每次出现时被自动创建。在终端浏览器 JavaScript 中，Window 对象是全局对象，所有的表达式都在当前的环境中计算。也就是说，要引用当前窗口根本不需要特殊的语法，可以把那个窗口的属性作为全局变量来使用。例如，可以只写 `document`，而不必写 `window.document`。同样，可以把当前窗口对象的方法当作函数来使用，如只写 `alert()`，而不必写 `Window.alert()`。

除了上面列出的属性和方法，Window 对象还实现了核心 JavaScript 所定义的所有全局属性和方法。Window 对象的 `window` 属性和 `self` 属性引用的都是它自己。当想明确地引用当前窗口，而不仅仅是隐式地引用它时，可以使用这两个属性。除了这两个属性之外，`top` 属性以及 `frame[]` 数组都引用了与当前 Window 对象相关的其他 Window 对象。

表 2 Window 对象说明

类别	方法/属性	说明
Property	closed	返回一个布尔型的数值，指出参考的窗口是否已经关闭
	document	该对象可以用来访问页面的所有元素
	frames	返回windows对象内的所有frame的名字
	history	包含用户在浏览器窗口访问过的页面的URL
	location	包含当前窗口访问的URL
	Navigator	对 Navigator 对象的只读引用。
	self	返回一个reference至当前窗口
	top	返回最顶层的先辈窗口。
Method	alert()	显示一个包含指定文字信息和确定按钮的警告窗口
	clearInterval()	删除由setInterval()方法创建的一个延时
	clearTimeout()	删除由setTimeout()方法创建的一个延时。浏览器只能够需要调用cleartimeout方法来删除定时器，不能要求其他处理
	close()	关闭子窗口

	focus()	非W3C定义的扩展方法，用于把焦点给予一个窗口，该窗口在最前端显示
	hide()	非W3C定义的扩展方法，用于隐藏当前窗口
	moveBy()	可相对窗口的当前坐标把它移动指定的像素。
	moveTo()	把窗口的左上角移动到一个指定的坐标。
	resizeBy()	按照指定的像素调整窗口的大小。
	resizeTo()	把窗口的大小调整到指定的宽度和高度。
	setInterval()	告诉函数每隔一个指定的时间间隔执行表达式
	setTimeout()	告诉一个函数在一个指定的毫秒级时间间隔后执行表达式
	show()	非W3C定义的扩展方法，用于显示当前窗口
Collection	frames[]	返回窗口中所有命名的框架。该集合是 Window 对象的数组，每个 Window 对象在窗口中含有一个框架或 <iframe>。属性 frames.length 存放数组 frames[] 中含有的元素个数。注意，frames[] 数组中引用的框架可能还包括框架，它们自己也具有 frames[] 数组。

6.6.3.2 Navigator

【说明】:

Navigator 对象包含有关浏览器的信息。

【注释】:

没有应用于 Navigator 对象的公开标准，不过所有终端浏览器都支持该对象。

表 3 Navigator 对象说明

类别	方法/属性	说明
Property	appName	返回浏览器的类型
	appVersion	返回浏览器的平台和版本信息。
	cpuClass	返回浏览器硬件系统的 CPU 等级。
	cookieEnabled	返回指明浏览器中是否启用 cookie 的布尔值。
	userAgent	返回由客户机发送服务器的 user-agent 头部的值。对于各终端浏览器应根据以下命名方式填写 UserAgent 字段：“<BrowserCoreName> + ‘ ’ + <BrowserCoreVersion> + ‘ ’ +

		<BrowserVandorName> + ' ' + <Version>”。 例如：Mozilla 4 iPanel 2.0.1, Opera 5 Opera 9.0.1.1
--	--	--

6.6.3.3 History

【说明】

History 对象是 window 对象的一部分，可通过 window.history 属性对其进行访问。History 对象实际上是 JavaScript 对象，而不是 HTML DOM 对象，它由一系列的 URL 组成，是用户在一个浏览器窗口内已访问的 URL。

表 4 History 对象说明

类别	方法/属性	说明
Property	length	返回历史清单上的元素个数
Method	back()	装载之前的URL
	go()	装载特定的URL

6.6.3.4 Location

【说明】

Location 对象是包含有关当前 URL 的信息，它是 Window 对象的一个部分，可通过 window.location 属性来访问。

表 5 Location 对象说明

类别	方法/属性	说明
Property	host	设置或返回主机名和端口号，只读
	hostname	设置或返回主机名，只读
	pathname	设置或返回由location对象指定的文件名称或路径，只读
	port	设置或返回关联URL的端口号，只读
	protocol	设置或返回URL的协议部分，只读
	href	设置或返回完整的URL
Method	reload()	重新装载当前文档
	replace()	用某一指定文档取代当前文档

6.6.3.5 Screen

【说明】

Screen 对象包含有关客户端显示屏幕的信息。每个 Window 对象的 screen 属性都引用一个 Screen 对象。Screen 对象中存放着有关显示终端屏幕的信息。JavaScript 程序将利用这些信息来优化它们的输出，以达到用户的显示要求。另外，JavaScript 程序还能根据有关屏幕尺寸的信息将新的浏览器窗口定位在屏幕中间。

【注释】:

没有应用于 Screen 对象的公开标准，不过所有终端浏览器都支持该对象。

表 6 Screen 对象说明

类别	方法/属性	说明
Property	availHeight	返回实际输出的屏幕的高度
	availWidth	返回实际输出可用的屏幕的宽度
	height	返回屏幕的高度。
	width	返回显示器屏幕的宽度。
	colorDepth	（如有，则）返回目标设备或缓冲器上的调色板的比特深度（，否则返回为-1）。
	pixelDepth	返回显示屏幕的颜色分辨率（比特每像素）。

6.6.4 DOM 对象

6.6.4.1 Anchor

【说明】

Anchor 对象代表一个 HTML 超链接。在 HTML 文档中 <a> 标签每出现一次，Anchor 对象就会被创建。

Anchor 对象可用于创建到另一个文档的链接（通过 href 属性），或者创建文档内的书签。可以通过搜索 Document 对象中的 anchors[] 数组来访问某个锚，或者使用 document.getElementById() 。

表 7 Anchor 对象说明

类别	方法/属性	说明
----	-------	----

Property	accessKey	设置或返回访问一个链接的快捷键，允许遥控器通过指定的快捷键访问或操作该元素
	className	设置或返回元素的 class 属性。
	coords	设置或返回逗号分隔列表，包含了图像映射中链接的坐标。
	id	设置或返回一个链接的 id 。
	innerHTML	设置或返回一个链接的内容。
	href	设置或获得锚点上的链接到指定页面的url
	name	设置或返回一个链接的名称。
	shape	设置或返回图像映射中某个链接的形状。
Method	blur()	把焦点从链接上移开。
	focus()	使对象得到焦点
Event Handler	onkeypress	当键盘被按下后又松开时的对象事件句柄
	onblur	当元素失去焦点时对象事件句柄
	onfocus	当元素获得焦点时对象事件句柄

6.6.4.2 Area

【说明】

Area 对象代表图像映射的一个区域（图像映射指的是带有可点击区域的图像）。在 HTML 文档中 **<area>** 标签每出现一次，就会创建一个 **Area** 对象。

表 8 Area 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问某个区域的快捷键。
	className	设置或返回元素的 class 属性。
	coords	设置或返回图像映射中可点击区域的坐标。
	href	设置area对象的url
	id	设置或获得area对象的标识符字符串
	shape	设置或返回图像映射中某个区域的形状。
Event Handler	onkeypress	当键盘被按下后又松开时的对象事件句柄
	onblur	当元素失去焦点时对象事件句柄
	onfocus	当元素获得焦点时对象事件句柄

6.6.4.3 Base

【说明】

Base 对象代表 HTML 的 base 元素。在 HTML 文档中 <base> 每出现一次，Base 对象就会被创建。

表 9 Base 对象说明

类别	方法/属性	说明
Property	href	设置或返回针对页面中所有链接的基准 URL。
	id	设置或返回 <base> 元素的 id。

6.6.4.4 Body

【说明】

Body 对象代表文档的主体 (HTML body) 。

表 10 Body 对象说明

类别	方法/属性	说明
Property	className	设置或返回元素的 class 属性。
	id	设置或获得area对象的标识符字符串
Event Handler	onload	当文档载入时执行脚本
	onunload	当文档卸载时执行脚本

6.6.4.5 Canvas

【说明】

Canvas 对象表示一个 HTML 画布元素 - <canvas>。它没有自己的行为，但是定义了一个 API 支持脚本化客户端绘图操作。

表 11 Canvas 对象说明

类别	方法/属性	说明
Property	height	画布的高度。和一幅图像一样，这个属性可以指定为一个整数像素值或者是窗口高度的

		百分比。当这个值改变的时候，在该画布上已经完成的任何绘图都会擦除掉。默认值是300。
	width	画布的宽度。和一幅图像一样，这个属性可以指定为一个整数像素值或者是窗口宽度的百分比。当这个值改变的时候，在该画布上已经完成的任何绘图都会擦除掉。默认值是300。
Method	getContext(contextID)	返回一个用于在画布上绘图的环境。参数 contextID 指定了画布上绘制的类型。当前唯一的合法值是 "2d"，它指定了二维绘图。返回值为一个 CanvasRenderingContext2D 对象，使用它可以绘制到 Canvas 元素中。
Event Handler	onkeypress	当键盘被按下后又松开时的对象事件句柄
	onblur	当元素失去焦点时对象事件句柄
	onfocus	当元素获得焦点时对象事件句柄

6. 6. 4. 6 Document

【说明】

Document 对象代表整个 HTML 文档，可用来访问页面中的所有元素，它是 Window 对象的一个部分，可通过 window.document 属性来访问。

表 12 Document 对象说明

类别	方法/属性	说明
Collection	anchors[]	返回对文档中所有 Anchor 对象的引用。
	forms[]	返回对文档中所有 Form 对象引用。
	images[]	返回对文档中所有 Image 对象引用。
	links[]	返回对文档中所有 Area 和 Link 对象引用。
Property	cookie	设置或返回所有的与文档关联的cookies
	domain	返回提供该页面的服务器的域名
	URL	返回当前文档的URL
	getElementById()	返回对拥有指定 id 的第一个对象的引用。
	getElementsByName()	返回带有指定名称的对象集合。

6.6.4.7 Event

【说明】

Event 对象代表事件的状态，比如事件在其中发生的元素、按键的状态等。事件通常与函数结合使用，函数不会在事件发生前被执行。

表 13 Event 对象说明

类别	方法/属性	说明
Property	bubbles	返回布尔值，指示事件是否是起泡事件类型。
	cancelable	返回布尔值，指示事件是否可拥可取消的默认动作。
	currentTarget	返回其事件监听器触发该事件的元素。
	target	返回触发此事件的元素（事件的目标节点）。
	timeStamp	返回事件生成的日期和时间。
	type	返回当前 Event 对象表示的事件的名称。
Method	initEvent()	初始化新创建的 Event 对象的属性。
	preventDefault()	通知浏览器不要执行与事件关联的默认动作。
	stopPropagation()	不再派发事件。
Event Handlers	onabort	图像的加载被中断。
	onblur	元素失去焦点。
	onchange	域的内容被改变。
	onclick	当用户点击某个对象时调用的事件句柄。
	onerror	在加载文档或图像时发生错误。
	onfocus	元素获得焦点。
	onload	一张页面或一幅图像完成加载。
	onreset	重置按钮被点击。
	onresize	窗口或框架被重新调整大小。
	onsubmit	确认按钮被点击。
	onunload	用户退出页面。

6.6.4.8 Form

【说明】

Form 对象代表一个 HTML 表单。在 HTML 文档中 <form> 每出现一次，Form 对象就会被创建。

表 14 Form 对象说明

类别	方法/属性	说明
Collection	elements[]	包含表单中所有元素的数组。
Property	action	设置或返回表单数据提交的地址的url
	className	设置或返回元素的 class 属性。
	id	设置或返回form的标识符字符串
	length	设置或返回form中元素的长度
	method	设置或返回form以何种方式提交给服务器（get or post）
	name	设置或返回form的名字
Method	reset()	重新设置form元素返回到默认值，（相当于点击reset按钮）
	submit()	提交表单（相当于点击submit按钮）
Event Handler	onSubmit()	在提交表单之前调用。
	onreset	在重置表单元素之前调用。
	onblur	当元素失去焦点时的对象事件句柄
	onfocus	当元素获得焦点时的对象事件句柄

6.6.4.9 Frame

【说明】

Frame 对象代表一个 HTML 框架。在 HTML 文档中 `<frame>` 每出现一次，就会创建一个 Frame 对象。

表 15 Frame 对象说明

类别	方法/属性	说明
Property	className	设置或返回元素的 class 属性。
	contentDocument	容纳框架的内容的文档。
	frameBorder	设置或返回是否显示框架周围的边框。
	id	设置或返回form的标识符字符串
	marginHeight	设置或返回框架的顶部和底部页空白。
	marginWidth	设置或返回框架的左边缘和右边缘的空白。
	name	设置或返回框架的名称。
	noresize	设置或返回能否缩放框架

	src	设置或返回应被加载到框架中的文档的 URL。
--	-----	------------------------

6.6.4.10 Frameset

【说明】

Frameset 对象代表一个 HTML 框架集。

表 16 Frameset 对象说明

类别	方法/属性	说明
Property	cols	设置或返回框架集中列的数目。
	id	设置或返回form的标识符字符串
	rows	设置或返回框架集中行的数目。
Event Handler	onload	当文档载入时执行的对象事件句柄
	onunload	当文档卸载时执行的对象事件句柄

6.6.4.11 IFrame

【说明】

IFrame 对象代表一个 HTML 的内联框架。在 HTML 文档中 <iframe> 每出现一次，一个 IFrame 对象就会被创建。

表 17 IFrame 对象说明

类别	方法/属性	说明
Property	align	根据周围的文字排列 iframe。
	className	设置或返回元素的 class 属性。
	contentDocument	容纳框架的内容的文档。
	frameBorder	设置或返回是否显示 iframe 周围的边框。
	id	设置或返回图片的标识符
	marginHeight	设置或返回 iframe 的顶部和底部的页空白。
	marginWidth	设置或返回 iframe 的左侧和右侧的页空白。
	name	设置或返回 iframe 的名称。
	scrolling	设置或返回 iframe 是否可拥有滚动条。
	src	设置或返回应载入 iframe 中的文档的 URL。
	height	设置或返回标签内定义的图片高度值
	width	设置或返回 iframe 的宽度。

6. 6. 4. 12 Image

【说明】

Image 对象代表一幅嵌入的图像。 标签每出现一次，一个 Image 对象就会被创建。

表 18 Image 对象说明

类别	方法/属性	说明
Property	align	设置或返回与内联内容的对齐方式。
	complete	返回浏览器是否已完成对图像的加载。
	className	设置或获得元素的Class属性
	border	采用象素值的方式设置或返回图片的边框宽度
	height	设置或返回标签内定义的图片高度值
	hspace	设置或返回标签内定义的图片上下边距的值
	id	设置或返回图片的标识符
	isMap	返回图像是否是服务器端的图像映射。
	name	设置或返回图片的名称
	src	设置或返回图片的路径
	useMap	设置或返回客户端图像映射的 usemap 属性的值。
	vspace	设置或返回标签内定义的图片左右边距的值
	width	设置或返回标签内定义的图片宽度值
Event Handler	onabort	当用户放弃图像的装载时调用的事件句柄。
	onload	当图像装载完毕时调用的事件句柄。
	onerror	在装载图像的过程中发生错误时调用的事件句柄。
	onkeypress	当键盘被按下后又松开时的对象事件句柄
	onblur	当元素失去焦点时对象事件句柄
	onfocus	当元素获得焦点时对象事件句柄

6. 6. 4. 13 Input Button

【说明】

Button 对象代表 HTML 文档中的一个按钮。该元素没有默认的行为，但是必须有一个 onclick 事件句柄以便使用。在 HTML 文档中 <input type="button"> 标签每出现一次，一个 Button 对象 就会被创建。

可以通过遍历表单的 elements[] 数组来访问某个按钮，或者通过使用

document.getElementById()。

表 19 Button 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问按钮的快捷键。
	className	设置或返回元素的 class 属性。
	disabled	设置或返回是否禁用按钮。
	form	返回对包含该按钮的表单对象的引用。
	id	设置或返回按钮的 id。
	name	设置或返回按钮的名称。
	type	返回按钮的表单元素类型。
	value	设置或返回在按钮上显示的文本。
Method	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
	focus()	使对象得到焦点
Event Handler	onBlur()	对象失去焦点时触发事件
	onClick()	当焦点置于控件上时，按下【OK】键，触发此方法
	onFocus()	对象得到焦点时触发事件

6.6.4.14 Input Checkbox

【说明】

Checkbox 对象代表一个 HTML 表单中的一个选择框。在 HTML 文档中 `<input type="checkbox">` 每出现一次，Checkbox 对象就会被创建。

可以通过遍历表单的 `elements[]` 数组来访问某个选择框，或者通过使用 `document.getElementById()`。

表 20 Checkbox 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问 checkbox 的快捷键。
	checked	设置或返回checkbox选项是否被选定
	className	设置或返回元素的 class 属性。
	defaultChecked	设置或返回checkbox选项是否默认情况下已经被选定
	form	返回一个reference到包含这个checkbox的form

	id	设置或返回checkbox对象的标识符字符串
	name	设置或返回checkbox对象的名字
	type	返回form元素的类型，此处应该返回“checkbox”
	value	设置或返回checkbox的属性值
Method	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
	focus()	使对象得到焦点
Event Handler	onBlur()	对象失去焦点时触发事件
	onChange()	改变选择则触发事件
	onClick()	当焦点置于控件上时，按下【OK】键，触发此方法
	onFocus()	对象得到焦点时触发事件
	onSelect	选择表单内容则触发事件

6.6.4.15 Input Hidden

【说明】

Hidden 对象代表一个 HTML 表单中的某个隐藏输入域。这种类型的输入元素实际上是隐藏的。这个不可见的表单元素的 value 属性保存了一个要提交给 Web 服务器的任意字符串。如果想要提交并非用户直接输入的数据的话，就是用这种类型的元素。

在 HTML 表单中 <input type="hidden"> 标签每出现一次，一个 Hidden object 就会被创建。可通过遍历表单的 elements[] 数组来访问某个隐藏输入域，或者通过使用 document.getElementById()。

表 21 Hidden 对象说明

类别	方法/属性	说明
Property	className	设置或返回元素的 class 属性。
	form	返回一个对包含隐藏域的表单的引用。
	id	设置或返回隐藏域的 id。
	name	设置或返回隐藏域的名称。
	type	返回隐藏输入域的表单类型。
	value	设置或返回隐藏域的 value 属性的值。

6.6.4.16 Input Password

【说明】

Password 对象代表 HTML 表单中的密码域。HTML 的 `<input type="password">` 标签在表单上每出现一次，一个 **Password** 对象就会被创建。

该文本输入字段供用户输入某些敏感的数据，比如密码等。当用户输入的时候，他的输入是被掩盖的（例如使用星号*），以防止旁边的人从他背后看到输入的内容。不过需要注意的是，当表单提交时，输入是用明文发送的。

与类型为 **"text"** 的元素类似，当用户改变显示值时，它会触发 **onchange** 事件句柄。可以通过遍历表单的 **elements[]** array 来访问密码域，或者通过使用 **document.getElementById()** 。

表 22 Input Password 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问密码字段的快捷键。
	className	设置或返回元素的 class 属性。
	defaultValue	设置或返回对象的初始状态下的值
	disabled	设置或返回是否应被禁用密码域。
	form	返回一个reference到包含该password的form
	id	设置或返回对象的标识符字符串
	maxLength	设置或返回密码域中字符的最大数目。
	name	设置或返回对象的名字
	readOnly	设置或返回密码域是否应当是只读的。
	size	设置或返回密码域的长度。
	type	设置或返回form表单元素的类型
	value	设置或返回password中value属性的属性值
Method	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
	focus()	使对象得到焦点
Event Handler	onClick()	当焦点在Password上时，点击遥控器【ok】键触发事件
	onFocus()	对象得到焦点触发事件
	onBlur()	对象失去焦点时触发事件
	onKeyPress()	当用户按下字面键时触发
	onChange()	改变passwordt内容则触发事件

6.6.4.17 Input Radio

【说明】

Radio 对象代表 HTML 表单中的单选按钮。在 HTML 表单中 `<input type="radio">` 每出现一次，一个 Radio 对象就会被创建。

单选按钮是表示一组互斥选项按钮中的一个。当一个按钮被选中，之前选中的按钮就变为非选中的。当单选按钮被选中或不选中时，该按钮就会触发 `onclick` 事件句柄。可通过遍历表单的 `elements[]` 数组来访问 Radio 对象，或者通过使用 `document.getElementById()`。

表 23 Radio 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问单选按钮的快捷键。
	checked	设置或返回当前radio的选择情况
	className	设置或返回元素的 class 属性。
	defaultChecked	设置或返回checked属性的默认值（true为已选）
	form	返回一个reference到包含该表单的form
	id	设置或返回radio表单的标识符字符串
	name	设置或返回radio表单的名字
	type	设置或返回form元素的类型，此处应该返回radio
	value	设置或返回radio按钮的属性值
Method	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
	focus()	使对象得到焦点
Event Handler	onBlur()	对象失去焦点时触发事件
	onChange()	改变选择则触发事件
	onClick()	点击对象触发事件
	onFocus()	对象得到焦点触发事件
	onSelect	选择表单内容则触发事件

