

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

V. 1.19

2011

Содержание.

Содержание.....	2
Информация о документе.	9
Введение.	10
Использование API.....	10
Типы данных и формат аргументов.	10
Вызов методов объекта stb.....	11
Доступность функций на IPTV-приставках.....	12
Методы объекта stb.	16
stb.InitPlayer	16
stb.DeinitPlayer	16
stb.Play	16
stb.Play с использованием прокси сервера	17
stb.PlaySolution	18
stb.Stop.....	18
stb.Pause	18
stb.Continue	19
stb.SetPosTime.....	19
stb.SetPosTimeEx.....	19
stb.SetPosPercent	20
stb.SetPosPercentEx	20
stb.GetPosTime	21
stb.GetPosTimeEx	21
stb.GetPosPercent	21
stb.GetPosPercentEx.....	22
stb.GetMediaLen.....	22
stb.GetMediaLenEx.....	22
stb.SetSpeed.....	23
stb.SetAudioPID.....	23
stb.SetSubtitlePID	24
stb.SetPIG.....	24
stb.SetAlphaLevel	25

stb.SetVolume	25
stb.SetUserFlickerControl	26
stb.SetFlicker	26
stb.SetDefaultFlicker	27
stb.SetLoop	27
stb.SetVideoControl	28
stb.SetVideoState	28
stb.SetChromaKey	29
stb.SetMode	29
stb.SetWinMode	30
stb.SetTopWin	30
stb.SetWinAlphaLevel	31
stb.SetAspect	31
stb.Rotate	32
stb.SetMute	33
stb.SetMicVolume	33
stb.GetMicVolume	34
stb.GetVolume	34
stb.GetVolume	34
stb.GetVolume	35
stb.GetMute	35
stb.Step	35
stb.SetupRTSP	36
stb.SetViewport	37
stb.IsPlaying	38
stb.Version	38
stb.SetupSPdif	39
stb.SetSubtitles	39
stb.SetSubtitlesSize	40
stb.SetSubtitlesFont	40
stb.SetSubtitlesOffs	41
stb.GetSpeed	41
stb.GetAudioPID	42
stb.GetSubtitlePID	42
stb.GetPIG	42

stb.GetAlphaLevel.....	43
stb.GetWinAlphaLevel.....	43
stb.SetTransparentColor.....	44
stb.GetTransparentColor.....	44
stb.IgnoreUpdates.....	45
stb.ExecAction.....	45
stb.SetCASType.....	46
stb.SetCASParam.....	46
stb.SetAdditionalCasParam.....	47
stb.LoadCASIniFile.....	47
stb.SetCASDescrambling.....	48
stb.GetAspect.....	49
stb.StandBy.....	50
stb.Rdir.....	50
stb.SetAudioLangs.....	51
stb.SetSubtitleLangs.....	52
stb.GetAudioPIDs.....	53
stb.GetSubtitlePIDs.....	53
stb.ReadCFG.....	54
stb.WriteCFG.....	55
stb.WritePrefs.....	55
stb.Debug.....	56
stb.SetListFilesExt.....	56
stb.ListDir.....	56
stb.SetBrightness.....	57
stb.SetSaturation.....	58
stb.SetContrast.....	58
stb.GetBrightness.....	58
stb.GetSaturation.....	59
stb.GetContrast.....	59
stb.DeleteAllCookies.....	59
stb.SetAudioOperationalMode.....	60
stb.SetHDMIAudioOut.....	60
stb.SetDRC.....	61
stb.SetStereoMode.....	61

stb.EnableJavaScriptInterrupt	62
stb.ShowSubtitle	62
stb.StartLocalCfg.....	63
stb.ShowVirtualKeyboard	63
stb.HideVirtualKeyboard	64
stb.EnableServiceButton	64
stb.EnableVKButton	64
stb.EnableSpatialNavigation	65
stb.EnableSetCookieFrom	66
stb.SetBufferSize	66
stb.GetBufferLoad	67
stb.SetWebProxy	67
stb.GetVideoInfo	68
Событийная модель в JavaScript.....	70
Настройка событийной системы	70
Список используемых событий	71
Приложение 1. Использование API.	72
Инициализация объекта stb.....	72
Инициализация плеера.....	72
Особенности версий JavaScript API >= 308.....	73
Инициализация плеера (Версия JavaScript API >= 308).....	73
Wrapper.js.....	73
Инициализация событийной системы.....	74
Пример использования API.	74
Приложение 2. Форматы видео контента и примеры использования.....	76
Формат параметров функции stb.Play.	76
solution	76
URL	78
atrack, vtrack и strack.....	79
position	79
Примеры:.....	79
Приложение 3. Использование и настройка CAS.....	81
Настройка CAS Verimatrix.....	81
Настройка CAS SecureMedia.....	82
Настройка внешнего CAS плагина.....	83

