

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

V. 1.16

2010

Содержание.

| | |
|---|----|
| Содержание..... | 2 |
| Информация о документе. | 7 |
| Введение. | 8 |
| Использование API..... | 8 |
| Типы данных и формат аргументов. | 8 |
| Вызов методов объекта stb..... | 9 |
| Доступность функций на IPTV-приставках..... | 10 |
| Методы объекта stb. | 13 |
| stb.InitPlayer | 13 |
| stb.DeinitPlayer | 13 |
| stb.Play | 13 |
| stb.PlaySolution | 14 |
| stb.Stop..... | 14 |
| stb.Pause | 15 |
| stb.Continue | 15 |
| stb.SetPosTime..... | 15 |
| stb.SetPosTimeEx..... | 16 |
| stb.SetPosPercent | 16 |
| stb.SetPosPercentEx | 16 |
| stb.GetPosTime | 17 |
| stb.GetPosTimeEx | 17 |
| stb.GetPosPercent | 18 |
| stb.GetPosPercentEx..... | 18 |
| stb.GetMediaLen..... | 18 |
| stb.GetMediaLenEx..... | 19 |
| stb.SetSpeed..... | 19 |
| stb.SetAudioPID..... | 20 |
| stb.SetSubtitlePID | 20 |
| stb.SetPIG..... | 21 |
| stb.SetAlphaLevel | 21 |
| stb.SetVolume..... | 21 |

| | |
|---------------------------------|----|
| stb.SetUserFlickerControl | 22 |
| stb.SetFlicker | 22 |
| stb.SetDefaultFlicker | 23 |
| stb.SetLoop | 24 |
| stb.SetVideoControl | 24 |
| stb.SetVideoState | 24 |
| stb.SetChromaKey | 25 |
| stb.SetMode | 26 |
| stb.SetWinMode | 26 |
| stb.SetTopWin | 27 |
| stb.SetWinAlphaLevel | 27 |
| stb.SetAspect | 27 |
| stb.Rotate | 29 |
| stb.SetMute | 29 |
| stb.SetMicVolume | 29 |
| stb.GetMicVolume | 30 |
| stb.GetVolume | 30 |
| stb.Step | 31 |
| stb.SetupRTSP | 31 |
| stb.SetViewport | 32 |
| stb.IsPlaying | 33 |
| stb.Version | 33 |
| stb.SetupSPdif | 34 |
| stb.SetSubtitles | 35 |
| stb.SetSubtitlesSize | 35 |
| stb.SetSubtitlesFont | 35 |
| stb.SetSubtitlesOffs | 36 |
| stb.GetSpeed | 36 |
| stb.GetAudioPID | 37 |
| stb.GetSubtitlePID | 37 |
| stb.GetPIG | 38 |
| stb.GetAlphaLevel | 38 |
| stb.GetWinAlphaLevel | 39 |
| stb.SetTransparentColor | 39 |
| stb.GetTransparentColor | 40 |

| | |
|-------------------------------------|----|
| stb.IgnoreUpdates | 40 |
| stb.ExecAction | 40 |
| stb.SetCASType..... | 41 |
| stb.SetCASParam | 41 |
| stb.SetAdditionalCasParam | 42 |
| stb.LoadCASIniFile | 43 |
| stb.SetCASDescrambling..... | 43 |
| stb.GetAspect..... | 44 |
| stb.StandBy..... | 45 |
| stb.RDir | 45 |
| stb.SetAudioLangs | 47 |
| stb.SetSubtitleLangs | 47 |
| stb.GetAudioPIDs..... | 48 |
| stb.GetSubtitlePIDs..... | 49 |
| stb.ReadCFG | 50 |
| stb.WriteCFG | 50 |
| stb.WritePrefs..... | 50 |
| stb.Debug..... | 51 |
| stb.SetListFilesExt..... | 51 |
| stb.ListDir | 52 |
| stb.SetBrightness | 53 |
| stb.SetSaturation..... | 53 |
| stb.SetContrast | 53 |
| stb.GetBrightness..... | 54 |
| stb.GetSaturation | 54 |
| stb.GetContrast..... | 54 |
| stb.DeleteAllCookies | 55 |
| stb.SetAudioOperationalMode | 55 |
| stb.SetHDMIAudioOut..... | 56 |
| stb.SetDRC | 56 |
| stb.SetStereoMode | 56 |
| stb.EnableJavaScriptInterrupt | 57 |
| stb.ShowSubtitle | 58 |
| Событийная модель в JavaScript..... | 59 |
| Настройка событийной системы | 59 |

| | |
|--|----|
| Список используемых событий | 60 |
| Приложение 1. Использование API. | 61 |
| Инициализация объекта stb..... | 61 |
| Инициализация плеера..... | 61 |
| Особенности версий JavaScript API >= 308..... | 62 |
| Инициализация плеера (Версия JavaScript API >= 308)..... | 62 |
| Wrapper.js..... | 62 |
| Инициализация событийной системы..... | 63 |
| Пример использования API. | 63 |
| Приложение 2. Форматы видео контента и примеры использования..... | 65 |
| Формат параметров функции stb.Play. | 65 |
| solution | 65 |
| URL | 67 |
| atrack, vtrack и strack..... | 68 |
| position | 68 |
| Примеры:..... | 68 |
| Приложение 3. Использование и настройка CAS..... | 70 |
| Настройка CAS Verimatrix. | 70 |
| Настройка CAS SecureMedia. | 71 |
| Установка дополнительных параметров CAS..... | 72 |
| Verimatrix. | 72 |
| SecureMedia. | 73 |
| Приложение 4. Особенности JS API при использовании браузера на базе WebKit. | 74 |
| Инициализация..... | 74 |
| Wrapper.js..... | 74 |
| Cookie..... | 75 |
| Использование альфа-прозрачности..... | 75 |
| Приложение 5. Коды кнопок пульта ДУ в JavaScript. | 77 |
| Таблица кодов кнопок для MAG100/MAG200 (ВЕРСИЯ РЕЛИЗА <= 0.1.4) | 77 |
| Таблицы кодов кнопок для MAG200 (ВЕРСИЯ РЕЛИЗА > 0.1.4)..... | 79 |
| Таблица для обработчика события onKeyPress..... | 79 |
| Таблица для обработчика событий onKeyDown и onKeyUp | 83 |
| Приложение 6. Управление индикацией на передней панели MAG200..... | 85 |
| Приложение 7. Использование кнопок на передней панели MAG200. | 86 |
| Приложение 8. Переключение режима видео выхода..... | 87 |

