# JavaScript API for controlling IPTV devices

# MAG100 and MAG200

# Specification

V. 1.20

2012

## *Contents.*

## *About this document.*

| | |
|---|---|
| Document revision | 1.20 |
| JavaScript API version | 325 |
| STB API version | 130 |
| MAG200 player version | 0x555 |
| MAG100 player version | 0x23 |

## *Overview.*

This document describes the program interface allowing controlling IPTV-device (including playing various types of video content and the event pattern of the IPTV-device) from JavaScript. The document assumes the knowledge of JavaScript.

## API use

It is assumed that the functions described are used from the JavaScript context on MAG100/MAG200 supplied with Mozilla Firefox 1.5 or WebKit as the browser.

### Data types and arguments format.

Hereinafter the following designations shall be used:
**int** – for digital types.
**bool** – for logical types.
**string** – for string types.

In this document it is understood as follows: if the argument type preceded by the keyword **out**, this parameter is used to return values from the function. It is sufficient to call the function with an empty object as this parameter from JavaScript and then receive the value from the field **value** of this object.
For example:

```
var tColor;
var x = {} ;
stb.GetTransparentColor(x);
tColor = x.value;
```

The example of use and initialization of API and IPTV-device events are described in appendix 1 and in the chapter Configuring the event system.
Any operations with IPTV-device are performed via the objects **stb** and **stbEvent**.

### stb object methods calling

More than one prototype of the object stb method can be described, due to different mechanisms of returning the result of the method operation. In this case the prototype shall be preceded with the following designations:

**FireFox** – the prototype is used when the method is called from the Mozilla FireFox browser.
**WK/FF+Wrapper** –The prototype is used when the method is called from the WebKit-based browser or from the Mozilla FireFox browser via **wrapper.js**.

To call any **stb** method from any JavaScript function, add the following string in the beginning of this function:

```
netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")
```

This rule is valid only when calling the method from the Mozilla FireFox browser without using **wrapper.js**.

### *Availability of the functions on IPTV-devices.*

Interface functions, their availability and specifics for each IPTV-device are shown below. The functions marked with "–" are present in API for compatibility but they do not provide full functionality.

**Table 1 Compatibility of the JavaScript API functions for MAG100 and MAG200**

| **stb** object methods | MAG100 | MAG200 |
|---|---|---|
| InitPlayer | + | + |
| DeinitPlayer | + | + |
| Play | + | + |
| Play using proxy server | - | +(WK) |
| PlaySolution | + | + |
| Stop | + | + |
| Pause | + | + |
| Continue | + | + |
| SetPosTime | + | + |
| SetPosTimeEx | – | +(WK) |
| SetPosPercent | + | + |
| SetPosPercentEx | + | + |
| GetPosTime | + | + |
| GetPosTimeEx | – | +(WK) |
| GetPosPercent | + | + |
| GetPosPercentEx | + | + |
| GetMediaLen | + | + |
| GetMediaLenEx | – | +(WK) |
| SetSpeed | + | + |
| SetAudioPID | + | + |
| SetPIG | + | + |
| SetAlphaLevel | + | + |
| SetVolume | + | + |
| SetUserFlickerControl | + | – |
| SetFlicker | + | + (distinction from MAG100) |
| SetDefaultFlicker | + | + (distinction from MAG100) |

| **stb** object methods | MAG100 | MAG200 |
|---|---|---|
| SetLoop | + | + |
| SetVideoControl | + | + |
| SetVideoState | + | + |
| SetChromaKey | + | + |
| SetMode | + | + |
| SetWinMode | + | + |
| SetTopWin | + | + |
| SetWinAlphaLevel | + | + |
| SetAspect | + | + (add. capabilities) |
| Rotate | + | – |
| SetMute | + | + |
| SetMicVolume | + | – |
| GetMicVolume | + | – |
| GetVolume | + | + |
| Step | + | – |
| SetupRTSP | + | + |
| SetViewport | + | + |
| IsPlaying | + | + |
| Version | + | + |
| SetupSPdif | + | + |
| SetSubtitles | + | + |
| SetSubtitlesSize | + | + |
| SetSubtitlesFont | + | + |
| SetSubtitlesOffs | + | + |
| GetSpeed | + | + |
| GetAudioPID | + | + |
| GetPIG | + | + |
| GetAlphaLevel | + | + |
| GetWinAlphaLevel | + | + |
| SetTransparentColor | + | + |
| GetTransparentColor | + | + |
| IgnoreUpdates | + | + |
| ExecAction | + | + |

| **stb** object methods | MAG100 | MAG200 |
|---|---|---|
| SetCASType | + | + |
| SetCASParam | + | + |
| SetAdditionalCasParam | – | + |
| LoadCASIniFile | + | + |
| SetCASDescrambling | – | + |
| GetAspect | + | + |
| StandBy | + | + |
| RDir | + | + |
| SetAudioLangs | + | + |
| GetAudioPIDs | + | + |
| GetSubtitlePIDs | + | + |
| ReadCFG | + | + |
| WriteCFG | + | + |
| WritePrefs | + | + |
| Debug | + | + |
| SetListFilesExt | - | +(WK) |
| ListDir | - | +(WK) |
| SetBrightness | - | +(WK) |
| SetSaturation | - | +(WK) |
| SetContrast | - | +(WK) |
| GetBrightness | - | +(WK) |
| GetSaturation | - | +(WK) |
| GetContrast | - | +(WK) |
| GetSubtitlePID | - | +(WK) |
| SetSubtitlePID | - | +(WK) |
| SetSubtitleLangs | - | +(WK) |
| DeleteAllCookies | - | +(WK) |
| SetAudioOperationalMode | - | +(WK) |
| SetHDMIAudioOut | - | +(WK) |
| SetDRC | - | +(WK) |
| SetStereoMode | - | +(WK) |
| EnableJavaScriptInterrupt | - | +(WK) |
| ShowSubtitle | - | +(WK) |

| **stb** object methods | MAG100 | MAG200 |
|---|---|---|
| GetMute | - | +(WK) |
| StartLocalCfg | - | +(WK) |
| ShowVirtualKeyboard | - | +(WK) |
| HideVirtualKeyboard | - | +(WK) |
| EnableServiceButton | - | +(WK) |
| EnableVKButton | - | +(WK) |
| EnableSpatialNavigation | - | +(WK) |
| EnableSetCookieFrom | - | +(WK) |
| SetBufferSize | - | +(WK) |
| GetBufferLoad | - | +(WK) |
| SetWebProxy | - | +(WK) |
| GetVideoInfo | - | +(WK) |

WK – only for WebKit.

## *stb object methods.*

### stb.InitPlayer

void InitPlayer()

Initializes the player. Call this function before using the player. The features are described in Appendix 1. API usage.

**Parameters:**

None.

**Returned value:**

None.

### stb.DeinitPlayer

void DeinitPlayer()

De-initialize the player.

**Parameters:**

None.

**Returned value:**

None.

### stb.Play

void Play(string playStr)

Start playing media content as specified in **playStr.**

**Parameters:**

**playStr** – string in the form: “<solution> <URL> [atrack:<anum>] [vtrack:<vnum>] [strack:<snum>] [subURL:<subtitleUrl>]“

| Parameter | Allowed value | Description |
|---|---|---|
| Solution | rtp, rtsp, mp3, auto, mpegps, mpegts, mp4 | Media content type. Depends on the IPTV-device type. See Appendix 2 for the table of supported formats and the description of media content types |

| Parameter | Allowed value | Description |
|---|---|---|
| URL | | Address of the content to be started for playing. Depends on the type. See more detailed information in Appendix 2. |
| atrack:<anum> | | Sets the number(PID) of audio track. Optional parameter. |
| vtrack:<vnum> | | Sets the number(PID) of audio track. Optional parameter |
| strack:<snum> | | Sets the number(PID) of subtitle track. Optional parameter |
| subURL: <subtitleURL> | | Sets the URL of external subtitles file. See stb.LoadExternalSubtitles Optional parameter |

**Returned value:**

None.

### stb.Play using proxy server

void Play(string playStr, string proxy_params)

Start playing media as described by **playStr**, using given proxy server for http playback.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| playStr | string | See stb.Play. |
| proxy_params | string in the following form:<br>'http://[username[:password]@]proxy_addr:proxy_port'<br>proxy_addr – proxy server address.<br>proxy_port – proxy server port.<br>username – username for proxy server.<br>password – password for proxy server..<br>Parameters in square brackets are optional and can be omited. | |

**Return value:**

None.

**Note.** Proxy server settings are valid till the next call of stb.Play().

**Note.** Proxy server settings affect only http playback.

### stb.PlaySolution

void PlaySolution(string solution, string URL)

Play media content of the preset type (**solution**) from the preset **URL**.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Solution | | Corresponds to the parameter **solution** from the function stb.Play |
| URL | | Address of the content to be started for playing. Depends on the type. See more detailed information in supplement 2. |

**Returned value:**

None.

### stb.Stop

void Stop()

Stops playing.

Continue() shall begin playing from the beginning.

**Parameters:**

None.

**Returned value:**

None.

### stb.Pause

void Pause()

Pauses current playback.

Continue() continues playing from the current position.

**Parameters:**

None.

**Returned value:**

None.

### stb.Continue

void Continue()

Continues playing (after Pause()) or begin anew (after Stop()).

**Parameters:**

None.

**Returned value:**

None.

### stb.SetPosTime

void SetPosTime(int time)

Sets the new position of playback in time

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Time | time >= 0 | The position in seconds from the beginning of the content where the playback should start (positioning in the content). |

**Returned value:**

None.

### stb.SetPosTimeEx

void SetPosTimeEx(int time)

Sets the current playback position in time, ms.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Time | time >= 0 | Position in ms from the beginning of the content where playback should start (positioning in the content) |

**Returned value:**

None.

### stb.SetPosPercent

void SetPosPercent(int prc)

Sets the current position in percent.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Prc | 0..100 | The position in percent of the total duration of the content where playback should start. |

**Returned value:**

None.

### stb.SetPosPercentEx

void SetPosPercentEx(int prc)

Set the current position in percent.

**Parameters:**

| Parameters | Returned value | Description |
|---|---|---|
| Prc | 0..10000 | Position in hundredth fractions of percent of the total duration of the content, from which the playback should start. |

**Returned value:**

None.

### stb.GetPosTime

**FireFox:** void GetPosTime(**out** int time);

**WK/FF+Wrapper:** int GetPosTime();

Gets the current position in time.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Time | | Current position in second from the beginning of content. |

### stb.GetPosTimeEx

**FireFox:** void GetPosTimeEx(**out** int time);

**WK/FF+Wrapper:** int GetPosTimeEx();

Gets the current position in time in ms

**Parameters:**

None.

**Returned value:**

| Parameter | Returned value | Description |
|---|---|---|
| Time | | The current position in ms from the beginning of content. |

### stb.GetPosPercent

**FireFox:** void GetPosPercent(**out** int prc);

**WK/FF+Wrapper:** int GetPosPercent();

Gets the current position in percent.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Prc | | The current position in percent of the whole duration of the content. |

### stb.GetPosPercentEx

**FireFox:** void GetPosPercentEx(**out** int prc);

**WK/FF+Wrapper:** int GetPosPercentEx();

Gets the current position in hundredth fractions of percent.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| Prc | 0..10000 | The current position in percent of the whole duration of content. |

### stb.GetMediaLen

**FireFox:** void GetMediaLen(**out** int len);

**WK/FF+Wrapper:** int GetMediaLen();

Gets the duration of the current content.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Len | | Total duration of the current content in seconds. |

### stb.GetMediaLenEx

**FireFox:** void GetMediaLenEx(**out** int len);

**WK/FF+Wrapper:** int GetMediaLenEx();

Gets the duration of the current content in ms.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Len | | Total duration of the current content in ms. |

### stb.SetSpeed

void SetSpeed(int speed)

Sets the rate of playing.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| Speed | -8..8 | Sets new playback speed:<br>1 - normal<br>2 - 2x<br>3 - 4x<br>4 - 8x<br>5 - 16x<br>6 - 1/2<br>7 - 1/4<br>8 – 12x<br>-1 – reverse<br>-2 - reverse 2x<br>-3 - reverse 4x<br>-4 - reverse 8x<br>-5 - reverse 16x<br>-8 - reverse 12x |

**Returned value:**

None.

## stb.SetAudioPID

void SetAudioPID(int pid)

Sets track number (PID) for audio.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Pid | | Sets the number or PID of the audio track to be played in the current content. If such track is absent the sound will be disabled. |

**Returned value:**

None.

### stb.SetSubtitlePID

void SetSubtitlePID(int pid)

Sets the number of track (PID) for subtitles.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Pid | | Set the number or PID for the subtitles track to be played in the current content. Is such track is absent subtitles will be disabled. |

**Returned value:**

None.

### stb.SetPIG

void SetPIG(int state,int scale,int x,int y)

Sets position and mode of video window.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| State | 0..1 | If state=1 – show the video on full screen.<br>If state=0 – show the video in the specified rectangle. |
| Scale | | The scale of the video window. The present multiplier of the video window size equals to **scale/256.** |
| X | | Horizontal offset of the upper left corner of the video window from the screen edge |
| Y | | Vertical offset of the upper left corner of the video window from the screen edge |

**Returned value:**

None.

### stb.SetAlphaLevel

void SetAlphaLevel(int alpha)

Sets alpha transparency of the video window.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Alpha | 0..255 | Transpareny of the video window:<br>0 – completely transparent;<br>255 – completely opaque. |

**Returned value:**

None.

### stb.SetVolume

void SetVolume(int volume)

Sets volume level.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Volume | 0..100 | Volume level:<br>0 – no sound;<br>100 – maximal level. |

**Returned value:**

None.

### stb.SetUserFlickerControl

void SetUserFlickerControl(int mode)

Sets the control mode of Flicker-filter.

Platform**: MAG100**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Mode | 0..1 | Control mode of flicker-filter:<br>0 – API user controls flicker-filter himself (see. stb.SetFlicker and stb.SetDefaultFlicker); |

| Parameter | Allowed value | Description |
|---|---|---|
| | | 1 – The player automatically switches on flicker-filter during pauses and stops and switches it off during playing. |

**Returned value:**

None.

### stb.SetFlicker

void SetFlicker(int state, int flk, int shp)

Sets Flicker-filter parameters.

Platforms: **MAG100,MAG200(**see. note**)**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| State | 0..1 | Flicker filter on/off<br>0 – switch off the flicker-filter;<br>1 – switch on the flicker-filter. |
| Flk | 0..15 | Flicker level. |
| Shp | 0..15 | Sharpness level. |

**Returned value:**

None.

**Note:**

Flicker filter on MAG200 is applicable only for graphic window, therefore it is advised to set it only once during loading and not to switch it off

**flk** and **shp** parameters are ignored for MAG 200

### stb.SetDefaultFlicker

void SetDefaultFlicker(int state)

Turns on/off flicker-filter with the default parameters.

**Platfoms: MAG100,MAG200(**see. note**)**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| state | 0..1 | Flicker-filter on/off: |

| Parameter | Allowed value | Description |
|---|---|---|
| | | 0 – switch off the Flicker-filter;<br>1 – switch on the Flicker-filter.<br>In this case default values for sharpness and flicker are set. |

**Returned value:**

None.

**Note:**

Flicker filter on MAG200 is applicable only for graphic window, this is why it is recommended to set its only once and keep it switched

### stb.SetLoop

void SetLoop(int loop)

Sets or cancels repeated playing.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Loop | 0..1 | 0 – switch off repeated playing on the content;<br>1 – switch on repeated playing on the content. |

**Returned value:**

None.

