

**Спецификация JavaScript API для управления  
IPTV-приставками MAG100 и MAG200**

V. 1.05

2009

## Содержание.

Содержание.....	2
Информация о документе. ....	6
Введение. ....	7
Использование API.....	7
Типы данных и формат аргументов. ....	7
Вызов методов объекта stb.....	8
Доступность функций на IPTV-приставках.....	9
Методы объекта stb. ....	12
stb.InitPlayer .....	12
stb.DeinitPlayer .....	12
stb.Play .....	12
stb.PlaySolution .....	13
stb.Stop.....	13
stb.Pause .....	14
stb.Continue.....	14
stb.SetPosTime.....	14
stb.SetPosPercent .....	15
stb.SetPosPercentEx .....	15
stb.GetPosTime .....	15
stb.GetPosPercent .....	16
stb.GetPosPercentEx.....	16
stb.GetMediaLen.....	17
stb.SetSpeed.....	17
stb.SetAudioPID.....	18
stb.SetPIG.....	18
stb.SetAlphaLevel .....	19
stb.SetVolume .....	19
stb.SetUserFlickerControl .....	19
stb.SetFlicker .....	20
stb.SetDefaultFlicker .....	20
stb.SetLoop .....	21

stb.SetVideoControl .....	21
stb.SetVideoState .....	22
stb.SetChromaKey .....	22
stb.SetMode .....	23
stb.SetWinMode .....	23
stb.SetTopWin .....	24
stb.SetWinAlphaLevel .....	24
stb.SetAspect .....	25
stb.Rotate .....	26
stb.SetMute .....	27
stb.SetMicVolume .....	27
stb.GetMicVolume .....	27
stb.GetVolume .....	28
stb.Step .....	28
stb.SetupRTSP .....	28
stb.SetViewPort .....	30
stb.IsPlaying .....	30
stb.Version .....	31
stb.SetupSPdif .....	31
stb.SetSubtitles .....	32
stb.SetSubtitlesSize .....	32
stb.SetSubtitlesFont .....	33
stb.SetSubtitlesOffs .....	33
stb.GetSpeed .....	33
stb.GetAudioPID .....	34
stb.GetPIG .....	35
stb.GetAlphaLevel .....	35
stb.GetWinAlphaLevel .....	35
stb.SetTransparentColor .....	36
stb.GetTransparentColor .....	36
stb.IgnoreUpdates .....	37
stb.ExecAction .....	37
stb.SetCASType .....	38
stb.SetCASParam .....	38
stb.LoadCASIniFile .....	39

stb.SetCASDescrambling.....	39
stb.GetAspect.....	40
stb.StandBy.....	41
stb.RDir.....	41
stb.SetAudioLangs.....	42
stb.GetAudioPIDs.....	43
stb.GetSubtitlePIDs.....	44
stb.ReadCFG.....	45
stb.WriteCFG.....	45
stb.WritePrefs.....	46
Событийная модель в JavaScript.....	47
Настройка событийной системы.....	47
Список используемых событий.....	47
Приложение 1. Использование API.....	48
Инициализация объекта stb.....	48
Инициализация плеера.....	48
Особенности версий JavaScript API >= 308).....	49
Инициализация плеера (Версия JavaScript API >= 308).....	49
Wrapper.js.....	49
Инициализация событийной системы.....	50
Пример использования API.....	50
Приложение 2. Форматы видео контента и примеры использования.....	52
Формат параметров функции stb.Play.....	52
solution.....	52
URL.....	54
atrack и vtrack.....	55
Примеры:.....	55
Приложение 3. Использование и настройка CAS Verimatrix.....	57
Приложение 4. Особенности JS API при использовании браузера на базе WebKit.....	58
Инициализация.....	58
Wrapper.js.....	58
Cookie.....	59
Использование альфа-прозрачности.....	59
Приложение 5. Коды кнопок пульта ДУ в JavaScript.....	61
Таблица кодов кнопок для MAG100/MAG200 (ВЕРСИЯ РЕЛИЗА <= 0.1.4).....	61

Таблица кодов кнопок для MAG200 (ВЕРСИЯ РЕЛИЗА > 0.1.4).....	63
--	----

### ***Информация о документе.***

Ревизия документа	1.05
Версия JavaScript API	309
Версия STB API	109
Версия плеера MAG200	0x503
Версия плеера MAG100	0x23

## **Введение.**

Данный документ описывает программный интерфейс, позволяющий из JavaScript управлять IPTV-приставкой (в том числе воспроизводить различные виды видео контента и включает событийную модель IPTV-приставки). Документ предполагает знание JavaScript.

## **Использование API.**

Предполагается, что описанные функции используются из JavaScript контекста на MAG100/MAG200, на которых в качестве браузера поставляется браузер Mozilla Firefox 1.5.

## **Типы данных и формат аргументов.**

Далее будут использоваться обозначения:

**int** – для числовых типов.

**bool** – для логических типов.

**string** – для строковых типов.

В документе принято следующее соглашение: если перед типом аргумента стоит ключевое слово **out**, то данный параметр используется для возврата значения из функции. Из JavaScript достаточно вызвать функцию с пустым объектом в качестве данного параметра, а затем получить значение из поля **value** данного объекта.

Например:

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

Примеры использования и инициализации API и событий IPTV-приставки описаны в [приложении 1](#) и в пункте "[Настройка событийной системы](#)".

Все действия над IPTV-приставкой производятся через объекты **stb** и **stbEvent**.

## **Вызов методов объекта stb.**

У метода объекта stb может быть описан не один прототип, что связано с различными механизмами возврата результата работы метода. В таком случае перед прототипом будет указано, где этот прототип действует, с помощью следующих обозначений:

**Firefox** – прототип действует при вызове метода из браузера Mozilla Firefox.

**WK/FF+Wrapper** – прототип действует при вызове метода из браузера WebKit или из браузера Mozilla Firefox **через wrapper.js**.

Для того, чтобы вызвать какой-либо метод **stb** из какой-либо JavaScript функции необходимо в начале этой функции добавить строку:

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

Это правило действует только при вызове метода из браузера Mozilla Firefox **без использования wrapper.js**.



## ***Доступность функций на IPTV-приставках.***

Ниже описаны функции интерфейса, их доступность и особенности для каждой IPTV-приставки. Функции отмеченные знаком “–” присутствуют в API для совместимости, но ничего не делают.

**Таблица 1 Совместимость функций JavaScript API для MAG100 и MAG200**

Методы объекта <b>stb</b>	MAG100	MAG200
<a href="#">InitPlayer</a>	+	+
<a href="#">DeinitPlayer</a>	+	+
<a href="#">Play</a>	+	+
<a href="#">PlaySolution</a>	+	+
<a href="#">Stop</a>	+	+
<a href="#">Pause</a>	+	+
<a href="#">Continue</a>	+	+
<a href="#">SetPosTime</a>	+	+
<a href="#">SetPosPercent</a>	+	+
<a href="#">SetPosPercentEx</a>	+	+
<a href="#">GetPosTime</a>	+	+
<a href="#">GetPosPercent</a>	+	+
<a href="#">GetPosPercentEx</a>	+	+
<a href="#">GetMediaLen</a>	+	+
<a href="#">SetSpeed</a>	+	+
<a href="#">SetAudioPID</a>	+	+
<a href="#">SetPIG</a>	+	+
<a href="#">SetAlphaLevel</a>	+	+
<a href="#">SetVolume</a>	+	+
<a href="#">SetUserFlickerControl</a>	+	–
<a href="#">SetFlicker</a>	+	+ (отличие от MAG100)
<a href="#">SetDefaultFlicker</a>	+	+ (отличие от MAG100)
<a href="#">SetLoop</a>	+	+
<a href="#">SetVideoControl</a>	+	+
<a href="#">SetVideoState</a>	+	+
<a href="#">SetChromaKey</a>	+	+