Установка дополнительных параметров CAS.....	84
Verimatrix.....	84
SecureMedia.....	85
Приложение 4. Особенности JS API при использовании браузера на базе WebKit.....	86
Инициализация.....	86
Wrapper.js.....	86
Cookie.....	87
Использование альфа-прозрачности.....	87
Приложение 5. Коды кнопок пульта ДУ в JavaScript.....	89
Таблица кодов кнопок для MAG100/MAG200 (ВЕРСИЯ РЕЛИЗА <= 0.1.4).....	89
Таблицы кодов кнопок для MAG200 (ВЕРСИЯ РЕЛИЗА > 0.1.4).....	91
Таблица для обработчика события onKeyPress.....	91
Таблица для обработчика событий onKeyDown и onKeyUp.....	95
Приложение 6. Управление индикацией на передней панели MAG200.....	97
Приложение 7. Использование кнопок на передней панели MAG200.....	98
Приложение 8. Переключение режима видео выхода.....	99
Установка режима видео выхода.....	99
Получение текущего режима видео выхода.....	99
Приложение 9. Управление размером и позицией окна браузера на базе WebKit.....	101
Приложение 10. Установка графического разрешения браузера на базе WebKit.....	102
Установка разрешения.....	102
Получение текущего графического разрешения.....	102
Приложение 11. Работа с переменными среды.....	104
Установка и получение переменных среды.....	104
Переменные среды, используемые в стандартном ПО.....	105
Приложение 12. JavaScript API для подсистемы обновления ПО.....	107
Варианты использования.....	107
Общий сценарий работы с объектом «stbUpdate».....	107
Обновление ПО.....	109
Автоматическое обновление ПО.....	109
Методы объекта «stbUpdate».....	110
stbUpdate.getStatusStr.....	110
stbUpdate.getStatus.....	110
stbUpdate.getPercents.....	111
stbUpdate.getActiveBank.....	111

stbUpdate.startCheck	112
stbUpdate.getImageVersionStr.....	112
stbUpdate.getImageDateStr	112
stbUpdate.getImageDescStr.....	113
stbUpdate.startUpdate.....	113
stbUpdate.startAutoUpdate	114
Пример использования API	114
Приложение 13. JavaScript API для подсистемы PVR.....	117
Описание объекта pvrManager	117
Таблица кодов ошибок	117
Таблица состояний задач	118
pvrManager.CreateRecordingTask.....	118
pvrManager.GetAllTasks	119
pvrManager.GetTasksByIDs.....	119
pvrManager.GetTaskByID	120
pvrManager.RemoveTask	121
pvrManager.ChangeEndTime.....	121
pvrManager.SetMaxRecordingCnt	121
Приложение 14. JavaScript API для управления загрузками.....	123
Методы объекта « stbDownloadManager ».....	123
DeleteJob	123
StartJob	123
StopJob	124
AdjustJobPriority.....	124
PlayDownloadedMedia	125
AddJob	125
GetQueueInfo	126
История изменений документа.	127
Версия 1.19.....	127
Версия 1.18.....	127
Версия 1.17.....	127
Версия 1.16.....	128
Версия 1.14.....	128
Версия 1.13.....	128
Версия 1.12.....	128

Версия 1.11 129
Версия 1.10 129

Информация о документе.

Ревизия документа	1.19
Версия JavaScript API	322
Версия STB API	126
Версия плеера MAG200	0x549
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	+	+
stb.Play с использованием прокси сервера	-	+(WK)
PlaySolution	+	+
Stop	+	+
Pause	+	+
Continue	+	+
SetPosTime	+	+
SetPosTimeEx	-	+(WK)
SetPosPercent	+	+
SetPosPercentEx	+	+
GetPosTime	+	+
GetPosTimeEx	-	+(WK)
GetPosPercent	+	+
GetPosPercentEx	+	+
GetMediaLen	+	+
GetMediaLenEx	-	+(WK)
SetSpeed	+	+
SetAudioPID	+	+
SetPIG	+	+
SetAlphaLevel	+	+
SetVolume	+	+
SetUserFlickerControl	+	-

Методы объекта stb	MAG100	MAG200
SetFlicker	+	+ (отличие от MAG100)
SetDefaultFlicker	+	+ (отличие от MAG100)
SetLoop	+	+
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	+	+