6.6.4.18 Input Reset

【说明】

Reset 对象代表 HTML 表单中的一个重置按钮。在 HTML 表单中 `<input type="reset">` 标签每出现一次，一个 Reset 对象就会被创建。

当重置按钮被点击，包含它的表单中所有输入元素的值都重置为它们的默认值。默认值由 HTML value 属性或 JavaScript 的 `defaultValue` 属性指定。

重置按钮在重置表单之前触发 `onclick` 句柄，并且这个句柄可以通过返回 `false` 来取消。可以通过遍历表单的 `elements[]` 数组来访问某个重置按钮，或者通过使用 `document.getElementById()`。

表 24 Reset 对象说明

类别	方法/属性	说明
Property	<code>accesskey</code>	设置或返回访问重置按钮的快捷键。
	<code>className</code>	设置或返回元素的 <code>class</code> 属性。
	<code>form</code>	返回reset按钮的父类的名字
	<code>disabled</code>	设置或返回重置按钮是否应被禁用。
	<code>id</code>	设置或获得对象的标识符字符串
	<code>name</code>	设置或获得reset按钮的名字
	<code>value</code>	设置reset按钮的value属性
	<code>type</code>	返回form表单元素的类型
Method	<code>blur()</code>	把焦点从元素上移开。
	<code>click()</code>	在某个按钮上模拟一次单击。
	<code>focus()</code>	使对象得到焦点
Event Handler	<code>onBlur()</code>	对象失去焦点时触发事件
	<code>onClick()</code>	点击对象触发事件
	<code>onFocus()</code>	对象得到焦点触发事件

6.6.4.19 Input Submit

【说明】

Submit 对象代表 HTML 表单中的一个提交按钮 (submit button)。在 HTML 表单中 `<input type="submit">` 标签每出现一次，一个 Submit 对象就会被创建。

在表单提交之前，触发 `onclick` 事件句柄，并且一个句柄可以通过返回 `false` 来取消表单提交。可以通过遍历表单的 `elements[]` 数组来访问某个提交按钮，或者通过使用 `document.getElementById()`。

表 25 Submit 对象说明

类别	方法/属性	说明
Property	<code>accessKey</code>	设置或返回访问提交按钮的快捷键。
	<code>className</code>	设置或返回元素的 <code>class</code> 属性。

	form	返回一个reference至包含该提交按钮的form
	disabled	设置或返回提交按钮是否应被禁用。
	id	设置或返回提交表单的标识符字符串
	name	设置或返回提交按钮的名字
	value	设置或返回submit按钮中value属性的值
	type	返回form元素的类型
Method	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
	focus()	使对象得到焦点
Event Handler	onBlur()	对象失去焦点时触发事件
	onClick()	点击对象触发事件
	onFocus()	对象得到焦点触发事件

6.6.4.20 Input Text

【说明】

Text 对象代表 HTML 表单中的文本输入域。在 HTML 表单中 `<input type="text">` 每出现一次，Text 对象就会被创建。

该元素可创建一个单行的文本输入字段。当用户编辑显示的文本并随后把输入焦点转移到其他元素的时候，会触发 `onchange` 事件句柄。

可以通过表单的 `elements[]` 数组来访问文本输入域，或者通过使用 `document.getElementById()`。

表 26 Text 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问提交按钮的快捷键。
	className	设置或返回元素的 class 属性。
	defaultValue	设置或返回文本域的默认值。
	disabled	设置或返回文本域是否应被禁用。
	form	返回一个reference至包含该提交按钮的form
	id	设置或返回提交表单的标识符字符串
	maxLength	设置或返回文本域中的最大字符数。
	name	设置或返回text的名字
	readOnly	设置或返回文本域是否应是只读的。
	size	设置或返回文本域的尺寸。

	type	设置或返回form表单元素的类型
	value	设置或返回text中value属性的属性值
Method	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
	focus()	使对象得到焦点
Event Handler	onClick()	当焦点在对象上时，点击遥控器【ok】键触发事件
	onFocus()	对象得到焦点触发事件
	onBlur()	对象失去焦点时触发事件
	onKeyPress()	当用户按下字面键时触发
	onChange()	改变内容则触发事件

6.6.4.21 Link

【说明】

Link 对象代表某个 HTML 的 <link> 元素。<link> 元素可定义两个链接文档之间的关系。<link> 元素被定义于 HTML 文档的 head 部分。

表 27 Link 对象说明

类别	方法/属性	说明
Property	id	设置或返回链接的标识符字符串
	disabled	设置或返回目标 URL 是否当被禁用
	href	设置或返回链接源的url地址
	type	设置或返回目标 URL 的 MIME 类型

6.6.4.22 Meta

【说明】

Meta 对象代表 HTML 的一个 <meta> 元素。<meta> 元素可提供有关某个 HTML 元素的元信息 (meta-information)，比如描述、针对搜索引擎的关键词以及刷新频率。

表 28 Meta 对象说明

类别	方法/属性	说明
Property	content	设置或获取与 HTTP-EQUIV 或 NAME 关联的资源信息
	httpEquiv	把 content 属性连接到一个 HTTP 头部。
	name	把 content 属性连接到某个名称。

6.6.4.23 Object

【说明】

目前仅限于调用 JVM 客户端。

表 29 Object 对象说明

类别	方法/属性	说明
Property	id	该Object的ID
	classid	该Object的注册类ID，对应不同类型的应用形式。使用“clsid: 72E6F181-D1B0-4C22-B0D7-4A0740EEAEF5”
	width	客户端应用运行时所占像素宽度
	height	客户端应用运行时所占像素高度

6.6.4.24 Option

【说明】

Option 对象代表 HTML 表单中下拉列表中的一个选项。在 HTML 表单中 <option> 标签每出现一次，一个 Option 对象就会被创建。

可通过表单的 elements[] 数组访问一个 Option 对象，或者通过使用 document.getElementById()。

表 30 Option 对象说明

类别	方法/属性	说明
Property	className	设置或返回元素的 class 属性。
	defaultSelected	设置或返回选项中已被选中的属性值
	disabled	设置或返回选项是否应被禁用。
	id	设置或返回选项的标识符字符串
	index	返回下拉列表中某个选项的索引位置。
	selected	设置或返回当前选项的状态
	text	设置或返回<option>标签的一个文本值
	value	设置或返回一个选项被选择后提交到服务器的值

6.6.4.25 Select

【说明】

Select 对象代表 HTML 表单中的一个下拉列表。在 HTML 表单中，<select> 标签每出现一次，一个 Select 对象就会被创建。

可通过遍历表单的 elements[] 数组来访问某个 Select 对象，或者通过使用 document.getElementById()。

表 31 Select 对象说明

类别	方法/属性	说明
Property	className	设置或返回元素的 class 属性。
	form	返回一个reference至包含下拉选择框的form
	disabled	设置或返回是否应禁用下拉列表。
	id	设置或返回对象的标识符字符串
	multiple	设置或返回是否选择多个项目。
	length	设置或返回下拉选择框中选项的数目
	name	设置或返回下拉列表的名字
	selectedIndex	设置或返回select框中被选中的选项的索引
	size	设置或返回下拉列表中的可见行数。
	type	返回form元素的类型
Method	blur()	从下拉列表移开焦点。
	focus()	使得select得到焦点
	add()	向下拉列表添加一个选项。
	remove()	从下拉列表中删除一个选项。
Event Handler	onClick()	当焦点置于控件上时，按下【OK】键，触发此方法
	onFocus()	select选择框获得焦点时触发事件
	onBlur()	对象失去焦点时触发事件
	onchange()	当改变选择时调用的事件句柄。
Collection	options[]	返回包含下拉列表中的所有选项的一个数组。

6.6.4.26 Style

【说明】

Style 对象代表一个单独的样式声明。可从应用样式的文档或元素访问 Style 对象。使用 Style 对象属性的语法：

```
document.getElementById("id").style.property="值"
```

表 32 Style 对象说明

类别	方法/属性	说明
Background 属性	background	在一行中设置所有的背景属性
	backgroundColor	设置背景图像是否固定或随页面滚动
	backgroundImage	设置元素的背景图像
	backgroundPosition	设置背景图像的起始位置
	backgroundRepeat	设置是否及如何重复背景图像
Border 和 Margin 属性	border	在一行设置四个边框的所有属性
	borderBottom	在一行设置底边框的所有属性
	borderBottomColor	设置底边框的颜色
	borderBottomStyle	设置底边框的样式
	borderBottomWidth	设置底边框的宽度
	borderColor	设置所有四个边框的颜色 (可设置四种颜色)
	borderLeft	在一行设置左边框的所有属性
	borderLeftColor	设置左边框的颜色
	borderLeftStyle	设置左边框的样式
	borderLeftWidth	设置左边框的宽度
	borderRight	在一行设置右边框的所有属性
	borderRightColor	设置右边框的颜色
	borderRightStyle	设置右边框的样式
	borderRightWidth	设置右边框的宽度
	borderStyle	设置所有四个边框的样式 (可设置四种样式)
	borderTop	在一行设置顶边框的所有属性
	borderTopColor	设置顶边框的颜色
	borderTopStyle	设置顶边框的样式
	borderTopWidth	设置顶边框的宽度
	borderWidth	设置所有四条边框的宽度 (可设置四种宽度)
	margin	设置元素的边距 (可设置四个值)
	marginBottom	设置元素的底边距
	marginLeft	设置元素的左边距
	marginRight	设置元素的右边距
	marginTop	设置元素的顶边距
	padding	设置元素的填充 (可设置四个值)

	paddingBottom	设置元素的下填充
	paddingLeft	设置元素的左填充
	paddingRight	设置元素的右填充
	paddingTop	设置元素的顶填充
Layout 属性	direction	设置元素的文本方向
	display	设置元素如何被显示
	clear	设置在元素的哪边不允许其他的浮动元素
	clip	设置元素的形状
	cssFloat	设置图像或文本将出现（浮动）在另一元素中的何处。（取值范围： <i>left right none</i> ）
	height	设置元素的高度
	overflow	规定如何处理不适合元素盒的内容
	verticalAlign	设置对元素中的内容进行垂直排列
	visibility	设置元素是否可见 （取值范围： <i>inherit visible collapse hidden</i> ）
	width	设置元素的宽度
List 属性	listStyle	在一行设置列表的所有属性
	listStyleImage	把图像设置为列表项标记
	listStylePosition	改变列表项标记的位置
	listStyleType	设置列表项标记的类型
Positioning 属性	bottom	设置元素的底边缘距离父元素底边缘的之上或之下的距离
	left	置元素的左边缘距离父元素左边缘的左边或右边的距离
	position	把元素放置在 <i>static, relative, absolute</i> 或 <i>fixed</i> 的位置
	right	置元素的右边缘距离父元素右边缘的左边或右边的距离
	top	设置元素的顶边缘距离父元素顶边缘的之上或之下的距离
Text 属性	color	设置文本的颜色
	font	在一行设置所有的字体属性
	fontFamily	设置元素的字体系列（只支持黑体）
	fontSizeAdjust	设置/调整文本的尺寸
	fontSize	在一行设置所有的字体属性
	fontStyle	设置元素的字体样式

	fontWeight	设置字体的粗细（只支持normal及bold）
	lineHeight	设置行间距
	textAlign	排列文本
	textDecoration	设置文本的修饰
	textTransform	对文本设置大写效果
	unicodeBidi	
	textIndent	缩紧首行的文本
	wordSpacing	设置文本中的词间距

【注】:

1、各属性支持的参数请参考第八章 CSS;

2、在下表中列出了 display 属性的可能值:

表 33 display 属性说明

值	描述
none	此元素不会被显示。
block	此元素将显示为块级元素，此元素前后会带有换行符。
inline	默认。此元素会被显示为内联元素，元素前后没有换行符。
list-item	此元素会作为列表显示。
run-in	此元素会根据上下文作为块级元素或内联元素显示。
compact	此元素会根据上下文作为块级元素或内联元素显示。
table	此元素会作为块级表格来显示（类似 <table>），表格前后带有换行符。
inline-table	此元素会作为内联表格来显示（类似 <table>），表格前后没有换行符。
table-row-group	此元素会作为一个或多个行的分组来显示（类似 <tbody>）。
table-header-group	此元素会作为一个或多个行的分组来显示（类似 <thead>）。
table-footer-group	此元素会作为一个或多个行的分组来显示（类似 <tfoot>）。
table-row	此元素会作为一个表格行显示（类似 <tr>）。
table-column-group	此元素会作为一个或多个列的分组来显示（类似 <colgroup>）。
table-column	此元素会作为一个单元格列显示（类似 <col>）

table-cell	此元素会作为一个表格单元格显示（类似 <td> 和 <th>）
table-caption	此元素会作为一个表格标题显示（类似 <caption>）

6.6.4.27 Textarea

【说明】

Textarea 对象代表 HTML 表单中的一个文本区 (text-area)。在表单中 <textarea> 标签每出现一次，一个 Textarea 对象就会被创建。可以通过索引相应表单的元素数组来访问某个 Textarea 对象，或者使用 getElementById()。

表 34 Textarea 对象说明

类别	方法/属性	说明
Property	accessKey	设置或返回访问按钮的快捷键。
	className	设置或返回元素的 class 属性。
	defaultValue	设置或返回文本框的初始内容
	id	设置或返回文本框的标识符
	form	返回一个reference到该文本框的父表单
	type	返回该表单元素的类型
	value	设置或返回textarea的文本
	cols	设置或返回 textarea 的宽度。
	disabled	设置或返回 textarea 是否应当被禁用。
	readOnly	设置或返回 textarea 是否应当是只读的。
Method	rows	设置或返回 textarea 的高度。
	blur()	把焦点从元素上移开。
	click()	在某个按钮上模拟一次单击。
Event Handler	focus()	使对象得到焦点
	onClick()	当焦点在对象上时，点击遥控器【ok】键触发事件
	onFocus()	对象得到焦点触发事件
	onBlur()	对象失去焦点时触发事件
	onKeyPress()	当用户按下字面键时触发
	onChange()	改变内容则触发事件

6.6.5 JavaScript 对象

6.6.5.1 Array

表 35 Array 对象说明

类别	方法/属性	说明
Property	constructor	由构造函数创建一个数组arr, arr.constructor返回创建这个arr对象的函数
	length	返回数组的长度, 即数组中有多少元素
	prototype	此属性可以为array对象原型添加方法和属性
	index	当一个数组是由正则表达式匹配创建的时候, 返回匹配的字符串所在源字符串中的起始位置
Method	concat()	将参数作为数组的元素合并到数组中, 并返回一个新数组
	join()	把数组中所有元素以separator为分隔符合并成一个字符串
	pop()	移除数组最后一个元素, 并返回这个元素, 同时改变数组的长度
	push()	添加一个或者多个元素到数组末尾, 返回数组的新长度
	reverse()	使数组中元素前后互调, 数组最后一个元素成为第一个, 第一个元素成为最后一个
	shift()	移除数组第一个元素, 并返回这个元素, 同时改变数组的长度
	slice()	截取数组中从begin开始, end结束的子数组, 并返回这个子数组
	sort()	把数组按指定排序顺序进行排序
	splice()	截取数组中从begin开始, end结束的子数组, 并返回这个子数组
	toString()	返回指定数组中所有元素组成的字符串
	unShift()	添加一个或者多个元素到数组的开始位置, 并返回数组的新长度
	valueOf()	返回array数组对象的原始值

6.6.5.2 Boolean

表 36 Boolean 说明

类别	方法/属性	说明
Method	toString()	返回指定对象的字符串
	valueOf()	返回Boolean对象的原始值

6.6.5.3 Date

表 37 Date 对象说明

类别	方法/属性	说明
Constructor parameters	milliseconds	毫秒
	dateString	日期字符串
	year, month, date	年、月、日
	hour, minute, second	小时、分、秒
Property	constructor	指出创建一个对象原型的函数
	prototype	表现一个类的原型,通过该原型能够对所有该类的实例增加属性或方法
Method	getDate()	返回 Date对象中用本地时间表示的一个月中的日期值
	getDay()	返回 Date对象中用本地时间表示的一周中的日期值
	getFullYear()	返回 Date对象中用本地时间表示的年份值
	getHours()	返回 Date对象中用本地时间表示的小时值
	getMilliseconds()	返回 Date对象中用本地时间表示的毫秒值
	getMinutes()	返回 Date对象中用本地时间表示的分钟值
	getMonth()	返回 Date对象中用本地时间表示的月份值
	getSeconds()	返回 Date对象中用本地时间表示的秒钟值
	getTime()	返回 Date对象中的时间值
	getTimezoneOffset()	返回用分钟表示的主计算机上的时间和全

		球标准时间 (UTC)之间的差别
	getUTCDate()	返回 Date 对象中用全球标准时间 (UTC)表示的日期
	getUTCDay()	返回 Date 对象中用全球标准时间 (UTC)表示的一周中的日期值
	getUTCFullYear()	返回 Date 对象中用全球标准时间 (UTC)表示的年份值
	getUTCHours()	返回 Date 对象中用全球标准时间 (UTC)表示的小时值
	getUTCMilliseconds()	返回 Date 对象中用全球标准时间 (UTC)表示的毫秒值
	getUTCMinutes()	返回 Date 对象中用全球标准时间 (UTC)表示的分钟值
	getUTCMonth()	返回 Date 对象中用全球标准时间 (UTC)表示的月份值
	getUTCSeconds()	返回 Date 对象中用全球标准时间 (UTC)表示的秒钟值
	setDate(val)	设置 Date 对象中用本地时间表示的数字日期
	setFullYear()	设置 Date 对象中用本地时间表示的年份值
	setHours(val)	设置 Date 对象中用本地时间表示的小时值
	setMilliseconds(val)	设置 Date 对象中用本地时间表示的毫秒值
	setMinutes(val)	设置 Date 对象中用本地时间表示的分钟值
	setMonth(val)	设置 Date 对象中用本地时间表示的月份值
	setSeconds(val)	设置 Date 对象中用本地时间表示的秒钟值
	setTime(val)	设置 Date 对象的日期和时间值
	setUTCDate()	设置 Date 对象中用全球标准时间 (UTC)表示的数值日期
	setUTCMilliseconds()	设置 Date 对象中用全球标准时间 (UTC)表示的年份值

	setUTCMinutes()	设置 Date 对象中用全球标准时间 (UTC) 表示的小时值
	setUTCMonth()	设置 Date 对象中用全球标准时间 (UTC) 表示的毫秒值
	setUTCSeconds()	设置 Date 对象中用全球标准时间 (UTC) 表示的分钟值
	setYear(val)	设置 Date 对象中用全球标准时间 (UTC) 表示的月份值。不鼓励使用
	toGMTString()	返回一个日期, 该日期用格林威治标准时间 (GMT) 表示并已被转换为字符串。不鼓励使用
	toLocaleString()	返回一个日期, 该日期使用当前区域设置并已被转换为字符串
	toLocaleTimeString()	以字符串的形式返回与宿主环境的当前区域设置相对应的日期
	toSource()	以字符串的形式返回与宿主环境的当前区域设置相对应的时间
	toString()	返回表现 Date 对象源代码的字符串
	toUTCString()	返回对象的字符串表示
	UTC (yy,MM,dd,hh,mm,ss,ms)	返回一个已被转换为字符串的, 用全球标准时间 (UTC)表示的日期
	valueOf()	返回全球标准时间 (UTC)(或 GMT) 的 1970年1月1日到所指定日期之间所间隔的毫秒数

6.6.5.4 Math

表 38 Math 对象说明

类别	方法/属性	说明
Property	E	返回 Euler 常数, 自然对数的底
	LN2	返回 2 的自然对数
	LN10	返回 10 的自然对数
	LOG2E	返回以 2 为底 e (自然对数的底) 的对数
	LOG10E	返回以 10 为底 e (自然对数的底) 的对数
	PI	返回圆的周长与其直径的比值, 约等于 3.141592653589793
	SQRT1_2	返回 0.5 的平方根, 或说 2 的平方根分之一

	SQRT2	返回 2 的平方根
Method	abs()	返回数字的绝对值
	acos()	返回数的反余弦值
	asin()	返回数字的反正弦值
	atan()	返回数字的反正切值
	atan2()	返回由 X 轴到 (y,x) 点的角度（以弧度为单位）
	ceil()	返回大于等于其数字参数的最小整数
	cos()	返回数的余弦值
	exp()	返回 e（自然对数的底）的幂
	floor()	返回小于等于其数值参数的最大整数
	log()	返回数字的自然对数
	max()	返回给出的零个或多个数值表达式中较大者
	min()	返回给出的零个或多个数值表达式中较小的值
	pow()	返回底表达式的指定次幂
	random()	返回介于 0 和 1 之间的伪随机数
	round()	返回与给出的数值表达式最接近的整数
	sin()	返回数字的正弦值
	sqrt()	返回数字的平方根
	tan()	返回数字的正切值