Методы объекта <b>stb</b>	MAG100	MAG200
<a href="#">SetMode</a>	+	+
<a href="#">SetWinMode</a>	+	+
<a href="#">SetTopWin</a>	+	+
<a href="#">SetWinAlphaLevel</a>	+	+
<a href="#">SetAspect</a>	+	+ (ДОП. ВОЗМОЖНОСТИ)
<a href="#">Rotate</a>	+	–
<a href="#">SetMute</a>	+	+
<a href="#">SetMicVolume</a>	+	–
<a href="#">GetMicVolume</a>	+	–
<a href="#">GetVolume</a>	+	+
<a href="#">Step</a>	+	–
<a href="#">SetupRTSP</a>	+	+
<a href="#">SetViewport</a>	+	+
<a href="#">IsPlaying</a>	+	+
<a href="#">Version</a>	+	+
<a href="#">SetupSPdif</a>	+	+
<a href="#">SetSubtitles</a>	+	+
<a href="#">SetSubtitlesSize</a>	+	–*
<a href="#">SetSubtitlesFont</a>	+	–*
<a href="#">SetSubtitlesOffs</a>	+	–*
<a href="#">GetSpeed</a>	+	+
<a href="#">GetAudioPID</a>	+	+
<a href="#">GetPIG</a>	+	+
<a href="#">GetAlphaLevel</a>	+	+
<a href="#">GetWinAlphaLevel</a>	+	+
<a href="#">SetTransparentColor</a>	+	+
<a href="#">GetTransparentColor</a>	+	+
<a href="#">IgnoreUpdates</a>	+	+
<a href="#">ExecAction</a>	+	+
<a href="#">SetCASType</a>	+	+
<a href="#">SetCASParam</a>	+	+
<a href="#">LoadCASIniFile</a>	+	+
<a href="#">SetCASDescrambling</a>	–	+

Методы объекта <b>stb</b>	MAG100	MAG200
<a href="#">GetAspect</a>	+	+
<a href="#">StandBy</a>	+	+
<a href="#">RDir</a>	+	+
<a href="#">SetAudioLangs</a>	+	+
<a href="#">GetAudioPIDs</a>	+	+
<a href="#">GetSubtitlePIDs</a>	+	+
<a href="#">ReadCFG</a>	+	+
<a href="#">WriteCFG</a>	+	+
<a href="#">WritePrefs</a>	+	+

\* – нет поддержки, пока не будут реализованы текстовые субтитры на MAG200.

## Методы объекта *stb*.

### **stb.InitPlayer**

void InitPlayer()

Инициализирует плеер. Необходимо вызвать эту функцию в самом начале работы с плеером. Особенности описаны в [приложении 1](#).

#### **Параметры:**

Нет.

#### **Возвращаемое значение:**

Нет.

### **stb.DeinitPlayer**

void DeinitPlayer()

Деинициализирует плеер.

#### **Параметры:**

Нет.

#### **Возвращаемое значение:**

Нет.

### **stb.Play**

void Play(string playStr)

Запустить воспроизведение медиа контента, как указано в **AStr**.

#### **Параметры:**

playStr – строка вида “<solution> <URL> [atrack:<anum>] [vtrack:<vnum>]”

Параметры	Допустимые значения	Описание
solution	rtp, rtsp, mp3, auto, mpegps, mpegts, mp4	Тип медиа контента. Зависит от типа IPTV-приставки. Таблица поддерживаемых форматов, а также описание типов медиа контента

Параметры	Допустимые значения	Описание
		находится в <a href="#">приложении 2</a> .
URL		Адрес контента, который необходимо запустить на воспроизведение. Зависит от типа. Более детальная информация в <a href="#">приложении 2</a> .
atrack:<anum>		Задаёт номер(PID) аудио дорожки. Необязательный параметр.
vtrack:<vnum>		Задаёт номер(PID) видео дорожки. Необязательный параметр.

**Возвращаемое значение:**

Нет.

## stb.PlaySolution

void PlaySolution(string solution, string URL)

Воспроизводит медиа контент, заданного типа (**solution**) по заданному **URL**.

**Параметры:**

Параметры	Допустимые значения	Описание
solution		Соответствует параметру <b>solution</b> из функции <a href="#">stb.Play</a>
URL		Адрес контента, который необходимо запустить на воспроизведение. Зависит от типа. Более детальная информация в <a href="#">приложении 2</a> .

**Возвращаемое значение:**

Нет.

## stb.Stop

void Stop()

Остановить воспроизведение.

[Continue\(\)](#) начнёт воспроизведение с начала.

**Параметры:**

Нет.

**Возвращаемое значение:**

Нет.

**stb.Pause**

void Pause()

Поставить воспроизведение на паузу.

[Continue\(\)](#) продолжит воспроизведение с текущего места.

**Параметры:**

Нет.

**Возвращаемое значение:**

Нет.

**stb.Continue**

void Continue()

Продолжить воспроизведение (после [Pause\(\)](#)) или начать заново (после [Stop\(\)](#)).

**Параметры:**

Нет.

**Возвращаемое значение:**

Нет.

**stb.SetPosTime**

void SetPosTime(int time)

Установить текущую позицию воспроизведения во времени.

**Параметры:**

Параметры	Допустимые значения	Описание
time	time >= 0	Позиция в секундах от начала контента, с которого необходимо начать/продолжить воспроизведение

Параметры	Допустимые значения	Описание
		(позиционирование по контенту).

**Возвращаемое значение:**

Нет.

### **stb.SetPosPercent**

void SetPosPercent(int prc)

Установить текущую позицию в процентах.

#### **Параметры:**

Параметры	Допустимые значения	Описание
prc	0..100	Позиция в процентах от общей длительности контента, с которой необходимо начать/продолжить воспроизведение.

**Возвращаемое значение:**

Нет.

### **stb.SetPosPercentEx**

void SetPosPercentEx(int prc)

Установить текущую позицию в процентах.

#### **Параметры:**

Параметры	Допустимые значения	Описание
prc	0..10000	Позиция в сотых долях процента от общей длительности контента, с которой необходимо начать/продолжить воспроизведение.

**Возвращаемое значение:**

Нет.

### **stb.GetPosTime**

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

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

Получить текущую позицию во времени.

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
time		Текущая позиция в секундах от начала контента.

**stb.GetPosPercent**

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

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

Получить текущую позицию в процентах.

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
prc		Текущая позиция в процентах от всей длительности контента.

**stb.GetPosPercentEx**

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

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

Получить текущую позицию в сотых долях процента.

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
prc	0..10000	Текущая позиция в процентах от всей длительности контента.



## stb.GetMediaLen

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

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

Получить длительность текущего контента.

### Параметры:

Нет.

### Возвращаемое значение:

Параметры	Допустимые значения	Описание
len		Общая длительность текущего контента в секундах.

## stb.SetSpeed

void SetSpeed(int speed)

Установить скорость воспроизведения.

### Параметры:

Параметры	Допустимые значения	Описание
speed	-8..8	Задаёт новую скорость воспроизведения: 1 - нормальная 2 - 2x 3 - 4x 4 - 8x 5 - 16x 6 - 1/2 7 - 1/4 8 - 12x -1 - обратное воспроизведение -2 - обратное воспроизведение 2x -3 - обратное воспроизведение 4x -4 - обратное воспроизведение 8x -5 - обратное воспроизведение 16x -8 - обратное воспроизведение 12x

**Возвращаемое значение:**

Нет.

**stb.SetAudioPID**

void SetAudioPID(int pid)

Установить номер дорожки (PID) для аудио.