Методы объекта stb	MAG100	MAG200
IgnoreUpdates	+	+
ExecAction	+	+
SetCASType	+	+
SetCASParam	+	+
SetAdditionalCasParam	-	+
LoadCASIniFile	+	+
SetCASDescrambling	-	+
GetAspect	+	+
StandBy	+	+
RDir	+	+
SetAudioLangs	+	+
GetAudioPIDs	+	+
GetSubtitlePIDs	+	+
ReadCFG	+	+
WriteCFG	+	+
WritePrefs	+	+
Debug	+	+
SetListFilesExt	-	+(WK)
ListDir	-	+(WK)
SetBrightness	-	+(WK)
SetSaturation	-	+(WK)
SetContrast	-	+(WK)
GetBrightness	-	+(WK)
GetSaturation	-	+(WK)
GetContrast	-	+(WK)
GetSubtitlePID	-	+(WK)
SetSubtitlePID	-	+(WK)
SetSubtitleLangs	-	+(WK)
DeleteAllCookies	-	+(WK)
SetAudioOperationalMode	-	+(WK)
SetHDMIAudioOut	-	+(WK)
SetDRC	-	+(WK)
SetStereoMode	-	+(WK)

Методы объекта stb	MAG100	MAG200
EnableJavaScriptInterrupt	-	+(WK)
ShowSubtitle	-	+(WK)
GetMute	-	+(WK)
StartLocalCfg	-	+(WK)
ShowVirtualKeyboard	-	+(WK)
HideVirtualKeyboard	-	+(WK)
EnableServiceButton	-	+(WK)
EnableVKButton	-	+(WK)
EnableSpatialNavigation	-	+(WK)
EnableSetCookieFrom	-	+(WK)
SetBufferSize	-	+(WK)
GetBufferLoad	-	+(WK)
GetVideoInfo	-	+(WK)
SetWebProxy	-	+(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.Play с использованием прокси сервера

void Play(string playStr, string proxy_params)

Запустить воспроизведение медиа контента, как указано в **playStr**, используя заданный прокси сервер для доступа к http контенту.

Параметры:

Параметры	Допустимые значения	Описание
playStr	строка	См stb.Play .
proxy_params	строка вида: 'http://[username[:password]@]proxy_addr:proxy_port' proxy_addr – адрес прокси сервера. proxy_port – порт прокси сервера. username – имя пользователя для авторизации на прокси сервере. password – пароль для авторизации на прокси сервере. Параметры указанные в квадратных скобках являются необязательными.	

Возвращаемое значение:

Нет.

Замечание. Прокси сервер устанавливается только до следующего вызова `stb.Play()`.

Действие данного прокси сервера распространяется только на воспроизведение `http` контента.

stb.PlaySolution

`void PlaySolution(string solution, string URL)`

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

Параметры:

Параметры	Допустимые значения	Описание
<code>solution</code>		Соответствует параметру solution из функции stb.Play
<code>URL</code>		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 (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
prc	0..10000	Позиция в сотых долях процента от общей длительности контента, с которой необходимо начать/продолжить воспроизведение.

Возвращаемое значение:

Нет.

stb.GetPosTime

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

WK/FF+Wrapper: int GetPosTime();

Получить текущую позицию во времени.

Параметры:

Нет.

Возвращаемое значение:

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

stb.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
-----------	---------------------	----------

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 -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 будет.

Возвращаемое значение:

Нет.

stb.SetPIG

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

Установить расположение и режим видео окна.

Параметры:

Параметры	Допустимые значения	Описание
state	0..1	Если state=1 – показывать видео на полный экран. Если state=0 – показывать видео в заданном прямоугольнике.
scale		Масштаб видео окна. Настоящий

Параметры	Допустимые значения	Описание
		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
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
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

Параметры	Допустимые значения	Описание
		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;

Параметры	Допустимые значения	Описание
		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	Задаёт соотношение сторон:																

Параметры	Допустимые значения	Описание
		0 – автоматическое; 1 – 20:9; 2 – 16:9; 3 – 4:3.
aspH	0..3	Задаёт преобразование формата видео: 0 – как есть, видео растягивается на весь экран; 1 – режим Letter Box, видео пропорционально увеличивается до размеров экрана по большей границе; 2 – режим Pan&Scan, видео пропорционально увеличивается до размеров экрана по меньшей границе; 3 – комбинированный режим, средний между Letter Box и Pan&Scan. 4 – режим увеличенный; 5 – режим оптимальный. Только для MAG200

Возвращаемое значение:

Нет.