| | |
|--|----|
| Установка режима видео выхода..... | 87 |
| Получение текущего режима видео выхода..... | 87 |
| Приложение 9. Управление размером и позицией окна браузера на базе WebKit..... | 89 |
| Приложение 10. Установка графического разрешения браузера на базе WebKit..... | 90 |
| Установка разрешения..... | 90 |
| Получение текущего графического разрешения..... | 90 |
| Приложение 11. Работа с переменными среды..... | 92 |
| Установка и получение переменных среды..... | 92 |
| Переменные среды, используемые в стандартном ПО..... | 93 |
| История изменений документа..... | 95 |
| Версия 1.16..... | 95 |
| Версия 1.14..... | 95 |
| Версия 1.13..... | 95 |
| Версия 1.12..... | 96 |
| Версия 1.11..... | 96 |
| Версия 1.10..... | 96 |

Информация о документе.

| | |
|-----------------------|-------|
| Ревизия документа | 1.16 |
| Версия JavaScript API | 319 |
| Версия STB API | 123 |
| Версия плеера MAG200 | 0x53B |
| 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 stb | MAG100 | MAG200 |
|---------------------------------------|--------|-----------------------|
| InitPlayer | + | + |
| DeinitPlayer | + | + |
| Play | + | + |
| PlaySolution | + | + |
| Stop | + | + |
| Pause | + | + |
| Continue | + | + |
| SetPosTime | + | + |
| SetPosTimeEx | – | +(WK) |
| SetPosPercent | + | + |
| SetPosPercentEx | + | + |
| GetPosTime | + | + |
| GetPosTimeEx | – | +(WK) |
| GetPosPercent | + | + |
| GetPosPercentEx | + | + |
| GetMediaLen | + | + |
| GetMediaLenEx | – | +(WK) |
| SetSpeed | + | + |
| SetAudioPID | + | + |
| SetPIG | + | + |
| SetAlphaLevel | + | + |
| SetVolume | + | + |
| SetUserFlickerControl | + | – |
| SetFlicker | + | + (отличие от MAG100) |
| SetDefaultFlicker | + | + (отличие от MAG100) |
| SetLoop | + | + |

| Методы объекта stb | MAG100 | MAG200 |
|-------------------------------------|--------|----------------------|
| SetVideoControl | + | + |
| SetVideoState | + | + |
| SetChromaKey | + | + |
| SetMode | + | + |
| SetWinMode | + | + |
| SetTopWin | + | + |
| SetWinAlphaLevel | + | + |
| SetAspect | + | + (ДОП. ВОЗМОЖНОСТИ) |
| Rotate | + | – |
| SetMute | + | + |
| SetMicVolume | + | – |
| GetMicVolume | + | – |
| GetVolume | + | + |
| Step | + | – |
| SetupRTSP | + | + |
| SetViewport | + | + |
| IsPlaying | + | + |
| Version | + | + |
| SetupSPdif | + | + |
| SetSubtitles | + | + |
| SetSubtitlesSize | + | + |
| SetSubtitlesFont | + | + |
| SetSubtitlesOffs | + | + |
| GetSpeed | + | + |
| GetAudioPID | + | + |
| GetPIG | + | + |
| GetAlphaLevel | + | + |
| GetWinAlphaLevel | + | + |
| SetTransparentColor | + | + |
| GetTransparentColor | + | + |
| IgnoreUpdates | + | + |
| ExecAction | + | + |
| SetCASType | + | + |

| Методы объекта stb | MAG100 | MAG200 |
|---|--------|--------|
| 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) |

WK – только для WebKit.

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 **playStr**.

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 приложении 2 . |
| URL | | Адрес контента, который необходимо запустить на воспроизведение. Зависит от типа. Более детальная информация в приложении 2 . |
| 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 solution из функции stb.Play |
| URL | | Адрес контента, который необходимо запустить на воспроизведение. Зависит от типа. Более детальная информация в приложении 2 . |

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.SetPosTimeEx

void SetPosTimeEx(int time)

Установить текущую позицию воспроизведения во времени в миллисекундах.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| time | time >= 0 | Позиция в миллисекундах от начала контента, с которого необходимо начать/продолжить воспроизведение (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 |
|-----------|---------------------|--|
| pos | 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.GetPosTimeEx

Firefox: void GetPosTimeEx(out int time);

WK/FF+Wrapper: int GetPosTimeEx();

Получить текущую позицию во времени в миллисекундах.

Параметры:

Нет.

Возвращаемое значение:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| time | | Текущая позиция в миллисекундах 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.GetMediaLenEx

Firefox: void GetMediaLenEx(out int len);

WK/FF+Wrapper: int GetMediaLenEx();

Получить длительность текущего контента в миллисекундах.

