

# 东方有线企业技术标准

---

## 东方有线 NGB 信息服务内容解析规范

(1280 版)

V1.0.2

20131226 修订

二〇一三年十二月发布

二〇一三年十二月实施

---

东方有线网络有限公司 发布



---

## 目 次

1 范围.....	1
2 规范性引用文件.....	1
3 术语和缩略语.....	1
4 NGB 互动信息服务系统架构 .....	2
5 NGB 互动信息服务的技术要求 .....	3
附录 A （规范性附录） 互动信息服务的分辨率设置和服务进入方式 .....	4
附录 B （规范性附录） 互动信息服务测试系统搭建方法 .....	5
附录 C （规范性附录） 机顶盒遥控器按键定义 .....	7
附录 D （规范性附录） 消息管理接口定义.....	10
附录 E （规范性附录） 媒体播放接口定义 .....	15
附录 F （规范性附录） DVB-SI 信息获取接口定义 .....	33
附录 G （规范性附录） 机顶盒 MAC 地址获取接口定义.....	36
附录 H （规范性附录） OCN 特定应用接口定义.....	38
规范修订记录.....	41

---

## 前 言

本规范由东方有线网络有限公司提出。

本规范由东方有线网络有限公司总工程师办公室起草。

本规范主要起草人：王明敏、陈宝霞、吴坚、周美英、项芳。

CONFIDENTIAL

# 东方有线 NGB 信息服务内容解析规范

## 1 范围

本规范规定了上海市符合东方有线 NGB 高清机顶盒及智能电视机顶盒（以下简称机顶盒）使用的 NGB 互动信息服务的内容解析要求。

本规范适用于上海市有线数字 NGB 系统。

本规范适合 NGB 互动信息服务的应用开发人员、机顶盒软件开发人员使用。

## 2 规范性引用文件

下列文件中的条款通过本部分的引用而成为本部分的条款。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部分。然而，鼓励根据本部分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部分。

GB 2312-1980 信息交换用汉字编码字符集 基本集

GB/T 7400.11 数字电视术语

ISO/IEC 13818-4 Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 4: Conformance Testing, 1998

W3C Extensible Markup Language (XML) 1.0 (第二版)

ISO 639-1, 639-2 – Codes for the Representation of Names of Languages- Part 1: Alpha-2 Code, Part 2: Alpha-3 Code

IETF STD0013 (RFC 1035), Domain Names–Implementation and Specification (Nov. 1987)

IETF RFC 1945, Hypertext Transfer Protocol -- HTTP/1.0 (May 1996)

IETF RFC 1738, Uniform Resource Locators(URL) (December 1994)

## 3 术语和缩略语

### 3.1 术语和定义

下列术语和定义适用于本标准。

#### 3.1.1

**数字电视** digital television

用数字信号表示电视的图象信息。

#### 3.1.2

**标准清晰度电视（标准数字电视）** standard definition television (SDTV)

其等效质量是由源于 ITU-R 601 建议书中 4:2:2 等级的图像经过码率压缩编码处理后达到的。对具有代表性的节目素材样本进行判断时，其主观评价质量优于 PAL-D 常规清晰度电视的数字电视系统。

#### 3.1.3

**高清晰度电视** high definition television (HDTV)

高清晰度电视在水平和垂直两个方向上分辨力均约为标准清晰度电视的两倍左右，并具有 16:9 的幅型比。

#### 3.1.4

**有线数字电视** cable digital television

利用射频电缆、光缆、多路微波或它们的组合，传输数字电视信号的一种电视系统。

#### 3.1.5

**电子节目指南** EPG (Electronic Program Guide)

传送流中描述某时间段内各个数字电视频道播出的所有节目和业务及其相关信息的信息表。该信息表能随着时间的推移而更新，接收端从传送流中将其正确的提取出来后，用户即可便捷的了解、选收相应的节目和得到相应的服务。

### 3.1.6

#### 视频点播 VOD (Video On Demand)

视频点播，即按需要的视频流播放。用户不必遵循传统的节目时间表，而是根据自己的意愿随时直接点播希望收看的节目，从根本上改变了过去被动收看电视的方式，实现了节目的按需收看和任意播放。

### 3.1.7

#### 节目导视

节目导视是指在基本视音频收看功能之外的其他界面和功能的总称，包括：主菜单、基本 EPG 信息以及各种信息条等。

### 3.1.8

#### 业务 Service

数字电视运营商提供的服务。

## 3.2 缩略语

下列缩略语适用于本标准。

EPG Electronic Program Guide 电子节目指南

HDMI High Definition Multimedia Interface 高清晰度多媒体接口

HDTV High Definition Television 高清晰度电视

OSD On-Screen Display 屏幕显示

STB Set-Top-Box 机顶盒

VOD Video On Demand 视频点播

## 4 NGB 互动信息服务系统架构

### 4.1 系统架构说明

互动信息服务系统的架构如下图所示。

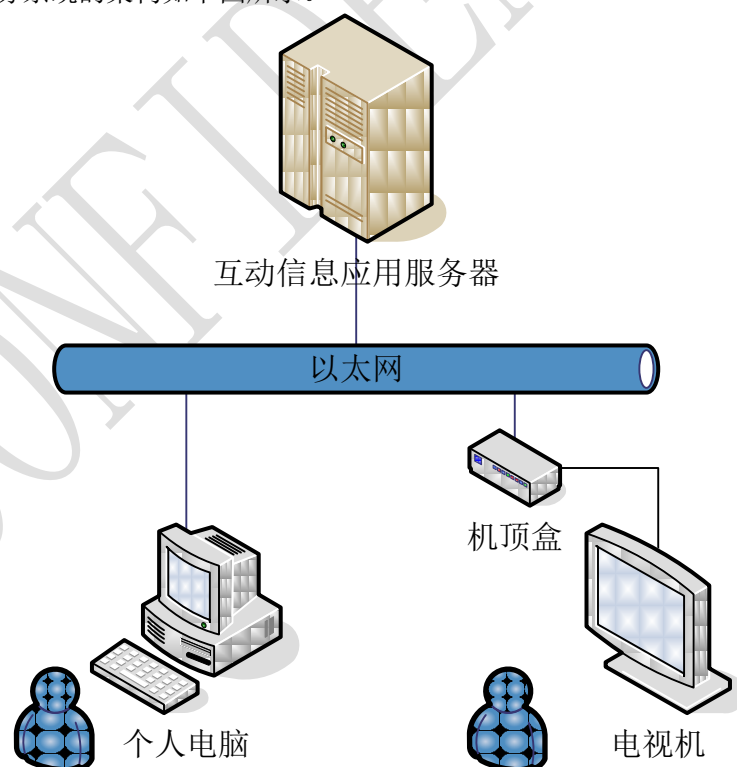


图1 系统架构图

系统架构与普通网站系统架构相同，应用服务器以网络服务器的方式搭建，用户可以通

---

过“机顶盒+电视机”浏览 WEBTV 应用内容，也可以通过 PC 直接访问。

## 5 NGB 互动信息服务的技术要求

由于发布的互动信息服务由机顶盒解析、显示在电视机上，因此对于开发的 DVBHTML 应用有比较严格的要求。

### 5.1 电视机显示要求

#### 5.1.1 电视显示区域

机顶盒浏览器的显示分辨率是 1280×720 像素。

由于每台电视机行、场都要保证不露边，因此会将部分有用的信号也调出去。电视机的图像重现率比计算机的显示器小，靠近边缘部分的文字与图像可能会在电视机中无法看到，所以在进行应用界面设计时，文字与图像尽量设计在屏幕中间的部位。

为了保证页面上的内容可以完全显示在电视机上，需要设置一个安全显示区域。各种型号的电视机，其重现率都是不同的，因此对于安全显示区没有固定的大小，推荐的安全显示区域分辨率为 1120×620，即左右各保留 80，上下各保留 50。

#### 5.1.2 电视画面闪烁

普通电视机采用的是 50Hz (PAL 制)，即每秒扫描 50 场，每幅图像是隔行扫描出来的。对于人眼来说，这是一个闪烁的图像。为了消除闪烁现象，在制作页面时需要尽量避免下列容易出现闪烁现象的情况发生：

- 颜色对比度太强烈
- 显示宽度为 1 像素的横向直线

### 5.2 机顶盒浏览器要求

#### 5.2.1 WEB 技术支持

机顶盒支持如下 WEB 技术：HTML 4.01 全集标签、JavaScript 1.5、CSS 2.0、Ajax 技术，建议尽量使用 JavaScript 原生脚本语言。

#### 5.2.2 字体支持

机顶盒默认使用 24 像素(px)矢量字体，支持 CSS 设置字体像素及样式，支持宋体字库。

#### 5.2.3 图片支持

##### 1. 格式

机顶盒浏览器支持 GIF 静态、GIF 动态、PNG、JPEG 四种格式，推荐使用 PNG 格式的图片。

##### 2. 效果

在进行图片设计时，建议尽量采用图形和色彩都相对简单的方案。

#### 5.2.4 遥控器按键响应

限于电视操作性限制，应用页面需要支持遥控器按键的响应，具体见附录。

#### 5.2.5 中文输入法

当页面的当前焦点处于文本输入框时，按遥控器确认键可以打开中文输入法。

附录 A  
(规范性附录)  
互动信息服务的分辨率设置和服务进入方式

### A.1 互动信息服务的分辨率设置

互动信息服务页面的默认显示分辨率应为1280×720。针对如下机顶盒，通过下述方式可以将机顶盒浏览器显示分辨率调整为1280×720：

- 针对采用茁壮中间件的机顶盒，机顶盒浏览器根据页面HTML中定义的元信息自动调整显示分辨率，如下红色标记代码所示：

```
<html>
  <head>
    <title>点播平台首页</title>
    <meta http-equiv="pragma" content="no-cache">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="expires" content="0">
    <meta name="page-view-size" content="1280*720">
    <link rel="stylesheet" type="text/css" href="{basePath}css/style.css">
```

- 针对采用全景中间件的机顶盒，通过对EPG应用栏目入口的配置可以使机顶盒浏览器自动显示成1280×720分辨率。配置通过节目导视应用头端下发数据来进行，由OCN来负责，因此在服务页面开发调试阶段，可以从【彩票天地】、【智慧闵行】、【特惠机票】三个栏目入口进入，此时机顶盒浏览器的分辨率为1280×720。

### A.2 互动信息服务进入方式说明

节目导视应用作为 OCN 机顶盒开机界面，具备多个交互应用入口，常用的应用入口有【视频点播】、【节目回看】等，每个交互应用入口带有服务类型 SrvType，取值为字符串，由机顶盒的节目导视应用来配置，例如视频点播的 SrvType 为 VOD，节目回看的 SrvType 为 RevTV。机顶盒浏览器使用如下规则拼接 HTTP 请求地址：

http://{网络服务器 IP 地址}/index.asp?SrvType={服务类型}，其中使用默认 80 端口号。

在互动信息服务页面开发调试阶段，可以将机顶盒设置为双向模式，这样每次机顶盒开机后就直接访问地址 http://{网络服务器 IP 地址}/index.asp?SrvType=VOD，以免去每次重启机顶盒后修改机顶盒系统菜单的麻烦。其操作方式如下：