**Параметры:**

Параметры	Допустимые значения	Описание
pid		Задаёт номер или PID аудио дорожки, которая будет воспроизводиться в текущем контенте. В случае отсутствия такой дорожки звук работать не будет.

**Возвращаемое значение:**

Нет.

**stb.SetPIG**

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

Установить расположение и режим видео окна.

**Параметры:**

Параметры	Допустимые значения	Описание
state	0..1	Если state=1 – показывать видео на полный экран. Если state=0 – показывать видео в заданном прямоугольнике.
scale		Масштаб видео окна. Настоящий множитель размеров видео окна равен <b>scale/256</b> .
x		Смещение верхнего левого угла видео окна от начала экрана
y		Смещение верхнего левого угла видео окна от начала экрана

**Возвращаемое значение:**

Нет.

### **stb.SetAlphaLevel**

void SetAlphaLevel(int alpha)

Установить альфа прозрачность видео окна.

#### **Параметры:**

Параметры	Допустимые значения	Описание
alpha	0..255	Прозрачность видео окна: 0 – полностью прозрачное; 255 – полностью непрозрачное.

#### **Возвращаемое значение:**

Нет.

### **stb.SetVolume**

void SetVolume(int volume)

Установить уровень громкости.

#### **Параметры:**

Параметры	Допустимые значения	Описание
volume	0..100	Уровень громкости: 0 – нет звука; 100 – максимальный уровень.

#### **Возвращаемое значение:**

Нет.

### **stb.SetUserFlickerControl**

void SetUserFlickerControl(int mode)

Установить режим управления Flicker-фильтром.

#### **Платформы: MAG100**

#### **Параметры:**

Параметры	Допустимые значения	Описание
mode	0..1	Режим управления Flicker-фильтром:

Параметры	Допустимые значения	Описание
		0 –пользователь API сам управляет Flicker-фильтром (см. <a href="#">stb.SetFlicker</a> и <a href="#">stb.SetDefaultFlicker</a> ); 1 – плеер сам включает Flicker-фильтр на время паузы и остановки и сам выключает во время воспроизведения.

**Возвращаемое значение:**

Нет.

### **stb.SetFlicker**

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

Установить параметры Flicker-фильтра.

**Платформы: MAG100, MAG200**(см. примечание)

**Параметры:**

Параметры	Допустимые значения	Описание
state	0..1	Включение/выключение Flicker-фильтра: 0 – выключить Flicker-фильтр; 1 – включить Flicker-фильтр.
flk	0..15	Уровень сглаживания.
shp	0..15	Уровень чёткости.

**Возвращаемое значение:**

Нет.

**Примечание:**

На MAG200 фликер фильтр включается только для графического окна, поэтому имеет смысл установить его только один раз при загрузке и не отключать.

Для MAG200 параметры flk и shp игнорируются.

### **stb.SetDefaultFlicker**

void SetDefaultFlicker(int state)

Установить параметры Flicker-фильтра по умолчанию.

**Платформы: MAG100, MAG200**(см. примечание)

**Параметры:**

Параметры	Допустимые значения	Описание
state	0..1	Включение/выключение Flicker-фильтра: 0 – выключить Flicker-фильтр; 1 – включить Flicker-фильтр. При этом устанавливаются значения по умолчанию для чёткости и сглаживания.

**Возвращаемое значение:**

Нет.

**Примечание:**

На MAG200 фликер фильтр включается только для графического окна, поэтому имеет смысл установить его только один раз при загрузке и не отключать.

### **stb.SetLoop**

void SetLoop(int loop)

Установить или снять повторное воспроизведение.

**Параметры:**

Параметры	Допустимые значения	Описание
loop	0..1	0 – выключить повторное воспроизведение контента; 1 – включить повторное воспроизведение контента.

**Возвращаемое значение:**

Нет.

### **stb.SetVideoControl**

void SetVideoControl (int mode)

Установить режим управления видео окном:

**Параметры:**

Параметры	Допустимые значения	Описание
mode	0..1	Режим управления: 0 – плеер сам включает видео окно при начале воспроизведения и выключает при остановке; 1 – пользователь API с помощью <a href="#">stb.SetVideoState</a> указывает показывать видео окно или нет.

**Возвращаемое значение:**

Нет.

### **stb.SetVideoState**

void SetVideoState (int state)

Включить или выключить видео окно.

#### **Параметры:**

Параметры	Допустимые значения	Описание
state	0..1	Разрешить/запретить отображение видео: 0 – видео окно не отображается; 1 – видео окно отображается при наличии потока.

**Возвращаемое значение:**

Нет.

#### **Примечания:**

Действует только если до этого разрешено управление пользователем с помощью [stb.SetVideoControl](#).

### **stb.SetChromaKey**

void SetChromaKey(int key,int mask)

Установить заданный цвет и маску для использования в качестве ChromaKey (прозрачность какого-либо цвета на всём окне).

#### **Параметры:**

Параметры	Допустимые значения	Описание
key	0..0xffffffff	Задаёт цвет в формате RGB. Если цвет пикселя окна после наложения маски <b>mask</b> совпадает с данным цветом, то пиксель считается прозрачным.
mask	0..0xffffffff	Задаёт маску для <b>key</b> . Если маска равна 0xffffffff, то прозрачным будет считаться цвет заданный параметром <b>key</b> .

**Возвращаемое значение:**

Нет.

**Примечания:**

Какие-либо изменения на экране будут видны только если включен режим ChromaKey функциями [stb.SetMode](#) или [stb.SetWinMode](#).

### **stb.SetMode**

void SetMode(int mode)

Включить (mode=1) или отключить (mode=0) режим ChromaKey для видео окна.

**Параметры:**

Параметры	Допустимые значения	Описание
mode	0..1	Режим ChromaKey для видео окна: 0 – режим отключен; 1 – режим включен. При включенном режиме будут действовать параметры заданные <a href="#">stb.SetChromaKey</a> или <a href="#">stb.SetTransparentColor</a> .

**Возвращаемое значение:**

Нет.

### **stb.SetWinMode**

void SetWinMode (int winNum, int mode)

Включить или отключить режим ChromaKey для заданного окна

## Параметры:

Параметры	Допустимые значения	Описание
winNum	0..1	Номер окна, для которого применяется данная функция: 0 – графическое окно; 1 – видео окно.
mode	0..1	Режим ChromaKey для видео окна: 0 – режим отключен; 1 – режим включен. При включенном режиме будут действовать параметры заданные <a href="#">stb.SetChromaKey</a> или <a href="#">stb.SetTransparentColor</a> .

**Возвращаемое значение:**

Нет.

## stb.SetTopWin

void SetTopWin(int winNum)

Установить заданное окно поверх остальных.

## Параметры:

Параметры	Допустимые значения	Описание
winNum	0..1	Номер окна, для которого применяется данная функция: 0 – графическое окно; 1 – видео окно.

**Возвращаемое значение:**

Нет.

## stb.SetWinAlphaLevel

void SetWinAlphaLevel(int winNum, int alpha)

Установить альфа прозрачность заданного окна.

## Параметры:



Параметры	Допустимые значения	Описание
winNum	0..1	Номер окна, для которого применяется данная функция: 0 – графическое окно; 1 – видео окно.
alpha	0..255	Прозрачность заданного окна: 0 – полностью прозрачное; 255 – полностью непрозрачное.

**Возвращаемое значение:**

Нет.

## **stb.SetAspect**

void SetAspect(int aspect)

Установить формат видео изображения.