Параметры:

Нет.

Возвращаемое значение:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| len | | Общая длительность текущего контента в миллисекундах. |

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 |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|-----------------------------------|
| | | -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.SetSubtitlePID

void SetSubtitlePID(int pid)

Установить номер дорожки (PID) для субтитров.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| pid | | Задаёт номер или PID дорожки субтитров, которая будет воспроизводиться в текущем контенте. В случае отсутствия такой дорожки субтитры работать не будут. |

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 scale/256 . |
| 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: 0 –пользователь API сам управляет Flicker-фильтром (см. stb.SetFlicker и stb.SetDefaultFlicker); 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 |
|-----------|---------------------|----------|
|-----------|---------------------|----------|

| 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 stb.SetVideoState 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 |
|-----------|---------------------|----------|
|-----------|---------------------|----------|

| 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 mask совпадает с данным цветом, то пиксель считается прозрачным. |
| mask | 0..0xffffffff | Задаёт маску для key . Если маска равна 0xffffffff, то прозрачным будет считаться цвет заданный параметром key . |

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 stb.SetChromaKey или stb.SetTransparentColor . |

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 stb.SetChromaKey или stb.SetTransparentColor . |

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.</p> <p>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;">aspH</td> <td colspan="4" style="text-align: center;">aspL</td> </tr> </table> <p>Для MAG100 aspH игнорируется.</p> | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | aspH | | | | aspL | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | |
| aspH | | | | aspL | | | | | | | | | | | | | | |
| aspL | 0..3 | <p>Задаёт соотношение сторон:</p> <p>0 – автоматическое;</p> <p>1 – 20:9;</p> <p>2 – 16:9;</p> <p>3 – 4:3.</p> | | | | | | | | | | | | | | | | |
| aspH | 0..3 | <p>Задаёт преобразование формата видео:</p> <p>0 – как есть, видео растягивается на весь экран;</p> <p>1 – режим Letter Box, видео пропорционально увеличивается до размеров экрана по большей границе;</p> <p>2 – режим Pan&Scan, видео пропорционально увеличивается до размеров экрана по меньшей границе;</p> <p>3 – комбинированный режим, средний между Letter Box и Pan&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 полноэкранный 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 |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|------------|
| | | 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 |
|-----------|---------------------|--|
| type | 0..6 | Тип поддерживаемого RTSP-сервера: 0 – RTSP сервер на базе VLC; 1 – BitBand RTSP сервер; 2 – Kasenna RTSP сервер; 3 – ARRIS (C-COR) RTSP сервер; 4 – Live555 RTSP сервер. 5 – ZTE RTSP сервер. 6 – Netup RTSP сервер. Сервера 3,4,5 и 6 поддерживаются только для MAG200. |
| flags | 0..0x3f | Управляющие флаги: 1 – включить режим поддержания соединения (keep-alive); |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| | | <p>2 – определение окончания потока по полю x-notice в сообщении ANNOUNCE от сервера;</p> <p>4 – определение окончания потока по полю x-notice в ответе на GET_PARAMETER;</p> <p>8 – определение окончания потока через некоторое время после отсутствия видео потока от сервера;</p> <p>16 (0x10) – определение окончания потока по полю согласно передаваемому в заголовке RTP пакета поля rtpptime. (Только для режима передачи видео над RTP);</p> <p>32 (0x20) – использовать UDP транспорт для передачи видео.</p> |

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). |

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

| Параметры | Допустимые значения | Описание |
|--|---------------------|---|
| version | | <p>Строка вида: “JS API version: <JS_API version>; STB API version: <STB_API version>; Player Engine version: <Player version>”.</p> <p>JS_API version – номер версии данного API;</p> <p>STB_API version – номер версии API плеера;</p> <p>Player version – версия плеера используемого в API, в шестнадцатеричном виде.</p> |
| 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 | <p>Режим вывода через SPdif:</p> <p>0 – звук выводится только на аналоговые выходы.</p> <p>1 – звук выводится на аналоговые 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</p> |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|-------------------|
| | | противном случае. |

Возвращаемое значение:

Нет.

stb.SetSubtitles

void SetSubtitles(bool enable);

Включить/выключить субтитры.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| enable | true, false | true – субтитры включены; false – субтитры выключены. |

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, MAG200.

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, MAG200.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| font | URL-адрес | URL адресуемый 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, MAG200.

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 |
|-----------|---------------------|--|
| speed | -8..8 | Текущая скорость воспроизведения: 1 - нормальная 2 - 2x 3 - 4x 4 - 8x 5 - 16x |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| | | 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.GetSubtitlePID

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

WK/FF+Wrapper: int GetAudioPID();

Получить получить номер (PID) текущей дорожки субтитров.

Параметры:

Нет.

Возвращаемое значение:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|----------------------------------|
| pid | 0..0x1fff | Номер текущей дорожки субтитров. |

Примечания:

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

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 blgnore);

Блокирует или разблокирует обновление экрана браузера:

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| blgnore | 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 | 0,1,2 | 0 – не задан; 1 – Verimatrix. 2 – SecureMedia. |

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. |
| 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.SetAdditionalCasParam

```
void SetAdditionalCasParam (string paramName, string paramValue);
```

Установить дополнительные параметры CAS сервера:

Платформы: MAG200.

Параметры:

| Параметры | Допустимые значения | Описание |
|------------|---------------------|---------------------|
| paramName | | Имя параметра. |
| paramValue | | Значение параметра. |

Возвращаемое значение:

Нет.

Примечания:

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

Набор дополнительных параметров зависит от конкретной CAS, текущий список параметров можно увидеть в пункте [Установка дополнительных параметров CAS](#).