### stb.SetVideoControl

void SetVideoControl (int mode)

Sets the video window control mode:

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Mode | 0..1 | Control mode:<br>0 – the device automatically switches on the video window at the beginning of playing and switches it off when stops;<br>1 – API user uses stb.SetVideoState |

| Parameter | Allowed value | Description |
|---|---|---|
| | | for instructing whether to show the video window or not. |

**Returned value:**

None.

### stb.SetVideoState

void SetVideoState (int state)

Switch on or switch off the video window.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| State | 0..1 | Allow/prohibit video dicplay:<br>0 – video window is not dosplayed;<br>1 – video window is displayed if the stream is present. |

**Returned value:**

None.

Notes:

Valid only if user control had been allowed with **stb.SetVideoControl**.

### stb.SetChromaKey

void SetChromaKey(int key,int mask)

Set the preset colour and mask for using as ChromaKey (the transparency of any colour on the whole window).

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| key | 0..0xffffff | Sets the colour in RGB. If ythe colour of a window pixel coincides with this colour after masking, the pixel is considered transparent. |

| Parameter | Allowed value | Description |
|---|---|---|
| mask | 0..0xffffff | Set the mask for **key**. If the mask is equal to 0xffffff, the colour set by the parameter key is considered transparent. |

**Returned value:**

None.

Notes:

Any changes on the screen shall be visible only subject to switching on the regime ChromaKey by the functions **stb.SetMode or stb.SetWinMode**.

### stb.SetMode

void SetMode(int mode)

Switch on (mode=1) or switch off (mode=0) the mode ChromaKey for the video window.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| Mode | 0..1 | ChromaKey mode for the video window:<br>0 – off;<br>1 – on.<br>The parameters set by stb.SetChromaKey ostb.SetTransparentColor shall be valid if the on-mode is used. |

**Returned value:**

None

### stb.SetWinMode

void SetWinMode (int winNum, int mode)

Switch on or switch off the ChromaKey mode for the preset window

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| winNum | 0..1 | The number of the window for which |

| Parameter | Allowed value | Description |
|---|---|---|
| | | this function is used:<br>0 – graphic window;<br>1 – video window. |
| Mode | 0..1 | ChromaKey mode for video window:<br>0 – off;<br>1 – on.<br>The parameters set by stb.SetChromaKey or stb.SetTransparentColor shall be active in the on-mode |

**Returned value:**

None.

### stb.SetTopWin

void SetTopWin(int winNum)

Set the preset window over others.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| winNum | 0..1 | Number of the window for which this function is used:<br>0 – graphic window;<br>1 – video window. |

**Returned value:**

None.

### stb.SetWinAlphaLevel

void SetWinAlphaLevel(int winNum, int alpha)

Set alpha transparency of the preset window.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| winNum | 0..1 | Number of the window for which this |

| Parameter | Allowed value | Description |
|---|---|---|
| | | function is used:<br>0 – graphic window;<br>1 – video window. |
| alpha | 0..255 | Transparency of the preset window:<br>0 – completely transparent;<br>255 – completely opaque |

**Returned value:**

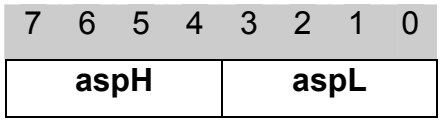
None.

## stb.SetAspect

void SetAspect(int aspect)

Set video picture format.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Aspect | | Sets the video picture format. Consists of 2 tetrads:<br>  <br><br>**aspH is ignored for MAG 100.** |
| aspL | 0..3 | Sets the aspect ratio:<br>0 – automatic;<br>1 – 20:9;<br>2 –16:9;<br>3 – 4:3. |
| aspH | 0..3 | Sets conversion of video format:<br>0 – as it is, video is stretched for the whole screen;<br>1 – Letter box mode, video is proportionally enlarged to the size of the screen along the larger edge;<br>2 – Pan&Scan mode, video is proportionally enlarged to the screen |

| Parameter | Allowed value | Description |
|---|---|---|
| | | size along the lesser edge;<br>3 – combined mode, intermediate between Letter Box Box and Pan&Scan.<br>4 – enlarged mode;<br>5 – optimal mode.<br>Only for MAG200 |

**Returned value:**

None.

Notes:

MAG100 ignores **aspH** .

MAG200 uses **aspL** only in windows mode, while aslH only in full screen mode, see. stb.SetPIG

### stb.Rotate

void Rotate(int angle)

Rotate video.

**Platform:** MAG100

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Angle | 0, 90, 180, 270 | Rotates the video window contents by the preset angle relative to the initial position. |

**Returned value:**

None.

### stb.SetMute

void SetMute(int mute)

Switch off or on the sound restoring the volume level.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| Mute | 0..1 | Switches on/switches off the sound:<br>0 – on;<br>1 – off. |
| After the cycle of switching off/on with this function is completed the volume level remains unchanged. | | |

**Returned value:**

None.

### stb.SetMicVolume

void SetMicVolume(int micvol)

Set the microphone volume level.

**Platform:** MAG100

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Micvol | 0..100 | Set the microphone volume level:<br>0 – minimal volume;<br>100 – maximal volume. |

**Returned value:**

None.

### stb.GetMicVolume

**FireFox:** void GetMicVolume(**out** int micvol);

**WK/FF+Wrapper:** int GetMicVolume();

Receive the current microphone volume level.

Platform : MAG100

**Parameters**

None

**Returned volume:**

| Parameter | Allowed volume | Description |
|---|---|---|
| Micvol | 0..100 | Returns the current microphone volume level. |

### stb.GetVolume

**FireFox:** void GetVolume(**out** int vol);

**WK/FF+Wrapper:** int GetVolume();

Receive the volume level.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| vol | 0..100 | Returns the current volume level. |

### stb.GetMute

**WK/FF+Wrapper:** int GetMute();

Receive the muted state of audio output.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| mute | 0..1 | Returns whether audio output is muted (mute==1) or not (mute==0). |

### stb.Step

void Step()

Display one next frame of video content.

**Platform: MAG100**

**Parameters:**

None.

**Returned value:**

None.

### stb.SetupRTSP

void SetupRTSP(int type, int flags)

Set-client to STB.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Type | 0..6 | Supported RTSP-server type:<br>0 – RTSP server based on VLC;<br>1 – BitBand RTSP server;<br>2 – Kasenna RTSP server;<br>3 – ARRIS (C-COR) RTSP server;<br>4 – Live555 RTSP server.<br>5 – ZTE RTSP server.<br>6 – Netup RTSP server.<br>The server types 3,4,5,6 are supported only for MAG200. |
| flags | 0..0x3f | Control flags:<br><br>1 – switch on the keep-alive mode;<br><br>2 – determination of the stream end by the field **x-notice** in the message ANNOUNCE from the server<br><br>4 – determination of te stream end by the field x-notice in the answer to GET_PARAMETER;<br><br>8 – determination of the stream end after a period of time of the video stream from the server absence;<br><br>16 (0x10) – determination of the stream end by the field according to the field rtptime sent in the RTP heading of the package (Only for the mode of sending video under RTP); |

| Parameter | Allowed value | Description |
|---|---|---|
| | | 32 (0x20) – Use UDP transport to send video. |

**Returned value:**

Нет.

### stb.SetViewport

void SetViewport(int xsize, int ysize, int x, int y)

Set the location and size of the video window.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| xsize | Depends on the screen resolution. | Horizontal size of the video window (width). |
| ysize | | Vertical size of the video window (height). |
| x | Must not exceed the screen width in sum with cxsize | Left upper corner of the video window horizontal offset from the screen edge. |
| y | Must not exceed the scxreen width in sum with ysize . | Left upper cornet of the video window vertical offset from the screen edge. |

**Returned value:**

None.

### stb.IsPlaying

**FireFox:** void IsPlaying(**out** bool bPlaying);

**WK/FF+Wrapper:** bool IsPlaying()

Receive the current state of display:

## Parameters:

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| bPlaying | true, false | Current state of display:<br>false – currently the content is not displayed;<br>true – currently the content is displayedв . |

### stb.Version

**FireFox:** void Version(**out** string version);

**WK/FF+Wrapper:** string Version();

Receive API version

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| version | | The string in ten form opf:<br>“JS API version: <JS_API version>;<br>STB API version: <STB_API version>;<br>Player Engine version: <Player version>”.<br>**JS_API version** – this API version number;<br>**STB_API version** – player API version<br>**Player version** – version of the player used in API in API in HEX code виде. |
| Example: “JS API version: 301; STB API version: 104; Player Engine version: 0x23” | | |

### stb.SetupSPdif

void SetupSPdif(int flags);

Set the mode of sound output through SPdif

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| flags | 0..2 | Output mode through SPdif:<br><br>0 – the sound is supplied only to analogue output.<br><br>1 – sound is supplied to analogue output and through SPdif in tne format 2- channel PCM<br><br>2 – sound is supplied to SPdif without decoding(AC3 …), if supported by codec, othjerwise through SPdif in te format of 2-channel PCM . |

**Returned value:**

None.

### stb.SetSubtitles

void SetSubtitles(bool enable);

Subtitle on/off.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Enable | true, false | true – subtitles on;<br>false – subtitles off. |

**Returned value:**

None.

**Notes**:

For MAG100 subtitles are displayed in full screen mode.

### stb.SetSubtitlesSize

void SetSubtitlesSize(int size);

Set the size of text subtitles – size in pixels.

**Platforms: MAG100, MAG200.**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| size | | Set the size of text subtitles. |

**Returned value:**

None.

### stb.SetSubtitlesFont

void SetSubtitlesFont(string font);

Set the font for displaying text subtitles.

**Platforms: MAG100, MAG200.**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| font | URL-адрес | URL addressing the font file in the root file system. For example: "/home/default/arial.ttf" |

**Returned value:**

None.

### stb.SetSubtitlesOffs

void SetSubtitlesOffs(int offs);

Set the offset for displaying text subtitles.

**Platforms: MAG100, MAG200.**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| offs | | Horizontal offset of subtitles. |

**Returned value:**

None.

### stb.GetSpeed

**FireFox:** void GetSpeed(**out** int speed);

**WK/FF+Wrapper:** int GetSpeed();

Receive the current speed of display

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| speed | -8..8 | Current speed of display:<br>1 - normal<br>2 - 2x<br>3 - 4x<br>4 - 8x<br>5 - 16x<br>6 - 1/2<br>7 - 1/4<br>8 – 12x<br>0 – stop or pause<br>-1 – reverse<br>-2 - reverse 2x<br>-3 - обратное воспроизведение 4x<br>-4 - reverse 8x<br>-5 - reverse 16x<br>-8 – reverse 12x |

**Returned value**.

None.

### stb.GetAudioPID

**FireFox:** void GetAudioPID(**out** int pid);

**WK/FF+Wrapper:** int GetAudioPID();

Receive the number (PID) of the current audio track.

## Parameters:

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| pid | 0..0x1fff | Current audio track number. |

## Notes:

The list of all audio tracks determined by the player can be received with stb.GetAudioPIDs.

### stb.GetSubtitlePID

**FireFox:** void GetAudioPID(**out** int pid);

**WK/FF+Wrapper:** int GetAudioPID();

Receive the number (PID) of the current subtitles track.

#### Parameters:

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| pid | 0..0x1fff | Current subtitles track number. |

#### Notes:

The list of all subtitles track determined by the player can be received with stb.GetSubtitlePIDs.

### stb.GetPIG

**FireFox:** void GetPIG(**out** bool isWindowed);

**WK/FF+Wrapper:** bool GetPIG();

Receive the video window state:

#### Parameters:

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| isWindowed | true, false | The resupt specifies whether full screen mode is set for the video window:<br>true – the content is displayed in a reduced window;<br>false – the content is displayed in a full screen mode. |

### stb.GetAlphaLevel

**FireFox:** void GetAlphaLevel(**out** int alpha);

**WK/FF+Wrapper:** int GetAlphaLevel();

Receive the video window alpha transparency level.

## Parameters:

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| alpha | 0..255 | Returne the current value of alpha transparency for the video window. |

### stb.GetWinAlphaLevel

**FireFox:** void GetWinAlphaLevel(int winNum, **out** int alpha);

**WK/FF+Wrapper:** int GetWinAlphaLevel(int winNum);

Receive the level of alpha transparency for the set window

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| winNum | 0..1 | Number of the window for which this function is used:<br>0 – graphic window;<br>1 – video window. |

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Alpha | 0..255 | Returns the current value of alpha transparency for video window. |

### stb.SetTransparentColor

void SetTransparentColor(int color);

Sets the colour considered transparent at the moment:

`Parameters`**:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| Color | 0..0xffffff | Colour in RGB format that can be considered transparent. |

**Returned value:**

None.

## Notes:

The function is a special case of stb.SetChromaKey.

Any changes on the screen are visible only provided the ChromaKey mode is switched on by functions **stb.SetMode or stb.SetWinMode**.

### stb.GetTransparentColor

**FireFox:** void GetTransparentColor(**out** int color);

**WK/FF+Wrapper:** int GetTransparentColor();

Returns the colour considered transparent at the moment:

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Color | 0..0xffffff | The colour in RGB format considered transparent at the moment |

### stb.IgnoreUpdates

void IgnoreUpdates(bool bIgnore);

Blocks or unblocks the screen browser upgrade:

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| bIgnore | true, false | **true** – after this call the graphic window stops upgrading till the next call with the parameter **false**;<br>**false** – after this call the graphic window resumes upgrading – passing to normal mode. |

**Returned value:**

None.

### stb.ExecAction

void ExecAction(string action);

Perform the script /home/default/action.sh with the parameters set.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Action | | String contains parameters for the script /home/default/action.sh.<br>Examle: stb.ExecAction('param 23 s') calls sh command from the shell "/home/default/action.sh param 23 s" |

**Returned value:**

None.

### stb.SetCASType

void SetCASType(int CAS_type);

Set default access server type after each start of the portal.

**Platforms: MAG100,MAG200**

**Parameters:**

| Parameters | Allowed velue | Description |
|---|---|---|
| Type | 0,1,2,4,5,6,7,8,9,10 | 0 – not set;<br>1 – Verimatrix;<br>2 – SecureMedia,<br>4-10 – custom CAS plugin with corresponding number. |

**Returned value:**

None.

**Notes:**

Set default server type once after each start of the portal.

### stb.SetCASParam

void SetCASParam(string serverAddr, int serverPort, string CompanyName, int opID, int errorLevel);

Set CAS server parameters:

**Platforms: MAG100,MAG200.**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| serverAddr | | CAS server URL. |
| serverPort | | CAS server port. |
| companyName | | Name of the company under which this operator is registered on CAS server. |
| opID | 1..255 | Operator identifier used by STB.<br>If opID is equal to -1, the value is not updated. |
| errorLevel | 0..5 | Level of error.<br>0 – minimal level.<br>If error Level equals to -1, it is not updated. |

**Returned value**

**None**.

**Notes:**

Call of the function becomes effective only if made before stb.SetCASType.

### stb.SetAdditionalCasParam

void SetAdditionalCasParam (string paramName, string paramValue);

Set additional CAS parameters:

**Platforms: MAG200.**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| paramName | | Additional parameter name. |
| paramValue | | Additional parameter value. |

**Returned value**

**None**.

**Notes:**

Call of the function becomes effective only if made before stb.SetCASType.

### stb.LoadCASIniFile

void LoadCASIniFile(string iniFileName);

Load CAS settings from the set file.

**Platforms: MAG100,MAG200.**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| iniFileName | | URL of the settings file in the root file systemф. |

**Returned value:**

None.

**Notes:**

**See instruction on adjusting CAS Verimatrix in the supplement**.

The call of the function becomes effective only if made before stb.SetCASType.