### **Параметры:**

Параметры	Допустимые значения	Описание																
aspect		<p>Задаёт формат видео изображения. Состоит из 2 тетрад:</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">7</td> <td style="text-align: center;">6</td> <td style="text-align: center;">5</td> <td style="text-align: center;">4</td> <td style="text-align: center;">3</td> <td style="text-align: center;">2</td> <td style="text-align: center;">1</td> <td style="text-align: center;">0</td> </tr> <tr> <td colspan="4" style="text-align: center;"><b>aspH</b></td> <td colspan="4" style="text-align: center;"><b>aspL</b></td> </tr> </table> <p><b>Для MAG100 aspH игнорируется.</b></p>	7	6	5	4	3	2	1	0	<b>aspH</b>				<b>aspL</b>			
7	6	5	4	3	2	1	0											
<b>aspH</b>				<b>aspL</b>														
aspL	0..3	<p>Задаёт соотношение сторон: 0 – автоматическое; 1 – 20:9; 2 – 16:9; 3 – 4:3.</p>																
aspH	0..3	<p>Задаёт преобразование формата видео: 0 – как есть, видео растягивается на весь экран; 1 – режим Letter Box, видео пропорционально увеличивается до</p>																

Параметры	Допустимые значения	Описание
		<p>размеров экрана по большей границе;</p> <p>2 – режим Pan&amp;Scan, видео пропорционально увеличивается до размеров экрана по меньшей границе;</p> <p>3 – комбинированный режим, средний между Letter Box и Pan&amp;Scan.</p> <p>4 – режим увеличенный;</p> <p>5 – режим оптимальный.</p> <p>Только для MAG200</p>

**Возвращаемое значение:**

Нет.

**Примечания:**

Для MAG100 **aspH** игнорируется.

Для MAG200 **aspL** имеет место только в оконном режиме, а **asIH** только в полноэкранном режиме, см. [stb.SetPIG](#)

**stb.Rotate**

void Rotate(int angle)

Повернуть видео изображение.

**Платформы: MAG100**

**Параметры:**

Параметры	Допустимые значения	Описание
angle	0, 90, 180, 270	Поворачивает содержимое видео окна на заданный угол относительно исходного положения.

**Возвращаемое значение:**

Нет.

## stb.SetMute

void SetMute(int mute)

Выключить или включить звук с восстановлением уровня громкости.

### Параметры:

Параметры	Допустимые значения	Описание
mute	0..1	Включает/отключает звук: 0 – звук включен; 1 – звук выключен.
После цикла выключение – включение звука данной функцией уровень громкости остаётся неизменным.		

### Возвращаемое значение:

Нет.

## stb.SetMicVolume

void SetMicVolume(int micvol)

Установить уровень громкость микрофона.

### Платформы: MAG100

### Параметры:

Параметры	Допустимые значения	Описание
micvol	0..100	Установить уровень громкости микрофона: 0 – минимальная громкость; 100 – максимальная громкость.

### Возвращаемое значение:

Нет.

## stb.GetMicVolume

FireFox: void GetMicVolume(out int micvol);

WK/FF+Wrapper: int GetMicVolume();

Получить текущий уровень громкости микрофона.

### Платформы: MAG100

### Параметры:

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
micvol	0..100	Возвращает текущий уровень громкости микрофона.

## **stb.GetVolume**

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

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

Получить уровень громкости.

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
vol	0..100	Возвращает текущий уровень громкости.

## **stb.Step**

void Step()

Отобразить один следующий кадр видео.

**Платформы: MAG100**

**Параметры:**

Нет.

**Возвращаемое значение:**

Нет.

## **stb.SetupRTSP**

void SetupRTSP(int type, int flags)

Настроить RTSP-клиент на STB.

**Параметры:**

Параметры	Допустимые значения	Описание
-----------	---------------------	----------

Параметры	Допустимые значения	Описание
type	0..2	Тип поддерживаемого RTSP-сервера: 0 – RTSP сервер на базе VLC; 1 – BitBand RTSP сервер; 2 – Kasenna RTSP сервер.
flags	0..0x3f	Управляющие флаги:  1 – включить режим поддержания соединения (keep-alive);  2 – определение окончания потока по полю <b>x-notice</b> в сообщении ANNOUNCE от сервера;  4 – определение окончания потока по полю x-notice в ответе на GET_PARAMETER;  8 – определение окончания потока через некоторое время после отсутствия видео потока от сервера;  16 (0x10) – определение окончания потока по полю согласно передаваемому в заголовке RTP пакета поля rtpptime. (Только для режима передачи видео над RTP);  32 (0x20) – использовать UDP транспорт для передачи видео.

**Возвращаемое значение:**

Нет.

## stb.SetViewPort

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

Установить расположение и размер видео окна.

### Параметры:

Параметры	Допустимые значения	Описание
xsize	зависит от разрешения экрана.	Размер видео окна по горизонтали (ширина).
ysize		Размер видео окна по вертикали (высота).
x	В сумме с xsize не должно превышать ширину экрана.	Смещение левого верхнего угла видео окна по горизонтали от начала экрана.
y	В сумме с ysize не должно превышать ширину экрана.	Смещение левого верхнего угла видео окна по вертикали от начала экрана.

### Возвращаемое значение:

Нет.

## stb.IsPlaying

Firefox: void IsPlaying(out bool bPlaying);

WK/FF+Wrapper: bool IsPlaying()

Получить текущее состояние воспроизведения:

### Параметры:

Нет.

### Возвращаемое значение:

Параметры	Допустимые значения	Описание
bPlaying	true, false	Текущее состояние воспроизведения: false – в данный момент контент не воспроизводится; true – в данный момент контент воспроизводится.

## stb.Version

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

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

Получить версию API.

### Параметры:

Нет.

### Возвращаемое значение:

Параметры	Допустимые значения	Описание
version		Строка вида: “JS API version: <JS_API version>; STB API version: <STB_API version>; Player Engine version: <Player version>”. <b>JS_API version</b> – номер версии данного API; <b>STB_API version</b> – номер версии API плеера; <b>Player version</b> – версия плеера используемого в API, в шестнадцатеричном виде.
Пример: “JS API version: 301; STB API version: 104; Player Engine version: 0x23”		

## stb.SetupSPdif

void SetupSPdif(int flags);

Установить режим вывода звука через SPdif

### Параметры:

Параметры	Допустимые значения	Описание
flags	0..2	Режим вывода через SPdif:  0 – звук выводится только на аналоговые выходы.  1 – звук выводится на аналоговые

Параметры	Допустимые значения	Описание
		<p>выходы и через SPdif в формате 2-канального PCM</p> <p>2 - звук выводится в SPdif не декодированным (AC3 ...), если кодек поддерживает, или через SPdif в формате 2-канального PCM в противном случае.</p>

**Возвращаемое значение:**

Нет.

### **stb.SetSubtitles**

void SetSubtitles(bool enable);

Включить/выключить субтитры.

#### **Параметры:**

Параметры	Допустимые значения	Описание
enable	true, false	<p>true – субтитры включены;</p> <p>false – субтитры выключены.</p>

**Возвращаемое значение:**

Нет.

#### **Примечания:**

Для MAG100 субтитры отображаются только в полноэкранном режиме.

### **stb.SetSubtitlesSize**

void SetSubtitlesSize(int size);

Установить размер текстовых субтитров – size в пикселях.

#### **Платформы: MAG100.**

#### **Параметры:**

Параметры	Допустимые значения	Описание
size		Устанавливает размер текстовых субтитров.



**Возвращаемое значение:**

Нет.

### **stb.SetSubtitlesFont**

void SetSubtitlesFont(string font);

Задать шрифт для отображения текстовых субтитров.

**Платформы: MAG100.**

**Параметры:**

Параметры	Допустимые значения	Описание
font	URL-адрес	URL адресующий в корневой файловой системе файл шрифтов. Например: "/home/default/arial.ttf"

**Возвращаемое значение:**

Нет.