- 机顶盒开机后选择“管理”，选择进入“系统设置”界面，密码为“4321”；
- 在系统设置页面，选择进入“系统选项”，将开机模式调整为双向模式；
- 在系统设置页面中，选择进入“系统选项”，选择“服务器设置”，进入服务器设置页面，将服务器地址设置为w.x.y.z，即开发测试用的入口服务器；
- 设置完成确认后，点击“导视”，选择栏目“视频点播”或者按遥控器“点播”键后，可直接访问上述地址。每次机顶盒重新启动后，也会直接访问上述地址。



## 附录 B (规范性附录)

### 互动信息服务测试系统搭建方法

为了配合互动信息服务的开发调试，还需要搭建对应的测试系统。搭建方法如下。

#### B.1 测试服务器端的配置

1. 第一步：测试环境中需要DHCP，以便为机顶盒分配IP地址。
2. 第二步：在测试环境中设置一台模拟入口服务器，用于响应机顶盒浏览器的请求，并对请求参数进行判断后重定向至测试页面。
  - 方法一：若互动信息服务系统使用IIS提供互动信息服务，由于ASP是其服务端脚本语言，对IIS配置即能正确响应机顶盒浏览器的请求，例如index.asp的代码如下

```
<html>
<head>
</head>
<body>
<%
dim ServiceType
ServiceType = "" & Request.QueryString("SrvType")
select case ucase(ServiceType)
    case "VOD"          '视频点播
        response.redirect "http://web-ip:port/path/url1"
    case "LOTTERY"      '彩票天地
        response.redirect "http://web-ip:port/path/url2"
    case else
        response.redirect "http://other-web-ip:other-port/other-path/defaulturl"
end select
%>
</body>
</html>
```

其中http://web-ip:port/path/url 对应的应用页面。这样将机顶盒系统菜单中的网络服务器设置为这台IIS 服务器地址后，每次按机顶盒遥控器的“点播”键或者从EPG栏目入口进入，就会自动跳转到测试环境的应用页面上。（机顶盒端的设置方法见附录A）

- 方法二：若使用其他web服务提供互动信息服务，可以通过URL映射方式来响应机顶盒浏览器的请求，以Apache Tomcat为例，在项目web.xml中进行如下servlet的配置。

```
<servlet>
<servlet-name>log4j-init</servlet-name>
<servlet-class>webepg.servlet.Log4jInitServlet</servlet-class>
<init-param>
<param-name>log4j</param-name>
<param-value>WEB-INF/classes/log4j.properties</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>

<servlet>
<servlet-name>portalServlet</servlet-name>
<servlet-class>webepg.servlet.PortalServlet</servlet-class>
<load-on-startup>2</load-on-startup>
</servlet>
<servlet-mapping>
<servlet-name>portalServlet</servlet-name>
<url-pattern>/index.asp</url-pattern>
```

---

```
</servlet-mapping>
```

另外，上述第二步服务器端配置还有一种简易配置方法。

按OCN要求，正式上线应用需定义相应的SrvType， OCN将在EPG应用栏目入口进行配置和业务访问控制。但是在应用开发调试阶段，可以不用关心交互应用入口，因此可以忽略请求参数SrvType，上述第二步的ASP代码可以按如下简化，Tomcat的Java代码简化原理与此相同。

```
<html>
<head>
</head>
<body>
<%
    response.redirect "http://web-ip:port/path/url1"
%>
</body>
</html>
```

附录 C  
(规范性附录)  
机顶盒遥控器按键定义

### C.1 遥控器按键键值定义

针对东方有线NGB网络内的三类高清机顶盒终端，其浏览器的键值定义分别有三种，因此互动信息服务的页面中要求同时支持以下三种键值定义：

- 针对2013年及之后发放的各类高清机顶盒（含智能电视机顶盒）：采用NGB-H国标对应键值（必须支持）
- 针对已有茁壮高清机顶盒：采用茁壮中间件对应键值
- 针对已有全景高清机顶盒：采用全景中间件对应键值

遥控器按键	NGB-H 国标 对应键值	茁壮中间件 对应键值	全景中间件 对应键值	备注
退出	27	339	123	全景中间件根据信息服务的需求，可以通过 Javascript 接口设置为 75(对应键盘 K 键)
返回	4096	340	8	全景中间件根据信息服务的需求，可以通过 Javascript 接口设置为 70(对应键盘 F 键)
上页	33	372	33	
下页	34	373	34	
待机	4106	258	76	
导视	4192	513	72	
点播	4211	517	86	
上键	38	1	38	
下键	40	2	40	
左键	37	3	37	
右键	39	4	39	
确定	4097	13	13	
时移	未定义	?	90	现有遥控器已删除该键
信息	4100	567	73	
频道+	4111	593	85	
频道-	4112	594	68	
音量+	4109	595	187	
音量-	4110	596	189	
静音	4108	597	87	
红键	2305	832	66	
绿键	2306	833	71	
黄键	2307	834	89	
蓝键	2308	835	81	
交替	4231	338	116	全景中间件根据信息服务的需求，可以通过 Javascript 接口设置为 79(对应键盘 O 键)
服务	4098	562	80	
收藏	4102	570	82	
快录	4166	1027	69	
配置	4101	519	67	
搜索	4117	770	83	

帮助	4099	515	78	
音效	4104	771	65	
电视	4209	1038	84	
音乐	4210	564	77	
标清	未定义	768	112	老版遥控器该键为 F1
高清	未定义	769	113	老版遥控器该键为 F2
*	2319	561	88	
#	2320	849	74	
0	48	48	48	
1	49	49	49	
2	50	50	50	
3	51	51	51	
4	52	52	52	
5	53	53	53	
6	54	54	54	
7	55	55	55	
8	56	56	56	
9	57	57	57	
注：复合键键值定义未包括在内。				

## C.2 按键监听的 JavaScript 脚本示例

```
//iPanel 茁壮
document.onkeypress = grabEvent;
document.onirkeypress = grabEvent;
function grabEvent(event){
    keycode = event.which;
    if (keycode == 38) {
        //do something
    } else if (keycode == 40) {
        //do something
    }
}
document.onkeydown = handleEvent;
function handleEvent(event){
    if (jShow) {
        //SHDV 全景
        keycode = event.keyCode || event.which;
        if (keycode == 38) {
            //do something
        } else if (keycode == 40) {
            //do something
        }
    } else {
        //NGB-H 国标
        if (event.type == 1002) {
            keycode = event.which;
            if (keycode == 38) {
                //do something
            } else if (keycode == 40) {
                //do something
            }
        }
    }
}
```

附录 D  
(规范性附录)  
消息管理接口定义

### D.1 接口定义

互动信息服务在调用各类业务接口后，需要捕获消息并进行相应的处理。

针对东方有线NGB网络内的三类高清机顶盒终端，其消息接口分别有三种，因此互动信息服务的页面中需要根据所适配的机顶盒，采用对应的消息接口定义。

- 针对2013年及之后发放的各类高清机顶盒（含智能电视机顶盒）：采用NGB-H国标接口定义（必须支持）
- 针对已有茁壮高清机顶盒：采用茁壮中间件接口定义
- 针对已有全景高清机顶盒：采用全景中间件接口定义

#### D.1.1 采用 NGB-H 国标接口定义

1. 与消息捕获相关的Javascript内置对象：event。

应用层捕获到的消息有以下两种来源：

- 系统消息，例如锁频成功、搜台完毕、网络信号中断等；
- 按键消息，例如遥控器、前面板等按键触发消息；

应用层通过以下方式捕获消息：

- 系统消息：应用通过document.onsystemevent捕获系统消息；
- 按键消息：
  - 遥控器：应用通过document.onkeydown和document.onkeyup捕获遥控器消息，消息码与PC方式保持兼容，扩展按键消息码定义见“（规范性附录）机顶盒遥控器按键定义”；
  - 前面板：应用通过document.onkeydown和document.onkeyup捕获遥控器消息，消息码与遥控器方式保持一致。

2. event对象的属性说明见下表

属性名称	类型	读写属性	说明
event.type	number	只读	表示发生的事件的类型，即当前event对象表示的事件的名称，它与注册的事件句柄同名，例如“onclick”；或者是事件句柄属性删除前缀“on”，例如“click”。同W3C定义。
event.source	number	只读	表示消息的来源，取值如下： 1001 — 表示系统消息； 1002 — 表示遥控器按键消息； 1003 — 表示前面板按键消息；
event.which	number	只读	表示消息的代码值。
event.modifiers	number	只读	表示消息的扩展属性。若该消息的扩展属性为空，则modifiers返回0；若该消息的扩展属性为number型，则modifiers返回一个ID值，该值由中间件内部生成，作为具体字符串内容的指针，应用可调用Utility.getEventInfo(ID)方法取出字符串内容。

3. 与消息获取相关的Javascript对象：Utility。Utility对象为内置对象，该对象可以完成消息扩展属性的获取。

原型：getEventInfo(id)。

描述：获取消息的详细描述信息。

参数：id — number型，表示消息描述字符串ID。

返回：string型，表示消息的详细描述信息。

4. 媒体播放模块与应用层交互的消息定义

消息名称	event.which	event.modifiers	消息说明
------	-------------	-----------------	------

MSG_MEDIA_URL_VALID	13001	-	媒体源路径有效。
MSG_MEDIA_URL_INVALID	13002	-	媒体源路径无效。
MSG_MEDIA_PLAY_SUCCESS	13003	-	开始播放成功。
MSG_MEDIA_PLAY_FAILED	13004	-	开始播放失败。
MSG_MEDIA_SETPACE_SUCCESS	13005	-	步长设置成功。
MSG_MEDIA_SETPACE_FAILED	13006	-	步长设置失败。
MSG_MEDIA_SEEK_SUCCESS	13007	-	设置播放时间点成功。
MSG_MEDIA_SEEK_FAILED	13008	-	设置播放时间点失败。
MSG_MEDIA_PAUSE_SUCCESS	13009	-	暂停播放成功。
MSG_MEDIA_PAUSE_FAILED	13010	-	暂停播放失败。
MSG_MEDIA_RESUME_SUCCESS	13011	-	恢复播放成功。
MSG_MEDIA_RESUME_FAILED	13012	-	恢复播放失败。
MSG_MEDIA_STOP_SUCCESS	13013	-	停止播放成功。
MSG_MEDIA_STOP_FAILED	13014	-	停止播放失败。
MSG_VOD_STREAM_START	13050	-	节目快退到头。
MSG_VOD_STREAM_END	13051	-	节目播放到尾。
MSG_VOD_STATE_CHANGE	13052	-	节目播放状态改变。
MSG_VOD_WARNING	13053	消息描述字符串的ID	E2304:网络连接失败,请稍后尝试
		消息描述字符串的ID	E2305:未订购该服务,请致电96877咨询
		消息描述字符串的ID	E2306:网络连接失败,请稍后尝试
		消息描述字符串的ID	E2307:网络连接失败,请稍后尝试
		消息描述字符串的ID	E2308:网络连接失败,请稍后尝试
		消息描述字符串的ID	E2309:网络连接失败,请稍后尝试
		消息描述字符串的ID	E2310:网络连接失败,请稍后尝试
		消息描述字符串的ID	E2311:网络连接失败,请稍后尝试
MSG_VOD_NEXT_ALERT	13054	-	是否继续播放下一个节目
MSG_VOD_SWITCHSONG	13055	-	Kalaoke切歌
保留	13015 ~ 13200		
event.modifiers 值由中间件内部自动给出,其数据类型: ——“number”,表示该值为消息描述字符串的 ID,可通过 Utility.getEventInfo()方法获取该消息描述字符串。若“消息说明”定义了消息字符串 JSON 格式,则按格式取出消息内容。 ——“-”,表示event.modifiers为undefined。			