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 дешифрование. 1 – использовать программное дешифрование. При отсутствии данного вызова будет использоваться аппаратное дешифрование. Для MAG100 может использоваться только программное дешифрование. |

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:

программный режим: RC4, AES;

аппаратный 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: <table border="1" style="margin: 10px auto; border-collapse: collapse;"> <tr> <td style="padding: 2px 5px;">7</td> <td style="padding: 2px 5px;">6</td> <td style="padding: 2px 5px;">5</td> <td style="padding: 2px 5px;">4</td> <td style="padding: 2px 5px;">3</td> <td style="padding: 2px 5px;">2</td> <td style="padding: 2px 5px;">1</td> <td style="padding: 2px 5px;">0</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">aspH</td> <td colspan="4" style="text-align: center; padding: 5px;">aspL</td> </tr> </table> <p>Для MAG100 aspH всегда равен 0.</p> | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | aspH | | | | aspL | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | | | | | | | | | |
| aspH | | | | aspL | | | | | | | | | | | | | | |
| 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, видео растягивается на | | | | | | | | | | | | | | | | |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| | | <p>весь экран;</p> <p>1 – режим Letter Box, видео пропорционально увеличивается до размеров экрана по большей границе;</p> <p>2 – режим Pan&Scan, видео пропорционально увеличивается до размеров экрана по меньшей границе;</p> <p>3 – комбинированный режим, средний между Letter Box и Pan&Scan.</p> <p>4 – режим увеличенный;</p> <p>5 – режим оптимальный.</p> <p>Только для MAG200</p> |

stb.StandBy

void StandBy(bool bStandby);

Войти в режим StandBy или выйти из него.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| bStandby | true, false | <p>true – перейти в режим Standby;</p> <p>false – выйти из режима Standby.</p> |

Возвращаемое значение:

Нет.

Примечания:

При входе в режим StandBy происходит следующее:

1. Выключаются все видео выходы.
2. Останавливается воспроизведение контента, если оно было.

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 |
|-----------|---------------------|---|
| par | Любая строка | Строка содержит параметры, с которыми будет запускаться скрипт <code>/home/default/rdi.cgi</code> . |

Возвращаемое значение:

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

Примечания:

В `rdir.cgi`, поставляемом 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.RDir("getenv v_name",x)` – получить значение переменной среды с именем `v_name`. Более детально работа с переменными среды описана в [приложении 11](#).

`stb.RDir("setenv v_name value")` – установить переменную среды с именем `v_name` в значение `value`. Более детально работа с переменными среды описана в [приложении 11](#).

stb.RDir("ResolveIP hostname") – получить IP адрес по имени хоста.

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 |
|-----------|--|--|
| 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.SetSubtitleLangs

```
void SetSubtitleLangs(string priLang, string secLang);
```

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

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|--|--|
| priLang | трёхсимвольные тэги ISO 639, например: "rus" или "eng" | При наличии информации о нескольких дорожках субтитров плеер будет выбирать дорожку заданную языком priLang. |
| secLang | | |

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| | | Если такая не найдена, то будет выбираться дорожка с языком secLang. Если и такой не найдено, то выбирается дорожка первая из списка. |

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: <pre> {{pid:<PID1>, lang:[<lang1_1>, <lang2_1>}}, ... , {pid:<PIDn>, lang:[<lang1_n>, <lang2_n>]}} </pre> |
| PIDn | | PID аудио дорожки с номером <i>n</i> . |
| lang1_n lang2_n | трёхсимвольные тэги ISO 639 | Первые два описания языков аудио дорожки 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 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: [[{pid:<PID1>, lang:[<lang1_1>, <lang2_1>}], ... , {pid:<PIDn>, lang:[<lang1_n>, <lang2_n>}]] |
| 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();

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 сохраняются 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

stb.Debug

```
void Debug(string debugString);
```

Вывести в поток стандартного вывода содержимое строки **debugString** в формате `DEBUG:: <время> debugString`

Параметры:

| Параметры | Допустимые значения | Описание |
|-------------|---------------------|--|
| debugString | | Данная строка выведется в поток стандартного вывода. |

Возвращаемое значение:

Нет.

stb.SetListFilesExt

```
void SetListFilesExt (string fileExts);
```

Установить список расширений файлов, которые будут возвращаться в функции .

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| fileExts | | Список расширений файлов, через пробел. Например: “.mkv .mov .mpg” |

Возвращаемое значение:

Нет.

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.ListDir

string ListDir (string dirName);

Возвращает список директорий и файлов, имеющих расширение установленное с помощью [SetListFilesExt](#), расположенных в директории **dirName**.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| dirName | | Путь к директории, содержимое которой необходимо получить. |

Возвращаемое значение:

Возвращается строка вида:

```
var dirs = [  
  "dir1/",  
  ...  
  "dirn/",  
  ""  
]  
var files = [  
  {"name" : "fileName1", "size" :size1},  
  ...  
  {"name" : "fileNamen", "size" :sizen},  
  {}  
]
```

Здесь dirn – имя n-й поддиректории,

fileNamen и sizen – имя и размер m-го файла.

Примечания:

Данная функция реализована только для браузера на базе WebKit. Для браузера на базе FireFox такую же функциональность можно получить, используя функцию [RDir](#) с параметром "rdir".

stb.SetBrightness

void SetBrightness (int bri);

Установить яркость видео выхода в режиме SD.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|----------------------------------|
| bri | 1..254 | яркость видео выхода в режиме SD |

Возвращаемое значение:

Нет.