6.6.5.5 Number

表 39 Number 对象说明

类别	方法/属性	说明
Property	MAX_VALUE	返回 JavaScript 能表达的最大的数
	MIN_VALUE	返回 JavaScript 中能够表示的最接近零的数
	NaN	表示算术表达式返回非数字值的特殊值
	NEGATIVE_INFINITY	返回比 JavaScript 能够表示的最小负数（-Number.MAX_VALUE）更小的值。
	POSITIVE_INFINITY	返回比在 JavaScript 中能够表示的最大的数 (Number.MAX_VALUE)更大的值
	prototype	返回对象类型原型的引用
Method	toExponential()	返回一个字符串，其中包含一个以指数记数法表示的数字
	toFixed()	返回一个字符串，代表一个以定点表示法表示的数字

	toPrecision()	返回一个字符串, 其中包含一个以指数记数法或定点记数法表示的, 具有指定数字位数的数字
	toString()	得到一个表示 Number 对象的字符串
	valueOf()	得到一个 Number 对象的初始值

6. 6. 5. 6 Object

表 40 Object 对象说明

类别	方法/属性	说明
Property	constructor	表示创建对象的函数
Method	eval()	检查 JScript 代码并执行
	toString()	返回对象的字符串表示
	valueOf()	返回指定对象的原始值

6. 6. 5. 7 RegExp

表 41 RegExp 说明

类别	方法/属性	说明
Property	constructor	返回创建 regexp 对象的函数
	global	返回 Boolean 值, 指出正则表达式使用的 global 标志(g)的状态(只读属性)
	ignoreCase	返回 Boolean 值, 指出正则表达式使用的 ignoreCase 标志(i)的状态(只读属性)
	lastIndex	返回字符位置, 它是在被查找字符串中下一次成功匹配的开始位置
	multiline	返回 Boolean 值, 指出正则表达式使用的 multiline 标志(m)的状态(只读属性)
	prototype	引用 regexp 对象的原型, 可以为原型增加新的方法和属性
	source	返回正则表达式模式的文本的复本(只读属性)
Method	exec()	用正则表达式模式在字符串中运行查找, 并返回包含该查找结果的一个数组
	test()	返回一个 Boolean 值, 它指出在被查找的字符串中是否存在模式
	toString()	返回对象的字符串表示

6.6.5.8 String

表 42 String 说明

类别	方法/属性	说明
Property	constructor	表示创建对象的函数
	length	返回 String 对象的长度
	prototype	返回对象类型原型的引用
Method	anchor()	在对象中的指定文本两端放置一个有 NAME 属性的 HTML 锚点
	big()	把 HTML <BIG> 标记放置在 String 对象中的文本两端
	blink()	把 HTML <BLINK> 标记放置在 String 对象中的文本两端
	bold()	把 HTML 标记放置在 String 对象中的文本两端
	charAt()	返回指定索引位置处的字符
	charCodeAt()	返回一个整数, 代表指定位置上字符的 Unicode 编码
	concat()	返回字符串值, 该值包含了两个或更多个提供的字符串的连接
	fixed()	把 HTML <TT> 标记放置在 String 对象中的文本两端
	fontcolor()	把带有 COLOR 属性的一个 HTML 标记放置在 String 对象中的文本两端
	fontsize()	把一个带有 SIZE 属性的 HTML 标记放置在 String 对象中的文本的两端
	fromCharCode()	从一些 Unicode 字符值中返回一个字符串
	indexOf()	返回 String 对象内第一次出现子字符串的字符位置
	Italics()	把 HTML <I> 标记放置在 String 对象中的文本两端
	lastindexOf()	返回 String 对象中子字符串最后出现的位置
	link()	把一个有 HREF 属性的 HTML 锚点放置在 String 对象中的文本两端
	match()	使用正则表达式模式对字符串执行查找, 并将包含查找的结果作为数组返回
	replace()	返回根据正则表达式进行文字替换后的字符串的复制
	search()	返回与正则表达式查找内容匹配的的第一个子字符串的位置
	slice()	返回字符串的片段
	small()	将 HTML 的<SMALL> 标识添加到 String 对象中的文本两端

	split()	将一个字符串分割为子字符串，然后将结果作为字符串数组返回
	strike()	将 HTML 的<STRIKE> 标识放置到 String 对象中的文本两端
	sub()	将 HTML 的 <SUB> 标识放置到 String 对象中的文本两端
	substr()	返回一个从指定位置开始的指定长度的子字符串
	substring()	返回位于 String 对象中指定位置的子字符串
	sup()	将 HTML 的 <SUP> 标识放置到 String 对象中的文本两端
	toLowerCase()	返回一个字符串，该字符串中的字母被转换为小写字母
	toString()	返回对象的字符串表示
	toUpperCase()	返回一个字符串，该字符串中的所有字母都被转化为大写字母
	valueOf()	返回指定对象的原始值

6.6.5.9 Kernel Property and Method

表 43 Kernel Property and Method

类别	方法/属性	说明
Property	Infinity	返回Number.POSITIVE_INFINITY 的初始值
	NaN	返回特殊值 NaN，指示一个表达式不是数字。使用方法举例：Var t= Number.NaN
	undefined	指示未定义的值。
Method	decodeURI()	返回一个已编码的统一资源标识符(URI) 的非编码形式
	encodeURI()	返回一个编码的URI
	eval()	检查JScript 代码并执行
	isFinite()	返回一个Boolean 值,指明所提供的数字是否是有限的
	isNaN()	返回一个Boolean 值，指明提供的值是否是保留值 NaN
	number()	返回或设置与特定错误相联系的数字值
	parseFloat()	返回由字符串转换得到的浮点数
	parseInt()	返回与保存在numString 中的数字值相等的整数
	string()	将一个对象转变为一个字符串
	undefined()	返回undefined 的值

	escape()	返回一个包含了 charstring 内容的字符串值（Unicode 格式）
	unescape()	从用 escape 方法编码的 String 对象中返回已解码的字符串

6.6.6 扩展 JavaScript 对象

6.6.6.1 Authentication

表 44 Authentication 对象说明

类别	方法/属性	说明
Method	CTCGetAuthInfo(EncryToken)	获取本地配置信息，使用 password 对 loginname 、 EncryToken 等信息进行加密，并按要求格式返回结果字符串，其详细加密过程参考表格后的说明 ● EncryToken : 采用 ASCII 编码的 32 位长字符串； ● 返回值: string 型加密结果
	CTCSetConfig(fieldname, fieldValue)	将需要设置的参数配置到终端中。目前可设置的参数请参考表格后的说明。 ● fieldname : 字符串类型，表示需要设置的参数名称； ● fieldValue : 字符串类型，表示需要设置的参数值；
	CTCSetConfig(fieldname)	将取出设置在终端中的参数。目前可取出的参数请参考表格后的说明。 ● fieldname : 字符串类型，表示需要取出的参数名称； ● 返回值: 字符串类型，表示取出的参数值；
	CTCStartUpdate()	后台启动升级功能。终端启动升级流程，并立即返回
	CTCLogin()	触发终端认证流程

● **CTCGetAuthInfo** 方法加密过程说明:

加密方法如下，其中 **Authenticator** 表示函数返回值，具体为:

Authenticator=3DES(Random+“\$”+EncryToken+”\$”+UserID
+”\$”+STBID+”\$”+IP+”\$”+MAC+”\$”+ Reserved+ ”\$”+ “CTC”)

其中:

- 在不采用机卡分离方式的情况下，使用用户的密码（由运营商统一分配，用户可通过终端操作界面进行配置）作为 3DES 密钥进行加密；
- 在采用机卡分离方式的情况下，采用为终端 SIM 卡内保存的密钥 IKey 作为 3DES 密钥；
- 密钥长度不足 24 字节时，右补 ASCII 字符“0”；
- 内容采用 PKCS5Padding 方式填充，即长度以 8 字节切分，不能被 8 整除的末尾部分，根据长度不足 8 字节的部分，填充“0x01”——“0x08”，如不足 1 字节，则填充 1 个“0x01”，如不足 2 字节，则填充 2 个“0x02”，以此类推，如整除，则填充 8 个“0x08”；
- 加密算法采用的鉴权算法应采用 3DES（168 位，ECB 方式）。
- Random 为十进制表示的随机数字，范围为 0~999999999

【注】返回结果使用 ASCII 形式的十六进制编码表示，采用大写“ABCDEF”，如加密结果为 8 字节二进制字符串：0x0123456789ABCDEF，则 Authenticator 为 16 字节 ASCII 字符串：“0123456789ABCDEF”。

● CTCSetConfig/ CTCGetConfig 方法说明：

目前 CTCSetConfig 及 CTCGetConfig 方法支持的参数列表如下：

表 45 CTCSetConfig/ CTCGetConfig 方法支持参数列表

参数列表	
UserToken	EPGDomain
EPGDomainBackup	UpgradeDomain
UpgradeDomainBackup	ManagementDomain
ManagementDomainBackup	NTPDomain
NTPDomainBackup	UserGroupNMB
EPGGroupNMB	ChannelCount
Channel	ServiceEntry

6.6.6.2 MediaPlayer

MediaPlayer 对象是对于终端播放直播、点播、即时时移、回看、音乐等这些媒体内容的能力进行的封装，并将其播放的基本行为抽象为 MediaPlayer 的相关方法以实现播放，

快进，跳转，暂停，停止等操作。

终端应能够支持同时创建至少 2 个 MediaPlayer 对象以支持音视频的同时播放，并且在终端能力允许的情况下可创建更多的 MediaPlayer 对象实例。

6.6.6.2.1 属性

表 46 MediaPlayer 属性列表

属性	属性值	说明
播放器的配置属性		
getNativePlayerInstanceId() D()	一个字节长度的无符号正整数（1—255）	获得 STB 本地播放器实例的 instanceID。
setSingleOrPlaylistMode (mode) getSingleOrPlaylistMode ()	0: 单媒体的播放模式(默认值) 1: 播放列表的播放模式	MediaPlayer 的播放模式。在初始化 MediaPlayer 对象时必须进行该参数的初始设定。在播放过程中，此属性不能再被修改。
setVideoDisplayMode(videoDisplayMode) getVideoDisplayMode()	0: 按 setVideoDisplayArea()中设定的 Height,Width,Left,Top 属性所指定的位置和大小来显示视频 1: 全屏显示，按全屏高度和宽度显示(默认值) 2: 按宽度显示，指在不改变原有图像纵横比的情况下按全屏宽度显示 3: 按高度显示，指在不改变原有图像纵横比的情况下按全屏高度显示 255: 视频显示窗口将被关闭。它将在保持媒体流连接的前提下，隐藏视频窗口。如果流媒体播放没有被暂停，将继续播放音频。	MediaPlayer 对象对应的视频窗口的显示模式。每次调用该函数后，视频显示窗口并不会被立即重新刷新以反映更改后的显示效果只有等到显式调用 refreshVideoDisplay()后才会刷新
setVideoDisplayArea(left, top, width, height)	Left: 相对于所在浏览器窗口左上角的右向偏移的像素点个数(默认值为 0) Top: 相对于所在浏览器窗口左上角的向下偏移的像素点个数(默认值为 0) Width: 显示视频的窗口宽度的像素点个数(默认值为默认页面设计空间的宽度，对 PAL 来说为 720) Height: 显示视频的窗口高度的像素点个数(默认值为默认页面设计空间的高度，对 PAL 来说为 576)	每次调用该函数后，视频显示窗口并不会被立即重新刷新以反映更改后的显示效果只有等到显式调用 refreshVideoDisplay()后才会刷新

getVideoDisplayLeft()	返回通过 setVideoDisplayArea() 设置的 left 值 像素值, (默认值为 0)	视频窗口的位置, 相对于所在浏览器窗口左上角的右向偏移 (像素点个数)
getVideoDisplayTop()	返回通过 setVideoDisplayArea() 设置的 top 值 像素值, (默认值为 0)	视频窗口的位置, 相对于所在浏览器窗口左上角的向下偏移 (像素点个数)
getVideoDisplayWidth()	返回通过 setVideoDisplayArea() 设置的 width 值 像素值, (默认值为 0)	显示视频的窗口宽度
getVideoDisplayHeight()	返回通过 setVideoDisplayArea() 设置的 height 值 像素值, (默认值为 0)	显示视频的窗口高度
setMuteFlag(muteFlag) getMuteFlag()	0: 设置为有声(默认值) 1: 设置为静音	MediaPlayer 对应的本地播放器实例是否静音 (sessionScope , Mute 键所触发的 MUTE 状态为全局 MUTE 状态, 不影响该值), 该值并不影响 STB 本地其它音频有关应用的 Mute 状态。 设置后立即生效。
setNativeUIFlag(nativeUIFlag) getNativeUIFlag()	0: 不使用 Player 的本地 UI 显示功能 1: 使用 Player 的本地 UI 显示功能(默认值)	可选的属性: 保留此属性, 但可以不实现功能 播放器是否显示缺省的 NativeUI , 如进度条/音量提示/静音提示/频道号 / 等。详细描述参考章节“ MediaPlayer 的行为”。设置后立即生效。
setMuteUIFlag(muteUIFlag) getMuteUIFlag()	0: 不使用静音提示的本地 UI 显示功能 1: 使用静音提示的本地 UI 显示功能 (默认值)	(可选): 保留此属性, 但可以不实现功能 播放器是否显示缺省的本地静音提示 UI。该属性与 nativeUIFlag 属性是逻辑与的关系。设置后立即生效。
setAudioVolumeUIFlag(audioVolumeUIFlag) getAudioVolumeUIFlag()	0: 不使用音量调节的本地 UI 显示功能 1: 使用音量调节的本地 UI 显示功能 (默认值)	(可选): 保留此属性, 但可以不实现功能 播放器是否显示缺省的本地音量调节 UI。该属性与 nativeUIFlag 属性是逻辑与的关系。设置后立即生效。
setAudioTrackUIFlag(audioTrackUIFlag) getAudioTrackUIFlag()	0: 不使用音轨选择的本地 UI 显示功能 1: 使用音轨选择的本地 UI 显示功能 (默认值)	(可选): 保留此属性, 但可以不实现功能 播放器是否显示缺省的本地音轨选择 UI。该属性与 nativeUIFlag 属性是逻辑与的关系。设置后立即生效。
setProgressBarUIFlag(progressBarUIFlag) getProgressBarUIFlag()	0: 不使用进度条的本地 UI 显示功能 1: 使用进度条的本地 UI 显示功能 (默认值)	(可选): 保留此属性, 但可以不实现功能 播放器是否显示缺省的本地进度条

getProgressBarUIFlag()		UI。该属性与 nativeUIFlag 属性是逻辑与的关系。设置后立即生效。
setChannelNoUIFlag(channelNoUIFlag) getChannelNoUIFlag()	0: 不使用频道号的本地 UI 显示功能 1: 使用频道号的本地 UI 显示功能(默认值)	(可选): 保留此属性, 但可以不实现功能 播放器是否显示缺省的本地频道号 UI。该属性与 nativeUIFlag 属性是逻辑与的关系。设置后立即生效。
setSubtitileFlag(subtitileFlag) getSubtitileFlag()	0: 不显示字幕(默认值) 1: 显示字幕	可选的属性: 保留此属性, 但可以不实现功能 播放器是否需要显示字幕。设置后立即生效。
setVideoAlpha(videoAlpha) getVideoAlpha()	0—100 之间的整数值, 0 表示不透明, 100 表示完全透明。 (默认值为 0)	可选的属性: 保留此属性, 但可以不实现功能 视频的透明度
setAllowTrickmodeFlag(allowTrickmodeFlag) getAllowTrickmodeFlag()	0: 允许 TrickMode 操做 1: 不允许 TrickMode 操作(默认值)	表示该播放器实例在生命周期内都是否允许任何 TrickMode 操作(包括快进/快退/暂停)。这个属性与媒体本身的 TrickMode 属性是逻辑与的关系
setCycleFlag(cycleFlag) getCycleFlag()	0: 设置为循环播放 1: 设置为单次播放 (默认值)	可选属性: 设置是否循环播放节目
setRandomFlag(randomFlag) getRandomFlag()	0: 设置为随机播放 (默认值) 1: 设置为随机播放	可选属性: 设置播放列表是否随机播放列表中的节目, 只在播放列表模式下起作用
setVendorSpecificAttr(vendorSpecificAttr) getVendorSpecificAttr()	字符串	留作厂商的扩展属性设置和获取, 格式由厂商自己确定
只读的播放器状态属性		
getMediaCode()	字符串, 媒体的标识	获取当前播放的媒体的标识。 MediaCode 是媒体的唯一标识(节目, 频道等), 使用系统定义的 ContentID 。当播放器在播放列表模式时, MediaCode 是动态改变的。
getMediaDuration()	整数, 以秒为单位	获取当前播放的媒体的总时长
getCurrentPlayTime()	媒体播放到的当前时间点	获取媒体播放到的当前时间点。对 VoD 为从该媒体起始点开始计算的相对时间, 以秒为单位; 对 TVoD 为

		当前播放点的绝对时间; 对 Channel 而言无意义 参见 RFC2326 中的 NormalPlayTime(NPT) 和 AbsoluteTime(ClockTime)两种时间类型的格式
getPlaybackMode()	播放器的当前播放模式。返回值为 JSON 字符串, 其中至少包括“播放模式”和“模式相关参数”两类信息, 播放模式分: NormalPlay, Pause, Trickmode; 当模式为 Trickmode 时必须带 2x/-2x, 4x/-4x, 8x/-8x, 16x/-16x, 32x/-32x 参数来表示快进/快退的速度参数, 如: {PlayMode: “NormalPlay”, Speed: “1x”}	当 STB 播放器的 playbackmode 发生改变的时候以虚拟键及事件的方式通知 EPG
getChannelNum()	整数	返回终端当前播放的频道号, 不能获得有效的频道号时, 返回-1。
getCurrentAudioChannel()	字符串: Left、Right、Stereo、JointStereo	获取当前的声道类型
getAudioTrack()	字符串: Chinese、English 等	获取当前的音轨类型
getSubtitle()	字符串: Chinese、English 等	获取当前选择的字幕类型
只读的播放器状态属性, 只在播放列表模式下有效		
getMediaCount()	整数	获取播放列表中的媒体数量
getCurrentIndex()	整数, 0: 表示媒体列表的顶端	获取当前播的媒体, 在播放列表中的索引值
getEntryID()	字符串, 媒体在播放列表中的唯一标识	获取当前播放的媒体的 EntryID
getPlaylist()	字符串, 用 JSON 格式的字符串, 封装播放列表数据。	参考: 媒体相关参数的描述和封装方式

6.6.6.2.2 方法

表 47 MediaPlayer 方法列表

方法	参数	注释
MediaPlayer()		MediaPlayer 的构造函数, new MediaPlayer()。以缺省值创建对象。
bindNativePlayerInstance(nativePlayerInstanceID)	<ul style="list-style-type: none"> ● nativePlayerInstanceID: 一个字节长度的无符号整数, 0—255 返回: <ul style="list-style-type: none"> ● 0: 表示绑定成功; ● -1: 表示绑定失败; 	终端上某个媒体流或播放器实例的标识, 由 STB 本地负责生成和维护, 一旦针对某个新生成的 MediaPlayer 对象绑定该 ID, 就意味着该对象和 STB 本地的某个媒体播放实例进行了绑定或关联。 如果无法找到 nativePlayerInstanceID 对应的本

	<ul style="list-style-type: none"> ● 其他值未定义 	<p>地播放器实例，则操作失败，返回-1。</p> <p>参考：MediaPlayer 控制的本地播放器实例的生命周期</p>
initMediaPlayer(nativePlayerinstanceID,playlistFlag,videoDisplayMode,height,width,left,top,muteFlag,useNativeUIFlag,subtitleFlag,videoAlpha,cycleFlag,randomFlag,)	<p>参考：</p> <p>播放器的配置属性</p>	<p>初始化 MediaPlayer 的属性。当新创建的 MediaPlayer 是为了和已有的 STB 本地某个媒体播放实例（由先前其它页面中创建 MediaPlayer 对象时创建的）进行绑定，无需调用该函数。</p>
releaseMediaPlayer(nativePlayerInstanceID)	<ul style="list-style-type: none"> ● nativePlayerInstanceID：一个字节长度的无符号整数，0—255 <p>返回：</p> <ul style="list-style-type: none"> ● 0：表示释放成功； ● -1：表示释放失败； ● 其他值未定义 	<p>释放终端 MediaPlayer 的对象，结束对应 MediaPlayer 的生命周期。</p>
setSingleMedia(mediaStr)	<ul style="list-style-type: none"> ● mediaStr, 字符串, 用 json 对象封装媒体的属性。 	<p>设置单个播放的媒体。</p> <p>传入字符串 mediaStr 中的媒体对象的 mediaURL。</p> <ul style="list-style-type: none"> ● rtsp: //的单播地址，则要求连接单播地址进行播放； ● igmp: //的组播地址，要求连接组播地址进行播放； ● http: //的地址用于播放 mp3、WAV 等音频，如 http: //xxxxx/test.mp3、test.wav
joinChannel(userchannelid)	<ul style="list-style-type: none"> ● Userchannelid, 整数, 表示用户频道号。 <p>返回值：</p> <ul style="list-style-type: none"> ● 0，表示成功； ● -1：表示频道号无效。 	<p>要求终端访问指定的频道，并立即返回。</p> <p>对由本地设置为跳过的频道，也返回-1。</p> <p>频道地址为通过 CTCSetConfig 设置的频道列表中的地址：</p> <ul style="list-style-type: none"> ● 如频道地址为 igmp: //的组播地址，则加入组播频道，播放器开始播放组播频道，并处理相应的时移等功能； ● 如频道地址为 rtsp: //的单播地址，则连接单播频道，播放器开始播放； ● 如频道地址为 http: //的地址，则浏览器直接发起请求，访问该页面。 <p>【注】：在加入一个频道之前已经加入另外一个频道，需先调用 leaveChannel 方法离开前一个频道</p>
leaveChannel()	<p>返回值：</p> <ul style="list-style-type: none"> ● 0，表示成功； ● -1：表示频道号无效。 	<p>要求终端离开指定的频道，并立即返回。</p> <ul style="list-style-type: none"> ● 如原频道地址为 igmp: //的组播地址，则立即离开组播频道，播放器停止播放组播频道，并断开相应的时移连接；