#### D.1.2 采用茁壮中间件接口定义

1. 与消息捕获相关的Javascript内置对象: event。  
应用层捕获到的消息有以下来源:

- 标准消息：指中间件终端通过某种协议（如http、rtsp等）与服务器进行交互时产生的一系列标准消息事件，如请求应答正常(200)、无法找到指定位置的资源(404)。
- 键盘按键消息：单击键盘上的一个按键时，中间件就会收到一个CHAR型 消息，即键盘按键消息。这些消息中间件会发送给页面处理。页面捕获到的消息值就是该键盘字符的ASSIC码。
- 遥控器按键消息：单击摇控器上按键时，中间件便会收到遥控按键消息，即irkey消息。
- 系统消息：中间件运行过程中会发生一系统列的功能性操作，如自动搜索、下载skin等等。

应用层通过以下方式捕获消息：

- 标准消息和系统消息：应用通过document.onsystemevent捕获消息。
- 键盘消息：应用通过document.onkeypress捕获消息。
- 遥控器按键消息：应用通过document.onirkeypress或document.onkeydown捕获消息。

## 2. event对象的属性说明见下表

属性名称	类型	读写属性	说明
event.type	字符串型	只读	这个用来区分消息的类型，用于在需要捕获的消息很多时进行一级过滤，目前主要针对system消息，支持的值为"dvb"和"iptv"；取值如下： "keyup": keyup消息。"keydown": keydown消息。 "keypress": keypress消息。"dvb": dvb消息。"iptv": iptv消息
event.which	number	只读	用于获取c code中event的p1参数的值。在char消息和irkey消息中指的是页面捕获的按键键值，在system消息中指的是dvb或iptv消息的消息值。
event.modifiers	number	只读	这个用来获取c code中event的p2参数的值，通常用于对同一消息提供更多的信息，供需要的时候获取。

## 3. 媒体播放模块与应用层交互的消息定义

消息名称	event.which	event.modifiers	消息说明
EIS_VOD_PREPAREPLAY_SUCCESS	5202	-	播放媒体成功。
EIS_VOD_CONNECT_FAILED	5203	-	连接服务器失败,建立会话失败或者服务器返回超时。
EIS_VOD_CHANGE_SUCCESS	5220	-	切换服务器成功。
EIS_VOD_CHANGE_FAIL	5221	-	切换服务器失败。
EIS_VOD_PLAY_SUCCESS	5205	-	播放媒体成功。
EIS_VOD_PLAY_FAILED	5206	-	播放媒体失败。
EIS_VOD_RELEASE_SUCCESS	5211	-	拆链成功。
EIS_VOD_RELEASE_FAILED	5212	-	拆链失败。
EIS_VOD_PROGRAM_BEGIN	5209	-	时移频道或VOD影片播放到了点播开始的位置。
EIS_VOD_PROGRAM_END	5210	-	时移频道或VOD影片播放到了点播结束的位置。

### D.1.3 采用全景中间件接口定义

使用全景中间件浏览器的内置JS对象jShow。

#### 1. signal.connect

原型：signal.connect(func)。



描述：通过接口注册消息处理方法，当应用层捕获到消息后，在该方式中执行相应的处理。

参数：func – function型，该方法的回调输入参数是一个对象，参数说明如下：

—— type – string型，消息类型。

—— msg – string型，消息内容。

—— info – string型，附加消息内容。

返回：无。

## 2. signal.disconnect

原型：signal.disconnect(func)。

描述：注销消息处理方法，应用层在离开当前页面前调用。

参数：func – function型，与signal.connect接口的输入参数是同一个方法

返回：无。

## 3. 媒体播放模块与应用层交互的消息定义

消息类型	消息内容	附加消息内容	消息说明
TYPE_VOD_MSG	MSG_VOD_ENTER	-	进入VOD播放状态。
	MSG_VOD_STATE_CHANGE	-	VOD播放状态发生变化。
	MSG_WARNING	-	VOD播放结束。

## H.2 Javascript 脚本示例

```
//国标中间件接口
document.onsystemevent = grabEvent;
function grabEvent(){
    try {
        $("#resultDiv").append("NGB-H Event type:"+ event.type +" which:"+ event.which +" modifiers:"+ event.modifiers + "<br />");

        try {
            if ("systemevent" == event.type) {
                $("#resultDiv").append("systemevent msg:"+ Utility.getEventInfo(event.modifiers) + "<br />");
            }
        } catch (e) {
            $("#resultDiv").append("Utility.getEventInfo exception:"+ e + "<br />");
        }
    } catch (e) {
        $("#resultDiv").append("grabEvent exception:"+ e + "<br />");
    }
}
```

## //茁壮中间件接口

```
document.onkeypress = grabevent; // 注册 char 消息处理函数,函数名可以随意取.
document.onirkeypress = grabevent; // 注册 irkey 消息处理函数,函数名可以随意取.
document.onsystemevent = grabevent2; // 注册系统消息处理函数,函数名可以随意取. function
grabevent(){ //按键消息处理函数定义
    var retv = 1;
    var keycode = event.which; //获取消息的 p1 参数即按键键值
    switch (keycode){
        case xxx: // 若为 xxx 按键
            ..... // 对 xxx 的处理代码
            break;
```

```

        case yyy: // 若为 yyy 按键
            ..... // 对 yyy 的处理代码
            break;
        default: ..... // 其他按键的处理代码
            break; }
        return retv; // 返回值决定是否继续下传该 event 到下层继续处理。
    }
}

function grabevent2(){ //系统消息处理函数定义
    var retv = 1;
    var eventCode = event.which; // 获取消息 p1 参数即系统消息的消息值
    var eventType = event.type; // 获取系统消息的消息的类型
    if(eventType == "iptv"){ // 如果是 iptv 消息
        case xxx: // 若为 xxx 消息 ..... // 对 xxx 的处理代码
            break;
        case yyy: // 若为 yyy 消息
            var paras = event.modifiers; // 获取消息 p2 参数
            if(paras == yyy1) ..... // 对 p2 参数为 yyy1 时进行的处理
            if(paras == yyy2) ..... //对 p2 参数为 yyy2 时进行的处理 .....
            break;
        default: ..... // 其他消息的处理代码
            break; }
        retrun retv; // 返回值决定是否继续下传该 event 到下层继续处理
    }
}

//全景中间件接口
jShow.signal.connect(signalCallback);
function signalCallback(o) {
    if (o.msg != "MSG_REC_Check" || o.type == "TYPE_VOD_MSG") {
        logToRtos("*****localwebpg vod*****receive message*****msg "+ o.msg
+ "*****type"+ o.type + "*****info
"+o.info+"*****\n");
    }
    //底层启动视频点播任务后的通知消息
    if (o.type == "TYPE_VOD_MSG" && o.msg == "MSG_VOD_ENTER") {
        showPlayState();
        showPlayControl(true);
    }
    //播控状态发生变化后的通知消息
    if (o.type == "TYPE_VOD_MSG" && o.msg == "MSG_VOD_STATE_CHANGE") {
        showPlayStateCallback();
        showPlayControl(true);
    }
    //快退到头、快进到尾、播放结束的通知消息
    if (o.type == "TYPE_VOD_MSG" && o.msg == "MSG_WARNING") {
        hidePlayControl();
        jShow.showAlertWithEnsure(o.info, function(){
            jShow.vodSendAction(3);
            exit();
        });
    }
}

window.onbeforeunload = function() {
    jShow.signal.disconnect(signalCallback);
}

```

附录 E  
(规范性附录)  
媒体播放接口定义

## E.1 接口定义

当互动信息服务中需要调用机顶盒浏览器进行视音频媒体播放时，需要用到该接口定义。

针对东方有线NGB网络内的三类高清机顶盒终端，其播放接口分别有三种，因此互动信息服务的页面中需要根据所适配的机顶盒，采用对应的播放接口定义。

- 针对2013年及之后发放的各类高清机顶盒（含智能电视机顶盒）：采用NGB-H国标接口定义（必选支持）
- 针对已有茁壮高清机顶盒：采用茁壮中间件接口定义
- 针对已有全景高清机顶盒：采用全景中间件接口定义

### E.1.1 采用 NGB-H 国标接口定义

#### 1. MediaPlayer

原型：MediaPlayer()

描述：MediaPlayer对象的构造方法。

参数：无。

返回：无。

#### 2. getPlayerInstanceID

原型：number getPlayerInstanceID()

描述：获得接收终端本地可用的播放器实例ID，由中间件内部自动分配。

参数：无。

返回：number型，成功返回0~255，失败返回-1。

#### 3. bindPlayerInstance

原型：number bindPlayerInstance(playerInstanceID)

描述：MediaPlayer对象与播放器实例绑定。

参数：playInstanceID — number型，指示本地播放器实例ID，取值范围0~255

返回：number型，成功返回0，失败返回-1。

#### 4. unbindPlayerInstance

原型：number unbindPlayerInstance(playerInstanceID)

描述：MediaPlayer对象和当前播放器实例解除绑定，并释放播放器的相关资源。

参数：playInstanceID — number型，指示本地播放器实例ID，取值范围0~255

返回：number型，成功返回0，失败返回-1。

#### 5. setMediaSource

原型：number setMediaSource(mediaURL)

描述：异步方法，设置待播放媒体的URL地址。设置此参数后，中间件自动检测设置的mediaURL的合法性。

——若URL合法，则向页面发送MSG\_MEDIA\_URL\_VALID消息；

——若URL不合法，则向页面发送MSG\_MEDIA\_URL\_INVALID消息。

页面在接收到URL合法消息后，再调用play()方法进行播放。

参数：mediaURL — string型，以URL格式表示的媒体路径，URL格式说明见下表

媒体源	mediaURL	说明
广播业务	dvb://<original_network_id>.<transport_stream_id>.<service_id>[.<component_tag>{&<component_tag>}];event_id>]{<path-element>}	可用于访问： ——广播业务； ——业务组件，例如视频、音频或字幕等基本流；