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.SetSaturation

void SetSaturation (int sat);

Установить насыщенность видео выхода в режиме SD.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---------------------------------------|
| sat | 1..254 | насыщенность видео выхода в режиме SD |

Возвращаемое значение:

Нет.

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.SetContrast

void SetContrast (int con);

Установить контраст видео выхода в режиме SD.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|-----------------------------------|
| con | -128..127 | контраст видео выхода в режиме SD |

Возвращаемое значение:

Нет.

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.GetBrightness

int GetBrightness ();

Получить текущую яркость видео выхода в режиме SD.

Параметры:

Нет.

Возвращаемое значение:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|----------------------------------|
| bri | 1..254 | яркость видео выхода в режиме SD |

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.GetSaturation

int GetSaturation ();

Получить текущую насыщенность видео выхода в режиме SD.

Параметры:

Нет.

Возвращаемое значение:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---------------------------------------|
| sat | 1..254 | насыщенность видео выхода в режиме SD |

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.GetContrast

void GetContrast (int con);

Получить текущую контраст видео выхода в режиме SD.

Параметры:

Нет.

Возвращаемое значение:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|-----------------------------------|
| con | -128..127 | контраст видео выхода в режиме SD |

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.DeleteAllCookies

```
void DeleteAllCookies ();
```

Удалить все cookie, сохранённые браузером.

Параметры:

Нет.

Возвращаемое значение:

Нет.

Примечания:

Данная функция реализована только для браузера на базе WebKit.

stb.SetAudioOperationalMode

```
void SetAudioOperationalMode(int mode);
```

Установить Operational Mode для DolbyDigital аудио.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| mode | 0..3 | 0 – RF режим. 1 – Линейный режим. 2 – Custom0. 3 – Custom1. |

Возвращаемое значение:

Нет.

Примечания:

Влияет на воспроизведения только **DolbyDigital** аудио.

stb.SetHDMIAudioOut

void SetHDMIAudioOut(int type);

Установить какой тип аудио передаётся через HDMI выход.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| type | 0..1 | 0 – в HDMI передаётся звук в формате PCM. 1 – в HDMI передаётся звук в формате SPdif. При этом режим работы SPdif устанавливается через stb.SetupSPdif |

Возвращаемое значение:

Нет.

Примечания:

Нет.

stb.SetDRC

void SetDRC(int high,int low);

Установить сжатие динамического диапазона для DolbyDigital аудио.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| high | 0..255 | Уровень сжатия верхнего диапазона. 0 – сжатие отключено. |
| low | 0..255 | Уровень сжатия нижнего диапазона. 0 – сжатие отключено. |

Возвращаемое значение:

Нет.

Примечания:

Нет.

stb.SetStereoMode

void SetStereoMode(int mode);

Установить режим стерео звука.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| mode | 0..4 | 0 – стерео режим. 1 – моно режим. Левый и правый каналы смешиваются и результат воспроизводится в обоих аудио выходах. 2 – моно левый. Звук из левого канала идёт на оба аудио выхода. 3 – моно правый. Звук из правого канала идёт на оба аудио выхода. 4 – Lt/Rt режим |

Возвращаемое значение:

Нет.

Примечания:

Режимы моно, моно левый и моно правый работают только в случае воспроизведения Dual Mono DolbyDigital аудио.

stb.EnableJavaScriptInterrupt

void EnableJavaScriptInterrupt(bool enable);

Включить/отключить диалог прерывания JavaScript кода, когда он не выходит из обработчика длительное время.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|--|
| enable | true, false | true – включить прерывание. false – выключить прерывание. |

Возвращаемое значение:

Нет.

Примечания:

Обычно используется для отладки.

stb.ShowSubtitle

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

Отобразить строку **text** в качестве субтитров на экране.

Параметры:

| Параметры | Допустимые значения | Описание |
|-----------|---------------------|---|
| start | | Время начала отображения данной строки на экране относительно начала файла в мс. |
| end | | Время окончания отображения данной строки на экране относительно начала файла в мс. |
| text | | Текст, который необходимо отобразить как субтитры. |

Возвращаемое значение:

Нет.

Примечания:

В случае, когда **start** и **end** равны 0, строка отображается немедленно и остаётся на экране до тех пор, пока не отобразится следующая строка или не пройдёт 30 секунд.

Если была вызвана эта функция, то до следующего запуска [stb.Play](#) субтитры будут работать только через эту функцию.