Примечания:

Для MAG100 **aspH** игнорируется.

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

stb.Rotate

void Rotate(int angle)

Повернуть видео изображение.

Платформы: MAG100

Параметры:

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

Возвращаемое значение:

Нет.

stb.SetMute

void SetMute(int mute)

Выключить или включить звук с восстановлением уровня громкости.

Параметры:

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

Возвращаемое значение:

Нет.

stb.SetMicVolume

void SetMicVolume(int micvol)

Установить уровень громкость микрофона.

Платформы: MAG100

Параметры:

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

Параметры	Допустимые значения	Описание
		100 – максимальная громкость.

Возвращаемое значение:

Нет.

stb.GetMicVolume

Firefox: void GetMicVolume(out int micvol);

WK/FF+Wrapper: int GetMicVolume();

Получить текущий уровень громкости микрофона.

Платформы: MAG100

Параметры:

Нет.

Возвращаемое значение:

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

stb.GetVolume

Firefox: void GetVolume(out int vol);

WK/FF+Wrapper: int GetVolume();

Получить уровень громкости.

Параметры:

Нет.

Возвращаемое значение:

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

stb.GetVolume

Firefox: void GetVolume(out int vol);

WK/FF+Wrapper: int GetVolume();

Получить уровень громкости.

Параметры:

Нет.

Возвращаемое значение:

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

stb.GetVolume

Firefox: void GetVolume(out int vol);

WK/FF+Wrapper: int GetVolume();

Получить уровень громкости.

Параметры:

Нет.

Возвращаемое значение:

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

stb.GetMute

WK/FF+Wrapper: int GetMute();

Получить состояние громкости (выкл/вкл).

Параметры:

Нет.

Возвращаемое значение:

Параметры	Допустимые значения	Описание
mute	0..1	0 – громкость включена. 1 – громкость выключена.

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); 2 – определение окончания потока по полю x-notice в сообщении ANNOUNCE от сервера; 4 – определение окончания потока по полю x-notice в ответе на

Параметры	Допустимые значения	Описание
		<p>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 rtp_time. (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).
x	В сумме с xsize не должно превышать ширину экрана.	Смещение левого верхнего угла видео окна по горизонтали от начала экрана.
y	В сумме с ysize не должно превышать ширину экрана.	Смещение левого верхнего угла видео окна по вертикали от начала экрана.

Возвращаемое значение:

Нет.

stb.IsPlaying

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

WK/FF+Wrapper: bool IsPlaying()

Получить текущее состояние воспроизведения:

Параметры:

Нет.

Возвращаемое значение:

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

stb.Version

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

WK/FF+Wrapper: string Version();

Получить версию API.

Параметры:

Нет.

Возвращаемое значение:

Параметры	Допустимые значения	Описание
version		Строка вида: “JS API version: <JS_API version>; STB API version: <STB_API version>; Player Engine version: <Player version>”. JS_API version – номер версии данного API;

Параметры	Допустимые значения	Описание
		STB_API version – номер версии API плеера; Player version – версия плеера используемого в API, в шестнадцатеричном виде.
Пример: “JS API version: 301; STB API version: 104; Player Engine version: 0x23”		

stb.SetupSPdif

```
void SetupSPdif(int flags);
```

Установить режим вывода звука через Spdif

Параметры:

Параметры	Допустимые значения	Описание
flags	0..2	Режим вывода через Spdif: 0 – звук выводится только на аналоговые выходы. 1 – звук выводится на аналоговые выходы и через Spdif в формате 2-канального PCM 2 – звук выводится в Spdif не декодированным (AC3 ...), если кодек поддерживает, или через Spdif в формате 2-канального PCM в 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 адресующий в корневой файловой системе файл шрифтов. Например: “/home/default/arial.ttf”

Возвращаемое значение:

Нет.

stb.SetSubtitlesOffs

void SetSubtitlesOffs(int offs);

Задать смещение для отображения текстовых субтитров.

Платформы: MAG100, 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 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

Параметры	Допустимые значения	Описание
		применяется данная функция: 0 – графическое окно; 1 – видео окно.

Возвращаемое значение:

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

stb.SetTransparentColor

void SetTransparentColor(int color);

Устанавливает цвет, который считается прозрачным в данный момент:

Параметры:

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

Возвращаемое значение:

Нет.

Примечания:

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

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

stb.GetTransparentColor

FireFox: void GetTransparentColor(out int color);

WK/FF+Wrapper: int GetTransparentColor();

Возвращает цвет, который считается прозрачным в данный момент:

Параметры:

Нет.