		——广播业务的事件； ——OC/DC 轮播中文件；
互动电视业务	rtsp://<会话管理器 IP>:<会话管理器端口>;EntitlementCode=<EntitlementCode>	可用于访问： ——视频点播； ——时移电视； ——频道回看；

返回：number型，成功返回0，失败返回其它。

## 6. play

原型：number play()

描述：异步方法，从媒体起始点开始播放，对电视广播以实时方式开始播放。

——若播放成功，则向页面发送MSG\_MEDIA\_PLAY\_SUCCESS消息；

——若播放失败，则向页面发送MSG\_MEDIA\_PLAY\_FAILED消息；

返回：number型，成功返回1，失败返回0。

## 7. seek

原型：number seek(type, timestamp)

描述：异步方法，从当前媒体的某个时间点开始播放，对实时播放的电视广播调用无效，但对于处于时移状态的电视广播有效。

——若调用成功，则发送MSG\_MEDIA\_SEEK\_SUCCESS消息；

——若调用失败，则发送MSG\_MEDIA\_SEEK\_FAILED消息；

仅在播放和暂停状态下可以调用该方法。

参数：type — number型，取值：

—— 1 – 表示Normal Play Time；

—— 2 – 表示Absolute Time。

timestamp — 见RFC2326中的Normal Play Time(NPT)和Absolute Time(ClockTime)两种时间类型格式。

——对VOD而言，是从媒体起始点开始计算的相对时间；

——对时移等有时间基的媒体而言就是绝对时间。

它可以表示起始时间（例如：“123-”），也可以表示时间段（例如：“123-333”）。

若是时间段的方式，播放到达结束时间时应为暂停状态，而不是停止状态。

返回：number型，成功返回0，失败返回其它。

## 8. setPace

原型：number setPace(pace)

描述：异步方法，设置播放步长。

——若设置成功，则向页面发送MSG\_MEDIA\_SET\_PACE\_SUCCESS消息；

——若设置失败，则向页面发送MSG\_MEDIA\_SET\_PACE\_FAILED消息。

仅在播放状态下可以调用该方法。

参数：pace — number型，播放步长，取值4、12、24、-4、-12、-24，对应快进4倍速、12倍速、24倍速，快退4倍速、12倍速、24倍速。

返回：number型，成功返回0，失败返回其它。

## 9. pause

原型：number pause()

描述：异步方法，暂停播放。

——若暂停播放成功，则向页面发送MSG\_MEDIA\_PAUSE\_SUCCESS消息；

——若暂停播放失败，则向页面发送MSG\_MEDIA\_PAUSE\_FAILED消息。

参数：无。

返回：number型，成功返回0，失败返回其它。

## 10. resume

---

原型: `number resume()`

描述: 异步方法, 恢复播放。

- 若恢复播放成功, 则向页面发送MSG\_MEDIA\_RESUME\_SUCCESS消息;
- 若恢复播放失败, 则向页面发送MSG\_MEDIA\_RESUME\_FAILED消息。

参数: 无。

返回: `number`型, 成功返回0, 失败返回其它。

#### 11. stop

原型: `number stop()`

描述: 异步方法, 停止播放。

- 若停止播放成功, 则向页面发送MSG\_MEDIA\_STOP\_SUCCESS消息;
- 若停止播放失败, 则向页面发送MSG\_MEDIA\_STOP\_FAILED消息。

参数: 无。

返回: `number`型, 成功返回0, 失败返回其它。

#### 12. refresh

原型: `number refresh(flag)`

描述: 调用`setVideoDisplayMode()`、`setVideoDisplayArea()`等属性调整视频参数后并不是立即生效, 而是在调用该方法后才会生效。

参数: 无。

返回: `number`型, 成功返回0, 失败返回其它。

#### 13. enableTrickMode

原型: `number enableTrickMode()`

描述: 设置MediaPlayer对象当前所绑定的播放器实例在其生命周期内是否允许特技操作(包括快进/快退/暂停等), 与媒体本身的特技模式属性是逻辑与的关系。

例如: 某个流前面30秒的广告不能进行特技操作, 则可以在该时间段内调用`enableTrickMode(0)`, 此时特技被禁止; 过了30秒后, Web页面中再调用`enableTrickMode(1)`, 以启动特技功能。若所播放的媒体源本身不支持特技模式, 则此方法无效。

参数: `flag` — `number`型, 取值0表示禁止特技操作(缺省值); 取值1表示允许特技操作。

返回: `number`型, 成功返回0, 失败返回其它。

#### 14. getTrickModeFlag

原型: `number getTrickModeFlag()`

描述: 获取特技模式操作标志。

参数: 无。

返回: `number`型, 0表示禁止特技操作(缺省值); 1表示允许特技操作。

#### 15. setVideoDisplayMode

原型: `number setVideoDisplayMode(mode)`

描述: 设置MediaPlayer对象对应的视频窗口的显示模式。每次调用该方法后, 视频显示窗口并不会立即显示设置的模式, 只有调用`refresh()`方法后才会刷新窗口来显示设置的模式。

参数: `mode` — `number`型, 视频显示模式, 取值如下:

- 0 — 按`setVideoDisplayArea()`方法中设定的`height`、`width`、`left`和`top`属性所指定的位置和大小来显示;
- 1 — 全屏显示, 按全屏高度和宽度显示(默认值);
- 2 — 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 — 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, 若媒体播放没有停止, 将继续播放音频。

---

返回: number型, 成功返回0, 失败返回其它。

#### 16. getVideoDisplayMode

原型: number getVideoDisplayMode()

描述: 获取MediaPlayer对象对应的视频窗口的显示模式。

参数: 无。

返回: number型, 取值如下:

- 0 - 按setVideoDisplayArea()方法中设定的height、width、left和top属性所指定的位置和大小显示;
- 1 - 全屏显示, 按全屏高度和宽度显示(默认值);
- 2 - 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 - 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。若媒体播放没有暂停, 将继续播放音频。
- 其它: 显示模式非法。

#### 17. setVideoDisplayArea

原型: number setVideoDisplayArea(rect)

描述: 设置窗口显示区域。每次调用该方法后, 视频显示窗口并不立即按设置的区域显示, 只有调用refresh() 方法后才按照设置的区域来显示视频。当需要将视频在setVideoDisplayArea()方法设置的区域显示时, 需要调用setVideoDisplayMode(0); 但二者没有调用的先后顺序, 只需要在调用refresh() 方法之前即可。

参数: rect — Rectangle对象, 描述显示区域的位置信息。

返回: number型, 成功返回0, 失败返回其它。

#### 18. getVideoDisplayArea

原型: Rectangle getVideoDisplayArea()

描述: 获取视频窗口显示区域的位置信息。

参数: 无。

返回: Rectangle对象。

#### 19. setVolume

原型: number setVolume(volume)

描述: 设置当前音量。若当前播放的是广播业务, 应记忆该频道相对于全局音量的增量。

参数: volume — number型, 取值范围0~100, 0表示最小音量 (静音), 100表示最大音量。

返回: number型, 成功返回0, 失败返回其它。

#### 20. getVolume

原型: number getVolume()

描述: 获取当前音量。

参数: 无。

返回: number型, 取值范围0~100, 0表示最小音量 (静音), 100表示最大音量。

#### 21. getMediaDuration

原型: string getMediaDuration()

描述: 获取当前播放媒体的总时长。

参数: 无。

返回: string型, 格式为“hh:mm:ss”。当媒体来源属性为本地媒体文件时, 该方法返回正播放媒体的总时长; 当媒体来源为广播业务、互动电视业务、IP-UDP码流时, 该方法返回undefined。

## 22. getCurrentPlayTime

原型: string getCurrentPlayTime()

描述: 获取媒体播放的当前时间点。

参数: 无。

返回: string型, 表示媒体播放的当前时间点, 格式见RFC 2326中的Normal Play Time和Absolute Time两种时间类型格式。点播用Normal Play Time格式, 时移用Absolute Time格式。

## 23. getPlaybackMode

原型: string getPlaybackMode()

描述: 获取播放器当前的播放模式。当接收终端播放器的播放模式发生改变的时候以虚拟按键事件的方式通知页面。

参数: 无。

返回: string型, 表示JSON字符串, 其中至少包括“PlayMode”和“Speed”两类信息。PlayMode可取值为Normal、Pause、Trickmode, 当为Trickmode时必须带4x/-4x、12x/-12x、24x/-24x参数来表示快进/快退的速度参数。

示例: {PlayMode: “Trickmode”, Speed:” 4x” }。

### E.1.2 采用茁壮中间件接口定义

#### 1. DVB.playLiveTV(frequency,symbolRate,modulation,servcieId)

原型: DVB.playLiveTV(frequency,symbolRate,modulation,servcieId)

描述: 播放一个指定service, 不论该service是否搜索过。

参数:

Frequency-number型, 表示频率, 单位hundred HZ, 取值如, 2590000;

symbolRate: number型, 表示符号率, 单位hundred symbol/s, 取值如68750;

modulation: 字符串类型, 表示调制方式, 格式“n-QAM”, 其中n为整数, “-QMA”一定要原样保留, 取值如“32-QAM”;

serviceId: 数值型, 十进制, 表示serviceID

返回: 无

#### 2. DVB.stopAV()

原型: DVB.stopAV()

描述: 关闭打开的视音频节目

参数: 无

返回值: 无

#### 3. media.AV.open(URL,mediatype)

原型: media.AV.open(URL,mediatype)

描述: 异步方法, 打开指定媒体源进入待命状态(同stop一样媒体源处于待命状态中, 执行media.AV.play()即可开始播放)。

若准备播放媒体成功, 则向页面发送[5202]EIS\_VOD\_PREPAREPLAY\_SUCCESS。

若连接服务器失败, 建立会话失败或者服务器返回超时, 则向页面发送[5203]EIS\_VOD\_CONNECT\_FAILED。

参数: mediatype: 字符串型, 表示媒体源的类型, 取值如VOD表示点播播放等。

URL:字符串型, 标示媒体源的地址。

返回值: 无

#### 4. VOD.changeServer(serverName, mode)

原型: VOD.changeServer(serverName, mode)

描述: 根据新的服务信息设置重新获取VOD的一些基本数据并进行更新。只要服务商、服务模式或入口地址中任一个发生改变则认为服务信息改变。当前VOD业务如果不

---

能正常运行(包括时移数据未请求到)也允许进行切换

若切换成功, 则向页面发送[5220]EIS\_VOD\_CHANGE\_SUCCESS.

若切换失败, 则向页面发送[5221]EIS\_VOD\_CHANGE\_FAIL.