### stb.SetCASDescrambling

void SetCASDescrambling(int isSoftware);

Set hard or soft mode of descrambling.

**PLatforms: MAG100,MAG200.**

**Parameters:**

| Parameter | Allowed value | DESCRIPTION |
|---|---|---|
| isSoftware | 0,1 | 0 –use hard descrambling.<br>1 – use soft descrambling.<br>In the absence of this call soft descrambling is used.<br>Only soft descrambling can be used for MAG100. |

**Returned value:**

None.

**Notes:**

At present the use of the function is expedient only for CAS **Verimatrix**. Depending on the mode set, the player can descramble only the streams scrambled by the following algorithm.:

Soft mode: RC4, AES;

Hard mode: AES, DVB-CSA.

This mode is set only once after the start of the portal.

The call of the function becomes effective only if it is called before stb.SetCASType.

## stb.GetAspect

**FireFox:** void GetAspect(**out** int aspect);

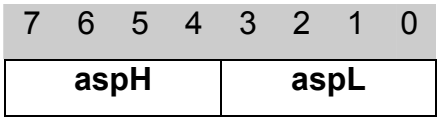
**WK/FF+Wrapper:** int GetAspect();

Return the current video content format.

### Parameters:

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| aspect | | Returns the current format of video content. Consists of 2 tetrads:<br>  <br><br>**ForMAG100 aspH is always equal to 0.** |
| aspL | 0..3 | Sets aspect ratio:<br>0 – automatically;<br>1 – 20:9;<br>2 –16:9;<br>3 – 4:3. |
| aspH | 0..3 | Sets video format conversion:<br>0 – as it is., the video is stretched to the whole screen;<br>1 – Letter Box mode, the video is proportionally enlarged to the screen |

| Parameter | Allowed value | Description |
|---|---|---|
| | | size along the larger edge;<br>2 – Pan&Scan mode, the video is proportionally enlarged to the screen size along the shorter edge;<br>3 – combined mode intermediate between Letter Box and Pan&Scan.<br>4 – enlarged mode;<br>5 – optimal mode.<br>Only for MAG200 |

### stb.StandBy

void StandBy(bool bStandby);

Enter or exit StandBy mode .

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| bStandby | true, false | true – enter Standby mode;<br>false – exit from Standby mode. |

**Returned value:**

None.

**Notes:**

When entering StandBy mode the following operations take place:

1. All video outputs switch off.
2. Content display, if it was on, stops.

### stb.RDir

**FireFox:** void RDir(string par, **out** string result);

**WK/FF+Wrapper:** string RDir(string par);

Perform script **/home/default/rdir.cgi** with set parameters and return the standard output of this script.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| par | Any string | The string contains parameters with |

| Parameter | Allowed value | Description |
|---|---|---|
| | | which the script is started /home/default/rdi.cgi. |

**Returned value:**

| Parameters | Allowed value | Description |
|---|---|---|
| result | | Standard output received when performing the script /home/default/rdi.cgi with parameters set. |

**Notes:**

The **rdir.cgi** supplied with the root file system has several commands preset**:**

stb.RDir("SerialNumber",x) – **x** returns serial number of this device to **x**.

stb.RDir("MACAddress",x) - receive MAC address

stb.RDir("IPAddress",x) - receive IP addressадрес

stb.RDir("HardwareVersion",x) – receive hardware version

stb.RDir("Vendor",x) – receive the name of STB manufacturer

stb.RDir("Model ",x) – receive the name of STB pattern

stb.RDir("ImageVersion",x) – receive the version of the software flash

imagestb.RDir("ImageDescription",x) – receive the information on the image of the software flash

stb.RDir("ImageDate",x) – receive the date of creation of the flash software image.

stb.RDir("getenv v_name",x) – receive the value of environment variable with the name **v_name**. See detailed description of operations with environment variables in supplement 11.

stb.RDir("setenv v_name value") – set environment variable with the name **v_name** to the value **value**. See detailed description of operations with environment variables in supplement 11.

stb.RDir("ResolveIP hostname") – resolve hostname to IP address.

## stb.SetAudioLangs

void SetAudioLangs(string priLang, string secLang);

Set languages of audio tracks to be automatically selected when receiving the information on the channel.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| priLang | 3 –symbol tags according to ISO 639, For example: “rus” или “eng” | If the information of several audio tracks is present the player selects the track preset by the language priLang. If such track is not found, the track with the language secLang is selected. If this one is not found either the first track from the list is selected. |
| secLang | | |

**Returned value:**

None.

### stb.SetSubtitleLangs

void SetSubtitleLangs(string priLang, string secLang);

Set the languages of subtitles tracks to be automatically selected when receiving the information on the channel.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| priLang | 3 –symbol tags according to ISO 639, For example: “rus” или “eng” | If the information of several audio tracks is present the player selects the track preset by the language priLang. If such track is not found, the track with the language secLang is selected. If this one is not found either the first track from the list is selected. |
| secLang | | |

**Returned value:**

None.

### stb.GetAudioPIDs

**FireFox:** void GetAudioPIDs(**out** string pidsList);

**WK/FF+Wrapper:** string GetAudioPIDs();

The function returns the list of audio tracks in the stream with the description of the language.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| pidsList | | List of the audio tracks found in the following format:<br><br>**[{pid:<***PID1***>, lang:[<***lang1_1***>, <***lang2_1***>]}, … , {pid:<***PIDn***>, lang:[** *<lang1_n>*, **<***lang2_n***>]}]** |
| PIDn | | *PID of audio track with the number n*. |
| lang1_n<br>lang2_n | 3-symbol tags according to ISO 639 | First two descriptions of languages in audio track with the number *n*. |

**Example**: the result in the form:

[{pid:114, lang:["rus", "ru"]}, {pid:115, lang:["eng", ""]}]

Means that 2 audio streams were found in the stream: Russian having PID=114 and English having PID=115;

**Notes:**

This stream can be easily converted into a structure array by calling the function **eval()**.

This function must be called after the event having the code 2 occurs (see description of events)

## stb.GetSubtitlePIDs

**FireFox:** void GetSubtitlePIDs(**out** string pidsList);

**WK/FF+Wrapper:** string GetSubtitlePIDs();

The function returns the list of subtitles track in the stream with the description of the language.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| pidsList | | List of subtitles tracks found in the following format:<br><br>**[{pid:<***PID1***>, lang:[<***lang1_1***>, <***lang2_1***>]}, … , {pid:<***PIDn***>, lang:[** <*lang1_n*>**, <***lang2_n***>]}]** |
| PIDn | | *PID of subtitle track with the number n*. |
| lang1_n<br>lang2_n | 3-symbol tags according to ISO 639 | First two descriptions of languages in subtitle track with the number *n*. |

**Example**: the result in the form:

[{pid:114, lang:["rus", "ru"]}, {pid:115, lang:["eng", ""]}]

means that 2 subtitle streams were found in the stream: Russian having PID=114 and English having PID=115;

**Notes:**

This string can be easily converted into a structure array by calling the function **eval()**.

This function must be called after the event having the code 2 occurs (see description of events)

### stb.ReadCFG

**FireFox:** void ReadCFG(**out** string result);

**WK/FF+Wrapper:** string ReadCFG();

Read the file of portal settings /etc/stb_params.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| result | | Returns the contents of the file /etc/stb_params. |

### stb.WriteCFG

void WriteCFG(string cfg);

Read the file of portal settings /etc/stb_params.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Cfg | | The data to be stored in the file /etc/stb_params. |

**Returned value:**

None.

**Notes:**

It must be kept in mind that the values PORTAL_IP, PORTAL_1, PORTAL_2 are used in the starting portal stored in /home/web of the root file system, therefore it is desirable to receive source values of these parameters via stb.ReadCFG before making the call and add them to the string cfg.

**stb.WritePrefs**

void WritePrefs(string prefs);

Save the string as the browser set up (prefs.js).

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| prefs | | Data to be saved in the file of browser settings. |

**Returned value:**

None.

**Notes:**

This function is not browser specific and it is used to set the right of access to the portal. This is performed in starting portal saved at /home/web of the root file system and it is recommended to avoid using it anywhere else.

**stb.Debug**

void Debug(string debugString);

Show the contents of the string **debugString** in the stream of standard output in the format:

DEBUG: <time> **debugString**

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| debugString | | This string is shown in the stream of standard output. |

**Returned value:**

None.

### stb.SetListFilesExt

void SetListFilesExt (string fileExts);

Set the list of file extensions for returning to the function stb.ListDir.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| fileExts | | List of files extensions followed by a space.<br>For example:<br>“.mkv .mov .mpg” |

**Returned value:**

None.

`Notes`**:**

This function is realized only for the browser based on WebKit.

### stb.ListDir

string ListDir (string dirName);

Returns the list of directories and files having the extension set with SetListFilesExt, located in the directory **dirName**.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| dirName | | Route to the directory the contents whereof must be received. |

**Returned value:**

The string in the following form is returned:

```
var dirs = [
"dir1/",
…
"dirn/",
```

""

]

var files = [

{"name" : "fileName1", "size" :size1},

…

{"name" : "fileNamem", "size" :sizem},

{}

]

Where dirn – the name of n-sub-directory,

fileNamem and sizem – name and size of m-file.

**Notes:**

This function is realized only for the browser based on WebKit. For browsers based on FireFox such function can be realized using the function RDir with the parameter “rdir”.

**stb.SetBrightness**

void SetBrightness (int bri);

Set the brightness of video output in SD mode.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Bri | 1..254 | Brightness in the SD mode. |

**Returned value:**

None.

**Notes:**

This function is realized only for the browsers based on WebKit.

**stb.SetSaturation**

void SetSaturation (int sat);

Set the saturation of video output in SD mode.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Sat | 1..254 | Saturation of video output in SD mode. |

**Returned value:**

None.

**Notes:**

This function is realized only for the browser based on WebKit.

### stb.SetContrast

void SetContrast (int con);

Set contrast of video output in SD mode.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Con | -128..127 | Video output contrast in SD mode |

**Returned value:**

None.

**Notes:**

This function is realized only for the browser based on WebKit.

### stb.GetBrightness

int GetBrightness ();

Receive current brightness of video output in SD.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| Bri | 1..254 | Brightness of video output in SD mode |

**Notes:**

This function is realized only for the browser based on WebKit.

### stb.GetSaturation

int GetSaturation ();

Receive current saturation of video output in SD mode

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| Sat | 1..254 | Saturation of video output in SD mode |

**Notes:**

This function is realized only for the browser based on WebKit.

**stb.GetContrast**

void GetContrast (int con);

Receive current contrast of video output in SD mode

**Parameters:**

None

**Returned value**

| Parameter | Allowed value | Description |
|---|---|---|
| Con | -128..127 | Contrast of video output in SD mode |

**Notes:**

This function is realized only for the browser based on WebKit.

**stb.DeleteAllCookies**

void DeleteAllCookies ();

Delete all cookie saved by the browser.

**Parameters:**

None.

**Returned value:**

None.

**Notes:**

This function is realized only for the browser based on WebKit.

## stb.SetAudioOperationalMode

void SetAudioOperationalMode(int mode);

Set Operational Mode for DolbyDigital audio.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| mode | 0..3 | 0 – RF mode.<br>1 – Line mode.<br>2 – Custom0.<br>3 – Custom1. |

**Returned value:**

None.

**Notes:**

Affects only **DolbyDigital** audio.

### stb.SetHDMIAudioOut

void SetHDMIAudioOut(int type);

Set HDMI audio format.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| type | 0..1 | 0 – HDMI transmits PCM audio.<br>1 – HDMI transmits SPdif audio. In that case SPDif output mode is set by stb.SetupSPdif |

**Returned value:**

None.

**Notes:**

None.

### stb.SetDRC

void SetDRC(int high,int low);

Set dynamic range compression for DolbyDigital audio.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| high | 0..255 | Compression level for high range.<br>0 – DRC is off. |
| low | 0..255 | Compression level for low range. |

| Parameter | Allowed value | Description |
|---|---|---|
| | | 0 – DRC is off. |

**Returned value:**

None.

**Notes:**

None.

### stb.SetStereoMode

void SetStereoMode(int mode);

Set stereo mode.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| mode | 0..4 | 0 – Stereo mode.<br>1 – Mono mode. Left and right channels are mixed and sent to both audio outputs.<br>2 – Mono left. Left channel audio are sent to both audio outputs.<br>3 – Mono right. Right channel audio are sent to both audio outputs.<br>4 – Lt/Rt mode |

**Returned value:**

None.

**Notes:**

Mono, Mono left and Mono right modes affect only Dual Mono DolbyDigital audio.

### stb.EnableJavaScriptInterrupt

void EnableJavaScriptInterrupt(bool enable);

Enable/disable Javascript Interrupt dialog, when Javascript code does not respond for some long time.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| enable | true, false | true – enable interrupt.<br>false – disable interrupt. |

**Returned value:**

None.

**Notes:**

Use this function only for debugging purpose.

### stb.ShowSubtitle

void ShowSubtitle(unsigned int start, unsigned int end,string text);

Show **text** string as a subtitle on screen.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| start | | String presentation start time in ms from start of current media. |
| end | | String presentation end time in ms from start of current media. |
| text | | This text will be shown on screen as a subtitle. |

**Returned value:**

None.

**Notes:**

In case when **start** and **end** equal 0, text is shown on screen immediately until next stb.ShowSubtitle is called or 30 seconds elapsed.

If this function was called then subtitles will work only via stb.ShowSubtitle until next call of stb.Play.

### stb.StartLocalCfg

void StartLocalCfg();

Start local configuration menu (Service Menu).

**Parameters:**

None.

**Returned value:**

None.

**Notes:**

Result of this function is similar to pressing "SET" ("service" on old RC) button, if automatic appearance of Service Menu is disabled via stb.EnableServiceButton.

### stb.ShowVirtualKeyboard

void ShowVirtualKeyboard();

Show virtual keyboard on screen.

**Parameters:**

None.

**Returned value:**

None.

**Notes:**

User can switch virtual keyboard from english symbols to symbols of the language that is set as the local language in Service Menu.

### stb.HideVirtualKeyboard

void HideVirtualKeyboard();

Hide virtual keyboard from screen.

**Parameters:**

None.

**Returned value:**

None.

### stb.EnableServiceButton

void EnableServiceButton(bool bEnable);

Enable or disable automatic start of Service Menu by pressing "SET" ("service" on old RC) button.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| bEnable | false, true | false – disable automatic start.<br>true – enable automatic start. |

**Returned value:**

None.

**Notes:**

If button "SET" ("service" on old RC) is already used by JavaScript code, there may be a conflict. To avoid this conflict JavaScript code should disable automatic start of Service Menu and call directly function stb.StartLocalCfg every time it is required.

### stb.EnableVKButton

void EnableVKButton(bool bEnable);

Enable or disable automatic show/hide of virtual keyboard by pressing "KB" ("empty" on old RC) button.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| bEnable | false, true | false – disable automatic show/hide.<br>true – enable automatic show/hide. |

**Returned value:**

None.

**Notes:**

If button "KB" ("empty" on old RC) is already used by JavaScript code, there may be a conflict. To avoid this conflict JavaScript code should disable automatic start of virtual keyboard and call directly functions stb.ShowVirtualKeyboard or stb.HideVirtualKeyboard every time it is required.

### stb.EnableSpatialNavigation

void EnableSpatialNavigation(bool bEnable);

Enable or disable 2D navigation (arrow navigation) on web pages.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| bEnable | false, true | false – disable 2D navigation.<br>true – enable 2D navigation. |

**Returned value:**

None.

**Notes:**

2D navigation is disabled by default, but could be enabled on previous web page, so it is recommended to disable 2D navigation if current page does not use it.