Возвращаемое значение:

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

stb.IgnoreUpdates

void IgnoreUpdates(bool bIgnore);

Блокирует или разблокирует обновление экрана браузера:

Параметры:

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

Возвращаемое значение:

Нет.

stb.ExecAction

void ExecAction(string action);

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

Параметры:

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

Возвращаемое значение:

Нет.

stb.SetCASType

void SetCASType(int CAS_type);

Установить тип сервера условного доступа:

Платформы: MAG100,MAG200

Параметры:

Параметры	Допустимые значения	Описание
type	0,1,2,4,5,6,7,8,9,10	0 – не задан; 1 – Verimatrix. 2 – SecureMedia, 4-10 – внешний CAS плагин с соответствующим номером.

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

Параметры	Допустимые значения	Описание
		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 зашифрованные 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" data-bbox="916 931 1350 1043"><tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="4">aspH</td><td colspan="4">aspL</td></tr></table> Для MAG100 aspH всегда равен 0.	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 весь экран; 1 – режим Letter Box, видео пропорционально увеличивается до размеров экрана по большей границе; 2 – режим Pan&Scan, видео пропорционально увеличивается до размеров экрана по меньшей																

Параметры	Допустимые значения	Описание
		границе; 3 – комбинированный режим, средний между Letter Box и Pan&Scan. 4 – режим увеличенный; 5 – режим оптимальный. Только для MAG200

stb.StandBy

void StandBy(bool bStandby);

Войти в режим StandBy или выйти из него.

Параметры:

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

Возвращаемое значение:

Нет.

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 /home/default/rdi.cgi .

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. Если такая не найдена, то будет выбираться дорожка с языком secLang. Если и такой не найдено, то выбирается дорожка первая из списка.
secLang		

Возвращаемое значение:

Нет.

stb.GetAudioPIDs

Firefox: void GetAudioPIDs(out string pidsList);

WK/FF+Wrapper: string GetAudioPIDs();

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

Параметры:

Нет.

Возвращаемое значение:

Параметры	Допустимые значения	Описание
pidsList		Список обнаруженных аудио дорожек в следующем формате: [{pid:<PID1>, lang:[<lang1_1>, <lang2_1>]}], ... , {pid:<PIDn>, lang:[<lang1_n>, <lang2_n>]}]
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: <pre>[{pid:<PID1>, lang:[<lang1_1>, <lang2_1>]}, ... , {pid:<PIDn>, lang:[<lang1_n>, <lang2_n>]}]</pre>
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 portalу. 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
-----------	---------------------	----------

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
-----------	---------------------	----------

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

Параметры	Допустимые значения	Описание
		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, string 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

Параметры	Допустимые значения	Описание
		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.

stb.StartLocalCfg

void StartLocalCfg();

Запустить меню настроек приставки (Service Menu).

Параметры:

Нет.

Возвращаемое значение:

Нет.

Примечания:

Действие функции аналогично нажатию кнопки "SET"("service" на старом ПДУ), если с помощью функции [stb.EnableServiceButton](#) разрешён автоматический вызов меню конфигурации.

stb.ShowVirtualKeyboard

void ShowVirtualKeyboard();

Показать виртуальную клавиатуру на экране.

Параметры:

Нет.

Возвращаемое значение:

Нет.

Примечания:

В виртуальной клавиатуре можно переключаться с ввода английских символов на ввод символов того языка, который выбран в качестве локализации в меню настроек.

stb.HideVirtualKeyboard

```
void HideVirtualKeyboard();
```

Убрать виртуальную клавиатуру с экрана.

Параметры:

Нет.

Возвращаемое значение:

Нет.

stb.EnableServiceButton

```
void EnableServiceButton(bool bEnable);
```

Включить/выключить автоматический вызов (независимо от JavaScript кода) меню настроек по нажатию на кнопку "SET"("service" на старом ПДУ).

Параметры:

Параметры	Допустимые значения	Описание
bEnable	true, false	false – автоматический вызов выключен. true – автоматический вызов включен.

Возвращаемое значение:

Нет.

Примечания:

Если заданные кнопки уже используются в JavaScript коде, то для избежания конфликтов можно выключить автоматический вызов и вызывать меню настроек когда необходимо из JavaScript кода с помощью функции [stb.StartLocalCfg](#).

stb.EnableVKButton

```
void EnableVKButton(bool bEnable);
```


Включить/выключить автоматический вызов (независимо от JavaScript кода) виртуальной клавиатуры по нажатию на кнопку "КВ"("пустая" на старом ПДУ).