参数: serverName - 字符串, 表示 vod 厂商。

Mode - 字符串, 表示模式。取值如下: “dvb” “ip” “ip\_rtp” “dvb\_rtp” “ip\_ts”

返回值:

1: 表示准备切换

0: 表示服务信息未改变, 不必重启

-1: 服务商名称变了,但是切换不成功

#### 5. media.AV.play()

原型: media.AV.play()

描述: 异步方法, 开始正常播放当前所打开的视频(被打开、被停止的、被暂停的、被快放的、被后退的视频), 但是不改变视频的播放时间、位置和模式等属性。

若播放媒体成功, 则向页面发送[5205]EIS\_VOD\_PLAY\_SUCCESS。

若播放媒体失败, 则向页面发送[5206]EIS\_VOD\_PLAY\_FAILED。

参数: 无

返回值: 无

接口要求:

1) 被打开、被停止的媒体都处于待命状态中, 这时播放是从开始处将0帧/s的速度改为正常速度开始播放;

2) 被暂停的媒体是在暂停当前点将0帧/s的速度改为正常速度开始播放;

3) 被快放的媒体是在播放当前点将快进的速度改为正常速度开始播放;

4) 被后退的媒体是在播放当前点将倒退的速度改为正常前进的速度开始播放。

#### 6. media.AV.duration

原型: media.AV.duration

描述: 获取播放媒体总的持续时间。

权限: 只读

类型: 数值型

单位: 秒

取值: 无

#### 7. media.AV.elapsed

原型: media.AV.elapsed

描述: 返回当前实时流播放的位置(离开始点的时间距离)

权限: 只读

类型: 数值型

单位: 秒

取值: 无

#### 8. media.AV.speed

原型: media.AV.speed

描述: 获取当前视频播放速度。speed是随机可取的, 即speed值会随媒体播放速度改变而立刻改变。

权限: 只读

类型: 数值型

单位: 秒

取值: 当前速度, 大于1的正整数表示快进, 负整数表示几倍速度回退, 分数表示慢放。

取值如下:

2, 4, 8, 16: 表示多少倍速度快进;

1: 表示正常速度播放



- 0: 表示停止播放
- 1: 表示正常速度回退
- 2, -4, -8, -16: 表示多少倍速度快退
- 1/2, 1/4, 1/8, 1/16: 表示以正常速度的多少进行慢放

#### 9. media.AV.progress

原型: media.AV.progress

描述: 进度条上显示的节目播放时间与节目总时长的比率, 便于进度条显示进度。

权限: 只读

类型: 数值型

单位: 百分率(%)

取值: [0,100]

#### 10. media.AV.forward()

原型: media.AV.forward()

描述: 设置快进播放

参数: 无

返回值: 无

接口要求:

从当前时间快进播放, 快进的倍数根据当前播放的倍数和按键同时决定, 有2, 4, 8, 16, 32。如: 当前倍数是1或非快进的其他情况, 执行快进后播放倍数就变为2, 再执行快进后播放倍数就变为4, 再执行快进后播放倍数就变为8, 再执行快进后播放倍数就变为16, 再执行快进后播放倍数就变为32, 这时再执行快进后播放倍数不变。

#### 11. media.AV.forward(n)

原型: media.AV.forward(n)

描述: 按照指定倍数进行快进播放。

参数: n,number型, 表示指定倍数。

返回值: 无

#### 12. media.AV.backward()

原型: media.AV.backward()

描述: 设置快退播放。

参数: 无。

返回值: 无。

接口要求:

从当前时间快退播放, 快退的倍数根据当前播放的倍数和按键同时决定, 有-2, -4, -8, -16, -32。如: 当前倍速是1或非快退的其他情况, 执行快退后播放倍速就变为-2, 再执行快退后播放倍速就变为-4, 再执行快退后播放倍速就变为-8, 再执行快退后播放倍速就变为-16, 再执行快退后播放倍速就变为-32, 这时再执行快退后播放倍数不变。

#### 13. media.AV.backward(n)

原型: media.AV.backward(n)

描述: 按照指定倍数进行快退播放。

参数: n,number型, 表示指定倍数。

返回值: 无

#### 14. media.AV.pause()

原型: media.AV.pause()

描述: 暂时停止正在播放的视频。

参数: 无

返回值: 无

---

15. `media.AV.seek(press)`

原型: `media.AV.seek(press)`

描述: 定位到某个进度开始播放媒体 (表示跳转到指定的进度开始正常播放), 针对整个电影或整个其他类型节目。

若播放媒体成功, 则向页面发送[5205]EIS\_VOD\_PLAY\_SUCCESS。

若播放媒体失败, 则向页面发送[5206]EIS\_VOD\_PLAY\_FAILED。

参数: `press` - 字符串型, 表示从指定进度开始播放节目, 这个参数值不能大于100%。

取值如: "18%"。

返回值: 无

16. `media.AV.seek(time)`

原型: `media.AV.seek(time)`

描述: 定位到`time`的时间开始播放媒体 (表示跳转到指定的时间开始正常播放), 针对整个电影或整个其他类型节目。

若播放媒体成功, 则向页面发送[5205]EIS\_VOD\_PLAY\_SUCCESS。

若播放媒体失败, 则向页面发送[5206]EIS\_VOD\_PLAY\_FAILED。

参数: `time` - 字符串类型, 格式: "HH:MM:SS", 表示从指定时间开始播放节目, 这个参数值不能大于整个节目的时间。

返回值: 无

17. `media.AV.status`

原型: `media.AV.status`

描述: 获取媒体目前的播放状态。

权限: 只读;

类型: 字符串型

单位: 无

取值:

"play" --播放中;

"pause"--暂停状态;

"forward"--快进;

"backward"--快退;

"repeat" --重绕;

"slow"--慢放;

"stop"--停止。

18. `media.AV.stop()`

原型: `media.AV.stop()`

描述: 将正在播放的、暂停的视频停止播放, 进入待命状态。

参数: 无。

返回值: 1: 表示成功 0: 表示失败。

19. `media.AV.close()`

原型: `media.AV.close()`

描述: 停止正在播放的、暂停的、停止的、快进的、后退的媒体源, 并且释放该媒体源信息及其相关信息。

若拆链成功, 则向页面发送[5211]EIS\_VOD\_RELEASE\_SUCCESS。

若拆链失败, 则向页面发送[5212]EIS\_VOD\_RELEASE\_FAILED

参数: 无。

返回值: 无。

20. 使用tv标签播放视频

原型: <tv>

描述: 根据标签内的属性播放相应的视频节目

属性: networkid 网络号

casystemid CA系统编号

ecmpid ECM编号

videopid 视频pid

videotype 视频类型,

audiopid 音频pid

audiotype 音频类型

pcrpid PCR的pid

freq 视频频点

sym 视频符号率

mod 调制方式

coords 视频所在位置, 分别代表left,top,width,height返回: 无.

示例: <tv networkid="2" casystemid="0" ecmpid="0" videopid="0x1F40" videotype="0x02" audiopid="0x1F41" audiotype="0x03" pcrpid="0x1F40" freq="387000000" sym="6900000" mod="64" play="all" coords="282,56,870,386">

备注: 还有两个消息

若时移频道或VOD影片播放到了点播开始的位置, 则向页面发送[5209]EIS\_VOD\_PROGRAM\_BEGIN。

若时移频道或VOD影片播放到了点播结束的位置, 则向页面发送[5210]EIS\_VOD\_PROGRAM\_END。

### E.1.3 采用全景中间件接口定义

使用全景中间件浏览器的内置JS对象jShow。

#### 1. vodSendAction

原型: number vodSendAction(action)

描述: 发送VOD操作, 例如对当前VOD点播发送停止、暂停、快进、快退等操作。

参数: action — number型, 表示操作类型, 取值如下:

——0 — 暂停播放。

——1 — 快进播放, 快进有4、12、24三种倍速, 执行顺序为1x -> 4x -> 12x -> 24x -> 1x。

——2 — 快退播放, 快退有-4、-12、-24三种倍速, 执行顺序为1x -> -4x -> -12x -> -24x -> 1x。

——3 — 停止播放。

——4 — 恢复播放。以正常1倍速进行播放。

返回: number型, 成功返回0, 失败返回其它。

#### 2. vodGetState(type)

原型: object vodGetState(type)

描述: 获取当前VOD点播状态信息, 包括VOD点播的播放倍速、播放时间、总时长等信息。

参数: type — number型, 表示操作类型, 这里固定取值为5。

返回: state — object型, 点播状态信息对象, 参数说明如下:

——speed — number型, 播放倍速, 取值为快退倍速-4、-12、-24, 正常倍速1, 快进倍速4、12、24。

——time — number型, 播放时间, 单位为秒。

——duration — number型, 总时长, 单位为秒。

——start\_time — number型, 起始播放时间, 单位为秒。

——str\_state — number型, 说明当前播放的状态, 取值如下:

- 
- 1 – 播放播放
  - 2 – 暂停播放
  - 3 – 快退播放
  - 4 – 快进播放
  - 5 – 播放结束

### 3. vodSetVodType(state)

原型: number vodSetVodType(state)

描述: 设置VOD点播模式。在VOD开始播放之前调用。

参数: state – number型, 这里固定取值为1。

返回: number型, 成功返回0, 失败返回其它。

### 4. vodSeek(time)

原型: number vodSeek(time)

描述: 指定时间点进行VOD点播的跳转播放。

参数: time – number型, 跳转的时间点, 单位为秒。

返回: number型, 成功返回0, 失败返回其它。

### 5. vodStartup(offering)

原型: number vodStartup(offering)

描述: 根据点播参数启动VOD点播。

参数: offering – string型, 多条点播节目参数采用|符号分隔, 各项点播参数采用&符号分隔。

返回: 无。

### 6. playByListIndex(type, index)

原型: playByListIndex(type, index)

描述: 根据直播频道类型和直播频道索引播放直播频道节目。

参数: type – number型, 取值如下:

——0 - 直播电视频道。

——1 - 收音机频道。

index – number型, 机顶盒数据库中频道表的频道索引。

返回: 无。

### 7. volume(vol)

原型: volume(vol)

描述: 设置音量值。

参数: vol – number型, 取值为0、1、2...98、99、100。

返回: 无。

### 8. volume()

原型: number volume()

描述: 获取音量值。

参数: 无。

返回: number型, 取值为0、1、2...98、99、100。

### 9. avWindow(area)

原型: avWindow(area)

描述: 设置视频显示区域。

参数: area – object型, 参数说明如下:

——x – number型, x坐标, 即显示区域的左上角横坐标, 单位为像素。

——y – number型, y坐标, 即显示区域的右上角纵坐标, 单位为像素。

---

——w - number型，显示区域的宽度，单位为像素。

——h - number型，显示区域的高度，单位为像素。

返回：无。

#### 10. avWindow()

原型：object avWindow()

描述：获取视频显示区域。

参数：无。

返回：object型，参数说明如下：

——x - number型，x坐标，即显示区域的左上角横坐标，单位为像素。

——y - number型，y坐标，即显示区域的右上角纵坐标，单位为像素。

——w - number型，显示区域的宽度，单位为像素。

——h - number型，显示区域的高度，单位为像素。

#### 11. avStopPlay()

原型：avStopPlay()

描述：停止直播频道节目播放。

参数：无。

返回：无。

#### 12. playByTs(ts)

原型：playByTs(ts)