### **stb.SetSubtitlesOffs**

void SetSubtitlesOffs(int offs);

Задать смещение для отображения текстовых субтитров.

**Платформы: MAG100.**

**Параметры:**

Параметры	Допустимые значения	Описание
offs		горизонтальное смещение субтитров.

**Возвращаемое значение:**

Нет.

### **stb.GetSpeed**

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

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

Получить текущую скорость воспроизведения

**Параметры:**

Параметры	Допустимые значения	Описание
-----------	---------------------	----------

Параметры	Допустимые значения	Описание
speed	-8..8	Текущая скорость воспроизведения: 1 - нормальная 2 - 2x 3 - 4x 4 - 8x 5 - 16x 6 - 1/2 7 - 1/4 8 – 12x 0 – остановлен или на паузе -1 – обратное воспроизведение -2 - обратное воспроизведение 2x -3 - обратное воспроизведение 4x -4 - обратное воспроизведение 8x -5 - обратное воспроизведение 16x -8 - обратное воспроизведение 12x

**Возвращаемое значение:**

Нет.

## stb.GetAudioPID

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

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

Получить получить номер (PID) текущей аудио дорожки.

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
pid	0..0x1fff	Номер текущей аудио дорожки.

**Примечания:**

Список всех аудио дорожек, которые плеер успел определить, можно получить с помощью [stb.GetAudioPIDs](#).

## stb.GetPIG

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

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

Получить состояние видео окна:

### Параметры:

Нет.

### Возвращаемое значение:

Параметры	Допустимые значения	Описание
isWindowed	true, false	Результат указывает на то, установлен на видео окно полноэкранный режим или нет: true – воспроизведение происходит в уменьшенном окне; false – воспроизведение происходит в полноэкранном режиме.

## stb.GetAlphaLevel

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

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

Получить уровень альфа прозрачности видео окна.

### Параметры:

Нет.

### Возвращаемое значение:

Параметры	Допустимые значения	Описание
alpha	0..255	Возвращает текущее значение альфа прозрачность для видео окна.

## stb.GetWinAlphaLevel

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

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

Получить уровень альфа прозрачности для заданного окна

### Параметры:

Параметры	Допустимые значения	Описание
winNum	0..1	Номер окна, для которого применяется данная функция: 0 – графическое окно; 1 – видео окно.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
alpha	0..255	Возвращает текущее значение альфа прозрачность для видео окна.

### **stb.SetTransparentColor**

```
void SetTransparentColor(int color);
```

Устанавливает цвет, который считается прозрачным в данный момент:

**Параметры:**

Параметры	Допустимые значения	Описание
color	0..0xffffffff	Цвет в формате RGB, который будет считаться прозрачным.

**Возвращаемое значение:**

Нет.

**Примечания:**

Функция является частным случаем [stb.SetChromaKey](#).

Какие-либо изменения на экране будут видны, только если включен режим ChromaKey функциями [stb.SetMode](#) или [stb.SetWinMode](#).

### **stb.GetTransparentColor**

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

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

Возвращает цвет, который считается прозрачным в данный момент:

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
color	0..0xffffffff	цвет в формате RGB, который на данный момент считается прозрачным.

## stb.IgnoreUpdates

void IgnoreUpdates(bool bIgnore);

Блокирует или разблокирует обновление экрана браузера:

### Параметры:

Параметры	Допустимые значения	Описание
bIgnore	true, false	true – после данного вызова графическое окно перестанет обновляться до следующего вызова с параметром false; false – после данного вызова графическое окно снова начнёт обновляться – перейдёт в обычный режим.

**Возвращаемое значение:**

Нет.

## stb.ExecAction

void ExecAction(string action);

Выполнить скрипт /home/default/action.sh с заданными параметрами.

### Параметры:

Параметры	Допустимые значения	Описание
action		Строка содержит параметры для скрипта /home/default/action.sh. Пример: stb.ExecAction('param 23 s') вызовет из оболочки sh команду "/home/default/action.sh param 23 s"

**Возвращаемое значение:**

Нет.

## stb.SetCASType

void SetCASType(int CAS\_type);

Установить тип сервера условного доступа:

**Платформы: MAG100, MAG200**

**Параметры:**

Параметры	Допустимые значения	Описание
type	0,1	0 – не задан; 1 – Verimatrix.

**Возвращаемое значение:**

Нет.

**Примечания:**

Устанавливать тип сервера условного доступа необходимо **один раз при старте** портала.

## stb.SetCASParam

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

Установить параметры CAS сервера:

**Платформы: MAG100, MAG200.**

**Параметры:**

Параметры	Допустимые значения	Описание
serverAddr		URL CAS сервера.
serverPort		Порт CAS сервера.
companyName		Имя компании, под которым данный оператор зарегистрирован на CAS сервере.
opID	1..255	Идентификатор оператора использующийся на STB. Если opID равен -1, то значение не обновляется.

Параметры	Допустимые значения	Описание
errorLevel	0..5	Уровень вывода ошибок. 0 – минимальный уровень. Если errorLevel равен -1, то уровень вывода ошибок не обновляется.

**Возвращаемое значение:**

Нет.

**Примечания:**

Вызов функции имеет эффект **только**, если она вызвана перед [stb.SetCASType](#).

### **stb.LoadCASIniFile**

void LoadCASIniFile(string iniFileName);

Загрузить настройки CAS из заданного файла.

**Платформы: MAG100,MAG200.**

**Параметры:**

Параметры	Допустимые значения	Описание
iniFileName		URL файла настроек на корневой файловой системе.

**Возвращаемое значение:**

Нет.

**Примечания:**

Инструкция по настройке CAS Verimatrix находится в [приложении](#).

Вызов функции имеет эффект **только**, если она вызвана перед [stb.SetCASType](#).

### **stb.SetCASDescrambling**

void SetCASDescrambling(int isSoftware);

Установить аппаратный либо программный режим дешифрования.

**Платформы: MAG100,MAG200.**

**Параметры:**

Параметры	Допустимые значения	Описание
isSoftware	0,1	0 –использовать аппаратное

Параметры	Допустимые значения	Описание
		декриптование. 1 – использовать программное декриптование. При отсутствии данного вызова будет использоваться аппаратное декриптование. Для MAG100 может использоваться только программное декодирование.

**Возвращаемое значение:**

Нет.

**Примечания:**

На данный момент использование функции имеет смысл только для CAS **Verimatrix**. При этом в зависимости от установленного режима плеер может декодировать **только** потоки закриптованные следующими алгоритмами:

аппаратный режим: RC4, AES;

программный режим: AES, DVB-CSA.

Данный режим устанавливается только один раз после старта портала.

Вызов функции имеет эффект **только**, если она вызвана перед [stb.SetCASType](#).

## stb.GetAspect

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

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

Вернуть текущий формат видео изображения.

**Параметры:**

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание																
aspect		Возвращает текущий формат видео изображения. Состоит из 2 тетрад: <div style="text-align: center; margin-top: 10px;"> <table border="1" style="margin: auto;"> <tr> <td style="padding: 2px 10px;">7</td> <td style="padding: 2px 10px;">6</td> <td style="padding: 2px 10px;">5</td> <td style="padding: 2px 10px;">4</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">0</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;"><b>aspH</b></td> <td colspan="4" style="text-align: center; padding: 5px;"><b>aspL</b></td> </tr> </table> </div>	7	6	5	4	3	2	1	0	<b>aspH</b>				<b>aspL</b>			
7	6	5	4	3	2	1	0											
<b>aspH</b>				<b>aspL</b>														



Параметры	Допустимые значения	Описание
		<b>Для MAG100 aspH всегда равен 0.</b>
aspL	0..3	Задаёт соотношение сторон: 0 – автоматическое; 1 – 20:9; 2 – 16:9; 3 – 4:3.
aspH	0..3	Задаёт преобразование формата видео: 0 – как есть; 1 – формат Letter Box; 2 – формат Pan&Scan; 3 – комбинированный. Только для MAG200

### stb.StandBy

void StandBy(bool bStandby);

Войти в режим StandBy или выйти из него.