Параметры:

Параметры	Допустимые значения	Описание
bEnable	true, false	false – автоматический вызов выключен. true – автоматический вызов включен.

Возвращаемое значение:

Нет.

Примечания:

Если заданные кнопки уже используются в JavaScript коде, то во избежание конфликтов можно выключить автоматический вызов и вызывать виртуальную клавиатуру, когда необходимо, из JavaScript кода с помощью функции [stb.ShowVirtualKeyboard](#).

stb.EnableSpatialNavigation

void EnableSpatialNavigation(bool bEnable);

Включить/выключить 2D навигацию (навигация стрелками "влево", "вправо", "вверх", "вниз") по web странице.

Параметры:

Параметры	Допустимые значения	Описание
bEnable	true, false	false – 2D навигация выключена. true – 2D навигация включена.

Возвращаемое значение:

Нет.

Примечания:

2D навигация по умолчанию выключена, однако она может быть включена на предыдущей странице, если таковые были, поэтому рекомендуется явным образом устанавливать режим 2D навигации при старте страницы, если она не должна использоваться.

stb.EnableSetCookieFrom

void EnableSetCookieFrom(string domain, bool bEnable);

Разрешить/запретить установку cookie с заданного домена.

Параметры:

Параметры	Допустимые значения	Описание
domain	URL-адрес домена	Если bEnable равно true, то любые попытки установить cookie с этого домена будут игнорироваться.
bEnable	true, false	false –запретить установку cookie с заданного домена. true – разрешить установку cookie с заданного домена.

Возвращаемое значение:

Нет.

Примечания:

По умолчанию для всех доменов разрешена установка cookie.

Каждый новый вызов функции добавляет домен (bEnable==false) или удаляет его (bEnable==true) из списка доменов, на которых запрещена установка cookie.

stb.SetBufferSize

void SetBufferSize(int sizeInMs, int maxSizeInBytes);

Установить размер буфера для буферизации входного потока.

Параметры:

Параметры	Допустимые значения	Описание
sizeInMs	int	Размер буфера в миллисекундах.
maxSizeInBytes	int	Максимальный размер буфера в байтах. Служит только для ограничения максимального объема выделяемой памяти. Механизм буферизации в первую очередь ориентируется на значение sizeInMs.

Возвращаемое значение:

Нет.

Примечания:

Буферизация работает только на следующих солюшенах:

ffmpeg, ffmt, ffmt2, ffmt3, fm, file.

При запуске любого из перечисленных солюшенов на воспроизведение происходят следующие действия:

- открытие контента на воспроизведение;
- определение информации о контенте (приходит событие 2 из [Список используемых событий](#));
- начинает наполняться входной буфер, пока буфер не наполнится полностью (см. sizeInMs);
- после полного наполнения данные начинают декодироваться;
- декодированные данные начинают поступать на выходы (приходит событие 4 из [Список используемых событий](#)).

stb.GetBufferLoad

```
int GetBufferLoad();
```

Получить текущую загрузку входного буфера в процентах.

Параметры:

Нет.

Возвращаемое значение:

Параметры	Допустимые значения	Описание
	0..100	Текущая загрузка входного буфера в процентах.

Примечания:

Имеет смысл вызывать только на солюшенах ffmpeg, ffmt, ffmt2, ffmt3, fm, file после получения события 2 из [Список используемых событий](#) и до наполнения буфера на 100% или прихода события 4 из [Список используемых событий](#).

stb.SetWebProxy

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

Параметры:

Параметры	Допустимые значения	Описание
proxy_addr	string	Адрес прокси сервера.
proxy_port	int	Порт прокси сервера.
user_name	string	Имя пользователя для авторизации на прокси сервере. Может быть пустым.
passwd	string	Пароль для авторизации на прокси сервере. Может быть пустым.
exlude_list	string	Список сайтов-исключений, разделённых пробелами, на которые запросы будут проходить минуя прокси сервер. Например: 'youtube.com .lenta.ru 192.168.1.1/24 192.168.1.*' Может быть пустым.

Возвращаемое значение:

Нет.

Замечание. Данные параметры действительны только для http:// или https:// запросов браузера, но не касаются воспроизведения контента с http сервера.

Для этого необходимо использовать расширенную функцию [stb.Play с использованием прокси сервера](#).

stb.GetVideoInfo

```
string GetVideoInfo();
```

Получить информацию о видео контенте.

Параметры:

Нет.