		<ul style="list-style-type: none"> ●如频道地址为 rtsp://的单播地址，则断开单播频道，播放器停止播放； ●如频道地址为 http://的地址，则浏览器不作操作。 <p>【注】：本方法只用于离开通过 joinChannel 方法加入的频道。</p>
addSingleMedia(index,mediaStr)	<ul style="list-style-type: none"> ●mediaStr，字符串，用JSON对象封装媒体的属性 	在播放列表末端，添加单个媒体。传入字符串。对于 index ：如果为负数（如-1），表示默认添加在列表尾部（ Append ）；如指定的 index 值大于或等于 playlist 中已有媒体数目，则添加至 playlist 尾部；其它情况为插入列表某个位置（ Insert ）； index 会随着条目的移动/增加/删除而动态变化，由 STB 保证当前 playlist 中所有条目的 index 是从 0 按序加一递增的。对于 entry_id ：唯一标识 playlist 中每个媒体条目，一旦成功加入某个媒体后，该值不会变化。
addBatchMedia(batchMediaStr)	<ul style="list-style-type: none"> ●batchMediaStr，字符串，用JSON对象封装批量媒体的属性。 	在播放列表末端，添加一批媒体。传入字符串
clearAllMedia()	N/A	清空播放列表
moveMediaByIndex(entryID,toIndex)	<ul style="list-style-type: none"> ●entryID：播放列表中某个媒体条目的唯一标识。 ●toIndex，整数，需要移动到的索引值。0：表示播放列表的顶端。 	根据 index 将指定的媒体在播放列表中移动
moveMediaByOffset(entryID,offset)	<ul style="list-style-type: none"> ●entryID：播放列表中某个媒体条目的唯一标识。 ●offset，偏移量，正整数表示从指定媒体向列表末端移动，负整数表示从指定媒体向列表起始端移动 	根据偏移量将指定的媒体在播放列表中移动
moveMediaByIndex1(index,toIndex)	<ul style="list-style-type: none"> ●index，整数，指定的媒体在播放列表中的索引值。0：表示播放列表的顶端。 ●toIndex，整数，需要移动到的索引值。0：表示播放列表的顶端。 	根据 index 将指定的媒体在播放列表中移动
moveMediaByOffset1(index,offset)	<ul style="list-style-type: none"> ●index，整数，指定的媒体在播放列表中的索引值。0：表示播放列表的顶端。 ●offset，偏移量，正整数表示从指定媒体向列表末端移动，负整数表示从指 	根据偏移量将指定的媒体在播放列表中移动

	定媒体向列表起始端移动	
moveMediaToNext(entryID)	●entryID: 播放列表中某个媒体条目的唯一标识。	将指定的媒体下移
moveMediaToPrevious(entryID)	●entryID: 播放列表中某个媒体条目的唯一标识。	将指定的媒体上移
moveMediaToFirst(entryID)	●entryID: 播放列表中某个媒体条目的唯一标识。	将指定的媒体移到列表顶端
moveMediaToLast(entryID)	●entryID: 播放列表中某个媒体条目的唯一标识。	将指定的媒体移到列表末端
moveMediaToNext1(index)	●index, 整数, 指定的媒体在播放列表中的索引值。 0: 表示播放列表的顶端。	将指定的媒体下移
moveMediaToPrevious1(index)	●index, 整数, 指定的媒体在播放列表中的索引值。 0: 表示播放列表的顶端。	将指定的媒体上移
moveMediaToFirst1(index)	●index, 整数, 指定的媒体在播放列表中的索引值。 0: 表示播放列表的顶端。	将指定的媒体移到列表顶端
moveMediaToLast1(index)	●index, 整数, 指定的媒体在播放列表中的索引值。 0: 表示播放列表的顶端。	将指定的媒体移到列表末端
selectMediaByIndex(index)	●index: 媒体在播放列表中的索引, 0: 表示播放列表的顶端。	按媒体在播放列表中的索引选中为当前候选播放节目
selectMediaByOffset(offset)	●offset, 偏移量, 正整数表示从当前媒体向列表末端跳转, 负整数表示从当前媒体向列表起始端跳转	按与当前媒体索引的偏移量选中媒体, 作为当前候选播放节目
selectNext()	N/A	选取播放列表中的下一个媒体, 作为当前候选播放节目
selectPrevious()	N/A	选取播放列表中的上一个媒体, 作为当前候选播放节目
selectFirst()	N/A	选取播放列表中的第一个媒体, 作为当前候选播放节目
selectLast()	N/A	选取播放列表中的最后一个媒体, 作为当前候选播放节目
SelectMediaByEntryID(entryID)	●entryID: 播放列表中某个媒体条目的唯一标识 (在添加媒体时设置并且在播放列表中保持不变)	按某个媒体条目的唯一标识选中为当前候选播放节目
playFromStart()	N/A	从媒体起始点开始播放。 对 TVchannel, 以实时 TV 的方式开始播放。

playByTime(type,timesta mp,speed)	<ul style="list-style-type: none"> ●type: 1: 参见 RFC2326 中的 Normal Play Time (NPT); 2: 参见 RFC2326 中的 Absolute Time (Clock Time) ●timestamp : 参见 RFC2326 中的 Normal Play Time (NPT) 和 Absolute Time (Clock Time)两种时间类型的格式。timestamp: 对 VoD 而言是从媒体起始点开始计算的相对时间; 对 TVoD 等有时间基的媒体而言就是绝对时间。 ●speed: 播放速度, 可选参数。 	参见 RFC2326 (RTSP 协议)。从当前媒体的某个时间点开始播放媒体 (对 playlist 而言是指当前已经选中的媒体)。对实时播放的 TV channel 该调用无效, 但对处于时移状态的 TV channel 有效。
pause()	N/A	暂停正在播放的媒体
fastForward(speed)	●speed: Float 类型, 2 至 32	快进
fastRewind(speed)	●speed: Float 类型, -2 至 -32	快退
resume()	N/A	从当前媒体的暂停/快进/快退状态恢复正常播放。
gotoEnd()	N/A	跳到媒体末端播放
gotoStart()	N/A	跳到媒体起始点播放
stop()	N/A	停止正在播放的媒体。本方法只能用于停止通过 playFromStart 或 playByTime 方法开始的流媒体对象。
refreshVideoDisplay()	N/A	根据 videoDisplayMode,vedioDisplayArea 属性,调整视频的显示。所以设定 Area 和 Mode 参数后并不是立即生效,而是要在显式调用该函数后才会生效。
switchAudioChannel()	N/A	可选的方法: 保留此接口, 但可以不实现功能切换声道, 循环进行。
switchAudioTrack()	N/A	可选的方法: 保留此接口, 但可以不实现功能切换音轨, 循环进行。
switchSubtitle()	N/A	可选的方法: 保留此接口, 但可以不实现功能切换字幕, 循环进行。
sendVendorSpecificCom mand(xml)	xml: XML 格式的字符串	留作厂商的命令扩展, 参数由厂商自定义
getAudioPID()		(可选): 返回音频 PID。如果正在播放返回当前播放的 PID, 否则返回缺省/第一个 PID。
getAudioPIDs()		(可选): 返回一个包含所有音频 PID 信息的对

		象。对象包含以下信息： ●lpids—当前媒体流中包含的音频PID个数 ●lpidarr[1..pids]—音频PID数组： ●AudioPID—PID值（数值） ●AudioLanguage—PID的语言标志（字符串）
setAudioPID()		指定使用的音频 pid。
getSubtitlePID()		返回字幕 PID。如果正在播放返回当前播放的 PID，否则返回缺省/第一个 PID。
getSubtitlePIDs()		返回一个包含所有字幕 PID 信息的对象。对象包含以下信息： ●lpids—当前媒体流中包含的字幕PID个数 ●lpidarr[1..pids]—字幕PID数组： ●SubtitlePID—PID值（数值） ●SubtitleLanguage—PID的语言标志（字符串）
setSubtitlePID()		指定使用的字幕 pid。
setVolume(intvolume)	Volume: 0-100, 表示音量, 0 为静音, 100 为最大音量。	设置系统音量
getVolume()	返回值: 当前系统音量, 0-100	获取当前系统音量
set(ioStr,wrStr)	ioStr: 要设置的参数的名称。(具体意义由终端定义) wrStr: 要设置的参数的值。	可选: 保留此属性, 但可以不实现功能。未来作为扩展使用。
get(ioStr)	ioStr: 要读取的参数的名称。(具体意义由终端定义)	可选: 保留此属性, 但可以不实现功能。未来作为扩展使用。

6.6.6.2.3 媒体对象相关参数

表 48 媒体对象相关参数列表

参数	参数值	是否必填	注释
mediaURL	字符串, 媒体的 url	Y	用于初始化多媒体会话。需要支持 rtsp、igmp 等等格式。例如： rtsp://server.domain.com:554/videos/movie.ts, igmp://239.255.250.2:11111
mediaCode	字符串, 媒体的唯一标识	Y	媒体的唯一标识
mediaType	整数值, 1:TYPE_CHANNEL 2:TYPE_VOD 3:TYPE_TVOD 4:TYPE_MUSIC	Y	媒体的类型
audioType	整数值 1:MPEG-1/2layer2(MP2)	Y	音频编码类型

	2:MPEG-1/2layer3(MP3) 3:MPEG-2LC-AAAC 4:MPEG-4LC-AAC 5:MPEG-4HE-AAC 6:AC-3 7:WMA9		
videoType	整数值 1:MPEG-2 2:MPEG-4 3:H.264 4:WMV9 5:VC-1 6:AVS	Y	视频编码类型
streamType	整数值 1:PS 2:TS 3:MP4 4:ASF	Y	流类型
drmType	整数值 1:DRM_TYPE_CLEAR_T EXT 2:DRM_TYPE_BESTDRM 3:DRM_TYPE_NDS 4:DRM_TYPE_MICROSO FT 5:DRM_WIDEVINE	Y	DRM 类型
fingerPrint	整数值 0:开启 fingerPrint 1:关闭 fingerPrint(缺省值)	N	是否支持水印保护
copyProtection	整数值 0:PROTECTION_NO(缺省 值) 1:PROTECTION_MACRO VISION 3:PROTECTION_CGMSA	N	防拷贝类型
allowTrickmode	0:允许 Trickmode 1:不允许 Trickmode(缺省 值)	N	表示该媒体是否允许任何 trickmode 操作(包 括快进/快退/暂停)
startTime	参见 RFC2326 中的 Normal Play Time (NPT)和 Absolute Time (Clock Time)两种时间类型的格式	Y	表示媒体的起始时间
endTime	参见 RFC2326 中的 Normal Play Time (NPT)和	Y	表示媒体的结束时间

	Absolute Time (Clock Time)两种时间类型的格式		
timeShiftURL	字符串，媒体 URL	O	媒体的时移地址，以 RTSP 协议开始
timeShift	0 或不存在：不支持时移； 其他：支持时移	O	媒体是否支持时移（单播时移/组播时移）
entryID	媒体加入 Playlist 时，在媒体列表中的唯一标识。	N	只在加入 Playlist 时用到

- 终端支持流媒体内容的逻辑播放，需注意以下几点：
 - 终端在发送 PLAY 消息时的 Range 字段应与 mediastream 中的 startTime 及 endTime 保持一致；
 - 当终端播放到 endTime 定义的播放时间时，平台侧应发起 ANNOUNCE 2101 命令，通知终端播放到达逻辑结束，此时终端应发送 TEARDOWN 消息关闭连接；
 - 终端再播放逻辑分段时，进行快进/快退等 TrickMode 操作时，在 PLAY 命令的 Range 字段中应明确表明操作的时间范围，例如（播放范围为 5s — 30s）：

```
PLAY rtsp://example.com/foo/bar/baz.mpg RTSP/1.0
CSeq: 5
Session: 719885386
Scale: 4.0
Range: npt=15-30

.....

PLAY rtsp://example.com/foo/bar/baz.mpg RTSP/1.0
CSeq: 10
Session: 719885386
Scale: -8.0
Range: npt=25-5
```

- 对于 mediastream 字段的 startTime 及 endTime 字段，分为以下两种场景进行处理：
 - 当节目需要整段播放时，startTime 与 endTime 取值相同，即为 NPT 格式时
 $\text{startTime} = \text{endTime} = 0$ ，UTC 格式时 $\text{startTime} = \text{endTime} = 197000T000000Z$ ；
 - 当节目需要分段逻辑播放时，startTime 与 endTime 取值不同，用于表示逻辑分段节目的开始与结束时间，且 $\text{startTime} < \text{endTime}$ 。
- 当播放到 startTime 时，终端抛出 EVENT_MEDIA_BEGINNING 事件，表示逻辑

媒体段播放到起始点。当播放到 `endTime` 时，终端抛出 `EVENT_MEDIA_END` 事件，表示逻辑媒体段播放结束。

- **Media** 对象可以采用 JSON 格式进行封装，一个包含有两个 **media** 对象的数组示例如下（【注】本示例只做参考用）：

```
[{mediaURL:"rtsp://servername:554/video1.ts",mediaCode:"code1",mediaType:2,audioType:1,videoType:1,streamType:1,drmType:1,fingerPrint:0,copyProtection:1,allowTrickmode:1,startTime:0,endTime:0,entryID:"entry1"},
{mediaURL:"rtsp://servername:554/video2.ts",mediaCode:"code1",mediaType:2,audioType:1,videoType:1,streamType:1,drmType:1,fingerPrint:0,copyProtection:1,allowTrickmode:1,startTime:200.2,endTime:400.3,entryID:"entry2"}]
```

6.6.6.2.4 媒体流的生命周期

EPG 页面中创建的 **MediaPlayer** 对象，控制终端上某个媒体流或播放器实例，且该播放器实例的生命周期允许跨页面存在，也就是说在一张页面上创建的对应该终端本地的一个播放器实例 **MediaPlayer** 对象，可以通过 **Session**、**Cookie**、**URL** 参数传递等方式在下一张页面中获得，并在下一个页面中继续操纵这个本地的一个播放器实例。

为支持以上功能，EPG 页面可以通过 **MediaPlayer** 的 `nativePlayerInstanceId` 属性来标识终端上的播放器实例。在设置 **MediaPlayer** 的 `nativePlayerInstanceId` 属性时，若终端上已存在一个相同 ID 的媒体播放实例，则 **MediaPlayer** 对象与它进行关联，并通过扩展脚本对这个终端上的媒体播放实例进行控制。

对于 **MediaPlayer** 对象生命周期的控制，示例如下（【注】本示例只做参考用）：

- 在第一张页面中,创建和初始化媒体流

```
//创建一个JS扩展的MediaPlayer对象，同时生成本地
var mp = new MediaPlayer();
//播放实例的标识
//读取本地的媒体播放实例的标识。保留这个标识以便在跨页面时，标识这个本地的媒体播放实例
var nativePlayerInstanceId = mp.getNativePlayerInstanceId();
//设置媒体播放器播放媒体内容
mp.setSingleMedia(mediaStr);
mp.playFromStart();//开始播放
```

- 在下一张页面中，根据 ID 获取已存在的媒体流

```

//创建一个JS 扩展的 MediaPlayer 对象
var mp= new MediaPlayer();
//根据从上一张页面传递过来的Id, 绑定本地的媒体播放实例
mp.bindNativePlayerInstance(nativePlayerInstanceId);
//设置这个媒体播放实例的参数
mp.setxxx(xxx);
//fastforward 播放
mp.fastforward(2);
//停止播放。
mp.stop();

```

6.6.6.2.5 相关的事件

MediaPlayer 对应的本地的播放器实例将一些需要 EPG 页面处理的状态，采用虚拟按键的方式抛出，虚拟按键定义为 0x0300，EPG 页面的 javascript 逻辑通过 onkeypress 函数进行响应。虚拟事件通过 Utility 对象的 getEvent 函数获取。

当发生虚拟事件 EVENT_MEDIA_ERROR 时，错误代码定义如下（待定）：

- RTSP_CONNECT_FAILED 连接 RTSP 服务器失败。可能是网线没有插好或者其他网络问题造成的。
- RTSP_SEND_FAILED RTSP 请求不能被发送到服务器。
- RTSP_READ_FAILED 无法读取从服务器发来的 RTSP 命令。
- RTSP_CONNECTION_DIED RTSP 连接断开。
- RTSP_STATUS_LINE 注释：该代码不再使用。
- RTSP_STATUS_PLAYING 视频正在播放。在播放时这个消息大约 20 秒收到一次
- RTSP_STATUS_ANNOUNCE RTSP 服务器发出的消息。这类消息中只有 END_OF_STREAM 不是错误消息。
- RTSP_STATUS_END_OF_STREAM 当前媒体资源文件播放结束。
- RTSP_STATUS_NOT_FOUND 服务器上找不到指定的资源文件。
- STATUS_NO_VIDEO 被请求的服务器不存在或者资源文件不存在。
- IGMP_STATUS_END_OF_STREAM IGMP 流到达了结尾，或者丢失了 IGMP 连接。
- RTSP_CONNECT_STOPPED 服务器关闭了 RTSP 连接。
- RTSP_STATUS_AUTHENTICATE 请求的媒体资源需要认证，并且认证信息没有传递到播放器，所以不能播放。

- RTSP_STATUS_REDIRECTED 请求被重定向。需要进一步的动作才能完成请求，例如需要访问另一台服务器。
- IGMP_STATUS_PLAYINGIGMP 流在播放当中。
- UDP_STATUS_END_OF_STREAMUDP 流播放结束。
- 17-UDP_STATUS_PLAYINGUDP 流正在播放。
- MP3S_STATUS_END_OF_STREAMMP3 流播放结束。
- AUD_DESCRIP_READY 这个回调表示当前资源文件的描述信息已经成功读取。
- RTSP_STATUS_START_OF_STREAMRTSP 流开始。
- MP3S_STATUS_START_OF_STREAMMP3 流开始播放。
- RTSP_STATUS_ANNOUNCE_PARTIAL 返回一个部分资源文件播放完毕的通知事件，发生在收到服务器的 END-OF-STREAMANNOUNCNE 消息时。在这种情况下是可能继续播放媒体流的，并不是停止播放。
- STATUS_PMT_CHANGED 接收到的传送流 PMT 信息改变。
- PVRSTATUS_PLAYING 正在播放的是 PVR 片段。
- PVRSTATUS_END_OF_STREAM 当前 PVR 片段播放结束。
- RTSP_STATUS_SET_PARAMETER 收到服务器发出的 SET_PARAMETER 消息。
- RTSP_STATUS_INVALID_RANGE 收到服务器 457InvalidRange 的消息回应。该消息不同于 RTSP_STATUS_END_OF_STREAM 和 RTSP_STATUS_START_OF_STREAM,这两个事件由 RTSP 的 Announce 信令产生，服务器端会终止流的发送。而 457 的应答虽然表示客户端指定了错误的范围，但服务端不会终止流的发送。

6.6.6.3 Navigation

浏览器提供了缺省的浏览方式：使用方向键移动焦点到可交互的页面元素，使用确认键激活页面元素。Navigation 对象提供了扩展的浏览方式：屏蔽浏览器缺省的浏览方式，页面逻辑使用 Navigation 对象直接操纵焦点移动和激活的过程。

Navigation 对象屏蔽浏览器缺省浏览方式的效果，只在创建 Navigation 对象的页面内有效，当离开这张页面后，浏览器恢复到缺省的浏览方式。

表 49 Navigation 对象说明

类别	方法/属性	说明
----	-------	----

Method	disableDefaultNavigation()	关闭缺省的浏览方式。浏览器通过 onkeypress 事件，将方向键传递给页面，由页面决定如何移动焦点
	enableDefaultNavigation()	恢复缺省的浏览方式。浏览器根据方向键，决定如何移动焦点
【注】：原有Navigation对象的其他方法均可由标准JavaScript及CSS实现，因此在此处不再继续要求。		