描述：根据transportId（传输码流的标识符）和serviceId（传输流中的业务信息标识符）使用SI索引播放TS流。

参数：ts - object型，参数说明如下：

——networkid - number型，网络标识符。

——transportid - number型，传输码流的标识符。

——serviceid - number型，传输流中业务信息标识符。

——casystemid - number型，条件接收系统标识符。

——ecmid - number型，加密节目标识符。

——state - number型，播放状态，这里固定值为2。

返回：无。

#### 13. playByPid(pid)

原型：playByPid(pic)

描述：使用PID索引方式播放视音频流。

参数：pid - object型，参数说明如下：

——networkid - number型，网络标识符。

——videopid - number型，视频包标识符。

——audiopid - number型，音频包标识符。

——pcrpId - number型，pcr标识符。

——casystemid - number型，条件接收系统标识符。

——ecmid - number型，加密节目标识符。

——audiotype - number型，音频类型，默认为MPEG2。

——videotype - number型，视频类型，默认为MPEG2。

——frequency - number型，频率，单位为Hz。

——symbolrate - number型，符号率，单位为sps。

——modulation - number型，QAM调制方式。

——state - number型，播放状态，默认为ALL。

返回：无。

#### 14. fullScreen()

原型: fullScreen()  
描述: 设置视频全屏显示。  
参数: 无。  
返回: 无。

## E.2 JavaScript 脚本示例

### ● 播放直播码流的接口示例

```
//国标中间件接口
var mp = new MediaPlayer();
var id = mp.getPlayerInstanceId();
mp.bindPlayerInstance(id);
mp.setMediaSource("dvb://<original_network_id>.<transport_stream_id>.<services_id>[.
<component_tar>{&<component_tar>}]<enent_id>]{</path-element>}");
mp.setVideoDisplayMode(0);
var rect = new Rectangle();
rect.left = 80;
rect.top = 50;
rect.width = 200;
rect.height = 200;
mp.setVideoDisplayArea(rect);
mp.refresh(); //设置小窗口播放
mp.setVideoDisplayMode(1);
mp.refresh(); //设置全屏播放
mp.stop(); //停止播放
mp.unbindPlayerInstance(id);

//茁壮中间件接口
//调用直播视频接口分两种
//播放指定的电视节目
DVB.playLiveTV(frequency,symbolRate,modulation,servcieId)
//停止播放
DVB.stopAV()

//全景中间件接口
jShow.playByListIndex(0, 15); //播放指定电视节目
jShow.avWindow({x:350, y:120,w:550,h:375}); //设置小窗口播放
jShow.fullScreen(); //设置全屏播放
jShow.avStopPlay(); //停止播放
```

### ● 播放点播码流的接口示例

```
//国标中间件接口
//初始化
var paras1 =
"rtsp://10.27.65.80:18082/id=1979779&Preview=0&ServiceName=SiTV_SVOD1&ServiceType=
SVOD&passthru_ip=10.27.65.50&mod_app_ip=10.27.65.50&srm_ip=10.27.65.50&poster_server
_ip=10.27.65.50&lsc_comm_proxy_ip=10.27.65.50&session_gateway_ip=10.27.65.50&freq1=56
2000&sym_rate1=6900&qam_mode1=64&freq2=834000&sym_rate2=6900&qam_mode2=64&r
un_time=01:38:17&content_name=V1511211110037.mpg&ServiceCode=20021&category=biz_6
2977698/300962310276&provider=SiTV&ApplicationType=vod&purchaseType=2&protocol_typ
e=rtsp&rtsp=10.27.65.80:18082&rtsp_vendor=OCN.RTSP&streamTransMode=hfc_stream";

var mp = new MediaPlayer();
var id = mp.getPlayerInstanceId();
```

```

mp.bindPlayerInstance(id);
mp.setMediaSource(paras1);
//停止播放
$("#resultDiv").append("MediaPlayer.stop() return:"+ mp.stop() + "<br />");
$("#resultDiv").append("MediaPlayer.unbindPlayerInstance() id:"+ id + " return:"+
    mp.unbindPlayerInstance(id) + "<br />");
//获取播放时间点
var strTime = mp.getCurrentPlayTime();
$("#resultDiv").append("MediaPlayer.getCurrentPlayTime() time:"+ strTime + "<br />");
//获取播放状态
var strPlaymode = mp.getPlaybackMode();
$("#resultDiv").append("MediaPlayer.getPlaybackMode() json string:"+ strPlaymode + "<br />");
var jsonPlaymode = JSON.parse(strPlaymode);
$("#resultDiv").append("MediaPlayer.getPlaybackMode() PlayMode:"+ jsonPlaymode.PlayMode
    + " Speed:"+ jsonPlaymode.Speed + "<br />");
//获取总时长
var duration = mp.getMediaDuration();
$("#resultDiv").append("MediaPlayer.getMediaDuration() duration:"+ duration + "<br />");
//设置视频显示指定区域
$("#resultDiv").append("MediaPlayer.setVideoDisplayMode(0) return:"+
    mp.setVideoDisplayMode(0) + "<br />");
var rect = new Rectangle();
rect.left = 80;
rect.top = 50;
rect.width = 200;
rect.height = 200;
mp.setVideoDisplayArea(rect);
$("#resultDiv").append("MediaPlayer.setVideoDisplayArea window<br />");
$("#resultDiv").append("MediaPlayer.refresh() return:"+ mp.refresh() + "<br />");
//设置视频全屏显示
$("#resultDiv").append("MediaPlayer.setVideoDisplayMode(1) return:"+
    mp.setVideoDisplayMode(1) + "<br />");
$("#resultDiv").append("MediaPlayer.refresh() return:"+ mp.refresh() + "<br />");
//暂停播放
$("#resultDiv").append("MediaPlayer.pause() return:"+ mp.pause() + "<br />");
//设置音量
volume = mp.getVolume();
volume = volume+10 > 100 ? 100 : volume+10;
$("#resultDiv").append("MediaPlayer.setVolume() volume:"+ volume + " return:"+
    mp.setVolume(volume) + "<br />");
//设置、获取特技模式
$("#resultDiv").append("MediaPlayer.enableTrickMode(0) return:"+ mp.enableTrickMode(0)
    + "<br />");
$("#resultDiv").append("MediaPlayer.getTrickModeFlag() return:"+ mp.getTrickModeFlag()
    + "<br />");
//恢复播放
$("#resultDiv").append("MediaPlayer.resume() return:"+ mp.resume() + "<br />");
//4 倍速快进播放
mp.setPace(4);
$("#resultDiv").append("MediaPlayer.setPace(4)<br />");
//24 倍速快退播放
mp.setPace(-24);
$("#resultDiv").append("MediaPlayer.setPace(-24)<br />");
//跳转播放
var secSeek = prompt("seek seconds", 1);
var timestamp = secSeek+"-";

```

```

mp.seek(1,timestamp);
$("#resultDiv").append("MediaPlayer.seek(1, "+ timestamp + "<\/>");
//开始播放
$("#resultDiv").append("MediaPlayer.play() return: "+ mp.play() + "<\/>");

//请茁壮补充脚本示例
<script>
function $(id)
{
    return document.getElementById(id);
}

var isSeek = false;
media.video.fullScreen();

document.onsystemevent = grabEvent;
function grabEvent(event)
{
    var code = event.which;
    switch(code){
        case 5202: // open 成功
            $('msg').innerText = "open--系统消息: open 成功(5202)";
            play_1();
            break;
        case 5203: // open 链接服务器失败
            $('msg').innerText = "open--系统消息: open 服务器失败(5203)";
            close_1();
            break;
        case 5205: // play 播放媒体成功
            $('msg').innerText = "play--系统消息: play 播放媒体成功(5205)";
            getCurrProgress();
            break;
        case 5206: // play 播放媒体失败
            $('msg').innerText = "play--系统消息: play 播放媒体失败(5206)";
            close_1();
            break;
        case 5211: // close 成功
            $('msg').innerText = "close--系统消息: close 关闭成功(5211)";
            break;
        case 5212: // close 失败
            $('msg').innerText = "close--系统消息: close 关闭失败(5212)";
            break;
        case 5210: // 播放完毕
            $('msg').innerText = "finish--系统消息: 播放完毕(5210)";
            close_1();
            break;
        case 5209: // 播放完毕
            $('msg').innerText = "begin--系统消息: 播放到头了(5209)";
            play_1();
            break;
        case 4404:
            $('msg').innerText = "播放的节目没有数据(4404)";
            break;
        default:

```



```

        $('msg').innerText = "捕获到其他的消息: "+code;
        break;
    }
    return 0;
}

function open_1()
{
    var playUrl = $('root').value;

    media.AV.open(playUrl,"vod");

    document.getElementById('state').innerText = "调用了 open 接口";
}

function play_1()
{
    if(media.AV.status == "play"){
        document.getElementById('msg').innerText = "当前已经是播放状态, 点击 play 无效";
    } else {
        media.AV.play();
        document.getElementById('state').innerText = "调用了 play 接口";
    }
}

function pause_1()
{
    if(media.AV.status == "pause"){
        document.getElementById('msg').innerText = "当前已经是播放状态, 点击 pause 无效";
    } else {
        media.AV.pause();
        document.getElementById('state').innerText = "调用了 pause 接口";
    }
}

function forward_1()
{
    media.AV.forward();

    document.getElementById('state').innerText = "调用了 forward 接口";
}

function backward_1()
{
    media.AV.backward();

    document.getElementById('state').innerText = "调用了 backward 接口";
}

function seek_1()
{
    var n = $('progress').value;

```

```

media.AV.seek(n);

document.getElementById('state').innerText = "调用了 seek(进度) 接口";
}

function seek_2()
{
    var hour_str = $('hour').value;
    var min_str = $('min').value;
    var sec_str = $('sec').value;

    var hour = parseInt(hour_str);
    var min = parseInt(min_str);
    var sec = parseInt(sec_str);
    var time = sec + min*60 + hour*60*60;
    var n = hour_str+":"+min_str+":"+sec_str;
    //iPanel.debug(" seek_2 time == "+time);
    //iPanel.debug(" seek_2 media.AV.duration == "+media.AV.duration);
    if(hour_str.length != 2){
        document.getElementById('msg').innerText = "小时的格式不对, 应该为 2 位, 请重新
输入";
    }else if(min_str.length != 2){
        document.getElementById('msg').innerText = "分钟的格式不对, 应该为 2 位, 请重新
输入";
    }else if(sec_str.length != 2){
        document.getElementById('msg').innerText = "秒的格式不对, 应该为 2 位, 请重新输
入";
    }else if(hour > 24 || min >= 60 || sec >= 60){
        document.getElementById('msg').innerText = "时间格式不对, 超出范围, 请重新输入
";
    }else if(time > media.AV.duration){
        document.getElementById('msg').innerText = "输入的时间超出视频总时长的范围, 请
重新输入";
    }else{
        iPanel.debug(" seek_2 n == "+n +", typeof(n) == "+ typeof(n));
        media.AV.seek(n);
        document.getElementById('state').innerText = "调用了 seek(时间) 接口";
    }
}

function close_1()
{
    media.AV.close();
    document.getElementById('state').innerText = "调用了 close 接口";
}

function exitPage(){
    clearTimeout(timer);
    media.AV.close();
}

var timer = null;

function getCurrProgress()
{