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>
```

Данный скрипт можно взять в директории /home/web/ nfs-образа корневой файловой системы.

Примечания:

Важно, чтобы содержимое этого скрипта попало в глобальную область видимости.

Для браузера на базе WebKit достаточно вместо подключения данного скрипта объявить в глобальной видимости объект stbEvent :

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

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. |
| 5 | Ошибка открытия контента: нет такого контента на сервере или произошёл отказ при соединении с сервером. |
| 0x20 | Было подключено HDMI устройство. |
| 0x21 | Было отключено HDMI устройство. |
| 0x81 | При воспроизведении RTP-потока получен разрыв в нумерации RTP-пакетов. |

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] [strack:num] [position:time]”, 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. Различия: MAG100: при автоопределении нельзя давать URL начинающийся с rtp://, udp://, rtsp://, т.е. автоопределение работает только с файлами. MAG200: правильно принимает URL начинающийся с rtp://, udp://, rtsp://. |
| "" (пустой) | + | + | Аналогичен auto . |
| rtp | + | + | Воспроизвести поток в формате MPEG2TS. Если URL начинается с rtp://, то будет воспроизводиться RTP поток, если с udp://, то UDP поток. Различия: MAG100 автоматически подключает декодер для MPEG2 Video и MPEG Audio. |

| Тип (solution) | MAG100 | MAG200 | Описание. |
|----------------|--------|--------|---|
| | | | MAG200 при наличии дополнительной информации о потоке устанавливает необходимые кодеки, например H.264, AC-3 и т.д. |
| rtsp | + | + | Воспроизвести контент с RTSP-сервера. Различия: MAG100 автоматически подключает декодер для MPEG2 Video и MPEG Audio. MAG200 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. Различия: MAG100 автоматически подключает декодер для MPEG2 Video и MPEG Audio. MAG200 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. Различия: |

| Тип (solution) | MAG100 | MAG200 | Описание. |
|----------------|--------|--------|--|
| | | | <p>MAG100 автоматически подключает декодер для MPEG2 Video и MPEG Audio.</p> <p>MAG200 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. |
| ffrt | – | + | Воспроизвести MPEG-TS поток с http сервера. |
| ffrt2 | – | + | Воспроизвести потоки с http, rtmp,... но контейнер при этом может быть не MPEG-TS Зацикливание контента всегда включено (SetLoop(true)). |
| ffrt3 | – | + | Аналогично ffrt2, но предполагает наличие не реалтайм потока, то есть с возможностью позиционироваться по контенту. Подходит для воспроизведения видео с YouTube. |

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 |
|------------------------------|---|
| /path | локальный адрес файла на корневой файловой системе, например /media/1.mp3 |
| rtp://addr:port | RTP-поток принимаемый из мультикастового или уникастового адреса <i>addr</i> и портом <i>port</i> . |
| udp://addr:port | UDP-поток принимаемый из мультикастового или уникастового адреса <i>addr</i> и портом <i>port</i> . |
| rtsp://addr:port/path | контент лежащий на RTSP-сервере по адресу <i>addr</i> с портом <i>port</i> по относительному пути <i>path</i> |

atrack, vtrack и strack

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

position

Необязательный параметр, задающий время **time** в секундах, начиная с которого необходимо начать воспроизводить контент.

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.

“**rtmpeg4 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.

“rtmpeg4_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.

Настройка CAS Verimatrix.

Для использования CAS Verimatrix необходимо:

1. Установить верное время, например с ntp сервера.
2. Настроить стартовые параметры CAS сервера одним из двух способов:
 - a. через функцию [LoadCASIniFile](#), тогда параметры автоматически возьмутся из указанного файла.
 - b. через функцию [SetCASParam\(\)](#) и/или [SetAdditionalCasParam](#)
3. для MAG200 установить режим дешифрования с помощью функции [stb.SetCASDescrambling](#)
4. Настроить тип CAS сервера, после установки стартовых параметров, то есть вызвать `stb.SetCASType(1);`

Замечание. Начиная с версии ПО 0.1.66 файл rootcert.pem уже лежит в директории /flash.

Настройка CAS SecureMedia.

Для использования CAS SecureMedia необходимо:

1. Установить верное время, например с ntp сервера.
2. Настроить стартовые параметры CAS сервера с помощью функции [SetCASParam\(\)](#) и/или [SetAdditionalCasParam](#)
3. Настроить тип CAS сервера, после установки стартовых параметров, то есть вызвать `stb.SetCASType(2);`

Примечание. В случае, если не установлены дополнительные параметры через [stb.SetAdditionalCasParam](#), строка запуска `smdaemon` может быть следующей:

- `smdaemon -daemon -l 0 -auto_register`
– если не было вызова [stb.SetCASParam](#);
- `smdaemon -daemon -l 0 -auto_register –rsurl serverAddr`
– если с помощью [stb.SetCASParam](#) был установлен только адрес сервера, но не установлен порт сервера.
- `smdaemon -daemon -l 0 -auto_register –rsurl serverAddr:port`
– если с помощью [stb.SetCASParam](#) был установлен адрес и порт сервера.

Установка дополнительных параметров CAS.

С помощью функции [SetAdditionalCasParam](#) можно установить дополнительные параметры для каждой CAS.

Verimatrix.

Реализован следующий набор дополнительных параметров:

- "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"

Имена параметров CAS совпадают с именами параметров, которые можно устанавливать через конфигурационный файл (**stb.LoadCASIniFile(filename)**).

Соответственно и совпадает их действие.

Например:

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

– устанавливает время ожидания соединения с сервером в 5 секунд.

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

– устанавливает количество попыток соединения с сервером в значение 3.

По умолчанию в параметре **STOREPATH** установлено значение **"/flash"**, а в **ROOTCERT** – **"/flash/rootcert.pem"**.

Не рекомендуется менять значения этих параметров без особой необходимости, особенно при использовании стандартного образа ПО.

SecureMedia.

Для SecureMedia реализованы два дополнительных параметра:

- **"sm_add_cmd"** – значение параметра добавляется параметром в строку запуска скрипта `/home/default/smd_start.sh` после добавления параметров, переданных в `stb.SetCASParam()`.
- **"sm_full_cmd"** – значение параметра полностью подменяет параметры командной строки для запуска `smdaemon`. То есть демон запустится только с этими параметрами.

Необходимо иметь ввиду, что запуск демона происходит в блокирующем режиме, поэтому в значении параметра **"sm_full_cmd"** **должна стоять опция -daemon**.

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=/';  
  
}
```

Если при этом указать корректный срок действия cookie ("expiry="), то браузер сохранит cookie в директории /flash. То есть cookie будет действовать, до тех пор, пока не выйдет срок действия или не вызовется функция [stb.DeleteAllCookies](#).