6.6.6.4 Utility

Utility 对象用于封装需浏览器整体支持、执行的属性和方法。

表 50 Utility 对象说明

类别	方法/属性	说明
Method	setBrowserWindowAlpha(alpha)	浏览器窗口（针对所有页面）的透明度，设置之后立即生效 ● Alpha: 0—100之间的整数值，0表示不透明，100表示完全透明
	getEvent()	获得虚拟按键事件的详细信息 - 返回值：封装 event 对象的 JSON 字符串
	startLocalCfg()	启动本地配置界面，用于进行网络、帐户、密码等内容的配置

6.6.6.5 Event

IPTV 终端在产生内部事件时，使用虚拟按键的方式传递该事件消息，同时产生这个事件的详细信息。页面 Javascript 逻辑可通过 Utiliy.getEvent()方法获取。

6.6.6.5.1 对象的结构

表 51 EVENT 对象的结构

属性名	属性类型	是否必选	注释
type	String	Y	Event 事件类型
instance_id	Int	Y	终端本地的播放器实例 ID
其他字段			不同 Event 事件定义不同

6.6.6.5.2 对象的类型

表 52 对象的类型

事件类型 (EVENT_TYPE)	注释
EVENT_GO_CHANNEL	当按下 CH+,CH-、数字键、页面链接等，终端将打开下一个频道时触发
EVENT_MEDIA_END	当媒体播放器中的媒体播放到末端时触发
EVENT_MEDIA_BEGINING	当媒体播放器中的媒体播放到起始端时触发
EVENT_MEDIA_ERROR	当媒体播放器，发生异常时触发
EVENT_PLAYMODE_CHANGE	当媒体播放器的 playbackmode 发生改变的时候触发
EVENT_REMINDER	当终端定时提醒时触发，
EVENT_JVM_CLIENT	当增值业务客户端产生下载、启动、退出、出错等状态发生时触发

6.6.6.5.3 对象消息

● EVENT_GO_CHANNEL 事件的定义

表 53 EVENT_GO_CHANNEL 事件

属性名	属性类型	是否必选	注释
type	String	Y	对于此事件 Type 的值为: "EVENT_GO_CHANNEL"
instance_id	Int	Y	终端本地的播放器实例 ID
channel_code	String	Y	频道的统一标识
channel_num	Int	N	频道号

● EVENT_MEDIA_END 事件的定义

表 54 EVENT_MEDIA_END 事件

属性名	属性类型	是否必选	注释
type	String	Y	对于此事件 Type 的值为: "EVENT_MEDIA_END"
instance_id	Int	Y	终端本地的播放器实例 ID
media_code	String	N	播放到末端的媒体的统一标识
entry_id	String	N	播放到末端的媒体，在播放列表中的唯一标识

● EVENT_MEDIA_BEGINING 事件的定义

表 55 EVENT_MEDIA_BEGINING 事件

属性名	属性类型	是否必选	注释
type	String	Y	对于此事件 Type 的值为: "EVENT_MEDIA_BEGINING"
instance_id	Int	Y	终端本地的播放器实例 ID
media_code	String	N	播放到起始端的媒体的统一标识
entry_id	String	N	播放到起始端的媒体, 在播放列表中的唯一标识

● EVENT_MEDIA_ERROR 事件的定义

表 56 EVENT_MEDIA_ERROR 事件

属性名	属性类型	是否必选	注释
type	String	Y	对于此事件 Type 的值为: "EVENT_MEDIA_ERROR"
instance_id	Int	Y	终端本地的播放器实例 ID
error_code	Int	Y	错误代码
error_message	String	N	错误详细信息
media_code	String	N	发生错误的媒体的统一标识

● EVENT_PLAYMODE_CHANGE 事件的定义

表 57 EVENT_PLAYMODE_CHANGE 事件

属性名	属性类型	是否必选	注释
type	String	Y	对于此事件 Type 的值为: "EVENT_PLAYMODE_CHANGE"
instance_id	Int	Y	终端本地的播放器实例 ID
new_play_mode	Int	Y	播放模式变化后的播放状态。参考: 播放状态 (PLAY_MODE) 的定义
new_play_rate	Num	N	当播放模式为 TRICK_MODE 时, 必须提供播放速率的信息
old_play_mode	Int	Y	播放模式变化前的播放状态。参考: 播放状态 (PLAY_MODE) 的定义
old_play_rate	Num	N	当播放模式为 TRICK_MODE 时, 必须提供播放速率的信息

上表中涉及的对于 PLAY_MODE 的定义如下:

表 58 PLAY_MODE 播放状态

播放状态值	PLAYMODE	注释
1	PAUSE	暂停状态
2	NORMAL_PLAY	正常播放状态

3	TRICK_MODE	正常播放之外，以指定播放速率快进，快退，慢进，慢退
4	MULTICAST_CHANNEL_PLAY	组播频道直播状态
5	UNICAST_CHANEL_PLAY	单播频道直播状态

以数值来表示播放的速率。其中整数表示正向播放，负数表示反向播放。示例如下：

表 59 PLAY_RATE 播放速率

播放速率值	注释
32	32 倍速快进
16	16 倍速快进
8	8 倍速快进
4	4 倍速率快进
2	2 倍速率快进
1	1 倍速正常播放
0	暂停状态
-2	2 倍速快退
-4	4 倍速快退
-8	8 倍速快退
-16	16 倍速快退
-32	32 倍速快退

● EVENT_REMINDER 事件的定义

表 60 EVENT_REMINDER 事件

属性名	属性类型	是否必选	注释
Type	String	Y	对于此事件 Type 的值为："EVENT_REMINDER"
Message	String	Y	定时提醒的消息

● EVENT_JVM_CLIENT 事件的定义

表 61 EVENT_JVM_CLIENT 事件

属性名	属性类型	是否必选	注释
Type	String	Y	对于此事件 Type 的值为："EVENT_JVM_CLIENT"
event_code	Int	Y	事件代码
event_result	Int	Y	事件操作结果
event_message	String	N	事件详细信息

上表中涉及的对于 event_code 的定义如下：

表 62 event_code 事件类型

事件状态值	event_code	注释
0	DOWNLOAD_START	开始下载
1	DOWNLOAD_END	下载结束
2	PLAY_START	开始运行
3	PLAY_END	退出应用
4	DOWNLOAD_ERROR	下载错误
5	PLAY_ERROR	运行错误
其他		待定

上表中涉及的对于 event_result 的定义如下：

表 63 event_result 事件操作结果

事件结果值	event_result	注释
0	SUCCESS	成功
1	FAIL	失败
其他		待定

6.6.6.5.4 封装方式

采用 JSON 格式来封装 EVENT 详细信息。示例如下：

表 64 EVENT 详细信息示例

属性名	属性类型	是否必选	属性值
Type	String	Y	"EVENT_MEDIA_END"
instance_id	Int	Y	1
media_code	String	N	"vod/2102"
entry_id	String	N	"12"

封装为如下 JSON 格式的字符串：

```
{type:"EVENT_MEDIA_END",instance_id:1,media_code:"vod/2102",entry_id:"12"}
```

6.6.6.5.5 事件时序图

● 点播 EVENT 时序图

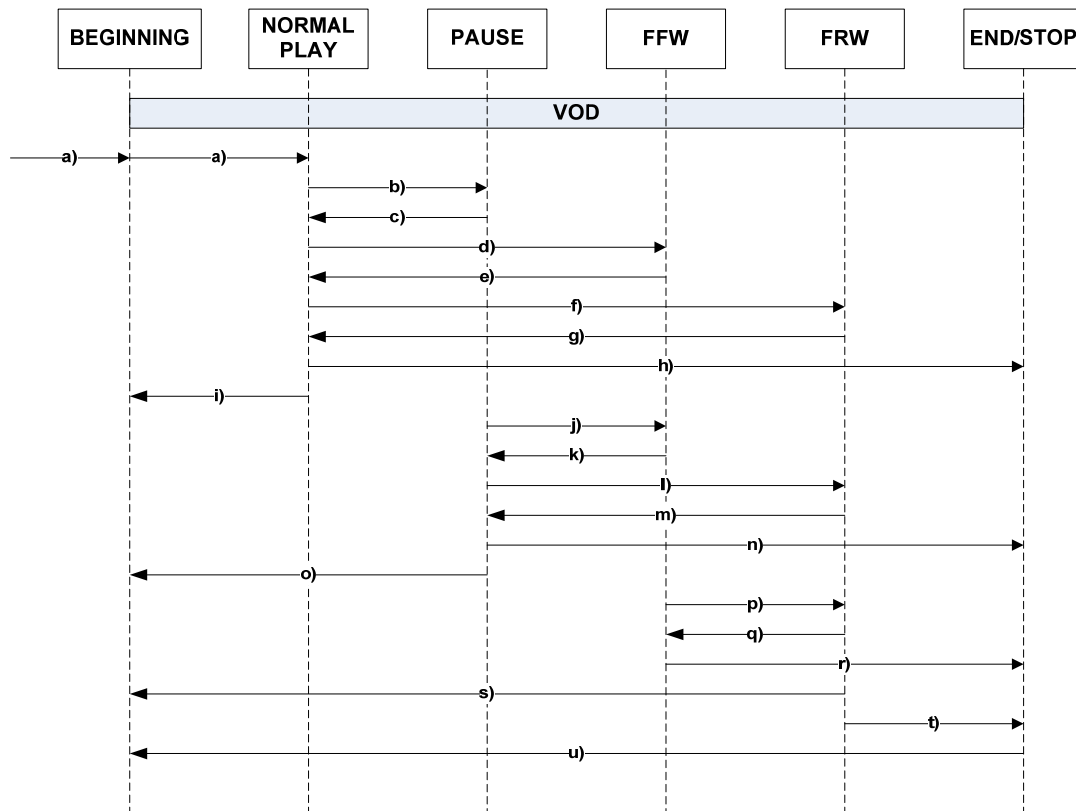


图 1 点播 EVENT 时序图

时序图中各状态切换情况如下：

a). 开始播放：

```
{type:"EVENT_MEDIA_BEGINING",instance_id:1,media_code:"vod/2102",entry_id:"12"};
```

b). 正常播放→→→暂停：

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:1,new_play_rate:0,old_play_mode:2,old_play_rate:1};
```

c). 正常播放←←←暂停：

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,new_play_rate:1,old_play_mode:1,old_play_rate:0};
```

d). 正常播放→→→快进：

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_play_rate:2,old_play_mode:2,old_play_rate:1}
```

e). 正常播放←←←快进：

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,new_pl
```

ay_rate:1,old_play_mode:3,old_play_rate:2}

f). 正常播放→→→快退:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_pl
ay_rate:-2,old_play_mode:2,old_play_rate:1}

g). 正常播放←←←快退:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,new_pl
ay_rate:1,old_play_mode:3,old_play_rate:-2}

h). 正常播放→→→停止/文件尾:

{type:"EVENT_MEDIA_END",instance_id:1,media_code:"vod/2102",entry_id:"12
"}

i). 正常播放→→→开始:

{type:"EVENT_MEDIA_BEGINING",instance_id:1,media_code:"vod/2102",entry
_id:"12"}

j). 暂停→→→快进:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_pl
ay_rate:2,old_play_mode:1,old_play_rate:0}

k). 暂停←←←快进:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:1,new_pl
ay_rate:0,old_play_mode:3,old_play_rate:2}

l). 暂停→→→快退:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_pl
ay_rate:-2,old_play_mode:1,old_play_rate:0}

m). 暂停←←←快进:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:1,new_pl
ay_rate:0,old_play_mode:3,old_play_rate:-2}

n). 暂停→→→停止:

{type:"EVENT_MEDIA_END",instance_id:1,media_code:"vod/2102",entry_id:"12
"}

o). 暂停→→→文件头:

1. {type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,ne

w_play_rate:1,old_play_mode:1,old_play_rate:0}

2. {type:"EVENT_MEDIA_BEGINING",instance_id:1,media_code:"vod/2102",entry_id:"12"}

p). 快进→→→快退:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_play_rate:-2,old_play_mode:3,old_play_rate:2}

q). 快进←←←快退:

{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_play_rate:2,old_play_mode:3,old_play_rate:-2}

r). 快进→→→停止/文件尾:

{type:"EVENT_MEDIA_END",instance_id:1,media_code:"vod/2102",entry_id:"12"}

s). 快进→→→文件头:

1. {type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,new_play_rate:1,old_play_mode:3,old_play_rate:2}
2. {type:"EVENT_MEDIA_BEGINING",instance_id:1,media_code:"vod/2102",entry_id:"12"}

t). 快退→→→停止/文件尾:

{type:"EVENT_MEDIA_END",instance_id:1,media_code:"vod/2102",entry_id:"12"}

u). 快退→→→文件头:

1. {type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,new_play_rate:1,old_play_mode:3,old_play_rate:-2}
2. {type:"EVENT_MEDIA_BEGINING",instance_id:1,media_code:"vod/2102",entry_id:"12"}

● 直播 EVENT 时序图

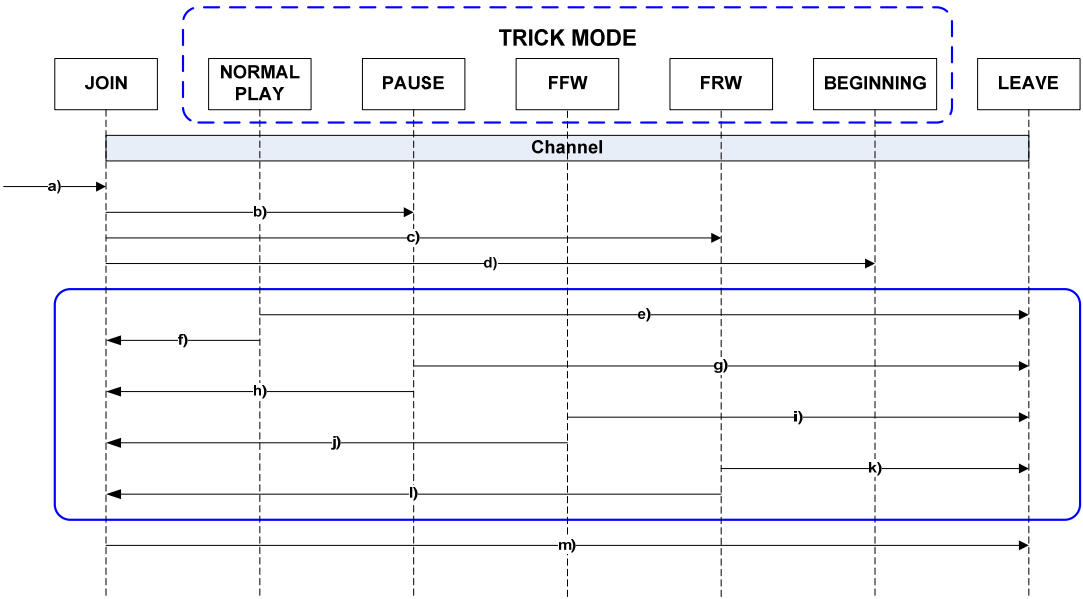


图 2 直播（以组播为例）EVENT 时序图

时序图中各状态切换情况如下（以组播为例）：

- a). 加入频道：
`{type:"EVENT_GO_CHANNEL",instance_id:1,channel_code:"433",
channel_num:1};`
 - b). 直播→→→暂停（TRICK MODE）：
`{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:1,new_pl
ay_rate:0,old_play_mode:4,old_play_rate:1}`
 - c). 直播→→→快退（TRICK MODE）：
`{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:3,new_pl
ay_rate:-2,old_play_mode:4,old_play_rate:1}`
 - d). 直播→→→录制左边界（TRICK MODE）：
`{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:2,new_pl
ay_rate:1,old_play_mode:4,old_play_rate:1}`
- 【注】以下描述的状态切换均为 *TRICK MODE* 状态下的切换，*TRICK MODE* 状态下的暂停、快进/快退等不涉及到加入/退出组播的操作请参考“D.5.5.1 点播 *EVENT* 时序图”中的定义。）
- e). 正常播放（TRICK MODE）→→→LeaveChannel：
`{type:"EVENT_MEDIA_END",instance_id:1}`
 - f). 正常播放（TRICK MODE）→→→一键追直播（JoinChannel）：


```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:4,new_play_rate:1,old_play_mode:2,old_play_rate:1}
```

【注】 无论是否关闭 RTSP 时移连接，均不发送 EVENT_MEDIA_END 消息

g). 暂停（TRICK MODE）→→→LeaveChannel:

```
{type:"EVENT_MEDIA_END",instance_id:1}
```

h). 暂停（TRICK MODE）→→→一键追直播（JoinChannel）:

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:4,new_play_rate:1,old_play_mode:1,old_play_rate:0}
```

i). 快进（TRICK MODE）→→→LeaveChannel:

```
{type:"EVENT_MEDIA_END",instance_id:1}
```

j). 快进（TRICK MODE）→→→一键追直播（JoinChannel）:

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:4,new_play_rate:1,old_play_mode:3,old_play_rate:2}
```

k). 快退（TRICK MODE）→→→LeaveChannel:

```
{type:"EVENT_MEDIA_END",instance_id:1}
```

l). 快退（TRICK MODE）→→→一键追直播（JoinChannel）:

```
{type:"EVENT_PLAYMODE_CHANGE",instance_id:1,new_play_mode:4,new_play_rate:1,old_play_mode:3,old_play_rate:-2}
```

m). 直播状态→→→LeaveChannel:

```
{type:"EVENT_MEDIA_END",instance_id:1}
```

● 音乐播放 EVENT 时序图

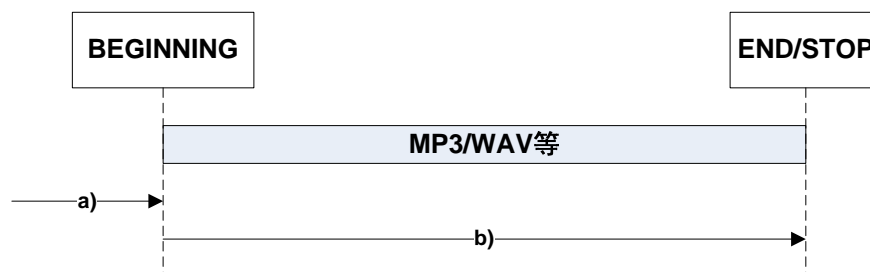


图 3 音乐播放 EVENT 时序图

时序图中各状态切换情况如下:

a). 开始播放:

```
{type:"EVENT_MEDIA_BEGINING",instance_id:1,media_code:"Music/MP3_001
```

”,entry_id:”33”};

b). 正常播放→→→停止/文件尾:

{type:”EVENT_MEDIA_END”,instance_id:1,media_code:”Music/MP3_001”,entry_id:”33”}

6.7 HTML 标签的使用

6.7.1 基本属性

以下列表中列出的 HTML 标签的基本属性，以下列出的所有属性被所有的标签所支持。

表 65 HTML 标准属性

属性	值	描述
class	<i>classname</i>	规定元素的类名（用于规定样式表中的类）
contextmenu	<i>menu_id</i>	规定元素的上下文菜单
hidden	hidden	规定该元素是无关的。被隐藏的元素不会显示
id	<i>id</i>	规定元素的唯一 ID
style	<i>style_definition</i>	规定元素的行内样式

6.7.2 基本事件

在本章节中定义了 HTML 标签所应支持的基本属性，在以下章节中，如无特殊说明，则表示该 HTML 标签应支持相应的基本事件。

6.7.2.1 窗口事件 (Window Events)

【说明】

本事件仅在 body 和 frameset 元素中有效。

表 66 窗口事件

属性	值	描述
onload	脚本	当文档载入时执行脚本
onunload	脚本	当文档卸载时执行脚本

6.7.2.2 表单元素事件 (Form Element Events)

【说明】

本事件仅在表单元素中有效。

表 67 表单元素事件

属性	值	描述
onsubmit	脚本	当表单被提交时执行脚本
onreset	脚本	当表单被重置时执行脚本
onblur	脚本	当元素失去焦点时执行脚本
onfocus	脚本	当元素获得焦点时执行脚本

6.7.2.3 键盘事件 (Keyboard Events)

【说明】

本事件在下列元素中无效: base, br, frame, frameset, head, html, iframe, meta, param, script, style,元素。

表 68 键盘事件

属性	值	描述
onkeypress	脚本	当键盘被按下后又松开时执行脚本
onblur	脚本	当元素失去焦点时执行脚本
onfocus	脚本	当元素获得焦点时执行脚本
【备注】: onkeypress 方法仅对能够点击的标签对象有效		

6.7.3 HTML 4 标签列表

6.7.3.1 <!--...-->

【说明】

注释标签用来在源文档中插入注释。注释会被浏览器忽略。可使用注释对代码进行解释，这样做有助于在以后的时间对代码的编辑。

也可以在注释内容存储针对程序所定制的信息。在这种情况下，这些信息对用户是不可见的，但是对程序来说是可用的。一个好的习惯是把注释或样式元素放入注释文本中，这样就可避免不支持脚本或样式的老式浏览器把它们显示为纯文本。

6. 7. 3. 2 <!DOCTYPE>

【说明】

<!DOCTYPE> 声明位于文档中的最前面的位置，处于 <html> 标签之前。此标签可告知浏览器文档使用哪种 HTML 或 XHTML 规范。该标签可声明三种 DTD 类型，分别表示严格版本、过渡版本以及基于框架的 HTML 文档。