```

```

var elapsed = media.AV.elapsed;
var sss = media.AV.progress;
var duration = media.AV.duration;
var spe = media.AV.speed;
var status = media.AV.status;
document.getElementById('durat').innerText = duration
+"S("+secondToStringTime(duration)+")";
document.getElementById('elsp').innerText = elapsed
+"S("+secondToStringTime(elapsed)+")";
document.getElementById('gress').innerText = sss+"%";
document.getElementById('spe').innerText = spe+"倍";
document.getElementById('stat').innerText = status ;
clearTimeout(timer);
timer = setTimeout("getCurrProgress()",1000);
}

function secondToStringTime(__sec){
var hour = Math.floor(__sec/3600);
var minute = Math.floor((__sec - hour*3600)/60);
var second = __sec - hour*3600 - minute*60;
hour = hour>9?hour:"0"+hour;
minute = minute>9?minute:"0"+minute;
second = second>9?second:"0"+second;
return hour+"."+minute+"."+second;
}

</script>
<body bgcolor="transparent" leftmargin="0" topmargin="0" onUnload="exitPage();">
<div id="play"
style="position:absolute;width:620px;height:140px;top:360px;left:10px;background:url(..image/bg.jpg);">
<table style="width:100%;height:100%; cellpadding="0" cellspacing="0">
<tr>
<td colspan="8" height="40px">
路径:
<select id="root">
<option value="rtsp://192.168.37.68:9554/purchaseToken=ocn.com-6204-VOD">test1</option>
<option value="rtsp://192.168.37.68:9554/purchaseToken=ovt.com-6232-VOD">test2</option>
</select>
seek 进度:<input type="text" class="input3" id="progress"/>
seek 时间: <input type="text" class="input4" id="hour" value="00"/>:<input
type="text" class="input4" id="min" value="00"/>:<input type="text" class="input4" id="sec"
value="00"/>
</td>
</tr>
<tr height="40px">
<td colspan="8">
<input type="button" value="Open" onClick="open_1()"/>
<input type="button" value="Close" onClick="close_1()"/>
<input type="button" value="Play" onClick="play_1()"/>
<input type="button" value="Pause" onClick="pause_1()"/>
<input type="button" value="Forward" onClick="forward_1()"/>
<input type="button" value="Backforward" onClick="backward_1()"/>
<input type="button" value="seek(进度)" onClick="seek_1()"/>
<input type="button" value="seek(时间)" onClick="seek_2()"/>
</td>
</tr>

```

```

        </tr>
        <tr>
            <td colspan="8" align="left" height="60px">
                当前动作: <span id="state" style="color:blue;font-size:16px;"></span> |
                捕获信息: <span id="msg" style="color:red;font-size:18px;"></span><br>
                播放进度: <span id="gress" style="color:blue;font-size:16px;"></span> |
                总时长: <span id="durat" style="color:blue;font-size:16px;"></span> |
                播放状态: <span id="stat" style="color:blue;font-size:16px;"></span> <br>
                播放位置: <span id="elsp" style="color:blue;font-size:16px;"></span> |
                播放速度: <span id="spe" style="color:blue;font-size:16px;"></span>
            </td>
        </tr>
    </table>
</div>
</body>

//全景中间件接口
//开始播放
jShow.vodSetVodType(1);
var r = jShow.vodStartup(paras1);
//停止播放
var r = jShow.vodSendAction(3);
//获取播放状态
var objState = jShow.vodGetState(5);
$("#resultDiv").append("vodGetState(5) speed:"+ objState.speed + " time:"+ objState.time + "
duration:"+ objState.duration + " audio_track:"+ objState.audio_track + " start_time:"+
objState.start_time + " str_state:"+ objState.str_state + "<br />");
//设置全屏显示
jShow.fullScreen();
//设置指定区域显示
jShow.avWindow({x:422, y:158, w:490, h:350});
//暂停播放
var r = jShow.vodSendAction(0);
//设置音量
volume = jShow.volume();
volume = volume+1 > 100 ? 100 : volume+1;
jShow.volume(volume);
//快退播放
var r = jShow.vodSendAction(2);
//快进播放
var r = jShow.vodSendAction(1);
//恢复播放
var r = jShow.vodSendAction(4);
//跳转播放
var secSeek = prompt("seek second to", 1);
var r = jShow.vodSeek(secSeek);

```

附录 F  
(规范性附录)  
DVB-SI 信息获取接口定义

## F.1 接口定义

当互动信息服务中需要调用机顶盒的直播频道信息和节目信息时，需要用到该接口定义。针对东方有线NGB网络内的三类高清机顶盒终端，其SI信息获取接口分别有三种，因此互动信息服务的页面中要求同时支持以下三种SI接口定义：

- 针对2013年及之后发放的各类高清机顶盒（含智能电视机顶盒）：采用NGB-H国标接口定义（必选支持）
- 针对已有茁壮高清机顶盒：采用茁壮中间件接口定义
- 针对已有全景高清机顶盒：采用全景中间件接口定义

### F.1.1 采用 NGB-H 国标接口定义

1. ChannelManager.getChannelByLogicalID

原型：Channel ChannelManager.getChannelByLogicalID(logicalId)

描述：根据逻辑频道号获取频道对象

参数：logicalId – string型，表示逻辑频道号，必须是大于0的十进制整数

返回：Channel对象

2. Channel.getService

原型：DvbService getService()

描述：获取当前频道对象对应的DvbService对象

参数：无

返回：DvbService对象

3. DvbService对象的属性定义表如下

属性名称	类型	属性	说明
original_network_id	number	只读	表示业务所在的原始网络ID。
transport_stream_id	number	只读	表示业务所在传送流ID。
service_id	number	只读	表示业务ID。

### F.1.2 采用茁壮中间件接口定义

4. user.channels.getChannelByNum(num,type)

原型：user.channels.getChannelByNum(num,type)

描述：根据输入的频道号获取相关的频道数据

参数：num – 整型，代表频道号；

type – 整型，可选，代表指定类型。取值，1 表示视频；2 表示音频；6 表示 mosaic；12 表示 OC。；

返回：object型，不存在频道时，返回undefined

5. channel.getService()

原型：channel.getService()

描述：根据相应的频道号获取相关的 Service

参数：无

返回：object 型，service 对象

6. service.presentProgram

原型：service.presentProgram

描述：获取相应service下的获取当前的节目信息

参数：无

返回：object型，program对象

7. service.followingProgram

原型: service.followingProgram  
 描述: 获取相应的service下的获取下一个的节目信息  
 参数: 无  
 返回: object型, program对象  
 service.programs原型: service.programs  
 描述: 获取相应的service下的所有的节目的列表  
 参数: 无  
 返回: object 型, program 对象数组

### F.1.3 采用全景中间件接口定义

使用全景中间件浏览器的内置JS对象jShow。

1. dbSQLQuery(sql)  
 原型: object dbSQLQuery(sql)  
 描述: 根据输入的sql语句, 获取数据库中符合查询条件的数据集  
 参数: sql – string型, sql语句字符串  
 返回: object型, 参数说明如下:  
 ——length – number型, 数据集的总数。
2. dbQueryResult(index)  
 原型: object dbQueryResult(index)  
 描述: 根据索引号获取数据集中的记录, 必须在dbSQLQuery()方法之后调用。  
 参数: index – number型, 数据集中的索引号, 取值为0~总数。  
 返回: object, 参数说明如下:  
 ——error – number型, 成功查询到记录返回0, 失败返回其他。  
 ——其他参数 – string型, 与sql语句中字段对应。
3. currentMediaType()  
 原型: string currentMediaType()  
 描述: 获取当前播放频道的媒体类型  
 参数: 无  
 返回: string, 0表示电视频道, 1表示收音机频道
4. currentChannelIndex()  
 原型: string currentChannelIndex()  
 描述: 获取当前播放频道的频道索引  
 参数: 无  
 返回: string, 频道索引值
5. 数据库频道表table\_ch的表结构

字段名	类型	说明
ch_id	number型	主键流水号
ch_index	number型	频道索引, 取值从0开始, playByListIndex()方法使用该参数播放直播频道节目。
ch_name	string型	频道名称, 可用于UI界面显示。
ch_bt	number型	频道类型, 0: 直播电视频道、1: 收音机频道, 可用于查询条件。
ch_l_num	number型	逻辑频道号, 可用于UI界面显示。
ch_code	string型	频道编码, 可用于UI界面显示。
ch_ads_groupnum	number型	频道广告分组号, 可用于广告编排。

#### 6. 数据库节目表table\_et的表结构

字段名	类型	说明
et_id	number型	主键流水号。
et_name	string型	节目名称, 可用于UI界面显示。
et_s_time	string型	节目开始的日期时间, 格式为

		yyyy/MM/dd-hh:mm:ss, 可用于UI界面显示。
et_e_time	string型	节目结束的日期时间, 格式为 yyyy/MM/dd-hh:mm:ss, 可用于UI界面显示。
et_detail	string型	节目详细信息, 可用于UI界面显示。

## F.2 JavaScript 脚本示例

```
//国标中间件接口示例
//get Channel
var channel = ChannelManager.getChannelByLogicalID(1);
//get DvbService and format data source
var srv = channel.getService();
var url = "dvb://" + srv.original_network_id + "." + srv.transport_stream_id + "." + srv.service_id;
mp.setMediaSource(url);

//茁壮中间件接口示例
var curr_channel = user.channels.getChannelByNum(0); //假设打开 0 频道
var service = curr_channel.getService();
var Pprogram = service.presentProgram; //当前节目
var followingProgram = service.followingProgram; //下一个节目
var tempPrograms = service.programs; //获取该频道的所有节目
for(var i = 0; i < tempPrograms.length; i++){
    var tmpProgram = tempPrograms[i]; //列出每一个节目
}

//全景中间件接口示例
//机顶盒数据库中 table_ch 表中的所有字段
var tableChField = ["ch_id", "ch_index", "ch_bt", "ch_bt_name", "ch_ot", "ch_ot_name",
    "ch_name", "ch_freq", "ch_encrypted", "ch_fav", "ch_vt", "ch_name_a", "ch_lock",
    "ch_hide", "ch_bt_special", "ch_l_num", "ch_display_type", "ch_ads_groupnum", "ch_code",
    "ch_hotel"];
//页面打印显示的字段
var htmlDisplayField = ["ch_index", "ch_name", "ch_l_num", "ch_ads_groupnum", "ch_code"];
//数据库查询语句的拼接
var sql = "select ";
for (a = 0; a < tableChField.length; a++) {
    sql += tableChField[a] + " ";
}
sql += "from table_ch where ch_bt = 0";
htmlInfo = htmlInfo + "sql: " + sql + "<br />";
//查询数据集的总数
var totalCount = jShow.dbSQLQuery(sql);
htmlInfo = htmlInfo + "total count: " + totalCount.length + "<br />";
//显示的字段名
for (c = 0; c < htmlDisplayField.length; c++) {
    htmlInfo += htmlDisplayField[c] + "-";
}
for(var i=0; i < totalCount.length; i++){
    var result = jShow.dbQueryResult(i);
    if (result.error == 0) {
        htmlInfo += "<li><a href='#">";
        for (c=0; c < htmlDisplayField.length; c++) {
            eval("htmlInfo += result." + htmlDisplayField[c] + " + \"-\"");
        }
    }
}
```

```

        htmlInfo += "</a></li>";
    }
}
htmlInfo = htmlInfo + "</ol>";
$("#PageBody").append(htmlInfo);

var curr_channel = "0";
var sql = "select et_id et_name et_s_time et_e_time et_detail et_remind et_rec_remind from
table_et where et_ch_index = " + curr_channel + " and et_ch_bt = 0";
var rs = jShow.dbSQLQuery(sql);
for (i = 0; i < rs.length; i++) {
    var result = jShow.dbQueryResult(i);
    arr[i] = {
        id: result.et_id,
        stime: result.et_s_time,
        etime: result.et_e_time,
        name: result.et_name,
        detail: result.et_detail
    };
}
}