### stb.EnableSetCookieFrom

void EnableSetCookieFrom(string domain, bool bEnable);

Allow or forbid to set cookie from given domain.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| domain | URL | If bEnable==true then any attempt to set cookie from given domain will be ignored. |
| bEnable | false, true | false – forbid to set cookie from given domain.<br>true – allow to set cookie from given domain. |

**Returned value:**

None.

**Notes:**

By default any domain is allowed to set cookie.

Each call of this function adds domain (bEnable==false) or removes it (bEnable==true) from the list of domains that are not allowed to set cookie.

### stb.SetBufferSize

void SetBufferSize(int sizeInMs, int maxSizeInBytes);

Set input buffer size for bufferring support.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| sizeInMs | int | Buffer size in ms. |
| maxSizeInBytes | int | Maximum buffer size in bytes. Used ONLY to limit maximum amount of allocated memory but not as the primary buffer size. |

**Returned value:**

None.

**Notes:**

Bufferring works only with the following solutions:

ffmpeg, ffrt, ffrt2, ffrt3, fm, file.

After start of playback with mentioned above solution the following stages take place:

- opening content for playback;
- determining content information (raised event 2 from List of the events  used);
- input buffer filling stage;
- after complete buffer filling stream data go to decoding;
- decoded data go to outputs (raised event 4 from List of the events  used).

### stb.GetBufferLoad

int GetBufferLoad();

Get current buffer loading in percents.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| | 0..100 | Current buffer loading in percents. |

**Notes:**

It makes sense to call this function only with the following solutions: **ffmpeg, ffrt, ffrt2, ffrt3, fm, file** after getting event 2 from List of the events  used and before complete buffer filling or before getting event 4 from List of the events  used.

### stb.SetWebProxy

void stb.SetWebProxy( string proxy_addr, int proxy_port, string user_name, string passwd, string exclude_list);

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| proxy_addr | string | Proxy server address. |
| proxy_port | int | Proxy server port. |
| user_name | string | Username for proxy server. Can be |

| Parameter | Allowed values | Description |
|---|---|---|
| | | empty. |
| passwd | string | Password for proxy server. Can be empty. |
| exlude_list | string | Proxy exlude list delimited by spaces. Access to any entry in the list is performed directly, without proxy. E.i.: 'youtube.com .lenta.ru 192.168.1.1/24 192.168.1.*' |

**Return value:**

None.

**Note.** Given proxy settings are only applied to http:// or https:// requests of the browser, but not applied to content playback from http server.

For this purpose please use extended stb.Play using proxy server.

### stb.GetVideoInfo

string GetVideoInfo();

Get information about current video content.

**Parameters:**

None.

**Return value:**

Returns string in the following form:

{frameRate:25000,pictureWidth:704,pictureHeight:576,hPAR:12,vPAR:11},

where

frameRate – video frame rate.

pictureWidth – encoded video width.

pictureHeight – encoded video height.

hPAR and vPAR – pixel aspect ratio coefficients. In example above these params mean that movie aspect ratio is:

(704*hPAR/vPAR)/576 = 1.333333333(3) = 4:3 for square pixels.

**Note:** Function must be called after receiving event 7 from List of the events used.

### stb.GetMetadataInfo

string GetMetadataInfo();

Get metadata information stored in current content. For example it can be data from ID3 tag from mp3 file.

**Parameters:**

None.

**Returned value:**

| Parameter | Allowed value | Description |
|---|---|---|
| | string in the following form: {"album":"album_1","album_artist":"artist_1", "artist":"artist_1","comment":"","composer":"", "copyright":"","date":"2000","disc":"","encoder":"," encoded_by":"","filename":"","genre":"", "language":"","performer":"","publisher": "publisher_1","title":"track_9","track":"9"} | Metadata from current content. |

**Notes:**

It makes sense to call this function after getting event 2 from List of the events used.

### stb.SetAutoFrameRate

void SetAutoFrameRate(int mode);

Set the current AutoFrameRate mode, which allows to automatically switch frame rate of HDMI video output according to a content frame rate.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| mode | int | contains bit flags which specify a set of frame rates to which video output can automatically switch:<br>1 – can switch to 1080p-24 mode.<br>2 – can switch to 720p-50, 1080i-50, 1080p-50 modes. |

| Parameter | Allowed value | Description |
|---|---|---|
| | | 4 – can switch to 720p-60, 1080i-60, 1080p-60.<br><br>For example:<br>stb.SetAutoFrameRate(0) disables auto frame rate switching.<br>stb.SetAutoFrameRate(7) enables switching to frame rates 24,50 and 60. |

**Returned value:**

None.

**Notes:**

Auto frame rate switching works with **ffmpeg**, **file**, **ffrt2**, **ffrt3** solutions if player has managed to detect frame rate of content.

Auto frame rate switching works only on HDMI output in the following modes: 720p-50/60, 1080i-50/60 and 1080p-50/60. In 720p-50/60 modes player can switch only to 720p-50/60 modes.

After stopping playback video output switches back to original frame rate.

**Warning:** Not every TV supports 1080p-24 video mode.

### stb.ForceHDMItoDVI

void ForceHDMItoDVI(int ForceDVI);

Force HDMI output to DVI mode.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| ForceDVI | 0,1 | 0 – auto detect HDMI mode.<br>1- force HDMI to DVI mode. |

**Returned value:**

None.

### stb.LoadExternalSubtitles

void LoadExternalSubtitles(string url);

Load text subtitles from external subtitle file of srt, sub, ass formats.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| url | string with external subtitles URL | URL which points to external subtitles.<br>Can be a local URL:<br>"/media/USB-.../subtitles.srt"<br>and URL from http server:<br>"http://192.168.1.1/subtitles.srt" |

**Returned value:**

None.

**Notes:**

If subtitles are loaded successfully then external subtitle track will be added to subtitle track list with number(PID) 0x2000.

If any error occurs while loading subtitles then JS API user will receive event with code 8 from List of the events used.

### stb.SetSubtitlesEncoding

void SetSubtitlesEncoding(string encoding);

Set the encoding which will be used to display external subtitles.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| encoding | string with encoding name, e.i.:<br>"utf-8", "cp1250", "cp1251", "cp1252", ..., "cp1258", "iso8859-1", ... , "iso8859-16". | Encoding for external subtitles. |

**Returned value:**

None.

### stb.GetEnv

string GetEnv (string args)

Read specified boot loader's variables

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| | | |
|---|---|---|
| Args | JSON object with pair named «varList» that has array of non empty strings as value | Read values for variables specified in the array<br><br>Example:<br>{<br>"varList":["a", "b", "timezone_conf_int", "wifi_ssid", "update" ]<br>} |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON object. Object is holding only pairs. Each pair name is equal to variable name. And value of the pair representing value of variable in string notation.<br><br>Example:<br>{<br>"result":<br>{<br>"a": "b",<br>"b": "",<br>"timezone_conf_int": "plus_02_00_13",<br>"wifi_ssid": "default_ssid",<br>"update": "erase $monitor_sec;cp.b $load_addr $monitor_base $monitor_len;protect on $monitor_sec"<br>},<br>"errMsg": ""<br>} |

### stb.SetEnv

boolean SetEnv (string args)

Setting up values of specified boot loader’s variables

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| Args | JSON object, which holding only pairs. Each pair has string value.<br>Each pair name is referencing variable name. And pair value will set new value<br>JSON объект, который содержит произвольное число пар, каждая из которых соответствует записываемой переменной среды бутлоадера | Each pair’s name is referencing name of variable. And pair value will be used as new value of variable.<br><br>If referencing variable does not exist it will be created.<br><br>If new value is empty string then value will be deleted<br><br>Example:<br>{<br>"a34":"b34", "c34":"", "c34":"d34"<br>} |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| boolean | True, false | Result of operation |

### stb.GetDeviceSerialNumber

string GetDeviceSerialNumber ()

Get serial number

**Parameters:**

Нет

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|

| | | |
|---|---|---|
| string | - | - |

### stb.GetDeviceVendor

string GetDeviceVendor ()

Get vendor information

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetDeviceModel

string GetDeviceModel ()

Get model of the device

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetDeviceVersionHardware

string GetDeviceVersionHardware ()

Get hardware information

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetDeviceMacAddress

string GetDeviceMacAddress ()

Get MAC address

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetDeviceActiveBank

string GetDeviceActiveBank()

Get active bank of NAND

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | - | Эквивалентно /home/default/rdir.cgi GetCurrentBank |

### stb.GetDeviceImageVersion

string GetDeviceImageVersion ()

Get image version

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetDeviceImageDesc

string GetDeviceImageDesc()

Get image description

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetDeviceImageVersionCurrent

string GetDeviceImageVersionCurrent()

Get current image version

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | |

### stb.GetLanLinkStatus

string GetLanLinkStatus ()

Get link status of LAN network interface (eth0)

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| boolean | - | True – link is active<br><br>False – no link connection |

### stb.GetWifiLinkStatus

string GetWifiLinkStatus ()

Get link status of WiFi network interface

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| boolean | - | True – link is active<br>False – no connection to WiFi access |

| | | |
|---|---|---|
| | | point |

### stb.GetWepKey64ByPassPhrase

string GetWepKey64ByPassPhrase(string passPhrase)

Return wep 64 bit keys for given passphrase

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| passPhrase | string | Passphrase, 1-32 symbols |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON object. Result object contain four 64 bit wep keys.<br><br>{<br>"errMsg": "",<br>"result":<br>{<br>"wep64-key1": "c6774663dd",<br>"wep64-key2": "af6bd13ecd",<br>"wep64-key3": "8e33fb2bf1",<br>"wep64-key4": "cf12611e1d"<br>}<br>} |

### stb.GetWepKey128ByPassPhrase

string GetWepKey128ByPassPhrase (string passPhrase)

Return wep 128 bit keys for given passphrase

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| passPhrase | string | Passphrase, 1-32 symbols |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|

| | | |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON object.<br>Result object contain 128 bit wep key.<br><br>{<br>"errMsg": "",<br>"result":<br>{<br>"wep128-key1": "46f04863257ac8040905ea0002"<br>}<br>} |

### stb.GetWifiGroups

string GetWifiGroups()

Make scan and return list of found wireless groups (SSID)

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair holding an array of JSON objects.<br><br>Each object representing wireless group and has following attributes:<br>"ssid" – name of group<br>"auth" – authentification method ("WPA", "WPA2", "WEPAUTO", "NONE")<br>"enc" – encoding ("CCMP", "TKIP", "NONE")<br>"signalInfo" – signal strength (numeric value)<br>"rfInfo" – string information about channel |

| | | |
|---|---|---|
| | | {<br>"errMsg": "",<br>"result":<br>[<br>{<br>"ssid": "dlink",<br>"auth": "WPA2",<br>"enc": "TKIP",<br>"signalInfo": "-47",<br>"rfInfo": "Frequency:2.412 GHz (Channel 1)"<br>},<br>{<br>"ssid": "linksys3E66",<br>"auth": "WEPAUTO",<br>"enc": "WEP",<br>"signalInfo": "-67",<br>"rfInfo": "Frequency:2.427 GHz (Channel 4)"<br>}<br>]<br>} |

### stb.ServiceControl

string ServiceControl(string serviceName, string action)

Execute control actions for background service

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| serviceName | string | Service name.<br>For example: "network", "pppoe" |
| Action | string | Action to execute for given service<br>For example: "start", "restart" |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON object.<br>Result object contain status pair.<br>Value is always "ok"<br><br>{<br>"errMsg": "",<br>"result":<br>{<br>"status": "ok"<br>}<br>} |

### stb.GetSmbGroups

string GetSmbGroups ()

Fetching available workgroups

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON array of groups.<br><br>Example:<br><br>{<br>"result":<br>[<br>"DUNE",<br>"WORKGROUP"<br>],<br>"errMsg": ""<br>} |

### stb.GetSmbServers

string GetSmbServers (string args)

Fetching available servers for given work group

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| args | String, JSON object | Pair "group" defines workgroup of interest<br><br>Example:<br>{<br>"group":"workgroup"<br>} |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON array of servers<br><br>Example:<br><br>{<br>"result":<br>[<br>"ARCHIVE",<br>"EDDY",<br>"SANDBOX"<br>],<br>"errMsg": ""<br>} |

### stb.GetSmbShares

string GetSmbShares (string args)

Fetching available shares for given server

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| args | String, JSON object | Pair "server" defines server of interest<br><br>Example:<br>{<br>"group":"workgroup"<br>} |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | JSON object.<br>Pair «errMsg» will have empty string as value in case of success execution and non localized string that representing error condition.<br>Pair «result» will hold result of operation. | Value of «result» pair is JSON object. Pair "shares" is an array of shares. Pair "serverIP" holds IP address of given server<br><br>Example:<br><br>{<br>"result":<br>{<br>"shares":<br>[<br>"share",<br>"foto"<br>],<br>"serverIP": "192.168.100.1"<br>},<br>"errMsg": ""<br>} |

### stb.IsFolderExist

bool IsFolderExist (string fileName)

Test is file name point to existing folder.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|

| Parameter | Allowed value | Description |
|---|---|---|
| fileName | String, absolute file path | Absolute file path which will be tested |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| bool | - | True if file name is absolute path and it point to existing folder |

### stb.IsFileExist

bool IsFileExist (string fileName)

Test is file name point to existing file.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| filename | String, absolute file path | Absolute file path which will be tested |

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| Bool | - | True if file name is absolute path and it point to existing IsFileExist |

### stb.SendEventToPortal

void SendEventToPortal(string pArgs)

Send an event to portal webkit window.

You can handle the event by defining event handler: «stbEvent.onPortalEvent(string)»

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| pArgs | String | This argument will be passed to event handler stbEvent.onPortalEvent(string). |

**Return value:**

None

### stb.IsWebWindowExist

bool IsWebWindowExist ()

This function indicating that Web Window is exists.

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| Bool | - | True when «wild web window» exist in the window stack |

### stb.IsInternalPortalActive

bool IsInternalPortalActive()

This function indicating that internal portal been started.

**Parameters:**

None

**Return value:**

| Data type | Allowed value | Description |
|---|---|---|
| Bool | - | True when internal portal has been started |

### stb.EnableAppButton

void EnableAppButton (bool pEnable)

Assign new status for “App” button handler

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| pEnable | Boolean | True – application will take control under “App” button<br><br>False - “App” button will be handled as regular button |

**Return value:**

None

***Event model in JavaScript.***

Event model in JavaScript assumes the possibility for API user to receive the events indicating some changes of the player playback state.

### Configuring the event system

To configure event system proceed as follows:

1. Include the script event.js in the portal:

**<script language="JavaScript" src="event.js"></script>**

This script can be taken from /home/web/ directory of the root file system nfs-image.

**Notes:**

The contents of this script **must** be in the global scope.

For the browser based on WebKit the declaration of the stbEvent object in the global scope is sufficient instead of including this script:

```
var stbEvent=
{
   onEvent : function(data){},
   event : 0
}
```

2. after the initialization of the player (see appendix 1) call **initEvents()** function

3. the function to be called when getting the event must be set through **stbEvent** object:

**stbEvent.onEvent** = EventCallback,

where EventCallback – is the function used for processing the events in the portal with the event code as the parameter.