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 * |

Таблицы кодов кнопок для MAG200 (ВЕРСИЯ РЕЛИЗА > 0.1.4)

Таблица для обработчика события onKeyPress

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

Таблица 3 Коды кнопок пульта ДУ для обработчика onKeyPress

| Кнопка ПДУ | Браузер | keyCode, dec(hex) | which, dec(hex) | Флаги |
|------------|-----------|-------------------|------------------|-------|
| 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 |

| Кнопка ПДУ | Браузер | keyCode, dec(hex) | which, dec(hex) | Флаги |
|------------|---------|-------------------|-----------------|-------|
| | 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 |

| Кнопка ПДУ | Браузер | keyCode, dec(hex) | which, dec(hex) | Флаги |
|------------|---------|-------------------|-----------------|-------|
| 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 |
| пустая | 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) | |
| Событие | FF | 0 | 112 (0x70) | ALT |

| Кнопка ПДУ | Браузер | keyCode, dec(hex) | which, dec(hex) | Флаги |
|---|---------|-------------------|-----------------|-------|
| генерируемое при подключении USB Flash Drive | WK | 112 (0x70) | 112 (0x70) | ALT |
| 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. По сравнению с предыдущей версией изменился код для кнопки “OK” для браузера на базе 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.

Таблица для обработчика событий onKeyDown и onKeyUp

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

Таблица 4 Коды кнопок пульта ДУ для обработчиков onKeyDown и onKeyUp

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

| Кнопка ПДУ | keyCode, dec(hex) | which, dec(hex) | Флаги |
|--|-------------------|-----------------|-------|
| 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 |
| пустая | 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) | |
| Событие генерируемое при подключении USB Flash Drive | 80 (0x50) | 80 (0x50) | ALT |
| Событие генерируемое при отключении USB Flash Drive | 81 (0x51) | 81 (0x51) | ALT |

Замечание 1. Обработка события `onKeyDown` гораздо проще, чем обработка события `onKeyPress`, поскольку в `onKeyDown` коды кнопок пульта не дублируются, кроме кнопок `channel+` и `channel-`.

Приложение 6. Управление индикацией на передней панели MAG200.

Для управления индикатором и светодиодом на передней панели можно воспользоваться вызовом функции [stb.ExecAction](#) следующим образом:

```
stb.ExecAction("front_panel param") ,
```

где param – строка параметров для скрипта setFpanel.sh, который выполняет вывод на переднюю панель. Параметры данного скрипта описаны в документе "Руководство_оператора_MAG200.pdf".

Приложение 7. Использование кнопок на передней панели MAG200.

Нажатия на кнопки на передней панели приводит к генерации события нажатия кнопки на клавиатуре. Для этого используются утилиты **fpchevent** для **Firefox** и **fpchevent** для **WebKit**. События нажатия транслируются согласно конфигурационным файлам `/etc/lirc/lircrc` для **Firefox** и `/etc/lirc/lircrc.wk` для **WebKit**. Более детальное описание работы этих утилит описано в документе “Руководство_оператора_MAG200.pdf”.

Приложение 8. Переключение режима видео выхода.

Установка режима видео выхода.

Переключение режима видео выхода можно осуществить с помощью вызова метода [stb.ExecAction](#) в следующем виде:

```
stb.ExecAction("tvsystem mode"),
```

где mode может принимать следующие значения:

PAL

576p-50

720p-50

1080i-50

NTSC

576p-60

720p-60

1080i-60

Например stb.ExecAction("tvsystem PAL") установит видео выход в режим PAL(576i).

Замечание. Изменения вступят в силу после перезапуска приставки.

Получение текущего режима видео выхода

Для получения текущего режима видео выхода можно воспользоваться функцией [stb.RDir](#) в виде:

```
var mode = stb.RDir('vmode')
```

При этом **mode** может принять следующие значения:

576i – режим PAL

576p – режим 576p

720p – режим 720p

1080i – режим 1080i

Замечание. В данном случае будет возвращаться текущий работающий режим, то есть поменяться он может только после перезагрузки.

Приложение 9. Управление размером и позицией окна браузера на базе WebKit.

При необходимости можно уменьшить размер окна браузера и изменить его позицию на экране. Для этого можно воспользоваться вызовом следующих функций:

window.moveTo(x, y) – сместить окно в позицию с координатами **x** и **y**.

window.resizeTo(width, height) – установить ширину окна в значение **width**, а высоту в значение **height**.

Приложение 10. Установка графического разрешения браузера на базе WebKit.

Установка разрешения

Установить графическое разрешение можно с помощью функции [stb.ExecAction](#) следующим образом:

```
stb.ExecAction('graphicsres mode'),
```

где **mode** может принимать следующие значения:

tvsystem_res – графическое разрешение совпадает с разрешением видео выхода (отображение 1:1)

720 – графическое разрешение 720x576, при этом выполняется аппаратное масштабирование данного разрешения на весь экран в режимах 1080i и 720p.

1280 – графическое разрешение 1280x720, при этом выполняется аппаратное масштабирование данного разрешения на весь экран в режиме 1080i.

1920 – графическое разрешение 1920x1080.

Замечание. Если разрешение видео выхода меньше, чем выставленное графическое разрешение, то графическое разрешение будет считаться равным разрешению видео выхода.

Замечание. Изменения вступят в силу после перезапуска приставки.

Получение текущего графического разрешения

Получить текущее графическое разрешение можно с помощью функции [stb.RDir](#) следующим образом:

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

при этом **gres** будет принимать значения: **tvsystem_res, 720, 1280, 1920** как описано в предыдущем пункте.

Также графическое разрешение можно получить с помощью **screen.width** и **screen.height**:

`screen.width` – горизонтальное разрешение.

`screen.height` – вертикальное разрешение.

Замечание. В данном случае будет возвращаться текущий работающий режим, то есть поменяться он может только после перезагрузки.