以下面这个 <!DOCTYPE> 标签为例：

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

在上面的声明中，声明了文档的根元素是 html，它在公共标识符被定义为 "-//W3C//DTD XHTML 1.0 Strict//EN" 的 DTD 中进行了定义。浏览器将明白如何寻找匹配此公共标识符的 DTD。如果找不到，浏览器将使用公共标识符后面的 URL 作为寻找 DTD 的位置。

【注】<!DOCTYPE> 标签没有结束标签！

6. 7. 3. 3 <A>

【说明】

<a> 标签可定义锚。一个锚有两种用法：

- 通过使用 href 属性，创建一个到另外一个文档的链接
- 通过使用 name 或 id 属性，创建一个文档内部的书签

表 69 <A>标签说明

类别	方法/属性	说明
Property	href	链接的目标 URL。浏览器对于 href 属性只能执行一次
Event	onClick	点击对象时触发事件

6. 7. 3. 4 <AREA>

【说明】

定义图像映射中的某个区域。

【注释】

此元素永远被嵌套在 <map> 标签内部。 中的 usemap 属性可引用 <map>

中的 id 或 name 属性（由浏览器决定），所以需要同时向 <map> 添加 id 和 name 两个属性。

表 70 <AREA>标签说明

类别	方法/属性	说明
Property	shape	设置或获取图像映射中超级链接区域的形状(方形区域)，其可选值为： <ul style="list-style-type: none">● default● rect● circ● poly
	coords	设置或获取图像映射中超级链接区域的坐标
	href	设置或获取目标 URL 或锚点
Event	onClick	点击对象时触发事件

【备注】: <area> 标签的 coords 属性

<area> 标签的 coords 属性定义了客户端图像映射中对鼠标敏感的区域坐标。坐标的数字及其含义取决于 shape 属性中决定的区域形状。可以将客户端图像映射中的超链接区域定义为矩形、圆形或多边形等。下面列出了每种形状的适当值：

● 圆形: *shape="circle", coords="x,y,z"*

这里的 x 和 y 定义了圆心的位置（"0,0" 是图像左上角的坐标），r 是以像素为单位的圆形半径。

● 多边形: *shape="polygon", coords="x1,y1,x2,y2,x3,y3,..."*

每一对 "x,y" 坐标都定义了一个多边形的一个顶点（"0,0" 是图像左上角的坐标）。定义三角形至少需要三组坐标；高维多边形则需要更多数量的顶点。多边形会自动封闭，因此在列表的结尾不需要重复第一个坐标来闭合整个区域。

● 矩形: *shape="rectangle", coords="x1,y1,x2,y2"*

第一个坐标是矩形的一个角的顶点坐标，另一对坐标是对角的顶点坐标，"0,0" 是图像左上角的坐标。请注意，定义矩形实际上是定义带有四个顶点的多边形的一种简化方法。例如，下面的 XHTML 片段在一个 100x100 像素图像的右下方四分之一处，定义了一个对鼠标敏感的区域，并在图像的正中间定义了一个圆形区域。

```
<map name="map">
  <area shape="rect" coords="75,75,99,99" nohref="nohref">
```

```
<area shape="circ" coords="50,50,25" nohref="nohref">
</map>
```

【注释】

如果某个 **area** 标签中的坐标和其他区域发生了重叠,会优先采用最先出现的 **area** 标签。浏览器会忽略超过图像边界范围之外的坐标。

6.7.3.5 <BASE>

【说明】

<base> 标签为页面上的所有链接规定默认地址或默认目标。通常情况下,浏览器会从当前文档的 URL 中提取相应的元素来填写相对 URL 中的空白。使用 **<base>** 标签可以改变这一点。浏览器随后将不再使用当前文档的 URL,而使用指定的基本 URL 来解析所有的相对 URL。这其中包括 **<a>**、****、**<link>**、**<form>** 标签中的 URL。

【注释】

<base> 标签必须位于 **head** 元素内部。

表 71 <base>标签说明

类别	方法/属性	说明
Property	Href	规定页面中所有相对链接的基准 URL。

6.7.3.6 <BODY>

【说明】

body 元素定义文档的主体。它包含文档的所有内容(比如文本、图像、颜色和图形等等。)

表 72 <BODY>标签说明

类别	方法/属性	说明
Property	Bgcolor	设置或获取对象后面的背景颜色,浏览器可支持但不赞成使用,请用 CSS 取代
	background	设置或获取页面上文本和图像后平铺的背景图片,浏览器可支持但不赞成使用,请用 CSS 取代

6.7.3.7

【说明】:

 可插入一个简单的换行符。
 标签是空标签（意味着它没有结束标签，因此这是错误的：
</br>）。

【注释】:

请使用
 来输入空行，而不是分割段落。

6.7.3.8 <DIV>

【说明】

<div> 可定义文档中的分区或节（division/section），用于把文档分割为独立的、不同的部分。它可以用作严格的组织工具，并且不使用任何格式与其关联。

<div> 作为一个块级元素，意味着它的内容自动地开始一个新行。实际上，换行是 <div> 固有的唯一格式表现。可以通过 <div> 的 id 应用额外的样式。

【注释】

浏览器通常会在 div 元素前后放置一个换行符，且浏览器统一对无内容的 DIV 隐藏。使用 <div> 元素来组合块级元素，就可以使用样式对它们进行格式化。

6.7.3.9 <FORM>

【说明】

form 元素可为用户输入创建表单。表单可包含文本域，复选框，单选按钮等等。表单用于向指定的 URL 传递用户数据。

表 73 <FORM>标签说明

类别	方法/属性	说明
Property	Action	设置或获取表单内容要发送处理的 URL
	Method	用于向 action URL 传送数据的 HTTP 方法。 默认为 get。
	enctype	enctype 属性取 mimetype 型值，用于对 form 的内容进行编码 <ul style="list-style-type: none">• application/x-www-form-urlencoded• text/plain

Event	Onsubmit	设置表单内容，并提交表单到服务器端处理
	Onreset	重置表单内容，并触发事件
【备注】：Method 属性的说明 <ul style="list-style-type: none">method="get"：此方法可传送 URL 中的表单内容： URL?name=value&name=value。但是，如果表单值包含非 ASCII 字符或者超过 100 个字符，则必须使用 method="post"。method="post"：此方法可传送 request 主体中的表单内容。		

6.7.3.10 <FRAME>

【说明】

在 <frameset> 中定义一个框架。

【注释】

在 HTML 中，<frame> 标签没有结束标签。且不能与 <frameset></frameset> 标签一起使用 <body></body> 标签。

表 74 <FRAME>标签说明

类别	方法/属性	说明
Property	Name	设置或获取框架的名称
	Src	设置或获取要由 frame 装入的 URL
	frameborder	定义浏览器边框显示 0（缺省值）、1
	marginwidht	定义页面与窗体间的左右距离
	marginheight	定义 frame 页面相对窗体顶部和底部的距离

6.7.3.11 <FRAMESET>

【说明】

frameset 元素可定义一个框架集。它被用来组织多个窗口（框架）。每个框架存有独立的文档。在其最简单的应用中，frameset 元素仅仅会规定在框架集中存在多少列或多少行。必须使用 cols 或 rows 属性。

【注释】

不能与 <frameset></frameset> 标签一起使用 <body></body> 标签。

表 75 <FRAMESET>标签说明

类别	方法/属性	说明
Property	Rows	使用象素值，横向定义 frame 的高度，只适用 pixels
	Cols	定义竖排（纵向 frame）的个数和大小，只适用 pixels

6. 7. 3. 12 <HEAD>

【说明】

head 元素可包含关于文档的信息。浏览器不会向用户显示这些头部信息（"head information"）。下面这些标签可用在 head 部分：<link>, <meta>, <script>, <style>, 以及 <title>。

6. 7. 3. 13 <HTML>

【说明】

此元素可告知浏览器其自身是一个 HTML 文档。

6. 7. 3. 14 <IFRAME>

【说明】

iframe 元素会创建一个包含另外一个文档的内联框架。

表 76 <IFRAME>标签说明


类别	方法/属性	说明
Property	Name	设置 iframe 的名称
	Width	采用象素值的方式设置 iframe 的宽度 <i>pixels</i>
	Height	用象素值的形式定义 iframe 高度 <i>pixels</i>
	Src	设置或获取要由 iframe 装入的 URL
	Frameborder	设置框架边框是否显示：0（缺省值）、1
	Marginwidth	设置或获取显示框架中文本之前的左右边距宽度 <i>pixels</i>
	Marginheight	设置或获取显示框架中文本之前的上下边距高度 <i>pixels</i>

6. 7. 3. 15

【说明】

img 元素可定义一副图像。

表 77 标签说明

类别	方法/属性	说明
Property	src	设置或获取要由  装入的图像的 URL
	usemap	设置或获取用于客户端图像映射的 URL
	width	设置图像的宽度
	height	设置图像的高度
	border	设置或获取绘制图片周围边框的宽度,取值必须为“0”

6.7.3.16 <INPUT>

【说明】

<input> 标签可定义输入域的开始,在其中用户可输入数据。对于大量通常的表单控件,可以使用 <input> 标签来进行定义,其中包括文本字段、多选列表、可单击的图像和提交按钮等。

虽然 <input> 标签中有许多属性,但是对每个元素来说,只有 type 属性和 name 属性是必需的(提交或重置按钮只有 type 属性)。

表 78 <INPUT>标签说明

类别	方法/属性	说明
Property	type	定义文本输入框类型的表单,默认值是 "text"。 <i>所有可选值为: button、checkbox、hidden、password、radio、reset、submit、text</i>
	name	定义用于此 input 元素的唯一名称。 <i>注: 此属性必需与 type="button"、type="checkbox"、type="hidden"、type="password"、type="text" 以及 type="radio" 一同使用。</i>
	value	对于按钮、重置按钮和确认按钮: 定义按钮上的文本。 对于复选框和单选按钮: 定义 input 元素被点击时的结果。 对于隐藏域、密码域以及文本域: 定义元素的默认值。 <i>注: 与 type="checkbox" 和 type="radio" 一同使用时,此元素是必需的。</i>
	maxlength	定义在某个文本域中所允许的最大字符数。 <i>注: 仅可与 type="text" 配合使用。</i>
	checked	创建复选框控件默认已选择

	inputmode	定义预期的输入类型。(<i>inputmode</i> 类型)
	list	引用 datalist 元素。如果定义，则一个下拉列表可用于向输入字段插入值。 <i>id of a datalist</i>
	readonly	指示此域的值不能被修改。 <i>注：仅可与 type="text" 配合使用。</i>
	size	定义 input 元素的尺寸。 <i>注：不能与 type="hidden" 一同使用。</i>
	disabled	当 input 元素首次加载时禁用此元素，这样用户就无法在其中写文本，或选定它。 <i>注：不能与 type="hidden" 一同使用。</i>
Event	onSelect	选择表单内容则触发事件 <i>注：仅可与 type="radio" 、 type="checkbox"配合使用。</i>
	onChange	改变表单内容则触发事件
	onClick	点击对象时触发事件

6. 7. 3. 17

【说明】

 标签定义了列表项目。 标签可用在有序列表() 和 unordered 列表() 中。

6. 7. 3. 18 <LINK>

【说明】

此元素定义了当前文档与 Web 集合中其他文档的关系。

【注释】

link 元素是空元素，它仅包含属性。且此元素只能存在于 head 部分，不过它可出现任何次数。

表 79 <LINK> 标签说明

类别	方法/属性	说明
Property	href	设置或获取目标 URL 或锚点
	type	设置或获取 link 的 MIME 类型，目前仅支持 text/css、text/javascript

6. 7. 3. 19 <MAP>

【说明】

定义一个客户端图像映射。图像映射（image-map）指带有可点击区域的一幅图像。

【注释】

<area> 元素永远嵌套在 map 元素内部。area 元素可定义图像映射中的区域。 中的 usemap 属性可引用 <map> 中的 id 属性，所以应同时向 <map> 添加 id 和 name 属性。

6. 7. 3. 20 <MARQUEE>

【说明】

创建一个滚动的文本字幕。微软 HTML 扩展,仅中兴支持。所有 marquee 均需要添加 nobr 标签。

【注释】

MARQUEE 元素的默认宽度与其父元素的宽度相等。如果 MARQUEE 位于没有指定宽度的 TD 内，就需要明确设置 MARQUEE 的宽度。如果 MARQUEE 和 TD 的宽度都没有指定，那么滚动字幕就将限定于 1 个像素宽。

要创建垂直滚动的字幕，请将其 scrollLeft 属性设定为 0。要创建水平滚动的字幕，请将其 scrollTop 属性设定为 0，这将覆盖任何脚本设置。

scrollLeft 和 scrollTop 属性当字幕滚动时为只读。当不处于滚动状态时，scrollLeft 对于设置为水平滚动的字幕来说为可读写，scrollTop 对于设置为垂直滚动的字幕来说为可读写。

表 80 <MARQUEE>标签说明

类别	方法/属性	说明
Property	align	表示元素的垂直对齐方式，值可以是 top, middle, bottom，默认为 middle
	bgcolor	表示运动区域的背景色，值是 16 进制的 RGB 颜色，默认为白色
	direction	表示滚动的方向，值可以是 left, right, up, down，默认为 left
	height	表示运动区域的宽度，值是正整数（单位

		是像素)或百分数,默认 width=100%
	width	表示运动区域的高度,值是正整数(单位是像素)或百分数,默认 height 为标签内元素的高度
	scrollamount	表示运动速度,值是正整数,默认为 6
	scrolldelay	表示停顿时间,值是正整数,默认为 0,单位是毫秒
	behavior	表示滚动的方式,值可以是 scroll(连续滚动) slide(滑动一次) alternate(来回滚动)
	loop	表示循环的次数,值是正整数,默认为无限循环
【注】:以标清为例,在默认状况下字母滚动全屏宽度 640px 的时间应该为 $640px/(6*10) = 10.6s$ 。高清分辨率下以此类推。		

6.7.3.21 <META>

【说明】

<meta> 元素可提供有关页面的元信息(meta-information),比如针对搜索引擎和更新频度的描述和关键词。<meta> 标签位于文档的头部,不包含任何内容。<meta> 标签的属性定义了与文档相关联的名称/值对。

【注释】

<meta> 标签永远位于 head 元素内部。元数据总是以名称/值的形式被成对传递的。

表 81 <META>标签说明

类别	方法/属性	说明
Property	content	设置或获取与 HTTP-EQUIV 或 NAME 关联的资源信息
	http-equiv	设置或获取用于将 META 标签的 content 绑定到 HTTP 响应头的信息,其取值为: content-type、expires、refresh、set-cookie
	name	设置或获取 meta 对象的 CONTENT 标签属性中指定的值: Author、description、keywords、generator、revised、others

6. 7. 3. 22 <OBJECT>

【说明】

定义一个嵌入的对象。

【注释】

Object 元素可位于在 head 元素或 body 元素内部。

表 82 <OBJECT>标签说明

类别	方法/属性	说明
Property	classid	Object的注册类ID，对应不同类型的应用形式。目前只使用“clsid：72E6F181-D1B0-4C22-B0D7-4A0740EEAEF5”用于表示J2ME应用
	width	设置对象的宽度
	height	设置对象的高度
【注】：在IPTV中，Object目前仅用于JVM的调用。		

6. 7. 3. 23 <OPTGROUP>

【说明】

定义选项组。此元素允许组合选项。当使用一个长的选项列表时，对相关的选项进行组合会使处理更加容易。

表 83 <OPTGROUP>标签说明

类别	方法/属性	说明
Property	disable	设置使得列表项处于 disabled 状态
	label	这里 label 用来对 option 进行分类

6. 7. 3. 24 <OPTION>

【说明】

option 元素可定义下拉列表中的一个选项（一个条目）。浏览器将 <option> 标签中的内容作为 <select> 标签的菜单或是滚动列表中的一个元素显示。

【注释】

<option> 标签可以在不帶有任何属性的情况下使用，但是通常需要使用 value 属性，

此属性会指示出被送往服务器的内容。请与 **select** 元素配合使用此标签，否则这个标签是没有意义的。

表 84 <OPTION>标签说明

类别	方法/属性	说明
Property	selected	设置列表框中的选项是已被选中的
	value	设置或获取当表单控件提交时返回给服务器的值
	label	定义当使用 <optgroup> 时所使用的标注
	disable	设置使得列表项处于 disabled 状态

6. 7. 3. 25 <PARAM>

【说明】

param 标签可为包含它的 <object>标签提供参数。

表 85 <PARAM>标签说明

类别	方法/属性	说明
Property	name	设置或获取元素的输入参数的名称
	value	设置或获取元素的输入参数值

6. 7. 3. 26 <SCRIPT>

【说明】

定义一段诸如 JavaScript 的脚本。

【注释】

假如此元素内部的代码没有位于某个函数中，那么这些代码会在页面被加载时被立即执行。<frameset> 标签之后的脚本会被忽略。

表 86 <SCRIPT>标签说明

类别	方法/属性	说明
Property	type	设置或获取用于解码对象的字符集，目前仅为 text/javascript
	language	设置或获取关联的脚本引擎的 MIME 类型，目前仅为 javascript

	src	设置或获取当前脚本编写用的语言
--	-----	-----------------

6. 7. 3. 27 <SELECT>

【说明】

`select` 元素可创建单选或多选菜单。当提交表单时，浏览器会提交选定的项目，或者收集用逗号分隔的多个选项，将其合成一个单独的参数列表，并且在将 `<select>` 表单数据提交给服务器时包括 `name` 属性。

【注释】

请在 `form` 元素中使用此标签来接受用户的输入。

表 87 <SELECT>标签说明

类别	方法/属性	说明
Property	name	设置或获取对象的名称
	disabled	设置列表框的状态为 <code>disabled</code>
	autofocus	在页面加载时使这个 <code>select</code> 字段获得焦点。 <ul style="list-style-type: none">● <code>true</code>● <code>false</code>

6. 7. 3. 28

【说明】

`` 标签被用来组合文档中的行内元素。

【注释】

请使用 `` 来组合行内元素，以便通过样式来格式化它们。`span` 没有固定的格式表现。当对它应用样式时，它才会产生视觉上的变化。

6. 7. 3. 29 <STYLE>

【说明】

`style` 元素位于 `head` 部分中。如果希望在页面中包含样式表，应当在外部定义样式表，并使用 `link` 标签来链接它。

表 88 <STYLE>标签说明

类别	方法/属性	说明
Property	type	定义内容类型，当前取值仅支持为 text/css

6. 7. 3. 30 <SUB>&<SUP>

【说明】

<sub> 标签可定义下标文本。<sup> 可定义上标文本。

6. 7. 3. 31 <TABLE>

【说明】

<table> 标签可定义表格。在 <table> 标签内部，可以放置表格的标题、表格行、表格列、表格单元以及其他的表格。

6. 7. 3. 32 <TD>

【说明】

定义表格中的一个单元格。

表 89 <TD>标签说明

类别	方法/属性	说明
Property	rowspan	设置或获取单元格要跨越表格的多少行
	colspan	设置或获取单元格要跨越表格的多少列

6. 7. 3. 33 <TEXTAREA>

【说明】

定义一个文本区域 (text-area) (一个多行的文本输入区域)。用户可在此文本区域中写文本。在一个文本区中，可输入无限数量的文本。文本区中的默认字体是等宽字体 (fixed pitch)。

【注释】

在文本输入区内的文本行间，用 "%OD%OA" (回车/换行) 进行分隔。可以通过 <textarea> 标签的 wrap 属性设置文本输入区内的换行模式。

表 90 <TEXTAREA>标签说明

类别	方法/属性	说明
Property	cols	设置或获取对象的宽度
	rows	设置或获取对象中包含的水平行数
	readonly	表明对象内容是为只读的值
	disabled	设置或获取控件的状态
	inputmode	定义该 <code>textarea</code> 所期望的输入类型
	name	为此文本区规定的一个名称
Event	onClick()	点击对象触发事件
	onChange()	改变 <code>textarea</code> 内容则触发事件

6. 7. 3. 34 <TITLE>

【说明】：此元素可定义文档的标题。

6. 7. 3. 35 <TR>

【说明】：在表格中定义一行。

6. 7. 3. 36

【说明】： 标签定义了无序列表。

6. 7. 4 HTML 5 标签列表

6. 7. 4. 1 <CANVAS>

【说明】：

<canvas> 标签定义图形，比如图表和其他图像。

表 91 <CANVAS>标签说明

类别	方法/属性	说明
Property	height	设置 <code>canvas</code> 的高度(像素)
	width	设置 <code>canvas</code> 的宽度(像素)

6.7.4.2 <DATALIST>

【说明】

<datalist> 标签定义选项列表。与 input 元素配合使用该元素，来定义 input 可能的值。datalist 及其选项不会被显示出来，它仅仅是合法的输入值列表。可使用 input 元素的 list 属性来绑定 datalist。

6.7.4.3 <KEYGEN>

【说明】

<keygen> 标签定义生成密钥。

表 92 <KEYGEN>标签说明

类别	方法/属性	说明
Property	autofocus	在页面加载时使输入字段获得焦点。 <i>注释：不能与 type="hidden" 一同使用。</i>
	challenge	如果使用，则将 keygen 的值设置为在提交时询问。
	disabled	在 input 元素首次载入时禁用该元素，这样用户就无法在其中输入或选取文本。 <i>注释：不能与 type="hidden" 一同使用。</i>
	form	定义一个或多个输入字段所属的表单。
	keytype	定义 keytype。rsa 生成 RSA 密钥。
	name	定义 input 元素的唯一的名称。name 属性用于在提交时搜集字段的值。