#### Параметры:

Параметры	Допустимые значения	Описание
bStandby	true, false	true – перейти в режим Standby; false – выйти из режима Standby.

#### Возвращаемое значение:

Нет.

### stb.RDir

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

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

Выполнить скрипт **/home/default/rdir.cgi** с заданными параметрами, и вернуть стандартный вывод данного скрипта.

#### Параметры:

Параметры	Допустимые значения	Описание
-----------	---------------------	----------

Параметры	Допустимые значения	Описание
par	Любая строка	Строка содержит параметры, с которыми будет запускаться скрипт /home/default/rdi.cgi.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
result		Стандартный вывод, полученный при выполнении скрипта /home/default/rdi.cgi с заданными параметрами.

**Примечания:**

В **rdir.cgi** поставляемым с корневой файловой системой уже предустановлены несколько команд:

stb.RDir("SerialNumber",x) – в x вернёт серийный номер данного устройства.

stb.RDir("MACAddress",x) - получить MAC адрес

stb.RDir("IPAddress",x) - получить IP адрес

stb.RDir("HardwareVersion",x) - получить версию аппаратного обеспечения

stb.RDir("Vendor",x) - получить имя производителя STB

stb.RDir("Model ",x) - получить имя модели STB

stb.RDir("ImageVersion",x) - получить версию образа прошитого программного обеспечения

stb.RDir("ImageDescription",x) - получить информацию о образе прошитого программного обеспечения

stb.RDir("ImageDate",x) - получить дату создания образа прошитого программного обеспечения.

**stb.SetAudioLangs**

void SetAudioLangs(string priLang, string secLang);

Установить языки аудио дорожек, которые будут автоматически выбираться при получении информации о канале.

**Параметры:**

Параметры	Допустимые значения	Описание
-----------	---------------------	----------

Параметры	Допустимые значения	Описание
priLang	трёхсимвольные тэги ISO 639, например: "rus" или "eng"	При наличии информации о нескольких аудио дорожках плеер будет выбирать дорожку заданную языком priLang.  Если такая не найдена, то будет выбираться дорожка с языком secLang.  Если и такой не найдено, то выбирается дорожка первая из списка.
secLang		

**Возвращаемое значение:**

Нет.

## stb.GetAudioPIDs

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

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

Функция возвращает список аудио дорожек в потоке с описанием их языка.

### Параметры:

Нет.

**Возвращаемое значение:**

Параметры	Допустимые значения	Описание
pidsList		Список обнаруженных аудио дорожек в следующем формате:  <b>[{pid:&lt;PID1&gt;, lang:[&lt;lang1_1&gt;, &lt;lang2_1&gt;}], ... , {pid:&lt;PIDn&gt;, lang:[&lt;lang1_n&gt;, &lt;lang2_n&gt;}]</b>
PIDn		PID аудио дорожки с номером <i>n</i> .
lang1_n	трёхсимвольные тэги ISO 639	Первые два описания языков аудио дорожки с номером <i>n</i> .
lang2_n		

**Пример:** результат в виде:

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

означает, что в потоке было обнаружено 2 аудио потока: русский с PID=114 и английский с PID=115;

### Примечания:

Данная строка может быть легко преобразована в массив структур с помощью вызова функции **eval()**.

Данную функцию необходимо вызывать после прихода события с кодом 2 (см. [описание событий](#))

## stb.GetSubtitlePIDs

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

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

Функция возвращает список дорожек субтитров в потоке с описанием их языка.

### Параметры:

Нет.

### Возвращаемое значение:

Параметры	Допустимые значения	Описание
pidsList		Список обнаруженных дорожек субтитров в следующем формате:  [ <b>{pid:&lt;PID1&gt;, lang:[&lt;lang1_1&gt;, &lt;lang2_1&gt;]}</b> ], ... , <b>{pid:&lt;PIDn&gt;, lang:[&lt;lang1_n&gt;, &lt;lang2_n&gt;]}</b> ]
PIDn		PID дорожки субтитров с номером <i>n</i> .
lang1_n lang2_n	трёхсимвольные тэги ISO 639	Первые два описания языков дорожки субтитров с номером <i>n</i> .

**Пример:** результат в виде:

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

означает, что в потоке было обнаружено 2 потока субтитров: русский с PID=114 и английский с PID=115;

### Примечания:

Данная строка может быть легко преобразована в массив структур с помощью вызова функции **eval()**.

Данную функцию необходимо вызывать после прихода события с кодом 2 (см. [описание событий](#))

## **stb.ReadCFG**

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

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

string ReadCFG();

Прочитать файл настроек портала /etc/stb\_params.

### **Параметры:**

Нет.

### **Возвращаемое значение:**

Параметры	Допустимые значения	Описание
result		Возвращает содержимое файла /etc/stb_params.

## **stb.WriteCFG**

void WriteCFG(string cfg);

Прочитать файл настроек портала /etc/stb\_params.

### **Параметры:**

Параметры	Допустимые значения	Описание
cfg		Данные, которые сохранятся в файл /etc/stb_params.

### **Возвращаемое значение:**

Нет.

### **Примечания:**

Необходимо иметь ввиду, что значения PORTAL\_IP, PORTAL\_1, PORTAL\_2 используются в стартовом портале, расположенном в /home/web корневой файловой системы, поэтому желательно перед данным вызовом получить исходные значения данных параметров через [stb.ReadCFG](#) и добавить их в строку cfg.

## **stb.WritePrefs**

`void WritePrefs(string prefs);`

Сохранить строку в качестве настроек браузера (prefs.js).

### **Параметры:**

Параметры	Допустимые значения	Описание
prefs		Данные, которые сохраняются в файл настроек браузера.

### **Возвращаемое значение:**

Нет.

### **Примечания:**

Данная функция является специфической для браузера и используется для того, чтобы настроить права доступа к порталу. Это делается в стартовом портале, расположенном в /home/web корневой файловой систем, и рекомендуется её больше нигде не применять

## **Событийная модель в JavaScript.**

Под событийной моделью в JavaScript подразумевается возможность пользователя API получать от плеера события, индицирующие некоторые изменения состояния воспроизведения.

### **Настройка событийной системы**

Для подключения событий на STB необходимо:

1. в портале подключить скрипт event.js:

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

2. после инициализации плеера (см. [приложение 1](#)) необходимо вызвать функцию **initEvents()**

3. функцию которая будет вызываться при получении события необходимо устанавливать через объект **stbEvent**:

```
stbEvent.onEvent = EventCallback,
```

где EventCallback функция, которая в портале будет обрабатывать события и принимающая в качестве параметра код события.

Например:

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

4. Код последнего события также хранится в **stbEvent.event**.

### **Список используемых событий**

Определены следующие события:

Код события	Описание
1	Плеер достиг окончания медиа контента или зафиксировал длительный разрыв потока.
2	Получена информация о аудио и видео дорожках медиа контента.
4	Начало отображаться видео и/или воспроизводиться звук.

## Приложение 1. Использование API.

### Инициализация объекта **stb**.

Прежде всего необходимо создать главный объект **stb**. Для этого нужно:

1. Объявить объект **stb**:

```
var stb;
```

2. В функции инициализации страницы инициализировать **stb** следующими строками:

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

После этого данный объект создан, но при этом плеер не инициализирован, т.е. на данном этапе нельзя воспроизводить контент, а управлять видео параметрами можно только через [stb.RDir](#) (используется на MAG100). Поскольку может быть инициализирован только один плеер одновременно, то данный режим используется для вспомогательных страниц, таких как страница /home/web/index.html.