For example:

```
function EventCallback(event) {_debug('event '+event)}
```

4. The code of the last event is also stored in the **stbEvent.event**.

### List of the events used

The following events are defined:

| Event code<br>события | Description |
|---|---|
| 1 | The player reached the end of the media content or detected a discontinuity of the stream. |
| 2 | Information on audio and video tracks of the media content is received. |
| 4 | Video and/or audio playback has begun. |
| 5 | Error when opening the content: content not found on the server or connection with the server was rejected. |
| 6 | Detected DualMono AC-3 sound. |
| 7 | Detected information about video content. |
| **8** | Error occurred while loading external subtitles. |
| 0x20 | HDMI connected. |
| 0x21 | HDMI disconnected. |
| 0x22 | Recording task has been finished successfully. See Appendix 13. JavaScript API for PVR subsystem. |
| **0x23** | Recording task has been finished with error. See Appendix 13. JavaScript API for PVR subsystem. |
| 0x81 | When playing RTP-stream the numbering of RTP-packets was broken. |

## *Appendix 1. API usage.*

### stb object initialization.

First of all the main object **stb** must be created. Proceed as follows:

1. Declare object **stb**:
   var stb;
2. Initialize **stb** in the page initialization function with the following lines:

```
netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")
const cid = "@mydomain.com/XPCOMSample/MyComponent;1"
stb = Components.classes[cid].createInstance()
stb = stb.QueryInterface(Components.interfaces.IMyComponent)
```

These lines created **stb** object but the player, so any media content cannot be played back at this stage and developer can use just a few set of function such as stb.RDir. Because only one player instance can be initialized simultaneously this mode is used only for auxiliary pages such as /home/web/index.html.

The following functions can be called in this mode:

stb.Version,
stb.ExecAction,
stb.RDir,
stb.ReadCFG,
stb.WriteCFG,
stb.WritePrefs,
stb.InitPlayer.

### Player initialization

For using all API functions initialize the player with stb.InitPlayer function. Only one player can be initialized during a certain period of time. To initialize another player (for example on another page) first call stb.DeinitPlayer for the player that had been already initialized.

**Specifics of JavaScript API >= 308 versions.**

Beginning from the version JavaScript API 308 the initialization scheme described above can be used or the strings:

```
const cid = "@mydomain.com/XPCOMSample/MyComponent;1"
stb = Components.classes[cid].createInstance()
stb = stb.QueryInterface(Components.interfaces.IMyComponent) in item 2)
```

can be replaced by:

```
stb = gSTB
```

Besides, beginning from version 308 the possibility of repeated call of stb.InitPlayer appeared; in this situation the player will be initialized when this function is called for the first time, and it will be de-initialized after the exit from the browser.

**Player initialization (Version  JavaScript API >= 308).**

To use all API functions initialize the player with stb.InitPlayer function. Only one player can be initialized at a time. To initialize another player (for example on another page) first call stb.DeinitPlayer for the player that had been already initialized

**Wrapper.js**

For those programmers who do not want to call the function

```
netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")
```

in every function that uses **stb** object, an auxiliary script wrapper.js, was written, which allows calling  the methods of object stb from any place of  JS code without setting privileges.

To include this script:

1. include this script at the very beginning
   <script language="JavaScript" src="wrapper.js"></script>
2. comment out the strings:
   var stb
   stb=gSTB

   const cid = "@mydomain.com/XPCOMSample/MyComponent;1"
   stb = Components.classes[cid].createInstance()
   stb = stb.QueryInterface(Components.interfaces.IMyComponent)
   if they are present in the main script of the page.

Thereafter **stb** object appears in the global scope of the script, which allows calling **stb** object methods without setting privileges.

### Event system initialization

This item is described in detail in the Event model in JavaScript chapter.

### API usage example.

The page below shows minimal HTML code of the page, which is simply loaded and starts playing rtp stream and stops playing or restarts playing the stream with the buttons **stop** and **continue** correspondingly.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
	<head>
		<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
		<title></title>
		<script language="JavaScript" src="event.js"></script>
		<script>
			var stb
			function init(){
				netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")
				const cid = "@mydomain.com/XPCOMSample/MyComponent;1"
				stb = Components.classes[cid].createInstance()
				stb = stb.QueryInterface(Components.interfaces.IMyComponent)
				stb.InitPlayer()
				stb.Play('rtp rtp://224.10.0.123:1234')
			}
			function getkeydown(e) {
				netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")
				ec = e.keyCode
				ew = e.which
				es = e.shiftKey
				pat = /^(\S+)_(\S+)/
				switch (ew){
					case 114: // Play
					{
						stb.Play('rtp rtp://224.10.0.123:1234')
						break;
```

```
                                        }
                                        case 115: // Stop
                                        {
                                                stb.Stop()
                                                break;
                                        }
                                }
                        }
                </script>
        <body onload="init()" onKeyPress="getkeydown(event)">
        </body>
</html>
```

Besides, the text page, which can be used for checking and seeing the operation of all API functions, is  contained in the root file system for MAG200 in the folder /home/web/ ..

***Appendix 2. Video content formats and examples of use.***

This appendix describes the types of the content played back and their use.

Playback can be started by two functions: stb.Play and stb.PlaySolution. The parameters of stb.PlaySolution function are included in the complex parameter **playStr** of the function stb.Play, therefore further description shall be based on the example stb.Play function and on the parameters of this function.

**stb.Play function parameters format.**

**playStr** has the following format:

**"solution URL [atrack:num] [vtrack:num] [strack:num] [position:time]"**, where

**solution**

The type of media content, which determines the content format, for example media container type and/or the method of broadcasting.

| Type (**solution**) | MAG100 | MAG200 | Description |
|---|---|---|---|
| auto | + | + | Automatic detection of the type of content, container, codec by the givern URL.<br>**Distinctions:**<br>**MAG100**: when using automatic defining URLs that begin with rtp://, udp://, rtsp:// cannot be used, i.e.. automatic definition is used only with files.<br>**MAG200**: correctly accepts the URLs beginning with rtp://, udp://, rtsp://. |
| "" (empty) | + | + | Similar to **auto**. |
| rtp | + | + | Play the string in the format MPEG2TS.<br>If **URL** begins with rtp:// , the RTP stream shall be played, if it begins with udp:// , the UDP stream shall be played.<br>**Distinctions:**<br>**MAG100** automatically connects decoder for MPEG2 Video and MPEG Audio.<br>**MAG200** set required codecs if addional |

| Type (**solution**) | MAG100 | MAG200 | Description |
|---|---|---|---|
| | | | information of stream is present, for example, H.264, AC-3 и т.д. |
| rtsp | + | + | Play the content from RTSP-server.<br>**Distinctions:**<br>**MAG100** automatically connects decoder for MPEG2 Video and MPEG Audio.<br>**MAG200** set required codecs if additional information on the stream is present, for example, H.264, AC-3 ,etc. |
| rtpac3 | + | + | Play the stream in the format MPEG2TS using decoders MPEG2 Video and AC-3 Audio |
| rtsp_ac3 | + | + | Play content from RTSP-server using decoders MPEG2 Video and AC-3 Audio |
| rtpmpeg4 | + | – | Play the stream in the format MPEG2TS using decoders MPEG4p2 Video and MPEG Audio |
| rtpmpeg4_aac | + | – | Play the stream in the format MPEG2TS using decoders MPEG4p2 Video and AAC Audio |
| mpegts | + | + | Play the file in the format MPEG2TS.<br>**Distinctions:**<br>**MAG100** automatically connects decoder for MPEG2 Video and MPEG Audio.<br>**MAG200** sets required codecs if additional information on the stream is present, for example H.264, AC-3 , etc. |
| mpegps | + | + | Play file in the format MPEG2 Program Stream.<br>**Distinctions:**<br>**MAG100** automatically connects decoder for MPEG2 Video and MPEG Audio.<br>**MAG200** sets required codecs if additional information on the stream is present, for example, AC-3 , etc |
| file | – | + | Play file at the URL set with automatic determining |

| Type (**solution**) | MAG100 | MAG200 | Description |
|---|---|---|---|
| | | | content type, container and codec. |
| mp4 | + | – | Play file in MP4 format using codecs MPEG4p2 Video and AAC Audio |
| mp4_mpa | + | – | Play file in MP4 format using codecs MPEG4p2 Video and MPEG Audio |
| fm | + | + | Play audio from MPEG-TS of the (udp://, rtp://) |
| ffmpeg | – | + | MAG200 allows playing files in various formats: avi, mkv, mpg, mp4, mov, wmv, AC-3, mp3. |
| **ffrt** | – | + | Play MPEG-TS realtime stream via http:// |
| **ffrt2** | – | + | Play realtime stream (not only MPEG-TS) via http://, rtmp:// ...<br>For this solution looping is always enabled. |
| **ffrt3** | – | + | Play non realtime internet video such as YouTube video. |

**Notes**

In contrast to MAG100, MAG200 can determine and change codecs during a playback, for example, when audio tracks compressed by different codecs are present.

**URL**

Specifies where the content is stored:

| URL format | Description |
|---|---|
| **/path** | Local address of the file in the root file system, for example, /media/1.mp3 |
| **rtp://addr:port** | RTP-stream received from multicast or unicast address ***addr*** and the port ***port***. |
| **udp://addr:port** | UDP-stream received from multicast or unicast address ***addr*** and the port ***port***. |
| **rtsp://addr:port/path** | The content stored in the RTSP-server at the address ***addr*** and port ***port*** with the relative route ***path*** |

**atrack, vtrack и strack**

Optional parameters setting the numbers of audio, video tracks and subtitle tracks (PID-s for MPEG2TS) of the content to be played.

**position**

Optional parameter setting the time **time** in seconds, from which the content is to be played.

**Examples:**

**“mpegps /media/1.mpg”** – plays the file Mpeg2 Program Stream файл /media/1.mpg.

**“mpegts /media/1.mpg”** – plays the file Mpeg2 Transport Stream файл /media/1.mpg.

**“mp4 /media/1.mp4”** – plays the file /media/1.mp4 in the format MP4.

**“rtp 224.10.0.30:5004”** – plays Mpeg2 in the format Transport Stream from the preset multicast address (224.10.0.30) and port (5004) using IGMP protocol for multicast broadcast.

**“auto /media/file1”** – an attempt to automatically determine the file format and play it.

**“rtpmpeg4 224.10.0.31:5004”** – plays Mpeg4 in the format Transport Stream from the preset multicast address (224.10.0.31) and port (5004) with Mpeg2 Audio using IGMP protocol for multicast broadcast.

**“rtpmpeg4_aac 224.10.0.32:5004 atrack:930 vtrack:920”** – plays Mpeg4 video in the format Transport Stream from the preset multicast address (224.10.0.32) and port (5004) with AAC audio using IGMP protocol for multicast broadcast. In this situation the stream with PID=920 is automatically selected as the video track and the stream with PID=930 – as the audio track, irrespective of the presence of information on the tracks in the stream.

**“rtsp rtsp://192.168.1.32:554/video/media003.mpg”** – plays the content /video/media003.mpg, located on RTSP-server with the address 192.168.1.32 and port 554.

## *Appendix 3. CAS usage and settings.*

### Setting up Verimatrix CAS.

For using Verimatrix CAS proceed as follows:

1. Set the correct time, for example, from ntp server.
2. Set initial parameters of the CAS server using one of the two methods:
    a. With LoadCASIniFile function, when the parameters are automatically taken from the specified file.
    b. With SetCASParam or/and stb. SetAdditionalCasParam functions.
3. For MAG200 set descrambling mode using stb.SetCASDescrambling function.
4. Set the type of the CAS server after setting initial parameters.

**Note.** Since release of 0.1.66 Software Image version, file rootcert.pem is already in /flash folder.

**Setting up SecureMedia CAS.**

For using SecureMedia CAS proceed as follows:

1. Set the correct time, for example, from ntp server.
2. Set initial parameters of the CAS server using SetCASParam or/and stb. SetAdditionalCasParam functions.
3. Set the type of the CAS server after setting initial parameters.

**Note**: In case when no additional parameters were set via stb.SetAdditionalCasParam, command line for smdaemon could be one of the following:

- smdaemon -daemon -l 0 -auto_register
  – if stb.SetCASParam was not called before stb.SetCASType.
- smdaemon -daemon -l 0 -auto_register –rsurl serverAddr
  – if via function stb.SetCASParam was set only server address but not server port.
- smdaemon -daemon -l 0 -auto_register –rsurl serverAddr:port
  – if via function stb.SetCASParam were set both server address and server port.

### Setting additional CAS parameters.

Some additional CAS parameters can be set by using SetAdditionalCasParam.

**Verimatrix.**

The following list of additional parameters is implemented:

- "COMPANY"
- "SERVERADDRESS"
- "SERVERPORT"
- "STOREPATH"
- "KEEP_NULL_PACKETS"
- "ERRORLEVEL"
- "TIMEOUT"
- "ENABLE_IPV6"
- "DISABLELOG"
- "CLEARLOG"
- "ROOTCERT"
- "MESSAGE_FORMAT"
- "PREFERRED_VKS"
- "CONNECTION_RETRIES"
- "MIN_KEY_RETRY_INTERVAL"
- "KEYMGR_DISABLED"

Listed above parameters are the same parameters which can be set using verimatrix.ini config file (**stb.LoadCASIniFile(filename)**).

Example:

**stb.SetAdditionalCasParam("TIMEOUT",5);**

– set connection timeout to 5 seconds.

**stb.SetAdditionalCasParam("CONNECTION_RETRIES",3);**

– set connection retries count to 3.

By default **STOREPATH** parameter is set to "/flash", and **ROOTCERT** – to "/flash/rootcert.pem". **It is not recommended to change those parameters.**

**SecureMedia.**

The following list of additional parameters is implemented:

- **"sm_add_cmd"** – value of this parameter is appended to command line arguments for /home/default/smd_start.sh script after all parameters from stb.SetCASParam().
- **"sm_full_cmd"** – value of this parameter completely overrides command line arguments for smdaemon. That is smdaemon starts **only** with specified in parameter value options. Please pay attention that smdaemon starts in blocking mode, so value for **"sm_full_cmd"** parameter should contain **-daemon option**.

**Custom CAS plugin.**

Custom CAS plugin interface description can be found by the following link http://soft.infomir.com.ua/mag200/CAS/Readme_en.txt.

To use custom CAS plugin put plugin dynamic library into /home/default folder. Ensure that the plugin has the name in the following form:

/home/default/libCasCustom_x.so,

where "_x" corresponds to given CAS type (seestb.SetCASType)

For example:

**stb.SetCASType(4);**

forces player to search /home/default/libCasCustom4.so library as a current CAS system.

## *Appendix 4. Specifics of JS API when using the browser based on WebKit.*

### Initialization.

To initialize **stb** object proceed as follows:

1. Declare the object **stb**:
   var stb;
2. Initialize **stb** in the page initialization function with the following string:
   stb = gSTB;

This method obviously corresponds with the new method used for Mozilla Firefox, as described above.

Also make sure that the lines in the file event.js

```
observerService = Components.classes["@mozilla.org/observer-
service;1"].getService(Components.interfaces.nsIObserverService);
observerService.addObserver(myObserver, "TeletecSTB", false);
```

are replaced with the following lines:

```
try
{
  observerService = Components.classes["@mozilla.org/observer-
service;1"].getService(Components.interfaces.nsIObserverService);
  observerService.addObserver(myObserver, "TeletecSTB", false);
}catch(e)
{}
```

(it has already been done for the default nfs image.)

### Wrapper.js

JS API for WebKit provides to the user (without the necessity of connecting **wrapper.js**) with the same interface as wrapper.js, i.e.:

1. No need to continuously call
   **netscape.security.PrivilegeManager.enablePrivilege("UniversalXPConnect")**

2. If the function returns the value, it can be received in a usual manner, for example :

**var tColor = stb.GetTransparentColor();**

In this case the required prototype of the method is denoted using **WK/FF+Wrapper** string according to the Object  stb methods calling chapter.

**Cookie**

When setting a cookie, as opposite to the code of Mozilla Firefox

```
function set_coookie(str)
{
        document.cookie = str
}
```

Add  path=/ :