Приложение 11. Работа с переменными среды.

Javascript API для приставки MAG 200 позволяет получать и устанавливать специальные переменные среды, которые сохраняются в ПЗУ. Далее для них будет использоваться обозначение «переменные среды».

Внимание. Переменные среды сохраняются в ПЗУ, которое имеет большое, но ограниченное количество перезаписей, поэтому настоятельно не рекомендуется сохранять параметры, которые меняются часто, например, при каждом запуске устройства.

Установка и получение переменных среды.

Для получения переменной среды необходимо воспользоваться функцией RDir с параметром `getenv` (см. [stb.RDir](#))

Для установки переменной среды необходимо воспользоваться функцией RDir с параметром `setenv` (см. [stb.RDir](#))

При необходимости установки нескольких переменных рекомендуется установить все эти переменные одним вызовом. Для этого необходимо между парами имя-значение вставлять строку «`|`» (3 символа), то есть вызывать RDir в следующем виде:

```
stb.RDir('setenv name_1 val_1 "|" name_2 val_2 "|" ... "|" name_n val_n'),
```

где `n` – количество переменных, `name_n` – имя переменной под номером `n`, `val_n` – значение, которое необходимо установить переменной с именем `name_n`.

Например:

```
stb.RDir('setenv mcip_conf 224.50.7.50 "|" mcip_img_conf 111.1.2.3 "|" portal2  
http://some\_portal.com')
```

установит переменные `mcip_conf`, `mcip_img_conf` и `portal2` в значения 224.50.7.50, 111.1.2.3 и http://some_portal.com соответственно.

Ниже приведён пример Javascript кода, для такого вызова:

```

.....
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);
.....
// Пакетная установка переменных окружения в ПЗУ
function batchSetEnvValues(str){
    stb.RDir ('setenv ' + str);
}
.....

```

Переменные среды, используемые в стандартном ПО.

Для обеспечения правильной работы стандартного ПО используются следующие переменные среды:

ipaddr_conf - статический IP адрес. Если эта переменная не установлена, то устройство при запуске будет получать IP адрес, маску подсети, шлюз, DNS и NTP автоматически (через DHCP).

netmask – маска подсети.

gatewayip – IP адрес шлюза по умолчанию.

dnsip – IP адрес DNS сервера.

ntpurl – URL NTP сервера.

mcip_conf – мультикаст адрес, из которого будет приниматься bootstrap.

mcport_conf – номер порта, из которого будет приниматься bootstrap.

mcip_img_conf – мультикаст адрес, из которого будет приниматься образ для обновления (imageupdate).

mcport_img_conf – номер порта, из которого будет приниматься образ для обновления (imageupdate).

mcip_mng_conf – мультикаст адрес канала управления.

mcport_mng_conf – номер порта канала управления.

portal1 – URL портала 1.

portal2 – URL портала 2.

volume – уровень громкости по умолчанию.

language – индекс языка интерфейса пользователя. 0 – английский, 1 – русский.

upnp_conf – запускать(true) или не запускать(false) UPnP клиент.

use_portal_dhcp – использовать(true) или не использовать(false) значение переменной

portal_dhcp в качестве стартового портала, если не установлены переменные

portal1 и portal2

portal_dhcp – URL портала, который устанавливается оператором с использованием протокола DHCP.

История изменений документа.

Версия 1.16

Добавлено описание команды ResolveIP в [stb.RDir](#).

Добавлены типы RTSP серверов в описании [stb.SetupRTSP](#).

Исправлено описание [stb.SetLoop](#).

Добавлены функции:

[stb.SetAudioOperationalMode](#),

[stb.SetHDMIAudioOut](#),

[stb.SetDRC](#),

[stb.SetStereoMode](#),

[stb.EnableJavaScriptInterrupt](#),

[stb.ShowSubtitle](#),

[stb.SetAdditionalCasParam](#).

Расширено описание в пункте [Приложение 3. Использование и настройка CAS](#).

Добавлены новые события в [Список используемых событий](#).

Расширилось описание в пункте [solution](#).

Версия 1.14

Добавлены функции:

[stb.SetPosTimeEx](#),

[stb.GetPosTimeEx](#),

[stb.GetMediaLenEx](#).

В таблице [Доступность функций на IPTV-приставках](#) обновилась информация о функциях [stb.SetSubtitlesSize](#), [stb.SetSubtitlesFont](#) и [stb.SetSubtitlesOffs](#).

Добавлено описание новых параметров для функции [stb.RDir](#).

Добавлено [Приложение 11. Работа с переменными среды](#).

Версия 1.13

Добавлена функция [stb.DeleteAllCookies](#).

Обновлена информация в разделе [Cookie](#)

Версия 1.12

Обновлено описание функции [stb.GetAspect](#).

Исправлено описание функции [stb.SetCASDescrambling](#).

Версия 1.11

Добавлены:

[Приложение 10. Установка графического разрешения браузера на базе WebKit.](#)

[Получение текущего режима видео выхода](#)

Версия 1.10

Добавлены функции:

[SetSubtitleLangs](#),

[GetSubtitlePID](#),

[SetSubtitlePID](#),

[SetBrightness](#),

[SetSaturation](#),

[SetContrast](#),

[GetBrightness](#),

[GetSaturation](#),

[GetContrast](#).

Добавлена информация по настройке событийной системы в пункте [Настройка событийной системы](#).

Добавлена информация о Standby режиме см. [StandBy](#).

Добавлено [Приложение 9. Управление размером и позицией окна браузера на базе WebKit](#).

Исправлена ошибка в описании функции [stb.SetViewport](#).