В данном режиме можно вызывать следующие функции:

[stb.Version](#),  
[stb.ExecAction](#),  
[stb.RDir](#),  
[stb.ReadCFG](#),  
[stb.WriteCFG](#),  
[stb.WritePrefs](#),  
[stb.InitPlayer](#).

### Инициализация плеера.

Для того, чтобы использовать все функции API необходимо инициализировать плеер с помощью функции [stb.InitPlayer](#). Одновременно можно инициализировать только один экземпляр плеера. Для того, чтобы инициализировать другой (например на другой странице) необходимо прежде вызвать [stb.DeinitPlayer](#) для уже инициализированного плеера.



## Особенности версий JavaScript API >= 308).

Начиная с версии JavaScript API 308 Можно использовать схему инициализации выше, а можно в пункте 2) вместо строк:

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

поставить строку:

```
stb = gSTB
```

Также начиная с версии 308 появилась возможность вызывать [stb.InitPlayer](#) много раз, при этом, сам плеер будет инициализироваться при первом вызове этой функции, а деинициализироваться только после выхода из браузера.

## Инициализация плеера (Версия JavaScript API >= 308).

Для того, чтобы использовать все функции API необходимо инициализировать плеер с помощью функции [stb.InitPlayer](#). Одновременно можно инициализировать только один экземпляр плеера. Для того, чтобы инициализировать другой (например на другой странице) необходимо прежде вызвать [stb.DeinitPlayer](#) для уже инициализированного плеера.

## Wrapper.js

Для тех разработчиков, кто не хочет постоянно вызывать функцию

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

написан вспомогательный скрипт-обёртка wrapper.js, после подключения которого можно вызывать методы объекта stb из любого места JS кода без установки привилегий.

Для подключения данного скрипта необходимо:

1. подключить данный скрипт в самом начале  

```
<script language="JavaScript" src="wrapper.js"></script>
```
2. закомментировать строки:

```
var stb  
stb=gSTB
```

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

если они есть в основном скрипте страницы.

После этого в глобальной видимости скрипта появится объект stb, у которого можно вызывать методы без установки привилегий.

## Инициализация событийной системы

Данный пункт подробно описан в пункте [“Настройка событийной системы”](#).

## Пример использования API.

Ниже приведён минимальный код HTML страницы, который просто загружается и запускает воспроизведение rtp потока, а по кнопкам stop и continue соответственно останавливает воспроизведение и заново начинает воспроизводить поток.

```
<!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>
```

Также в корневой файловой системе для MAG200 в папке /home/web/ содержится тестовая страница, на которой можно проверить и увидеть, как работают все функции API.

## Приложение 2. Форматы видео контента и примеры использования.

В данном приложении описаны типы воспроизводимого контента и их использование.

Запустить воспроизведение можно двумя функциями: [stb.Play](#) и [stb.PlaySolution](#). Параметры функции [stb.PlaySolution](#) входят в составной параметр playStr функции [stb.Play](#), поэтому далее описание будет на примере [stb.Play](#) и параметров данной функции.

### Формат параметров функции [stb.Play](#).

playStr имеет следующий формат:

“solution URL atrack:num vtrack:num”, где

#### solution

Тип медиа контента, который определяет, в каком формате находится контент, например тип контейнера и/или способ вещания.

Тип (solution)	MAG100	MAG200	Описание.
auto	+	+	Автоматическое определение типа контента, контейнера, кодека по заданному URL. <b>Различия:</b> <b>MAG100:</b> при автоопределении нельзя давать URL начинающийся с rtp://, udp://, rtsp://, т.е. автоопределение работает только с файлами. <b>MAG200:</b> правильно принимает URL начинающийся с rtp://, udp://, rtsp://.
"" (пустой)	+	+	Аналогичен <b>auto</b> .
rtp	+	+	Воспроизвести поток в формате MPEG2TS. Если URL начинается с rtp://, то будет воспроизводиться RTP поток, если с udp://, то UDP поток. <b>Различия:</b> <b>MAG100</b> автоматически подключает декодер для MPEG2 Video и MPEG Audio.

Тип (solution)	MAG100	MAG200	Описание.
			<b>MAG200</b> при наличии дополнительной информации о потоке устанавливает необходимые кодеки, например H.264, AC-3 и т.д.
rtsp	+	+	Воспроизвести контент с RTSP-сервера. <b>Различия:</b> <b>MAG100</b> автоматически подключает декодер для MPEG2 Video и MPEG Audio. <b>MAG200</b> при наличии дополнительной информации о потоке устанавливает необходимые кодеки, например H.264, AC-3 и т.д.
rtspac3	+	+	Воспроизвести поток в формате MPEG2TS используя декодеры MPEG2 Video и AC-3 Audio
rtsp_ac3	+	+	Воспроизвести контент с RTSP-сервера используя декодеры MPEG2 Video и AC-3 Audio
rtmpeg4	+	-	Воспроизвести поток в формате MPEG2TS используя декодеры MPEG4p2 Video и MPEG Audio
rtmpeg4_aac	+	-	Воспроизвести поток в формате MPEG2TS используя декодеры MPEG4p2 Video и AAC Audio
mpegts	+	+	Воспроизвести файл в формате MPEG2TS. <b>Различия:</b> <b>MAG100</b> автоматически подключает декодер для MPEG2 Video и MPEG Audio. <b>MAG200</b> при наличии дополнительной информации о потоке устанавливает необходимые кодеки, например H.264, AC-3 и т.д.
mpegps	+	+	Воспроизвести файл в формате MPEG2 Program Stream. <b>Различия:</b>

Тип (solution)	MAG100	MAG200	Описание.
			<p><b>MAG100</b> автоматически подключает декодер для MPEG2 Video и MPEG Audio.</p> <p><b>MAG200</b> при наличии дополнительной информации о потоке устанавливает необходимые кодеки, например AC-3 и т.д.</p>
file	–	+	Воспроизвести файл по заданному URL с автоматическим определением типа контента, контейнера и кодека.
mp4	+	–	Воспроизвести файл формата MP4 используя кодеки MPEG4p2 Video и AAC Audio
mp4_mpa	+	–	Воспроизвести файл формата MP4 используя кодеки MPEG4p2 Video и MPEG Audio
fm	+	+	Воспроизвести аудио из MPEG-TS потока (udp://, rtp://)
ffmpeg	–	+	MAG200 позволяет воспроизводить файлы различных форматов: avi, mkv, mpg, mp4, mov, wmv, AC-3, mp3.

## Примечания

MAG200, в отличие от MAG100, может на ходу определять кодеки и менять их, например при наличии аудио дорожек сжатых разными кодеками.

## URL

Указывает, где находится контент:

Формат URL	Описание
<b>/path</b>	локальный адрес файла на корневой файловой системе, например /media/1.mp3
<b>rtp://addr:port</b>	RTP-поток принимаемый из мультикастового или уникастового адреса <i>addr</i> и портом <i>port</i> .
<b>udp://addr:port</b>	UDP-поток принимаемый из мультикастового или уникастового адреса <i>addr</i> и портом <i>port</i> .

Формат URL	Описание
<b>rtsp://addr:port/path</b>	контент лежащий на RTSP-сервере по адресу <i>addr</i> с портом <i>port</i> по относительному пути <i>path</i>

### **atrack и vtrack**

Необязательные параметры, задающие номера аудио и видео дорожек (PID-ы для MPEG2TS) контента, которые будут воспроизводиться.