```
function set_coookie(str)
{
        document.cookie = str+'; path=/;'
}
```

If the correct expiry date of the  cookie (”expiry=”), is specified, the browser shall save the cookie into /flash directory. That is, the cookie shall be valid till the term expires or stb.DeleteAllCookies function is called.

**Use of alpha-transparency.**

To create transparent or semi-transparent applications based on WebKit browser add the following attribute to BODY:

background-color: none;

For setting transparency use **opacity** attribute or set the colour to **transparent**.

Alpha-transparency shall be functioning only if the following mode is specified in /etc/directfbrc:

```
pixelformat=ARGB
depth=32
bg-color=0
#bg-none
```

If alpha-transparency is not required (when ChromaKey is sufficient), 16-bit mode can be set by changing the mode in /etc/directfbrc to:

**pixelformat=RGB16**

**depth=16**

**bg-none**

**#bg-color=0**

The performance of graphic sub-system will be increased in this mode, while the memory load will be reduced.

### *Appendix 5. Remote control key codes in JavaScript.*

Remote control key codes sent to JavaScript are completely determined by the settings of the program irxevent. Mozilla Firefox uses the file/etc/lirc/lircrc, while WebKit uses the file /etc/lirc/lircrc.wk.

**The table of key codes for MAG100/MAG200 (release version <= 0.1.4)**

Table below shows the key codes received by the events processor of JavaScript:

**Table 2 Remote control keys codes (version <= 0.1.4)**

| Remote key control key | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|
| exit | 27 (0x1b) | 0 | |
| ok | 13 (0x0d) | 0 | |
| right | 39 (0x27) | 0 | |
| left | 37 (0x25) | 0 | |
| up | 38 (0x26) | 0 | |
| down | 40 (0x28) | 0 | |
| PageUp | 33 (0x21) | 0 | |
| PageDown | 34 (0x22) | 0 | |
| menu | 122 (0x7a) | 0 | |
| back | 8 (0x08) | 0 | |
| refresh | 116 (0x74) | 0 | |
| red | 112 (0x70) | 0 | |
| green | 113 (0x71) | 0 | |
| yellow | 114 (0x72) | 0 | |
| blue | 115 (0x73) | 0 | |
| channel+ | 9 (0x09) | 0 | |
| channel– | 9 (0x09) | 0 | SHIFT |
| service | 120 (0x78) | 0 | |
| tv | 121 (0x79) | 0 | |
| phone | 119 (0x77) | 0 | |
| web | 123 (0x7b) | 0 | |
| frame | 117 (0x75) | 0 | |

| Remote key control key | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|
| vol+ | 0 | 43 (0x2b) | |
| vol– | 0 | 45 (0x2d) | |
| rew | 0 | 98 (0x62) | CTRL, ALT * |
| ffwd | 0 | 102 (0x66) | CTRL, ALT * |
| stop | 0 | 115 (0x73) | CTRL, ALT * |
| play/pause | 0 | 114 (0x72) | CTRL, ALT * |
| rec | 0 | 119 (0x77) | CTRL, ALT * |
| mic | 0 | 32 (0x20) | CTRL, ALT * |
| mute | 0 | 96 (0x60) | CTRL, ALT |
| power | 0 | 117 (0x75) | CTRL, ALT |
| info | 0 | 121 (0x79) | CTRL, ALT * |
| Empty | 0 | 107 (0x6b) | CTRL, ALT * |
| 1 | 0 | 49 (0x31) | |
| 2 | 0 | 50 (0x32) | |
| 3 | 0 | 51 (0x33) | |
| 4 | 0 | 52 (0x34) | |
| 5 | 0 | 53 (0x35) | |
| 6 | 0 | 54 (0x36) | |
| 7 | 0 | 55 (0x37) | |
| 8 | 0 | 56 (0x38) | |
| 9 | 0 | 57 (0x39) | |
| 0 | 0 | 48 (0x30) | |
| The event generated when connecting USB Flash Drive | 0 | 112 (0x70) | CTRL, ALT * |
| The event generated when disconnecting USB Flash Drive | 0 | 113 (0x71) | CTRL, ALT * |

**The table of keys codes for MAG200 (RELEASE VERSION > 0.1.4)**

**The table for event processor onKeyPress**

The table below shows the keys codes received by the event processor of JavaScript onKeyPress for the browsers Mozilla Firefox and WebKit:

**Table 3 Remote control keys codes for the processor onKeyPress**

| Remote control key | Browser | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|---|
| Exit | FF | 27 (0x1b) | 0 | |
| | WK | 27 (0x1b) | 27 (0x1b) | |
| **Ok** | **FF** | **13 (0x0d)** | **0** | |
| | **WK** | **13 (0x0d)** | **13 (0x0d)** | |
| right | FF | 39 (0x27) | 0 | |
| | WK | 39 (0x27) | 39 (0x27) | |
| left | FF | 37 (0x25) | 0 | |
| | WK | 37 (0x25) | 37 (0x25) | |
| Up | FF | 38 (0x26) | 0 | |
| | WK | 38 (0x26) | 38 (0x26) | |
| down | FF | 40 (0x28) | 0 | |
| | WK | 40 (0x28) | 40 (0x28) | |
| PageUp | FF | 33 (0x21) | 0 | |
| | WK | 33 (0x21) | 33 (0x21) | |
| PageDown | FF | 34 (0x22) | 0 | |
| | WK | 34 (0x22) | 34 (0x22) | |
| menu | FF | 122 (0x7a) | 0 | CTRL |
| | WK | 122 (0x7a) | 122 (0x7a) | CTRL |
| back | FF | 8 (0x08) | 0 | |
| | WK | 8 (0x08) | 8 (0x08) | |
| refresh | FF | 116 (0x74) | 0 | CTRL |
| | WK | 116 (0x74) | 116 (0x74) | CTRL |
| red | FF | 112 (0x70) | 0 | CTRL |
| | WK | 112 (0x70) | 112 (0x70) | CTRL |
| green | FF | 113 (0x71) | 0 | CTRL |

| Remote control key | Browser | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|---|
| | WK | 113 (0x71) | 113 (0x71) | CTRL |
| yellow | FF | 114 (0x72) | 0 | CTRL |
| | WK | 114 (0x72) | 114 (0x72) | CTRL |
| blue | FF | 115 (0x73) | 0 | CTRL |
| | WK | 115 (0x73) | 115 (0x73) | CTRL |
| channel+ | FF | 9 (0x09) | 0 | |
| | WK | 9 (0x09) | 9 (0x09) | |
| channel– | FF | 9 (0x09) | 0 | SHIFT |
| | WK | 9 (0x09) | 9 (0x09) | SHIFT |
| service | FF | 120 (0x78) | 0 | CTRL |
| | WK | 120 (0x78) | 120 (0x78) | CTRL |
| Tv | FF | 121 (0x79) | 0 | CTRL |
| | WK | 121 (0x79) | 121 (0x79) | CTRL |
| phone | FF | 119 (0x77) | 0 | CTRL |
| | WK | 119 (0x77) | 119 (0x77) | CTRL |
| web | FF | 123 (0x7b) | 0 | CTRL |
| | WK | 123 (0x7b) | 123 (0x7b) | CTRL |
| frame | FF | 117 (0x75) | 0 | CTRL |
| | WK | 117 (0x75) | 117 (0x75) | CTRL |
| vol+ | FF | 0 | 43 (0x2b) | |
| | WK | 43 (0x2b) | 43 (0x2b) | |
| vol– | FF | 0 | 45 (0x2d) | |
| | WK | 45 (0x2d) | 45 (0x2d) | |
| rew | FF | 0 | 98 (0x62) | ALT |
| | WK | 98 (0x62) | 98 (0x62) | ALT |
| ffwd | FF | 0 | 102 (0x66) | ALT |
| | WK | 102 (0x66) | 102 (0x66) | ALT |
| stop | FF | 0 | 115 (0x73) | ALT |
| | WK | 115 (0x73) | 115 (0x73) | ALT |
| play/pause | FF | 0 | 114 (0x72) | ALT |
| | WK | 114 (0x72) | 114 (0x72) | ALT |
| rec | FF | 0 | 119 (0x77) | ALT |
| | WK | 119 (0x77) | 119 (0x77) | ALT |

| Remote control key | Browser | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|---|
| mic | FF | 0 | 32 (0x20) | ALT |
| | WK | 32 (0x20) | 32 (0x20) | ALT |
| mute | FF | 0 | 96 (0x60) | ALT |
| | WK | 96 (0x60) | 96 (0x60) | ALT |
| power | FF | 0 | 117 (0x75) | ALT |
| | WK | 117 (0x75) | 117 (0x75) | ALT |
| info | FF | 0 | 121 (0x79) | ALT |
| | WK | 121 (0x79) | 121 (0x79) | ALT |
| Empty | FF | 0 | 108 (0x6c) | ALT |
| | WK | 108 (0x6c) | 108 (0x6c) | ALT |
| 1 | FF | 0 | 49 (0x31) | |
| | WK | 49 (0x31) | 49 (0x31) | |
| 2 | FF | 0 | 50 (0x32) | |
| | WK | 50 (0x32) | 50 (0x32) | |
| 3 | FF | 0 | 51 (0x33) | |
| | WK | 51 (0x33) | 51 (0x33) | |
| 4 | FF | 0 | 52 (0x34) | |
| | WK | 52 (0x34) | 52 (0x34) | |
| 5 | FF | 0 | 53 (0x35) | |
| | WK | 53 (0x35) | 53 (0x35) | |
| 6 | FF | 0 | 54 (0x36) | |
| | WK | 54 (0x36) | 54 (0x36) | |
| 7 | FF | 0 | 55 (0x37) | |
| | WK | 55 (0x37) | 55 (0x37) | |
| 8 | FF | 0 | 56 (0x38) | |
| | WK | 56 (0x38) | 56 (0x38) | |
| 9 | FF | 0 | 57 (0x39) | |
| | WK | 57 (0x39) | 57 (0x39) | |
| 0 | FF | 0 | 48 (0x30) | |
| | WK | 48 (0x30) | 48 (0x30) | |
| Event generated when connecting USB Flash Drive | FF | 0 | 112 (0x70) | ALT |
| | WK | 112 (0x70) | 112 (0x70) | ALT |

| Remote control key | Browser | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|---|
| Event generated when disconnecting USB Flash Drive | FF | 0 | 113 (0x71) | ALT |
| | WK | 113 (0x71) | 113 (0x71) | ALT |

Where *keyCode* – is the *keyCode* field of the event received by the processor, and *which* – is the *which field of the event received by the processor*.

FF – means Mozilla Firefox, and WK – WebKit.

**Remark 1.** As compared to the previous version the code of “OK” key for the browser based on WebKit was changed.

**Remark 2.** With the purpose of making the processor independent of the browser it is recommended to add the following code at the beginning of the processor:

**var code = e.keyCode | e.which;**

and further to analyze the meaning of the **code** as the key code taking into account the modifiers specified in the table.

**Remark 3**. As compared to the previous releases for all keys having the modifier CTRL+ALT it was replaced with ALT for Mozilla Firefox.

**The table for event processor onKeyDown and onKeyUp**

The table below lists the keys codes received by the event processor JavaScript onKeyDown and onKeyUp for browsers Mozilla Firefox and WebKit:

**Table 4. Remote control keys codes for processors onKeyDown и onKeyUp**

| Remote control key | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|
| exit | 27 (0x1b) | 27 (0x1b) | |
| ok | 13 (0x0d) | 13 (0x0d) | |
| right | 39 (0x27) | 39 (0x27) | |
| left | 37 (0x25) | 37 (0x25) | |
| up | 38 (0x26) | 38 (0x26) | |
| down | 40 (0x28) | 40 (0x28) | |
| PageUp | 33 (0x21) | 33 (0x21) | |
| PageDown | 34 (0x22) | 34 (0x22) | |
| menu | 122 (0x7a) | 122 (0x7a) | CTRL |
| back | 8 (0x08) | 8 (0x08) | |
| refresh | 116 (0x74) | 116 (0x74) | CTRL |
| red | 112 (0x70) | 112 (0x70) | CTRL |
| green | 113 (0x71) | 113 (0x71) | CTRL |
| yellow | 114 (0x72) | 114 (0x72) | CTRL |
| blue | 115 (0x73) | 115 (0x73) | CTRL |
| channel+ | 9 (0x09) | 9 (0x09) | |
| channel– | 9 (0x09) | 9 (0x09) | SHIFT |
| service | 120 (0x78) | 120 (0x78) | CTRL |
| tv | 121 (0x79) | 121 (0x79) | CTRL |
| phone | 119 (0x77) | 119 (0x77) | CTRL |
| web | 123 (0x7b) | 123 (0x7b) | CTRL |
| frame | 117 (0x75) | 117 (0x75) | CTRL |
| vol+ | 107(0x6b) | 107(0x6b) | |
| vol– | 109(0x6d) | 109(0x6d) | |
| rew | 66 (0x42) | 66 (0x42) | ALT |
| ffwd | 70 (0x46) | 70 (0x46) | ALT |

| Remote control key | keyCode, dec(hex) | which, dec(hex) | Flags |
|---|---|---|---|
| stop | 83(0x53) | 83(0x53) | ALT |
| play/pause | 82(0x52) | 82(0x52) | ALT |
| rec | 87(0x57) | 87(0x57) | ALT |
| mic | 32 (0x20) | 32 (0x20) | ALT |
| mute | 192(0xC0) | 192 (0xC0) | ALT |
| power | 85 (0x55) | 85 (0x55) | ALT |
| info | 89 (0x59) | 89 (0x59) | ALT |
| Empty | 76 (0x4c) | 76 (0x4c) | ALT |
| 1 | 49 (0x31) | 49 (0x31) | |
| 2 | 50 (0x32) | 50 (0x32) | |
| 3 | 51 (0x33) | 51 (0x33) | |
| 4 | 52 (0x34) | 52 (0x34) | |
| 5 | 53 (0x35) | 53 (0x35) | |
| 6 | 54 (0x36) | 54 (0x36) | |
| 7 | 55 (0x37) | 55 (0x37) | |
| 8 | 56 (0x38) | 56 (0x38) | |
| 9 | 57 (0x39) | 57 (0x39) | |
| 0 | 48 (0x30) | 48 (0x30) | |
| Event generated when connecting USB Flash Drive | 80 (0x50) | 80 (0x50) | ALT |
| Event generated when disconnecting USB Flash Drive | 81 (0x51) | 81 (0x51) | ALT |

**Remark 1.** Event processing with onKeyDown is much simpler than event processing with onKeyPress, because remote control keys codes are not duplicated in onKeyDown, excluding the keys channel+ и channel-.

### *Appendix 6. MAG200 front panel indication control*

For controlling the indicator and LED on the front panel call the function stb.ExecAction as follows:

stb.ExecAction(“front_panel param”) ,

where param – is the parameter string for the script setFpanel.sh, which performs output to the front panel. The parameters of this script are described in the document “Operator guide_MAG200.pdf”.

## *Appendix 7. Use of keys on the  MAG200 front panel*

Pressing the keys on the front panel generates the event of pressing the key on the keyboard. Utilities **fpxevent** for **FireFox** and **fpqevent** for **WebKit** are used for this purpose. The events of pressing are translated according to configuration files  /etc/lirc/lircrc wk for FireFox and /etc/lirc/lircrc.wk for WebKit. These utilities are described in more detail in the documents “Operator guide _MAG200.pdf”.

## *Appendix 8. Switching video output modes.*

### Setting video output mode.

Switching video output mode is performed by calling the method **stb.ExecAction** in the following form:

stb.ExecAction(“tvsystem mode”),

where mode can accept the following values:

PAL

576p-50

720p-50

1080i-50

1080p-50 (for MAG250)

NTSC

576p-60

720p-60

1080i-60

1080p-60 (for MAG250)

For example, stb.ExecAction(“tvsystem PAL”) sets the video output in the mode PAL(576i).

**Remark.** The changes shall come into force after the device is restarted.

### Receiving the current mode of video output

Use the function stb.RDir in the form:

var mode = stb.RDir('vmode') for receiving te current mode of te video output.

Where **mode** can take the following values:

576i – PAL mode

576p – 576p@50 mode

720p – 720p@50 mode

1080i – 1080i@50 mode

1080p –1080p@50 mode (for MAG250)

480i –NTSC mode

720p60 –720p@60 mode

1080i60 –1080i@60 mode

1080p60 –1080p@60 mode(for MAG250)

**Remark.** In this case the current operating mode shall be returned, that is, it can be changed only after reloading.

## *Appendix 9. Control of the size and position of the browser window on the basis of WebKit.*

If necessary, the size of the browser window can be reduced and its position on the screen changed. To do so call the following functions:

**window.moveTo( x, y)** – offsets the window to the position with the coordinates **x** and **y**.

**window.resizeTo( width, height) –** sets the window width in the value **width**, and the height - in the value **height**.

## *Appendix 10. Setting graphic resolution of the browser based on the WebKit.*

### Setting resolution

Graphic resolution can be set using the function stb.ExecAction in the following manner:

stb.ExecAction('graphicres **mode**'),

where **mode** can take the following values:

tvsystem**_res** – graphic resolution agrees with the resolution of the video output (aspect 1:1)

**720** – graphic resolution 720x576, with hard scaling of this resolution to the whole screen in the modes 1080i and 720p.

**1280** – graphic resolution 1280x720, with hard scaling of this resolution to the whole screen in the mode 1080i.

**1920** – graphic resolution 1920x1080.

**Remark.** If the resolution of the video output is less than the graphic resolution set, the graphic resolution shall be considered equal to the video output resolution.

**Remark.** The changes shall come into force after the device is restarted.

### Receiving current graphic resolution

Current graphic resolution can be received using the function stb.RDir in the following way:

var gres = stb.RDir('gmode'),

where **gres** shall take the values: **tvsystem_res, 720, 1280, 1920** as described in the previous item.

Graphic resolution can be also received using **screen.width** и **screen.height** and:

screen.width – horizontal rersolution.

screen.height – vertical resolution.

**Remark.** In this case the current operating mode shall be returned because it can be changed only after reloading.

## *Appendix 11. Operation with environment variables.*

Javascript API for MAG 200 device allows receiving and setting special environment variables stored in ROM. Hereinafter they shall be defined as “environment variables”.

**Attention.** Environment variables are stored in ROM possessing a large but limited number of re-recordings, therefore, it is ultimately recommended not to save the parameters that often change, for example, at each start of the device.

### Setting and getting environment variables.

**Warning**: starting from JS API version 325 there is new method for accessing variables available. New method does not rely on «rdir.cgi» script. It has time and usability optimization. You can read about new function here: stb.GetEnv, stb.SetEnv

For getting an environment variable use the function RDir with the parameter getenv (see. stb.RDir)

To set the environment variable use the function RDir with the parameter setenv (see. stb.RDir)

If several variables must be set, it is recommended to set all these variables in one call. Simply insert the line «"|"» (3 symbols), between the pairs “name-value”, i.e. call RDir in the following form:

stb.RDir(‘setenv name_1 val_1 "|" name_2 val_2 "|" … "|" name_n val_n’),

where n – the number of variables, name_n – the name of variable under number n, val_n – the value to be set to the variable with the name name_n.

For example:

stb.RDir(‘setenv mcip_conf 224.50.7.50 "|" mcip_img_conf 111.1.2.3 "|" portal2 http://some_portal.com’)

set variables  mcip_conf, mcip_img_conf and portal2 to the values 224.50.7.50, 111.1.2.3 and http://some_portal.com corerspondingly.

Example of Javascript code for such call is shown below:

```
    ........
var CONCAT = ' "|" ',
str    = '',
ipaddr = '',    // variable 1
mcip   = '',    // variable 2
mcport = '';    // variable 3
str += 'ipaddr_conf ' + ipaddr;
str += CONCAT + 'mcip_conf ' + mcip;
str += CONCAT + 'mcport_conf ' + mcport;
batchSetEnvValues(str);
    ........
// Packet setting of environment variables in ROM
function batchSetEnvValues(str){
  stb.RDir ('setenv ' + str);
}
    ........
```

**Environment variables used in standard program**

To ensure correct operation of the standard program use the following environment variables:

**ipaddr_conf** – static  IP address. If this variable is not set the device receives IP address, netmask, gateway, DNS and NTP automatically (via DHCP) when starting.

**netmask** – subnetwork mask.

**gatewayip** – IP address of the default gateway.

**dnsip** – IP address of DNS server.

**ntpurl** – URL of NTP server.

**mcip_conf** – multicast address, from which bootstrap is received.

**mcport_conf** – port number from which bootstrap is received

**mcip_img_conf** – multicast address from which the image for updating is received (imageupdate).

**mcport_img_conf** – port number from which the image for updating is received (imageupdate).

**mcip_mng_conf** – multicast address of the managing channel.

**mcport_mng_conf** – port number of the managing channel.

**portal1** – URL of portal 1.

**portal2** – URL of portal 2.

**volume** – default volume level.

**language** – language index of user interface. 0 – English, 1 – Russian.

**upnp_conf** – Start(true) or not start(false) UPnP client.

**use_portal_dhcp** – use(true) or not use(false) the variable valueportal_dhcp as starting portal if the variables portal1 and portal2 are not set.

**portal_dhcp** – URL of the portal set by the operator using protocol DHCP.

## *Appendix 12. Software updates JavaScript API*

Software updates subsystem operations available by using «**stbUpdate**» object. The object provides an interface to the update manager.

Update Manager allows you to initiate and display the status of the software upgrade process

**Attention!** Before any software update operation you must stop every single process of media content accessing and displaying.

### Use cases

### Common scenario using of an object

#### *Possible states of an object*

Update manager is a finite state machine.

State is accessible via **getStatus** method. Initial state – «Idle» (value «21»)

Any active operation upon the update system is allowed only in «Idle» state

Right after starting of an operation state machine leaves «Idle» state and must be considered as busy until «Idle» state turned back. So, after every start of operation that been committed user should wait for «Idle» state back

#### *User interface hints*

#### *Additional information*

Besides general state of an object there is additional information available to user (in terms of machine-user interface).

Additional information can be accessed via **getStatusStr** method

This information should be sampled on periodic basis (recommended period is 1 sec) to keep user interface up to date with the actual data behind update process

***Progress indicator***

Progress indicator reflects measure of completion of current operation and can be read via **getPercents** method

***Update file check***

There is update file check procedure available before actual update process will be committed.

Update file checking available via **startCheck** method. It take path (either URL or path to file) to update file as parameter. For example,

«/media/usbdisk/mag200/imageupdate»,

or

«http://mag.infomir.com.ua/mag200/imageupdate»

After checking procedure completion available information could be read via **getImageVersionStr**, **getImageDateStr**, **getImageDescStr** methods.

***Bank selection procedure***

Active bank could be found via **getActiveBank** method.

In case when active bank defined (**getActiveBank** method returned either «0» or «1») update should be applied to bank opposite to active. In case of undefined active bank (**getActiveBank** method returned «-1») selection of bank to update depends on implementation.

***Software update***

Software update procedure could be started via **startUpdate** method. It takes bank number and path (either URL or path to file) to update file as parameters. For example,

«/media/usbdisk/mag200/imageupdate»

or

«http://mag.infomir.com.ua/mag200/imageupdate»

In general, it takes few minutes to complete the operation. So, it is recommended to show status of the operation to user during the process.

During process execution additional information is available via **getPercents**, **getStatus**, **getStatusStr**, **getImageVersionStr**, **getImageDateStr**, **getImageDescStr** methods.

### Software update

Software update procedure include following stages:

1. Waiting for «Idle» state of manager.
2. Verification of update file (using **startCheck** method).
3. Analyzing of available update file's attributes and making a decision about update start.
4. Selecting memory bank where update will take place (using data received via **getActiveBank** method).
5. Initiate update operation by using **startUpdate** method. During execution of operation state of manager is available via **getStatus** method, additional information is available via **getStatusStr** method and the progress indicator via **getPercents** method.
6. In case of any trouble object will be set to «Idle» state and additional information would help to understand what happened
7. In case of success device will be forced to reboot

### Automatic software update

Initiated by **startAutoUpdate** method

Dedicated status form will be summoned to user interface during execution of this operation. Form will show overall progress.

Automatic software update procedure include following stages:

1. Verification if the update file's attributes. In case of any trouble process will be terminated and related status provided
2. Automatic memory bank selection taking place. When active bank equals «0» then memory bank «1» will be selected. And memory bank «0» will be selected in any other cases.
3. If there additional software version check was specified then update will happen only if current software version is older than available for update.
4. Actual software update taking place.

**Methods of the «stbUpdate» object**

### stbUpdate.getStatusStr

string getStatusStr()

Returns status of update subsystem as string

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | Localized string.<br>Localization done according to the settings of internal configuration portal. | String is describing current operation status |

### stbUpdate.getStatus

int getStatus()

Returns status of update subsystem as code

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| int | -1, 1..26 | -1: not defined<br>1: Signature init error (final state error)<br>2: Wrong device model<br>3: Section size exceed partition size on FLASH<br>4: Required FLASH section not found. Aborting update<br>5: Updating Kernel<br>6: Updating Image<br>7: Internal error (final state error)<br>8: Inspecting environment section<br>9: Updating environment variables<br>10: Updating Bootstrap section<br>11: Skipping Bootstrap section<br>12: Updating User FS section<br>13: Skipping User FS section<br>14: Updating second boot<br>15: Updating logotype<br>16: Update finished OK (final state OK)<br>17: Wrong Signature (final state OK)<br>18: Erasing flash section |

| Data type | Allowed value | Description |
|---|---|---|
| | | 19: Flash write error (final state error)<br>20: File write error (final state error)<br>21: Idle (final state OK)<br>22: Invalid file header (final state error)<br>23: Inspecting update file<br>23: File check finished<br>24: File check finished (final state OK)<br>25: File not found (final state error)<br>26: Preparing for work<br>27: Read error (final state error) |

**stbUpdate.getPercents**

int getPercents()

Returns progress indicator value expressed in percents

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| int | 0-100 | Value expressed in per cents |

**stbUpdate.getActiveBank**

int getActiveBank()

Returns memory bank number, which was used for current software loading

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| Integer | -1, 0, 1 | 0 – first memory bank<br>1 – second memory bank<br>-1 – memory bank is undefined (it could be possible if device was booted from network storage. For examples, NFS) |

**stbUpdate.GetFlashBankCount**

int GetFlashBankCount ()

Returns total number of flash banks

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| Integer | 1,2 | There is always one bank exist |

**stbUpdate.startCheck**

void startCheck (string image)

Initiate update file check operation.

Operation should be started only from «Idle» state

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| image | Either:<br>- URL pointing to update file using HTTP scheme (for example, http://test.com/imageupdate)<br>- or file path to update file (for example, /media/usbdisk/mag200 /imageupdate) | File will be verified and available data will be read |

**Returned value:**

None

**stbUpdate.getImageVersionStr**

string getImageVersionStr()

Returns version of the Image

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | Returns version of the Image, which was assigned upon image creation procedure |

**stbUpdate.getImageDateStr**

string getImageDateStr()

Returns date of then Image creation

**Parameters:**

Нет

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | Returns date of the Image, which was assigned upon image creation procedure |

**stbUpdate.getImageDescStr**

string getImageDescStr()

Returns description of the Image.

**Parameters:**

None

**Returned value:**

| Data type | Allowed value | Description |
|---|---|---|
| string | | Returns description of the Image, which was assigned upon image creation procedure |

**stbUpdate.startUpdate**

void startUpdate(int bank, string image)

Initiating software update procedure from given update file to given memory bank.

Operation should be started only from «Idle» state

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| bank | 0,1 | «0» – update using first memory bank<br>«1» - update using second memory bank |
| image | Either:<br>- URL pointing to update file using HTTP scheme (for example, http://test.com/imageupdate)<br>- or file path to update file (for example, | Software update procedure will be started using given update file |

| | | |
|---|---|---|
| | /media/usbdisk/mag200 /imageupdate) | |

**Returned value:**

None

**stbUpdate.startAutoUpdate**

void startAutoUpdate(string image, bool checkVersion)

Initiating automatic software update procedure from given update file.

Memory bank will be selected automatically.

During update procedure there is dedicated user interface form will be displayed.

**Parameters:**

| Parameter | Allowed value | Description |
|---|---|---|
| image | Either:<br>- URL pointing to update file using HTTP scheme (for example, http://test.com/imageupdate)<br>- or file path to update file (for example, /media/usbdisk/mag200 /imageupdate) | Software update procedure will be started using given update file |
| checkVersion | False, true | «True» – update procedure will be committed only if current version of software older than available version<br><br>«False» – do not do version check |

**Returned value:**

None

**stbUpdate API usage example**

Following HTML page code demonstrate basic usage of update API.

«Update file» field is specified update file location (could be either URL or file path).

Check procedure could be started by pressing «Check» button (or button «0» on RC).

Update procedure could be started by pressing «Update» button (or button «1» on RC).

Automatic update procedure could be started by pressing button «2» on RC.