Возвращаемое значение:

Возвращает строку вида:

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

где frameRate – частота кадров.

pictureWidth – горизонтальный размер видео контента (исходного).

pictureHeight – вертикальный размер видео контента (исходного).

hPAR и vPAR – коэффициенты горизонтальной и вертикальной пропорции пиксела.

В данном примере эти параметры означают, что соотношение сторон контента следующее:

$$(704 * hPAR / vPAR) / 576 = 1.333333333(3) = 4:3 \text{ для квадратных пикселей.}$$

Примечания:

Имеет смысл вызывать только после получения события 7 из [Список используемых событий](#).

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.
6	Началось воспроизведение DualMono AC-3 звука.
7	Получена информация о видео контенте. См. stb.GetVideoInfo
0x20	Было подключено HDMI устройство.
0x21	Было отключено HDMI устройство.
0x22	Завершилась задача записи. См. Приложение 13. JavaScript API для подсистемы PVR.
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.

Настройка внешнего CAS плагина.

Инструкцию по интерфейсу внешнего CAS плагина можно найти по ссылке <http://soft.infomir.com.ua/mag200/CAS/Readme.txt>. Там описано, как написать свой CAS плагин

Для использования внешнего CAS плагина необходимо положить динамическую библиотеку плагина в директорию /home/default корневой файловой системы приставки с именем

/home/default/libCasCustom_x.so,

где "_x" соответствует типу CAS (см. [stb.SetCASType](#))

Например при вызове:

```
stb.SetCASType(4);
```

будет искаться библиотека /home/default/libCasCustom4.so

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 **fpkevent** для **Firefox** и **fpqevent** 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

1080p-50 (для MAG250)

NTSC

576p-60

720p-60

1080i-60

1080p-60 (для MAG250)

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

Замечание. Изменения вступают 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@50

1080i – режим 1080i@50

1080p – режим 1080p@50 (для MAG250)

480i – режим NTSC

720p60 – режим 720p@60

1080i60 – режим 1080i@60

1080p60 – режим 1080p@60 (для MAG250)

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.

Приложение 12. JavaScript API для подсистемы обновления ПО

Операции с подсистемой производятся с помощью объекта «**stbUpdate**».

Объект **stbUpdate** не нуждается в дополнительной инициализации и всегда доступен из JavaScript контекста.

Объект предоставляет интерфейс к менеджеру обновления.

Менеджер обновления позволяет инициировать и отображать состояние процесса обновления ПО устройства.

Внимание! Перед операцией обновления ПО необходимо остановить проигрывание медиаконтента.

Варианты использования

Общий сценарий работы с объектом «stbUpdate»

Возможные состояния объекта

Менеджер обновления представляет собой конечный автомат.

Состояние автомата доступно через метод [getStatus\(\)](#). Начальное состояние автомата – «Свободен» (значение 21).

Выполнение активных операций с подсистемой обновления разрешается только, если состояние имеет значение «Свободен».

После начала операции (например, обновление) автомат покидает состояние «Свободен» и считается занятым до тех пор, пока опять не попадет в состояние «Свободен». Таким образом, после инициации операции необходимо периодически запрашивать состояние и ждать пока оно не перейдет в состояние «Свободен».

Взаимодействие с пользователем

Дополнительная информация

Помимо основного состояния доступна дополнительная информация, которая отражает общий ход выполнения операций.

Дополнительная информация представляет собой локализованную строку символов и доступна через вызов метода [getStatusStr\(\)](#).

Для корректного отображения дополнительной информации в контексте пользовательского интерфейса рекомендуется периодически запрашивать метод (рекомендуемое время 1 сек) и при необходимости выводить полученную информацию пользователю.

Прогресс индикатор

Индикатор прогресса отражает степень завершенности текущей операции в процентах и доступен через метод [getPercents\(\)](#).

Проверка файла обновления

Имеется возможность предварительной (до начала операции обновления) проверки основных атрибутов файла прошивки.

Проверка атрибутов файла прошивки инициируется вызовом метода [startCheck\(\)](#).

Параметром передается путь к файлу обновления (URL либо путь к файлу). Например

«/media/usbdisk/mag200/imageupdate»

либо

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

После завершения операции проверки необходимая информация может быть получена через методы [getImageVersionStr\(\)](#), [getImageDateStr\(\)](#), [getImageDescStr\(\)](#).

Процедура выбора банка памяти для обновления

Используя метод [getActiveBank\(\)](#) можно определить какой банк памяти является активным.