```

## 附录 G (规范性附录) 机顶盒 MAC 地址获取接口定义

### G.1 接口定义

当互动信息服务中获取机顶盒的MAC地址时，需要用到该接口定义。针对东方有线NGB网络内的三类高清机顶盒终端，其MAC地址获取接口分别有三种，因此互动信息服务的页面中要求同时支持以下三种接口定义：

- 针对2013年及之后发放的各类高清机顶盒（含智能电视机顶盒）：采用NGB-H国标接口定义（必选支持）
- 针对已有茁壮高清机顶盒：采用茁壮中间件接口定义
- 针对已有全景高清机顶盒：采用全景中间件接口定义

#### G.1.1 采用 NGB-H 国标接口定义

1. Broadband.getAllEthernets()  
原型：Ethernet[] Broadband.getAllEthernets()  
描述：获取接收终端所有的网络设备对象  
参数：无  
返回：Ethernet对象数组
2. Ethernetd对象的属性定义表如下：

属性名称	类型	属性	说明
MACAddress	string	只读	表示当前网卡6个字节的物理地址，每两个十六进制数之间以“-”分隔开，例如“00-15-F2-63-7A-83”。

#### G.1.2 采用茁壮中间件接口定义

1. network.ethernets[0].MACAddress  
原型：network.ethernets[0].MACAddress  
描述：获取当前机顶盒 MAC 地址  
参数：无



---

返回：string型，以“-”分隔开。

### G.1.3 采用全景中间件接口定义

使用全景中间件浏览器的内置JS对象jShow。

1. `systemInfo()`

原型：object `systemInfo()`

描述：获取系统信息对象

参数：无

返回：object型，参数说明如下：

——mac – string型，机顶盒MAC地址，以“-”分隔开。

### G.2 JavaScript 脚本示例

```
//茁壮中间件接口示例
var stbMac = network.ethernets[0].MACAddress;

//全景中间件接口示例
stbMac = jShow.systemInfo().mac;

//国标中间件接口示例
var ethernets = Broadband.getAllEthernets();
if (ethernets.length > 0) {
    var ethernet = ethernets[0];
    alert(ethernet.MACAddress.replace(/-/g, ""));
    mac = ethernet.MACAddress.replace(/-/g, "");
} else {
    mac = ethernet.MACAddress.replace(/-/g, "");
}
```

## 附录 H (规范性附录)

### OCN 特定应用接口定义

如下应用已完成开发并上线，当机顶盒要求支持这些应用中，需要支持以下接口定义。

#### H.1 信息发布应用接口定义

在东方有线智慧闵行的信息发布应用中采用了http方式的系统接口。所有接口地址URL通过节目导视应用下发数据进行修改。

##### H.1.1 接口说明

- 所有 url 请求中 params 的 value，对于其中的中文都需要进行编码，也就是 E4%B8%AD%E6%96%87 这样的标准格式。
- 对于日期类接口，要求统一转换成 iso8601 格式也就是“2012-06-26T10:31:00+08:00”。
- 由于目前接口需要跨域使用，所以这里所有 json 接口都会提供 jsonp 支持，回调为 callback。如果是 jquery 的话直接使用 jsonp 作为参数就可以了。若无法使用 jquery 请参考 jquery 的 jsonp 接口实现。

##### H.1.2 接口定义

###### 1. 获取机顶盒所属用户的小区信息

说明：

查询当前机顶盒所属用户的小区信息

地址：

[URL\_PREFIX]/co\_pub/api/v1/accounts/{uid}.json

举例：

[http://10.27.89.9/co\\_pub/api/v1/accounts/025681bed76f.json?callback=?](http://10.27.89.9/co_pub/api/v1/accounts/025681bed76f.json?callback=?)

参数：

{uid}为机顶盒 mac 地址，格式为 025681bed76f

返回：

```
{
  "uid": "025681bed76f", //机顶盒 mac 地址
  "region_code": "mhxq-1" //这个就是该机顶盒所属用户小区代码，可能会包含中文
}
```

###### 2. 获取机顶盒所属用户的发布信息

说明：

获取头端发布给该机顶盒的所有信息。

地址：

[URL\_PREFIX]/co\_pub/api/v1/stacks/{stack\_id\_or\_code}/entries.json[?[PARAMS1=VALUE1]]

举例：

[http://10.27.89.9/co\\_pub/api/v1/stacks/mh-sqgg/entries.json?available\\_at=2013-07-19T11:31:00+08:00&region\\_code eq=mhxq\\_1&created\\_at\\_gt=2013-07-19T10:31:00+08:00&callback=?](http://10.27.89.9/co_pub/api/v1/stacks/mh-sqgg/entries.json?available_at=2013-07-19T11:31:00+08:00&region_code eq=mhxq_1&created_at_gt=2013-07-19T10:31:00+08:00&callback=?)

参数：

{stack\_id\_or\_code}为所要查询内容的分类，这里固定为 mh-sqgg。

PARAMS 参数 available\_at: 查询该时间之前所有的发布信息，一般传入机顶盒当前时间，取值为 iso8601 标准的日期时间组合格式，例如 2013-07-19T10:31:00+08:00。

PARAMS 参数 created\_at\_gt: 查询该时间之后所有的发布信息，可以传入最近一次成功请求后的 last\_created\_at 返回参数，取值为 iso8601 标准的日期时间组合格式，例

如 2013-07-19T10:31:00+08:00。如果是第一次请求，则传入 0。

PARAMS 参数 region\_code\_eq: 机顶盒所属用户的小区代码。

返回:

```
{
  last_created_at: "2012-07-22T00:00:00+08:00", //所有发布信息中最新一个信息的
  创建时间, 不存在为 null
  "entries": [{ //发布信息数据集
    "title": "空气质量周评（2012 年 6 月 19 日~2012 年 6 月 25 日）", //信息名
    称
    "released_at ": "2012-06-26T10:31:00+08:00", //信息发布时间
    "content": "可吸入颗粒物 38 \r\n 二氧化硫 18 \r\n 二氧化氮 25 ",//信息详
    细内容
    "updated_at": "2012-07-20T00:00:00+08:00", //信息更新时间
    "created_at": "2012-07-20T00:00:00+08:00" //信息创建时间
  },{
    "title": "闵行区城市规划建设讲师团赴云南保山讲课 ",
    "released_at ": "2012-06-26T03:28:00+08:00",
    "content": "6 月 14 日至 20 日，闵行区第二批对口支援讲师...",
    "updated_at": "2012-07-22T00:00:00+08:00",
    "created_at": "2012-07-22T00:00:00+08:00"
  }]
}
```

## H.2 电视账单应用接口定义

在东方有线电视账单应用中采用了该 JavaScript 接口。

### 1. OCN.TVBill.setAccount(Domain\_name,address, password)

说明:

用于设置邮件服务器、邮件账户等各项参数。该接口调用仅作保存动作，以备后续访问服务器时使用。

输入:

Domain\_name: 字符串，邮件域名，如@ocn.com

address: 字符串，邮件服务器的 IP 地址

Password: 字符串，邮件账户密码，默认为 123

输出:

整型。0 代表成功，其他为失败

样例:

```
OCN.TVBill.setAccount("@ocn.com","10.0.3.203","123");
```

### 2. OCN.TVBill.setReceiveInterval(IntervalTime)

说明:

设置机顶盒终端访问头端邮件服务器时间间隔。机顶盒终端根据这个时间间隔，周期访问头端邮件服务器。

输入:

IntervalTime: 整型，单位 分钟，取值范围 0~1440 分钟。0 表示不需要定时判断是否有新邮件

输出:

整数。0 代表成功，其它为失败

样例:

```
OCN.TVBill.setReceiveInterval(10); //设置 10 分钟检测一次是否有新邮件
```

### 3. OCN.TVBill.refreshTime()

说明:

更新保存在机顶盒终端里的校验时间。该接口每次进入头端电视账单应用系统时调用,即由头端来调用,调用该函数后,机顶盒终端将校验时间更新为当前时间。校验时间用来与邮件服务器该账户最新一封邮件的接收时间进行比较,以判断是否有新邮件。

输入:

无

输出:

整型。0 代表成功,其他为失败

备注:

特殊情况,机顶盒首次开机时,OCN.TVBill.refreshTime()在没调用时,使用默认值0,只要从邮件服务器上能获取邮件,都判断为有新邮件,

### 4. OCN.TVBill.newBillFlag();

说明:

机顶盒终端是否有新邮件的状态值。该接口只是获取机顶盒终端保存的状态值,不是要求机顶盒终端马上去判断是否有新邮件。

输入:

无

输出:

整形。1 表示有新邮件,0 表示没有新邮件

备注:

当判断有新邮件后,在机顶盒节目导视 miniEPG 菜单的频道号左侧显示新账单提醒图标。

该应用的说明:

机顶盒终端按 IntervalTime 时间间隔,定期访问头端邮件服务器,以获得头端邮件服务器中最后一封邮件的时间。需要做的工作如下:

- 访问头端邮件服务器时需要用如下参数:MAC 地址为用户名(类似:0019684372ac 的字符串),域名、IP 地址、密码(之前已经设置的参数);
- 通过 pop3 协议获取头端邮件服务器上该用户的邮件列表;
- 判断邮件列表中的最近一份邮件的时间,保存这个最新时间;
- 再将最新时间跟校验时间进行对比,如果最新时间更新,则更新机顶盒终端的新邮件状态值。

---

### 规范修订记录

日期	版本	作者	修改描述
2013/8/22	V1.0.1	陈宝霞、李嘉晨、陈志	修改稿，全文修订
2013/12/26	V1.0.2	李嘉晨	(1) 更新附录 C：退出键的国标键值更新为 27 (IRKEY_ESC/取消键) (2) 更新附录 F：新增国标接口定义、补充全景接口定义 (3) 新增附录 G

CONFIDENTIAL