### **Примеры:**

“**mpegps /media/1.mpg**” – проигрывает Mpeg2 Program Stream файл /media/1.mpg.

“**mpegts /media/1.mpg**” – проигрывает Mpeg2 Transport Stream файл /media/1.mpg.

“**mp4 /media/1.mp4**” – проигрывает файл /media/1.mp4 формата MP4.

“**rtp 224.10.0.30:5004**” – проигрывает Mpeg2 в формате Transport Stream из указанных мультикастового адреса (224.10.0.30) и порта (5004) с использованием IGMP протокола для мультикастового вещания.

“**auto /media/file1**” – попытка автоматически определить формат файла и проиграть его.

“**rtpmpeg4 224.10.0.31:5004**” – проигрывает Mpeg4 в формате Transport Stream из указанных мультикастового адреса (224.10.0.31) и порта (5004) со звуком Mpeg2 Audio с использованием IGMP протокола для мультикастового вещания.

“**rtpmpeg4\_aac 224.10.0.32:5004 atrack:930 vtrack:920**” – проигрывает Mpeg4 в формате Transport Stream из указанных мультикастового адреса (224.10.0.32) и порта (5004) со звуком AAC с использованием IGMP протокола для мультикастового вещания. При этом автоматически выберется поток с PID=920 в качестве видео дорожки и поток с PID=930 в качестве аудио дорожки, независимо от наличия информации о дорожках в потоке.

**“rtsp rtsp://192.168.1.32:554/video/media003.mpg”** – проигрывает контент /video/media003.mpg, находящийся на RTSP-сервере с адресом 192.168.1.32 и портом 554.



### **Приложение 3. Использование и настройка CAS Verimatrix.**

Для использования CAS Verimatrix необходимо:

1. В каталог /flash/ положить файл сертификата с сервера rootcert.pem
2. Установить верное время, например с ntp сервера.
3. Настроить стартовые параметры CAS сервера одним из двух способов:
  - a. через функцию [LoadCASIniFile](#), тогда параметры автоматически возьмутся из указанного файла.
  - b. через функцию [SetCASParam\(\)](#)
  - c. для MAG200 установить режим дешифрования с помощью функции [stb.SetCASDescrambling](#)
4. Настроить тип CAS сервера, после установки стартовых параметров.

## **Приложение 4. Особенности JS API при использовании браузера на базе WebKit.**

### **Инициализация.**

Для инициализации объекта stb достаточно следующего:

1. Объявить объект **stb**:

```
var stb;
```

2. В функции инициализации страницы инициализировать **stb** следующей строкой:

```
stb = gSTB;
```

Этот метод очевидно совпадает с новым методом используемым для Mozilla Firefox, описанном [выше](#).

Также необходимо убедиться, что в файле event.js строки

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

заменены следующими:

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

(В образе по умолчанию это уже сделано.)

### **Wrapper.js**

JS API для WebKit предоставляет пользователю (**без необходимости подключать wrapper.js**) точно такой же интерфейс как и wrapper.js, то есть:

1. Нет необходимости постоянно вызывать

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

2. Если функция возвращает значение, то оно может быть получено обычным способом, например:

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

В этом случае необходимый прототип метода обозначен с помощью **WK/FF+Wrapper** согласно пункту [Вызов методов объекта stb.](#)

## **Cookie**

При установке cookie, в отличии от кода Mozilla Firefox

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

необходимо добавлять path=/ :

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

## **Использование альфа-прозрачности.**

Для того, чтобы создавать прозрачные и полупрозрачные приложения на базе WebKit браузера, необходимо у BODY поставить атрибут:

```
background-color: none;
```

Для указания прозрачности следует использовать атрибут opacity или задавать цвет transparent.

Альфа-прозрачность будет работать, только если в /etc/directfbrc установлен следующий режим:

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

В случае, когда альфа-прозрачность не нужна (когда достаточно использовать ChromaKey) можно переключить в режим 16-bit, изменив в /etc/directfbrc режим на:

**pixelformat=RGB16**

**depth=16**

**bg-none**

**#bg-color=0**

В этом режиме производительность графической подсистемы увеличится, а нагрузка на память уменьшится.

## Приложение 5. Коды кнопок пульта ДУ в JavaScript.

Коды кнопок ПДУ присылаемые в JavaScript полностью определяются настройками программы irxevent. Для Mozilla Firefox используется файл /etc/lirc/lircrc, а для WebKit используется /etc/lirc/lircrc.wk.

### Таблица кодов кнопок для MAG100/MAG200 (ВЕРСИЯ РЕЛИЗА <= 0.1.4)

В таблице ниже перечислены коды кнопок, которые получает обработчик событий JavaScript:

Таблица 2 Коды кнопок пульта ДУ (версия <= 0.1.4)

Кнопка ПДУ	keyCode, dec(hex)	which, dec(hex)	Флаги
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	

Кнопка ПДУ	keyCode, dec(hex)	which, dec(hex)	Флаги
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 *
пустая	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)	
Событие генерируемое при подключении USB Flash Drive	0	112 (0x70)	CTRL, ALT *
Событие генерируемое при отключении USB Flash Drive	0	113 (0x71)	CTRL, ALT *

## Таблица кодов кнопок для MAG200 (ВЕРСИЯ РЕЛИЗА > 0.1.4)

В таблице ниже перечислены коды кнопок, которые получает обработчик событий JavaScript для браузеров Mozilla Firefox и WebKit:

Таблица 3 Коды кнопок пульта ДУ

Кнопка ПДУ	Браузер	keyCode, dec(hex)	which, dec(hex)	Флаги
exit	FF	27 (0x1b)	0	
	WK	27 (0x1b)	27 (0x1b)	
ok	<b>FF</b>	<b>13 (0x0d)</b>	<b>0</b>	
	<b>WK</b>	<b>32 (0x20)</b>	<b>32 (0x20)</b>	<b>CTRL</b>
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
	WK	113 (0x71)	113 (0x71)	CTRL

Кнопка ПДУ	Браузер	keyCode, dec(hex)	which, dec(hex)	Флаги
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
mic	FF	0	32 (0x20)	ALT



Кнопка ПДУ	Браузер	keyCode, dec(hex)	which, dec(hex)	Флаги
	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
<b>пустая</b>	<b>FF</b>	<b>0</b>	<b>108 (0x6c)</b>	<b>ALT</b>
	<b>WK</b>	<b>108 (0x6c)</b>	<b>108 (0x6c)</b>	<b>ALT</b>
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)	
Событие генерируемое при подключении USB Flash Drive	FF	0	112 (0x70)	ALT
	WK	112 (0x70)	112 (0x70)	ALT

Кнопка ПДУ	Браузер	keyCode, dec(hex)	which, dec(hex)	Флаги
Событие генерируемое при отключении USB Flash Drive	FF	0	113 (0x71)	ALT
	WK	113 (0x71)	113 (0x71)	ALT

Здесь *keyCode* – это поле *keyCode* события полученного обработчиком, а *which* – поле *which* события полученного обработчиком.

Здесь FF – означает Mozilla Firefox, а WK – WebKit.

**Замечание 1.** Особое внимание необходимо обратить на отличие кодов кнопки «Ok» для Mozilla Firefox и WebKit.

**Замечание 2.** Для того, чтобы обработчик кнопок не зависел от браузера, рекомендуется вначале обработчика добавить следующий код:

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

и в дальнейшем анализировать значение **code** как код клавиши, принимая во внимание указанные в таблице модификаторы.

**Замечание 3.** По сравнению с предыдущими релизами для Mozilla Firefox у всех клавиш с модификаторами CTRL+ALT, модификатор заменён на ALT.

**Замечание 4.** По сравнению с предыдущими релизами для Mozilla Firefox изменён код кнопки «пустая».