```
<html>
<head>
<script>
function onLoad()
{
   setTimeout("timerHandler()", 1000);
   document.getElementById("btn1").focus();
}
function timerHandler()
{
       document.getElementById("div0").innerHTML=stbUpdate.getStatus();
       document.getElementById("div1").innerHTML="\"" + stbUpdate.getStatusStr() + "\"";
       document.getElementById("div2").innerHTML=stbUpdate.getPercents() + "%";
       document.getElementById("div3").innerHTML=stbUpdate.getImageVersionStr();
       document.getElementById("div4").innerHTML=stbUpdate.getImageDateStr();

       setTimeout("timerHandler()", 1000)
}
function onKeyPress(e)
{
       if(e.which==48)
       {
          startCheck();
          return;
       }
       if(e.which==49)
       {
          startUpdate();
          return;
       }
       if(e.which==50)
       {
          stbUpdate.startAutoUpdate(document.getElementById("input1").value, true);
          return;
       }
}
function startCheck()
{
   stbUpdate.startCheck(document.getElementById("input1").value);
}
function startUpdate()
```

```
{
   stbUpdate.startUpdate(0, document.getElementById("input1").value);
}
</script>
</head>
<body onload="onLoad()" onkeypress="onKeyPress(event)" style="background:silver; padding: 40px">

<table cellspacing="10" cellpadding="10" border="3">
<tr>
<td><div>Status code: </div></td>
<td><div id="div0"></div></td>
</tr>
<tr>
<td><div>Status str: </div></td>
<td><div id="div1"></div></td>
</tr>
<tr>
<td><div>Progress: </div></td>
<td><div id="div2"></div></td>
</tr>
<tr>
<td><div>ImageVersionStr: </div></td>
<td><div id="div3"></div></td>
</tr>
<tr>
<td><div>ImageDateStr: </div></td>
<td><div id="div4"></div></td>
</tr>
</table>

<p>
<hr>
File to update: <input id = "input1" value="http://mag.infomir.com.ua/200/imageupdate" style="width: 350px"><br>
<hr>
<input id = "btn1" value="Check" onclick="startCheck()" type="button">
<input id = "btn2" value="Update" onclick="startUpdate()" type="button">
</body>
</html>
```

## *Appendix 13. JavaScript API for PVR subsystem*

All operations with PVR subsystem are performed via **pvrManager** object.

**pvrManager** object does not need any additional initialization. It is always accessible from JavaScript context.

This object provides API for channel recording manager (PVR).

Recording manager allows to schedule a task, which will record specified channel(stream) into local storage during the specified time range.

**Note.** It is allowed to record only channels that contain mpeg-ts stream. It can be mulicast stream or stream from http server.

### Description of pvrManager object

#### Error codes table

| Allowed values | Description |
|---|---|
| 0 | Operation successful. |
| -1 | Bad argument. |
| -2 | Not enough memory. |
| -3 | Wrong recording range (start or end time).<br>e.i. recording duration must be less or equal than 24 hours. |
| -4 | Task with specified ID was not found. |
| -5 | Wrong file name. Folder where you want to save recording must exist and begin with /media/USB- or /ram/media/USB-. |
| -6 | Duplicate tasks. Recording with that file name already exists. |
| -7 | Error opening stream URL. |
| -8 | Error opening output file. |
| -9 | Maximum number of simultaneous recording is exceeded. It does not mean task number but number of simultaneous recording. See also SetMaxRecordingCnt. |
| -10 | Manager got end of stream and recording has finished earlier keeping the recorded file. |
| -11 | Error writing output file. E.i. disk is full or has been disconnected during recording. |

**Task state table**

| Allowed values | Description |
|---|---|
| 1 | Waiting for a start of actual recording. |
| 2 | Recording. |
| 3 | Error occurred. Recording is stopped. |
| 4 | Recording completed. |

**pvrManager.CreateTask**

string CreateTask (string url, string fileName, string startTime, string endTime)

Schedule channel recording task.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| url | http://...<br>rtp://...<br>udp://... | URL of the stream, that wiil be recorded. |
| fileName | /media/USB-...<br>or<br>/ram/media/USB-... | Full file name of recording. |
| startTime | UTC time in "YYYYMMDDThhmmss" format or number of seconds since Epoch (1970/01/01 UTC). | Recording start time. |
| endTime | UTC time in "YYYYMMDDThhmmss" format or number of seconds since Epoch (1970/01/01 UTC). | Recording end time. |

**Return value:**

| Data type | Allowed values | Description |
|---|---|---|
| string | Unique task identifier if operation was successful, otherwise return value is a string representing error code (<0) from Error codes table | Unique task identifier or error code. |

**Note.** Number of seconds since Epoch can be obtained via Date object:

```
var date = new Date();
var startTime = date.getTime()/1000;
```

**pvrManager.GetAllTasks**

string GetAllTasks()

Get the list of all tasks.

**Parameters:**

None.

**Return value:**

| Data type | Allowed values | Description |
|---|---|---|
| string | [task_1,...,task_n],<br>where task_n has the following form:<br><br>{"id":1,"state":0,"errorCode":0,"filename":"/media/USB-1/1.ts", "url":"http://192.168.1.1/mpegts", "startTime":"3452344145","endTime":"3452345345"}<br><br>Here,<br>id – unique task identifier.<br>state – current task state. See Task state table.<br>errorCode – error code. See Error codes table.<br>fileName – requested recording file name.<br>url – URL of recorded stream.<br>startTime and endTime – start and end recording time. See CreateTask. | List of all recording tasks in form of JSON array. |

### pvrManager.GetTasksByIDs

string GetTasksByIDs (string idList)

Get task list by identifier list.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| idList | string in form:<br>[id_1,...,id_n] | List of task identifiers in form of JSON array. |

**Return value:**

| Data type | Allowed values | Description |
|---|---|---|
| string | [task_1,...,task_n],<br>where task_n has the following form:<br><br>{"id":1,"state":0,"errorCode":0,<br>"filename":"/media/USB-1/1.ts",<br>"url":"http://192.168.1.1/mpegts",<br>"startTime":"3452344145",<br>"endTime":"3452345345"}<br><br>Here,<br>id – unique task identifier.<br>state – current task state. See Task state table.<br>errorCode – error code. See Error codes table.<br>fileName – requested recording file name.<br>url – URL of recorded stream.<br>startTime and endTime – start and end recording time. See CreateTask. | List of all matched recording tasks in form of JSON array. |

**pvrManager.GetTaskByID**

string GetTaskByID (string id)

Get recording task by its identifier.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| id | string | Task identifier. |

**Return value:**

| Data type | Allowed values | Description |
|---|---|---|
| string | Return value in the following form:<br>{"id":1,"state":0,"errorCode":0,<br>"filename":"/media/USB-1/1.ts",<br>"url":"http://192.168.1.1/mpegts",<br>"startTime":"3452344145",<br>"endTime":"3452345345"}<br>For more details see GetAllTasks.<br>Rturn value will be the string '{}' if no task found. | |

**pvrManager.RemoveTask**

void RemoveTask (string id, int removeType)

Remove recording task by its identifier.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| id | | Task identifier. |
| removeType | 0-3 | 0 – do not remove any files.<br>1 – if temporary file exists, rename it into resulting file.<br>2 – remove only temporary file, if it exists.<br>3 – remove both temporary and resulting files. |

**Return value:**

None.

**pvrManager.ChangeEndTime**

int ChangeEndTime(string id, string endTime)

Change recording end time.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| id | | Task identifier. |
| endTime | See CreateTask. | New recording end time. |

**Return value:**

| Data type | Allowed values | Description |
|---|---|---|
| int | See Error codes table. | |

### pvrManager.SetMaxRecordingCnt

void SetMaxRecordingCnt (int maxCnt)

Set maximum number of simultaneous recording.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| maxCnt | | Maximum number of simultaneous recording. |

**Return value:**

None.

## *Appendix 14. JavaScript API for download manager*

All operations with PVR subsystem are performed using «**stbDownloadManager**» object.

**stbDownloadManager** object itself does not require any additional initialization. It is always accessible from JavaScript context.

The object provides API to download manager.

Download manager allows adding and scheduling download tasks, which will try to download and store remote file into local storage.

### Methods of the «stbDownloadManager» object

#### DeleteJob

bool DeleteJob(double id, boolean deleteFile)

Delete given job.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| Id | Numeric (-1, 0-4294967295) | ID of a job assigned for this operation |
| deleteFile | boolean | True – delete associated local file<br>False – just delete the job, keep local file |

**Return value(s):**

| Data type | Allowed values | Description |
|---|---|---|
| Boolean | True, false | Result of operation |

#### StartJob

bool StartJob(double id)

Change state of given job to «waiting for queue».

This will cause job to start downloading process once queue will be ready to schedule the job

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|

| | | |
|---|---|---|
| Id | Numeric (-1, 0-4294967295) | ID of a job assigned for this operation |

**Return value(s):**

| Data type | Allowed values | Description |
|---|---|---|
| Boolean | True, false | Result of operation |

**StopJob**

bool StopJob(double id)

Change state of given job to «stopped».

This state will cause the job will never be selected by scheduler for downloading

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| Id | Numeric (-1, 0-4294967295) | ID of a job assigned for this operation |

**Return value(s):**

| Data type | Allowed values | Description |
|---|---|---|
| Boolean | True, false | Result of operation |

**AdjustJobPriority**

void AdjustJobPriority(double id, Boolean rise)

Change priority of given job.

Priority can either be increased or decreased

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| Id | Numeric (-1, 0-4294967295) | ID of a job assigned for this operation |
| rise | boolean | True – increase priority<br>False – decrease priority |

**Return value(s):**

None

**PlayDownloadedMedia**

void PlayDownloadedMedia (double id)

Will play given job in dedicated «media player» window of the internal portal.
This effectively generate «stbEvent.onMediaAvailable(…)» event

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| Id | Numeric (-1, 0-4294967295) | ID of a job assigned for this operation |

**Return value(s):**

None

**AddJob**

bool AddJob(String urlToDownload, String filePath)

Will add the job for file downloading using URL *urlToDownload*
In case of success local file will be stored in *filePath*
By the time of the operation local file should not exist.

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| urlToDownload | URL in form of the string | Remote file will be download using URL |
| filePath | Path to local storage pointing to non-existed file | Dowloaded file will be stored using this path to local storage |

**Return value(s):**

| Data type | Allowed values | Description |
|---|---|---|
| Boolean | True, false | Returns true if job were added |

**AddMeasureJob**

bool AddMeasureJob(String urlToDownload)

Similar to «AddJob». Create a special job, that will download given file without saving the result to file storage.

This job is using as connection test facility. You can calculate download speed (once job is finished) using «timeWasted» и «sizeDone» attributes
You can only create one such connection at the moment. So, you have to delete it each time you want to create next one

**Arguments:**

| Agrument | Allowed values | Description |
|---|---|---|
| urlToDownload | URL | URL must point to remote file |

**Return value(s):**

| Data type | Allowed values | Description |
|---|---|---|
| Boolean | True, false | Result of operation |

**GetQueueInfo**

string GetQueueInfo(string idList = «»)

Get info about queue of jobs

**Parameters:**

| Parameter | Allowed values | Description |
|---|---|---|
| idList | List of object's IDs.<br>For example «1.0, 2.0, 3.0» | If list is not empty information for given jobs will be returned.<br>Whole queue will be returned in other case. |

**Return value(s):**

| Data type | Allowed values | Description |
|---|---|---|
| string | Contain JavaScript array of objects.<br>Size of the array depends on operation's result.<br><br>Each object should have these fields:<br><br>«id» - numeric (0-4294967295)<br>«state» - numeric<br>«stateStr» - string | «id» - ID of the job<br>«state» - state number ( 0 – Stopped, 1 – WaitingQueue, 2 – Running, 3 – Completed, 4 – TemporaryError, 5 - PermanentError)<br>«stateStr» - state string (localization is supported for this string resource)<br>«url» - URL of remote file<br>«filePath» - path to local |

| | | |
|---|---|---|
| | «url» - string<br>«filePath» - string<br>«progressPct» - numeric (0-100)<br>«sizeDone» - numeric<br>«sizeTotal» - numeric (value -1 if undefined)<br>«prio» - numeric<br>«attempt» - numeric | storage<br>«progressPct» - progress of downloading process (expressed in per cent)<br>«sizeDone» - size of already downloaded data<br>«sizeTotal» - total file size<br>«prio» - priority of the job<br>«attempt» - number of the download attempt |

**GetMeasureInfo**

string GetMeasureInfo()

Similar to GetQueueInfo.

Get information about special (test) job

If test job does not exist then return nothing

**Arguments:**

Нет

**Return:**

| Data type | Allowed value | Description |
|---|---|---|
| string | See GetQueueInfo | See GetQueueInfo |

## *Appendix 15. Support for external media (FLASH drives, USB drives*

Device is capable for automatic detection and use of supported external drives.

There will be an attempt to mount all available partitions in case when supported external drive connected.

Contents of every valid partition will be available for file operations in so-called «mount points». Those mount points located in «/media/» folder and have format «USB-»<XXX>-<YYY>, where XXX is the serial number of connected drive and YYY is the partition number (count from 1)

There is an additional information available for mount points using «/home/default/rdir.cgi get_storage_info» sub function. Without arguments «get_storage_info» returns an array of JSON objects, which holds an information about every mount point available at the moment. Or you can supply «get_storage_info» sub function with the argument of interested mount point (excluding /media/ prefix)

***Change history.***

**Version 1.20**

Added events 8 and 0x23 in List of the events used.

Added the following functions:

stb.GetMetadataInfo,

stb.SetAutoFrameRate,

stb.ForceHDMItoDVI,

stb.LoadExternalSubtitles,

stb.SetSubtitlesEncoding,

stb.GetEnv,

stb.SetEnv,

stb.GetDeviceSerialNumber,

stb.GetDeviceVendor,

stb.GetDeviceModel,

stb.GetDeviceVersionHardware,

stb.GetDeviceMacAddress,

stb.GetDeviceActiveBank,

stb.GetDeviceImageVersion,

stb.GetDeviceImageDesc,

stb.GetDeviceImageVersionCurrent,

stb.GetLanLinkStatus,

stb.GetWifiLinkStatus,

stb.GetWepKey64ByPassPhrase,

stb.GetWepKey128ByPassPhrase,

stb.GetWifiGroups,

stb.ServiceControl,

stb.GetSmbGroups,

stb.GetSmbServers,

stb.GetSmbShares,

stb.IsFolderExist,

stb.IsFileExist,

stb.SendEventToPortal,

stb.IsWebWindowExist,

stb.IsInternalPortalActive,

stb.EnableAppButton,

stbDownloadManager.AddMeasureJob,

stbDownloadManager.GetMeasureInfo.

Fixed error: function which starts recording task pvrManager.CreateTask had wrong name.

In stb.Play added possibility to specify external subtitle file.

In pvrManager.GetAllTasks, pvrManager.GetTasksByIDs, pvrManager.GetTaskByID format of return value to comply with JSON notation.

Added Appendix 15. Support for external media (FLASH drives, USB drives

Updated description of stbUpdate.getStatus

Updated Setting and getting environment variables.

**Version 1.19**

Added events 7 and 0x22 in List of the events used.

Added the following functions:

Play  using proxy server,

SetWebProxy,

GetVideoInfo.

Added Appendix 13. JavaScript API for PVR subsystem.

Added Appendix 14. JavaScript API for download manager.

**Version 1.18**

Added DualMono event description in List of the events used.

Added the following functions:

stb.SetBufferSize,

stb.GetBufferLoad.

Added custom CAS plugin description in stb.SetCASType and Custom CAS plugin.

**Version 1.17**

Added Appendix 12. Software updates JavaScript API.

Added the following functions:

stb.GetMute,

stb.StartLocalCfg,

stb.ShowVirtualKeyboard,

stb.HideVirtualKeyboard,

stb.EnableServiceButton,

stb.EnableVKButton,

stb.EnableSpatialNavigation,

stb.EnableSetCookieFrom.

**Version 1.16**

Added ResolveIP command description in stb.RDir.

Added RTSP server types in stb.SetupRTSP.

Corrected description of stb.SetLoop.

Added the following functions:

stb.SetAdditionalCasParam,

stb.SetAudioOperationalMode,

` stb.SetHDMIAudioOut,

stb.SetDRC,

stb.SetStereoMode,

stb.EnableJavaScriptInterrupt,

stb.ShowSubtitle.

Extended description in Appendix 3. CAS usage and settings.

Added new events in List of the events  used.

Added some "solution" in solution.

**Version 1.14**

The functions added:

stb.SetPosTimeEx,

stb.GetPosTimeEx,

stb.GetMediaLenEx.

Information on the functions stb.SetSubtitlesSize, stb.SetSubtitlesFont and stb.SetSubtitlesOffs was updated in the table “Accessibility of function on IPTV devices”. Description of new parameters for the function stb.RDir. was added .

Supplement  11. Operation with environment variables was added.

**Version 1.13**

Function stb.DeleteAllCookies was added.

Information in the section Cookie is updated.

### Version 1.12

Description of the function stb.GetAspect. was updated

Description of the function stb.SetCASDescrambling. was corrected.

### Version 1.11

Added:

Appendix 10. Setting graphic resolution of the browser based on WebKit.

Getting current video output mode

### Version 1.10

Functions added:

SetSubtitleLangs,

GetSubtitlePID,

SetSubtitlePID,

SetBrightness,

SetSaturation,

SetContrast,

GetBrightness,

GetSaturation,

GetContrast.

Information on setting event system was added to the item “Setting event system”.

Information on Standby mode was added. See .StandBy.

*Appendix 9. Control of the size and position of the browser window on the basis of WebKit.* *was added.*

Error in the description of the function stb.SetViewport was corrected.