6.8 CSS 的使用

6.8.1 尺寸单位

虽然在 W3C 的定义中，CSS 的尺寸定义支持百分比、英寸、磅、像素等多种单位，但是对于 IPTV 业务来说，为保证各终端浏览器在业务显示上的一致性，要求在 IPTV EPG 页面制作中统一采用“像素 (电视机屏幕上的一个点)”作为 CSS 的唯一尺寸单位。

6.8.2 颜色单位

虽然在 W3C 的定义中，允许 CSS 采用多种颜色单位，但是对于 IPTV 业务来说，为保

证各终端浏览器在业务显示上的一致性，要求在 IPTV EPG 页面制作中统一采用以下单位进行颜色的配置单位。

表 93 CSS 颜色单位说明

单位	描述
rgb(x,x,x)	RGB 值 (比如 rgb(255,0,0))
#rrggbb	十六进制数 (比如 #ff0000)

6.8.3 CSS 2 属性

6.8.3.1 Background

【说明】

CSS 允许应用纯色作为背景，也允许使用背景图像创建相当复杂的效果。

表 94 Background 元素说明

属性	说明	取值
Background	复合属性。设置或检索对象的背景特性	background-color background-image background-repeat background-position
background-color	设置或检索对象的背景颜色	color-rgb color-hex transparent
background-image	设置或检索对象的背景图像	url none
background-position	设置或检索对象的背景图像位置	xpos ypos
background-repeat	设置或检索对象的背景图像是否及如何铺排	repeat (默认) no-repeat repeat-x repeat-y

6.8.3.2 Border

【说明】

元素的边框 (border) 是围绕元素内容和内边距的一条或多条线。CSS border 属性允

许规定元素边框的样式、宽度和颜色。

表 95 Border 元素说明

属性	说明	取值
border	简写属性。作用是在一个声明中用来设置四个边框的所有属性。	border-width border-style border-color
border-bottom	简写属性。作用是在一个声明中用来设置下边框的所有属性。	border-bottom-width border-style border-color
border-bottom-color	设置或检索对象底边颜色	同 border-color
border-bottom-style	设置或检索对象底边样式	同 border-style
border-bottom-width	设置或检索对象底边宽度	thin (2px) medium (4px) thick (6px) <i>length(必须支持)</i>
border-color	设置四个边框的颜色，可以设置一到四个颜色。	Color <i>注：主要颜色值都要支持 至少支持黑色#000000</i>
border-left	简写属性。用于在一个声明中设置左边框的所有属性。	border-left-width border-style border-color
border-left-color	设置或检索对象左边颜色	同 border-color
border-left-style	设置或检索对象左边样式	同 border-style
border-left-width	设置或检索对象左边宽度	thin (2px) medium (4px) thick (6px) <i>length(必须支持)</i>
border-right	简写属性。将所有用于右边框的属性设置于一个声明中。	border-right-width border-style border-color
border-right-color	设置或检索对象右边颜色	同 border-color
border-right-style	设置或检索对象右边样式	同 border-style
border-right-width	设置或检索对象右边宽度	thin (2px) medium (4px) thick (6px) <i>length(必须支持)</i>
border-style	设置四个边框的样式，可以设置一到四个样式。	none solid
border-top	简写属性。将所有用于上边框的属性设置于一个声明中。	border-top-width border-style border-color

border-top-color	设置或检索对象顶边颜色	同 border-color
border-top-style	设置或检索对象顶边样式	同 border-style
border-top-width	设置或检索对象顶边宽度	thin (2px) medium (4px) thick (6px) <i>length(必须支持)</i>
border-width	简写属性。在一个声明中设置四个边框的宽度,可以设置一到四个值。	thin (2px) medium (4px) thick (6px) <i>length(必须支持)</i>

6.8.3.3 Dimension

【说明】

CSS 尺寸 (Dimension) 属性允许控制元素的高度和宽度。同样, 它允许增加行间距。

表 96 Dimension 元素说明

属性	说明	取值
width	设置元素的宽度	auto (默认 0px) <i>length</i>
height	设置元素高度	auto (默认 0px) <i>length</i>
line-height	设置元素高度	normal (2px) number <i>length</i>
max-height	设置元素的最大高度。	<i>length</i>
max-width	设置元素的最大宽度。	<i>length</i>
min-height	设置元素的最小高度。	<i>length</i>
min-width	设置元素的最小宽度。	<i>length</i>

6.8.3.4 Font

【说明】

设置字体属性是样式表的最常见用途之一。CSS 字体属性允许设置字体的大小、字体风格(如斜体)和字体变形(如小型大写字母)。

表 97 Font 元素说明

属性	说明	取值
----	----	----

font	简写属性。作用是将所有针对字体的属性设置在一个声明中	font-style font-weight(仅支持 nomal 及 bold) font-size/line-height(px) font-family（只支持黑体）
font-size	设置字体的尺寸	xx-small x-small small medium large x-large xx-large 注：建议尽量使用以 px 为基础的字号
font-family	字体类型名称或者针对某元素的类属族名名称的优先列表	黑体
font-style	设置字体样式	normal italic
font-weight	设置字体的粗细	normal（根据黑体字体设置） bold

6.8.3.5 List

【说明】

CSS 列表属性允许放置、改变列表项标志，或者将图像作为列表项标志。

表 98 List 元素说明

属性	说明	取值
list-style	复合属性。设置列表项目相关内容	list-style-type list-style-position list-style-image
list-style-image	设置或检索作为对象的列表项标记的图像	none url
list-style-type	设置或检索对象的列表项所使用的预设标记	none disc circle square decimal lower-roman upper-roman lower-alpha

		upper-alpha
list-style-type	设置列表中列表项标记被放置的位置	inside outside

6.8.3.6 Margin

【说明】

围绕在元素边框的空白区域是外边距。设置外边距会在元素外创建额外的“空白”。设置外边距的最简单的方法就是使用 `margin` 属性，这个属性接受任何长度单位，甚至负值。

表 99 Margin 元素说明

属性	说明	取值
margin	简写属性。在一个声明中设置外边距属性。	margin-top margin-right margin-bottom margin-left
margin-top	检索或设置对象顶边的外延边距	auto(默认 2px) <i>length</i>
margin-right	检索或设置对象右边的外延边距	auto (默认 2px) <i>length</i>
margin-bottom	检索或设置对象底边的外延边距	auto (默认 2px) <i>length</i>
margin-left	检索或设置对象左边的外延边距	auto (默认 2px) <i>length</i>

6.8.3.7 Padding

【说明】

元素的内边距在边框和内容区之间。控制该区域最简单的属性是 `padding` 属性。CSS `padding` 属性定义元素边框与元素内容之间的空白区域。

表 100 Padding 元素说明

属性	说明	取值
padding	检索或设置对象四边的外延边距	padding-top padding-right padding-bottom padding-left
padding-top	检索或设置对象顶边的外延边距	<i>length</i>

padding-right	检索或设置对象右边的外延边距	<i>length</i>
padding-bottom	检索或设置对象底边的外延边距	<i>length</i>
padding-left	检索或设置对象左边的外延边距	<i>length</i>

6.8.3.8 Position

【说明】

CSS 为定位提供了一些属性，利用这些属性，可以建立列式布局，将布局的一部分与另一部分重叠。

表 101 Position 元素说明

属性	说明	取值
position	把元素放置到一个静态的、相对的、绝对的、或固定的位置中。	static relative absolute fixed
top	检索或设置对象与其最近一个定位的父对象右边相关的位置	auto (默认 0px) <i>length</i>
left	检索或设置对象与其最近一个定位的父对象顶部相关的位置	auto (默认 0px) <i>length</i>
right	检索或设置对象与其最近一个定位的父对象右边相关的位置	auto (默认 0px) <i>length</i>
bottom	设置定位元素下外边距边界与其包含块下边界之间的偏移。	auto (默认 0px) <i>length</i>
vertical-align	设置元素的垂直排列	top middle bottom
clip	设置元素的形状。	shape auto
overflow	设置当元素的内容溢出其区域时发生的事情。	visible hidden scroll auto

6.8.3.9 Pseudo-Class

【说明】

CSS 伪类用于向某些选择器添加特殊的效果。

表 102 Pseudo-Class 元素说明

属性	说明
:active	将样式添加到被激活的元素。
:focus	将样式添加到被选中的元素。

6.8.3.10 Text

【说明】

CSS 文本属性可定义文本的外观。通过文本属性，您可以改变文本的颜色、字符间距，对齐文本，装饰文本，对文本进行缩进等等。

表 103 Text 元素说明

属性	说明	取值
color	设置文本颜色	color
direction	设置文本方向（可以配合 unicode-bidi 使用）	ltr rtl
text-align	对齐元素中的文本	left right center justify
text-decoration	向文本添加修饰	none underline overline line-through blink
text-transform	控制元素中的字母	none capitalize uppercase lowercase
word-spacing	设置字间距	normal <i>length</i>
unicode-bidi	设置文本方向	normal embed bidi-override

6.8.4 CSS 3 属性

6.8.4.1 属性列表

6.8.4.1.1 Background

表 104 Background 元素说明

属性	说明	取值
background-linear-gradient	背景颜色线性渐变填充	[[[[top bottom] [left right]] <angle>],]? <color-stop>[, <color-stop>]+
background- radial-gradient	背景颜色放射性渐变填充	[<bg-position>],)? [[<shape> <size>] [<length> <percentage>]{2}],]? <color-stop>[, <color-stop>]+

6.8.4.1.2 Border

表 105 Border 元素说明

属性	说明	取值
border-radius	边框导角定义	[<length> <percentage>]{1,4} [/ [<length> <percentage>]{1,4}]?
border-*-*-radius	四导角分别定义: top-left top-right bottom-left bottom-right	[<length> <%>] [<length> <%>]?
box-shadow	边框阴影	h-shadow v-shadow blur spread

		color inset
--	--	----------------

6.8.4.1.3 Text

表 106 Text 元素说明

属性	说明	取值
text-shadow	文本阴影	none color length

6.8.4.2 CSS 变换

通过使用 CSS 变换，可以实现页面元素样式在指定时间区间中平滑过渡的动画效果。

以下列表中给出了能够进行 CSS 变换的属性：

表 107 支持变换的 CSS 属性列表

属性名称	值类型
background-color	color<(r,g,b) #rrgbb>
background-position	percentage,length
border-bottom-color	color
border-bottom-width	percentage,length
border-color	color
border-left-color	color
border-left-width	length
border-right-color	color
border-right-width	length
border-spacing	length
border-top-color	color
border-top-width	length
border-width	length
bottom	percentage,length
color	color
font-size	length,precentage
font-weight	number
height	length,precentage
left	length,precentage
letter-spacing	length
line-height	number, length,precentage
margin-bottom	length
margin-left	length

margin-right	length
margin-top	length
max-height	length,percentage
max-width	length,percentage
min-height	length,percentage
min-width	length,percentage
opacity	number
padding-bottom	length
padding-left	length
padding-right	length
padding-top	length
right	length,percentage
text-shadow	shadow
top	length,percentage
visibility	visibility
width	length,percentage
word-spacing	length,percentage
zoom	number

能够进行 CSS 变换的属性属性详细说明如下：

表 108 CSS 变换的属性

属性	说明	取值
transition	复合属性。设置或检索对象的变化特性	[<'transition-property'> <'transition-duration'> <'transition-timingfunction'> <'transitiondelay'> [, [<'transitionproperty'> <'transitionduration'> <'transitiontiming- function'> <'transition-delay'>]]*
transition-delay	变换延迟执行时间。默认值为 0	<time> [, <time>]*
transition-duration	变换执行时间	<time> [, <time>]*
transition-property	变换操作的 css 属性	none all [<IDENT>] [';' <IDENT>]*
transition-timing-function	变换中时间值与属性值的之间 计算方式，可实现线性、加速、 减速等变换效果。	ease linear ease-in ease-out ease-in-out cubic-bezier(<number>, <number>, <number>, <number>) [, ease linear ease-in ease-out

		ease-inout cubicbezier(< number>, <number>, <number>, <number>)]*
--	--	---

● **transition-property 属性**

该属性指定执行变换的 CSS 属性，具体属性参照本章以上的说明。

● **transition-timing-function 属性**

该属性值申明新、旧值之间的中间值随着时间的变化如何计算得到。即允许变换过程中改变运动速度。常见的动画效果为加速减速或淡入淡出。

表 109 transition-timing-function 属性值

函数名	描述
cubic-bezier	立方贝塞尔曲线定义.
ease	等同于 cubic-bezier(0.25, 0.1, 0.25, 1.0)
linear	等同于 cubic-bezier(0.0, 0.0, 1.0, 1.0)
ease-in	等同于 cubic-bezier(0.42, 0, 1.0, 1.0)
ease-in-out	等同于 cubic-bezier (0.42, 0, 0.58, 1.0)

该属性指定为使用立方贝塞尔曲线公式，该曲线由四个控制点来定义。P₀ 和 P₃ 代表起始位置和结束位置，固定值 (0, 0) 和 (1, 1)。 P₁ 和 P₂ 的位置则是由本属性值来定义的。

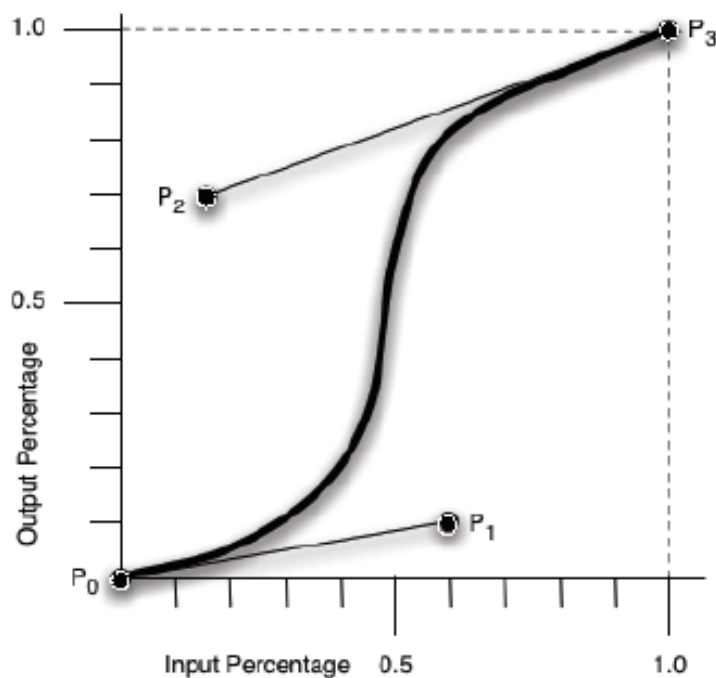


图 4 立方贝塞尔曲线图形

- **transition-duration** 属性

该属性定义变换执行的时长，如果指定值为‘0’，表示立即完成旧值到新值的变换（即没有动画效果）。所有负数值设置按照‘0’值处理。

- **transition-delay** 属性

本属性定义变换延迟执行的时间。默认值为‘0’，表示当 **transition-property** 中指定的 CSS 属性值改变时变换立即执行。如果取值为负数值，则表示当属性值改变时变换立即执行，且 CSS 属性变换开始值即为该负数值根据 **transition-timing-function** 计算所得值。

- **transition** 属性

表示以上四个属性的缩写。数据格式：

```
[<'transition-property'> ||
<'transition-duration'> ||
<'transition-timingfunction'>
|| <'transitiondelay'>
[, [<'transitionproperty'>
|| <'transitionduration'>
|| <'transitiontiming-
function'> ||
<'transition-delay'>]]*
```

6.9 注意事项

针对 IPTV EPG 制作过程中经常会出现的问题、影响到浏览器执行效率的情况及相关建议，将在本章节中进行简单的说明。

● Frame 的使用

在页面制作中涉及到多个 FRAME 的嵌套时，建议不要超过 3 层，且不能循环嵌套（A 嵌套 B，B 嵌套 A），页面 FRAME 的总个数不要超过 8 个（包括嵌套的 FRAME）。

为尽量避免预留给 FRAME 的大小跟实际包含的页面内容大小不同，而造成滚动的情况。在设定子 FRAME 的尺寸的时候，应考虑字体大小对实际显示尺寸的影响。如果预先设置的尺寸小于实际计算的尺寸，会造成子 FRAME 的内部滚动。

由于当用户由一个有多 Frame 的页面跳转到一个无多个 FRAME 的页面，并跳转回原来存在多 FRAME 的页面时，浏览器可能会在某个 FRAME 中显示整个页面的内容，因此建议在页面跳转过程中通过“top.window.location”来进行页面跳转。

● <MAP>标签的使用

禁止多个 IMG 使用同一个 MAP，也不应该在一个页面不能出现同名的多个 MAP，且 MAP 中不能嵌套 AREA 元素。

● <IFRAME>标签的使用

Iframe 所在页面的尺寸不应因 Iframe 的存在而超出一屏的显示，从而导致 Iframe 页面的滚动。

以标清为例，iframe 的大小只要不超过 640*530 就可以，但必须设置将边框设置为 frameborder="0"，滚动方式 scrolling="no"。Iframe 内部页面的尺寸不应超出 Iframe 尺寸，从而导致 Iframe 内部页面的滚动，或视频显示不全。

7 浏览器要求

7.1 分辨率

浏览器对于 IPTV 业务的分辨率支持应区分标清及高清两种：

- 标清业务来说，终端浏览器在电视机的 PAL 制下应支持 644×534 的分辨率，浏览器对于 EPG 页面显示分辨率为 640×530；
- 对于高清业务来说，终端浏览器在采用高清输入时应至少支持 1280×720 的分辨率。

对于标清及高清业务来说，浏览器应该能够应用【附录 A】中给出的 CSS 对于页面整体显示布局进行设置。同时，终端浏览器在分辨率确定的情况下进行页面整体或页面元素的显示过程中符合以下原则：

- 若页面未设置 `bgcolor` 属性的，则浏览器底色默认显示为黑色；
- 若页面将 `body` 中的 `leftmargin`、`topmargin`、`marginwidth`、`marginheight` 等元素设置为 0，则浏览器将紧贴屏幕上方及左侧显示页面；
- 若页面上的元素采用 CSS 的 `absolute` 进行 `style` 定义时，浏览器应从页面大小 640×530 或 1280×720 分辨率下的左上角 (0, 0) 作为元素显示位置计算的起点；
- 若 EPG 页面元素未采用 CSS 的 `absolute` 进行 `style` 定义时，浏览器则基于基本 CSS 再计算实际的位置作为起点进行页面元素的编排；
- 尽管许多屏幕分辨率被调成 4:3，但是有些电视可能宽或窄一些。当为支持这些分辨率调整时，终端会在边缘加入一些额外的空，使用黑色填充；对于标清页面来说，若屏幕分辨率为 16: 9，那么终端在显示时会将画面置中，其余区域使用黑色填充。
- 终端浏览器在高清 EPG 和标清 EPG 之间应进行平滑切换，在 EPG 页面上 `head` 中增加 `meta` 通知浏览器当前页面的大小：
 - 高清 EPG 页面应填写 `<meta name="page-view-size" content="1280*720" />`
 - 标清 EPG 页面应填写 `<meta name="page-view-size" content="640*530" />`
 - 如果页面中未通过 `<META>` 标签说明页面大小，则浏览器默认将页面作为标清页面进行显示；

【注】： 为方便服务器对于支持不同分辨率的浏览器预先进行处理，浏览器应在所有 HTTP 请求的 `User-Agent` 中增加对于支持分辨率的说明。在原有的 `User-Agent` 字段的说明后添加半角分号“;”进行区分，并增加“`Resolution(PAL,720p)`”字段进行说明，其中多个分辨率之间采用英文半角逗号“,”进行分隔。

其示例代码如下：

```
Host: download.*****.de
Accept: */*
Pragma: no-cache
Cache-Control: no-cache
User-Agent: Mozilla/4.04[en](Win95;I;Nav);Resolution(PAL,720P,1080i)
Range: bytes=554554-
```

7.2 语言文字编码

终端浏览器对于语言文字的编码支持应与 EPG 制作中采用的编码一致。

7.3 字体字号

浏览器在显示 EPG 中的字体时，在 EPG 未通过 CSS 进行设置的情况下，应能够应用【附录 A】中给出的 CSS 对于页面整体显示布局进行设置。对于浏览器支持的字体字号其他要求如下：

- 浏览器只允许页面通过绝对大小 px、字号两种方式来设置页面字体大小，浏览器默认长度单位为像素（px），针对标清及高清页面来说，浏览器的最小字体大小均为 20px

【注】在 IPTV 中一个像素（px）被定义为，以 640X530（标清）或 1280x720（高清）为输出大小的屏幕上的构成影像的最小单位。

页面制作过程中通常可以通过 Font 标签属性或 CSS 进行字体属性的设置，在 CSS 未通过绝对大小设置字体时，浏览器对于各种设置方式对应的字体大小参见下表。

表 110 HTML 字体对照表

FONT SIZE 【Html】	4	5	6	7
字体大小	21px	30px	32px	36px
字间距	4px	6px	6px	6px
字高	20px	28px	29px	33px
行间距	6px	9px	11px	11px
【备注】：1-3 号字体大小小于 20px，因此在此不要求浏览器支持。				

表 111 ASCII 字符显示对照表

FONT SIZE 【Html】	4	5	6	7
最宽字符【大写 W】	19px	28px	30px	34px
最窄字符【小写 l】	3px	3px	3px	4px
最窄数字【1】	6px	8px	9px	10px
最宽数字【2~9，0】	11px	16px	17px	19px
【注 1】对于 ASCII 部分的规则，由于英文数字字符属于变宽，所以统计值上仅仅为				

字符灰度部分的取值。对于字符的占宽其实还包括内边距、提升度、降低度等参数，暂时无法反映在该表中。

【注 2】：1-3 号字体大小小于 20px，因此在此不要求浏览器支持。

除页面中一般文字的显示外，EPG 页面中一些特殊状态下的文字在浏览器中的显示状态如下：

- 文字链接、选中状态文字、已选中状态文字等根据【附录 A】提供的 CSS 进行设置；
- 页面中多个空格和回车显示为一个英文字符宽度空格，每一个“ ps”显示为一个英文字符宽度的空格。

7.4 图片格式

终端浏览器支持 BMP、GIF（包括动态 GIF）、JPG、PNG、APNG 格式的图片。且支持分辨率为 720×576、352×288、640×480、1280×720、1920×1080（其中 1280×720、1920×1080 等分辨率仅在高清业务所使用）的图形的缩放显示，图片上的每个象素至少支持 16 位彩色。

- 当标签中的图片地址不存在或由于图片过大导致下载过长时，直接显示页面背景色，则无论该图片是否设置边框 border 属性默认边框均设置为 0px；
- 当图片作为连接存在时，若页面没有设置默认的边框 border 属性则浏览器会将边框默认设置为 2px。

7.5 Cookie 的支持

浏览器对 cookie 的支持是符合 RFC 2109 规范的，其支持的属性/值对如下：

- Name = VALUE
- domine = DOMAIN_NAME
- path = PATH
- Max-Age = value
- expires = DATE

浏览器支持的 cookie 总数为 30 个，在同一个域下尽量避免使用超过 5 个 Cookie。若同一个于中设置 Cookie 超过 5 个或 Cookie 总数超过 30 个时，浏览器将自动删除最早的 cookie；

7.6 浏览器脚本的支持

浏览器在对于脚本的支持应与 EPG 制作要求相关章节保持一致，且浏览器一旦发现脚本运行错误，应立即停止运行该脚本。

7.7 HTML 标签的支持

浏览器在对于 HTML 标签的支持应与 EPG 制作要求相关章节保持一致，且浏览器一旦发现标签书写错误，则错误书写的标签则不再显示，但仍可继续解析之后的标签。并注意以下问题：

- **<DIV>元素的嵌套**

浏览器支持<div>标签中设置的 type 或样式，且对<div>中的标签也有效。浏览器对该属性的实现为透明到视频层字体浏览器需要支持<div>标签中设置的 style 或样式，能够对<div>中的标签也有效。

- **标签容错性及建议**

对于页面中出现的语法错误，浏览器会尽可能的实现页面语意，但不能保证完全达到页面设计人员的目的。页面中常见错误有如下类型：

- **标签交错：**两个标签的的起始和结束相互交错会造成解析上的错误并进而导致排版错误。（处理原则：如果中间出现标签未结束的时候，重新出现新的同一标签且标签头尾个数不匹配的情况，以首次出现的标签头及首次出现的标签尾为一个 Block）；
- **热链接嵌套**（处理原则同上条），且为提高效率，应避免使用空标签。

7.8 CSS 及 CSS 变换的支持

浏览器在对于 CSS 及 CSS 变换的支持应与 EPG 制作要求相关章节保持一致，且浏览器一旦发现 CSS 书写错误，则错误书写的 CSS 则不起作用，但仍可继续执行之后的 CSS。

7.8.1 盒子模型的尺寸

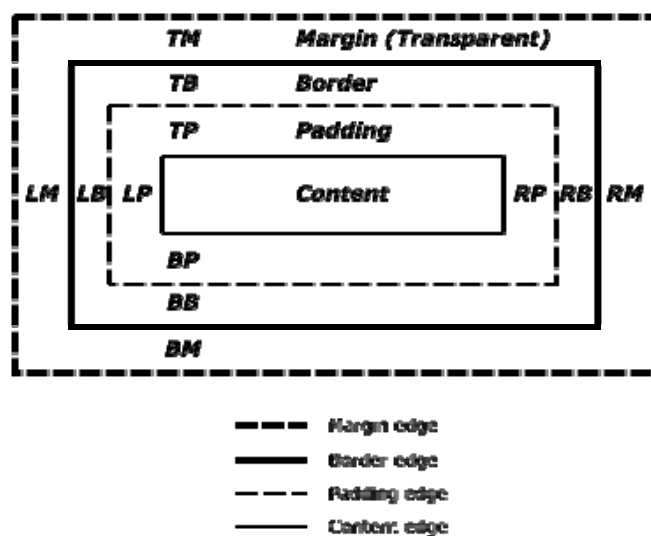


图5 盒子模型尺寸说明

如上图所示，盒模型包括内容区域（content，比如文本、图片等）以及内边距(padding)，边框(border)，外边距(margin)。

- CSS 属性 `background` 指定属性作用于 border, padding, content 区域，margin 区域背景始终做透明处理
- CSS 属性指定的 `width` 和 `height` 值代表 content 区域的宽和高
- padding, border, margin 的默认值为 '0'

7.8.2 盒子模型示例

```
<html>
<head>
  <title>examples of margins, padding, and borders</title>
  <style type="text/css">
    ul {
      background: yellow;
      margin: 12px 12px 12px 12px;
      padding: 3px 3px 3px 3px;
      /* no borders set */
    }
    li {
      color: white;                /* text color is white */
      background: blue;           /* content, padding will be blue */
      margin: 12px 12px 12px 12px;
      padding: 12px 0px 12px 12px; /* note 0px padding right */
      list-style: none            /* no glyphs before a list item */
                                   /* no borders set */
    }
    li.withborder {
      border-style: dashed;
      border-width: medium;       /* sets border width on all sides */
      border-color: lime;
    }
  </style>
</head>
<body>
  <ul>
    <li>first element of list
    <li class="withborder">second element of list is
      a bit longer to illustrate wrapping.
    </li>
  </ul>
</body>
</html>
```

以上代码渲染效果示例如下：

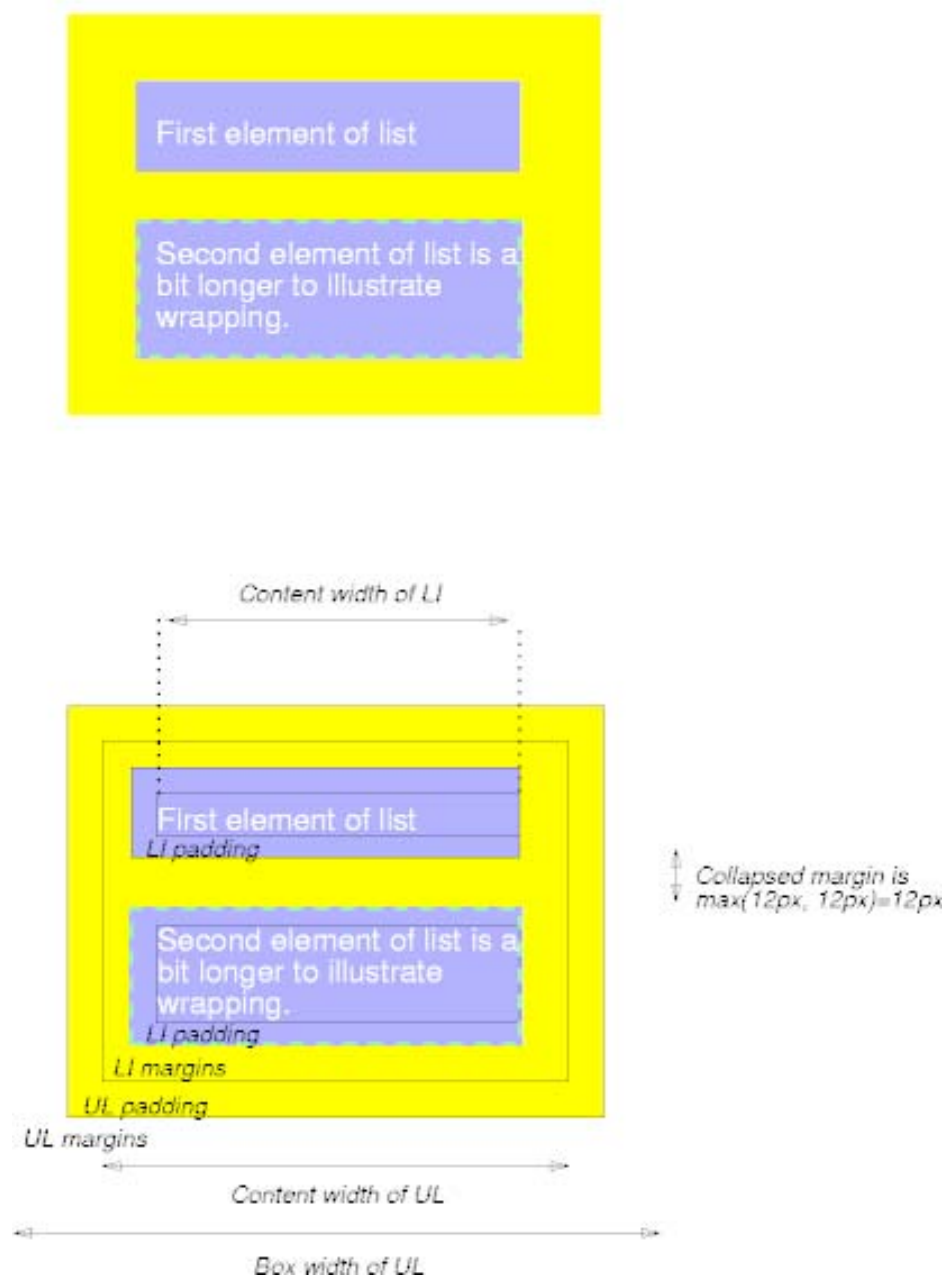


图 6 盒子模型示例代码渲染效果

7.8.3 外边距折叠

如上节图中标签 li 的渲染效果所示，2 个或以上盒模型之间（关系可以是相邻或嵌套）相邻的 `margin` 属性（这之间不能有非空内容、`padding` 区域、`border` 边框或使用清除分离方法）结合表示为一个单独的 `margin`。

- 水平 `margin` 不可折叠
- 垂直 `margin` 可折叠，除了一下例外：
 - 设置了 `display:inline-block` 的元素

- 相邻的盒模型中，其中的一个是浮动的（floated）
- 设置了 overflow 属性的元素和它的子元素之间的 margin
- 设置了绝对定位（position:absolute）的盒模型
- 根元素的垂直 margin
- margin 值计算方法如下：
 - 全部都为正值，取最大者；
 - 不全是正值，则都取绝对值，然后用正值减去最大值；
 - 没有正值，则都取绝对值，然后用 0 减去最大值。

【注】：相邻的盒模型可能由 DOM 元素动态产生并没有相邻或继承关系。

7.9 浏览器焦点处理原则

7.9.1 焦点显示规则

为了在不同浏览器中实现一致的焦点展现，对焦点的显示需要按以下规则实现：

根据“CSS 盒子模型”的描述，焦点区域大小由本身包含的内容区域的盒子决定，焦点盒子的 padding、margin、border 值均为 0，焦点的光标显示在此盒子外，默认样式为 4px、颜色值为 #ffff00 的实线条

7.9.2 焦点移动规则

在 IPTV 业务中，对于页面焦点的处理有两种方式：1) 页面通过 JavaScript 根据用户按键进行处理焦点移动；2) 浏览器根据用户按键及页面可聚焦元素键的排列关系自行处理。在本节中描述的是第 2 种方式。在浏览器根据用户的按键处理焦点移动时，应遵循以下原则：

- 只有在 EPG 页面中，除 base, br, frame, frameset, head, html, iframe, meta, param, script, style 元素外均可以被聚焦，且焦点框的颜色、粗细等可以通过 CSS 进行设置；
- 当 EPG 页面被加载完成后，如果未通过 JavaScript 设置某页面元素被聚焦，则浏览器应该自动将焦点放置在页面左上角第一个可以被聚焦的元素上；
- 当某个元素被聚焦时，用户按下【上】、【下】、【左】、【右】后，浏览器应该根据以下原则判断焦点移动的位置（假设以页面左下角为坐标轴 0，0 点）：

- 如果用户按下的是【左】、【右】时，浏览器应以当前被聚焦元素在画面中的左上角坐标为起点（X，Y），对比在该元素左侧或右侧的所有可聚焦元素，将所有元素在画面左上角的 Y 坐标与当前聚焦元素的 Y 坐标相比较，并将焦点聚焦至距离当前元素 Y 坐标最近的一个元素。如果存在多个可聚焦元素与当前元素的 Y 坐标差值一致的情况，则再计算各元素间的绝对距离，将焦点移动到绝对距离最小的一个元素上。如果存在多个元素的 Y 坐标差值及绝对距离均相等的情况下，则将焦点置于 Y 坐标最大的一个元素；
- 如果用户按下的是【上】、【下】时，浏览器应以当前被聚焦元素在画面中的左上角坐标为起点（X，Y），对比在该元素左侧或右侧的所有可聚焦元素，将所有元素在画面左上角的 X 坐标与当前聚焦元素的 X 坐标相比较，并将焦点聚焦至距离当前元素 X 坐标最近的一个元素。如果存在多个可聚焦元素与当前元素的 X 坐标差值一致的情况，则再计算各元素间的绝对距离，将焦点移动到绝对距离最小的一个元素上。如果存在多个元素的 X 坐标差值及绝对距离均相等的情况下，则将焦点置于 X 坐标最小的一个元素。

在下图中给出了一个示例用于说明终端处理焦点移动时的计算方法：



图 7 浏览器焦点处理示例

在上图的左侧为一个 EPG 页面，该页面上所有采用红框标出的元素为所有可聚焦的元素，采用黄框标识的元素表示浏览器当前聚焦的元素。在上图以用户按下【左】、【右】键为例，分别采用蓝色及紫色的虚线表示出浏览器根据用户操作时可能聚焦的元素（其中蓝色表示用户按下【左】键时，紫色表示用户按下【右】键时）。

根据之前说明的原则，当用户按下【左】键时由于路径 3 标识的元素的左上角 Y 坐标与当前聚焦元素最小，因此该元素将被聚焦。而当用户按下【右】键时路径 1 标识的左上角 Y 坐标与当前元素的差值明显大于路径 2 及路径 3 标识的元素，但路径 2 及路径 3 标识

的元素的 Y 坐标差值相同，且与当前元素的绝对距离也相同，在此情况下，应该将焦点放置在 Y 坐标更大的路径 2 标识的元素上。

【注】：对于用户按下【上】、【下】键时的聚焦方式，在此不再描述。

7.10 “多窗口”功能

浏览器应支持“多窗口”功能，即允许在业务需要时实现多个窗口的同时运行。支持“多窗口”功能的终端浏览器窗口间的关系如下图所示：

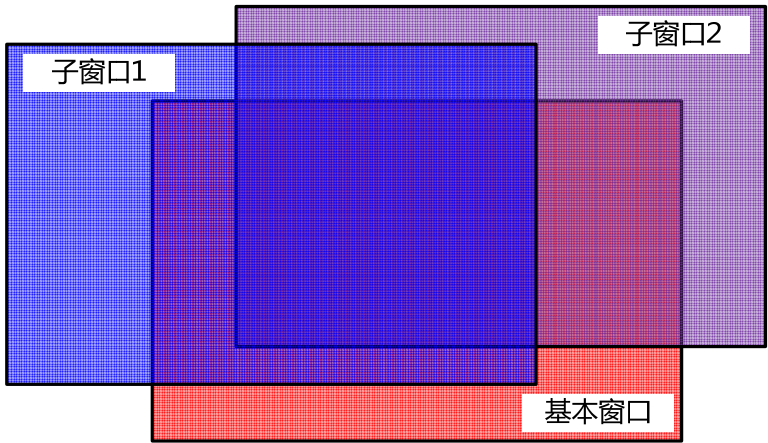


图 8 浏览器多窗口关系示意

如上图所示，终端浏览器在“多窗口”功能下的多个窗口被区分为一个“基本窗口”与多个“子窗口”。支持“多窗口”功能的浏览器各窗口间的关系及其展现要求如下所述：

- “基本窗口”：是指用于展现基础 EPG 浏览、音视频播放的窗口。该窗口在终端开机过程中即启动并执行业务认证相关操作，且在终端使用过程中该窗口不可被关闭、移动、改变大小等；
- “子窗口”：是指在业务使用过程中由浏览器底层启动的窗口。这类窗口可以叠加在“基本窗口”及其他“子窗口”上进行展现，且允许通过浏览器相关窗口控制模块被新建、关闭、移动、改变大小等，但不支持通过“基本窗口”对于“子窗口”进行相应操作。就目前情况来说，除“基本窗口”外，最多允许新建 3 个“子窗口”；
- 除在窗口是否可被改变关闭、移动、改变大小等方面存在差异外，“基本窗口”与“子窗口”共享浏览器的所有资源，其中包括 HTML 解析器、图形库、字库、Cookie、Session 等；

- 由业务逻辑需要而新建的“子窗口”自动获取浏览器的焦点并在最前端进行显示，放置在后端的“子窗口”或“基本窗口”可以通过 `window.focus` 方法放置在最前端显示，并获取焦点；
- “基本窗口”及“子窗口”可以通过调用 `window.hide` 方法进行隐藏，也可以调用 `window.show` 方法进行显示。当在最前端显示的窗口被隐藏时，浏览器焦点自动放置在该窗口后已显示的“子窗口”或“基本窗口”上；
- “子窗口”可以通过调用本窗口内的 `window.close` 方法进行关闭，但“基本窗口”不支持通过 `window.close` 方法进行关闭。

附录 A
(规范性附录)
浏览器默认 CSS

终端浏览器应根据提供的默认 CSS 进行页面元素的显示，默认 CSS 示例如下：

<pre>img{ border:0px; } body,div,p,font,span{ margin:0px; padding:0px; border:0px; display:block; font-size:20px; } body { background-color:#000000; }a{ color:#ffffff; text-decoration:none; } a:link { color:#ffffff; text-decoration:none; }</pre>	<pre>a:visited{ color:#ffffff; text-decoration: none; } a:hover{ color:#ffffff; text-decoration:none; } ol { display: block; list-style-type: none; } li { display: list-item; }</pre>	<pre>frameset { display: block; position: static; float: none; border: none; } frame { border: none; } iframe { border: none; } noframes { display: none; }</pre>
---	--	---

附录 B
(资料性附录)
规范版本修订记录

序号	操作	章节	说明	修订人	修订日期
1	新增	全文	此规范为新增规范	沈昕	2009-5-11
2	修订	全文	将文档内容区分为 EPG 制作及浏览器显示 2 个大部分,并将 MediaPlayer、Navigation 等扩展 JavaScript 对象从原有的接口规范中移植到本标准中。	沈昕	2011-05-29
3	修订	全文	增加了对于 Ajax、部分 HTML5、CSS3 的支持要求	沈昕	2011-05-29
4	新增	CSS 部分	增加了对于 CSS Transition 的支持要求及说明	沈昕	2011-05-29
5	新增	8.10	新增了浏览器多窗口的功能要求	沈昕	2011-05-29
6	新增	全文	增加了对于高清 EPG 的制作及显示支持要求	沈昕	2011-05-29
7	新增	8.9	新增了对于浏览器焦点处理方面的要求	沈昕	2011-05-29
8	新增	8.8	新增了对于 CSS 解析过程中盒子模型的说明	沈昕	2011-05-29
9	删除	8.10	删除在允许基本窗口与子窗口通过 window.open 方法新建子窗口的要求	沈昕	2011-06-03
10	删除	7.6.3.1	删除对于 window.open 方法的支持,删除 Window 的 Parent 属性	沈昕	2011-06-03
11	新增	8.1	对于浏览器通过页面<META>标签进行标清、高清页面的切换进行了进一步的说明	沈昕	2011-06-09
12	新增	8.1	要求浏览器在发送 HTTP 请求的 User-Agent 字段中对于其支持的分辨率进行说明以方便服务器进行预先处理	沈昕	2011-06-09
13	修订	6.7	将 HTML 标签区分为 HTML4 及 HTML5,因此新增 6.7.4 章节,将 Canvas 及 Datalist、KEYGEN 元素作为 HTML5 的要求加入该章节	沈昕	2011-07-10
14	修订	6.8	将 CSS 区分为 CSS2 及 CSS3,并将 CSS Tansition 章节合并如 CSS 3 章节中进行描述	沈昕	2011-07-10
15	修订	全文	对于文中笔误进行修订	包盛 沈昕	2011-08