В случае, если активный банк известен (метод [getActiveBank](#) вернул либо «0» либо «1») обновление необходимо инициировать в банк памяти отличный от активного банка. Если активный банк неизвестен (метод [getActiveBank](#) вернул «-1»), то выбор банка оставляется на усмотрение разработчиков процедуры обновления.

Обновление ПО

Процедура обновления инициируется вызовом метода [startUpdate\(\)](#). Параметром передается банк памяти и путь к файлу обновления (URL либо путь к файлу). Например

«/media/usbdisk/mag200/imageupdate»

либо

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

Типичное время выполнения данной операции может достигать нескольких минут. Поэтому в ходе выполнения данной операции рекомендуется предоставление пользователю информации о ходе ее выполнения (в частности прогресс индикатор).

В ходе выполнения операции дополнительная информация доступна через методы [getPercents\(\)](#), [getStatus\(\)](#), [getStatusStr\(\)](#), [getImageVersionStr\(\)](#), [getImageDateStr\(\)](#), [getImageDescStr\(\)](#).

Обновление ПО

Рекомендуемая процедура обновления должна состоять из следующих этапов:

1. Для выполнения операции необходимо дождаться состояния объекта «Свободен».
2. Инициировать проверку атрибутов файла прошивки (вызвать [startCheck](#) и дождаться окончания операции).
3. Проанализировать полученные атрибуты файла прошивки и принять решение о том, следует ли начинать процесс обновления.
4. Выбрать банк памяти для обновления (на основании данных, полученных после вызова метода [getActiveBank](#)).
5. Инициировать процесс обновления вызовом метода [startUpdate\(\)](#). В процессе выполнения операции обновления состояние объекта доступно через метод [getStatus\(\)](#), дополнительная информация доступна через метод [getStatusStr\(\)](#), прогресс индикатор через метод [getPercents\(\)](#).
6. В случае неуспешного завершения операции объект переходит в состояние «Свободен» и дополнительная информация содержит причину ошибки.
7. В случае успешного завершения операции устройство автоматически перезапустится.

Автоматическое обновление ПО

Иницируется вызовом метода [startAutoUpdate\(\)](#)

В ходе выполнения операции пользователю будет отображена форма автоматического обновления, на которой будет отображаться процесс обновления

Операция автоматического обновления представляет собой следующие этапы:

1. Проверку атрибутов файла прошивки.
2. В случае, если не удалось получить атрибуты файла прошивки процесс прерывается и индицируется ошибка.

3. Определяется банк памяти для обновления. Если активный банк равен «0», то выбирается банк 1 и 0 в во всех остальных случаях.
4. В случае, если была задана проверка версии ПО (аргумент **checkVersion**), то обновление производится только в случае, если активная версия ПО старше доступной для обновления.
5. Производится обновление ПО.

Методы объекта «stbUpdate»

stbUpdate.getStatusStr

string getStatusStr()

Возвращает статус подсистемы в виде строки.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string	Локализованная строка символов. Язык соответствует настройкам встроенного портала конфигурации	Возвращается произвольная строка статуса, которая отражает ход выполнения текущей операции

stbUpdate.getStatus

int getStatus()

Возвращает статус подсистемы.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
int	-1, 1..23	-1: не определено 1: Ошибка инициализации крипто библиотеки 2: Неправильная модель устройства 3: Размер NAND слишком мал для размещения данных 4: Необходимый раздел NAND не найден 5: Обновление ядра 6: Обновление образа

		7: Внутренняя ошибка 8: Анализ прошивки 9: Обновление переменных среды 10: Обновление bootstrap 11: Обновление bootstrap 12: Обновление файловой системы пользователя 13: Обновление файловой системы пользователя 14: Обновление второго загрузчика 15: Обновление логотипа 16: Обновление завершено 17: Неправильная сигнатура 18: Подготовка NAND для записи 19: Ошибка записи в NAND 20: Ошибка записи в файл 21: Свободен 22: Неправильный заголовок файла 23: Анализ версии файла обновления
--	--	--

stbUpdate.getPercents

int getPercents()

Возвращает статус завершенности текущего процесса, выраженный в процентах.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
int	0-100	Значение выраженное в процентах

stbUpdate.getActiveBank

int getActiveBank()

Возвращает банк памяти, который был использован для загрузки текущего ПО.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
int	-1, 0, 1	0 – первый банк памяти 1 – второй банк памяти -1 – банк памяти не определен (такое возможно, когда приставка была загружена с сетевых носителей информации.

		Например, NFS)
--	--	----------------

stbUpdate.startCheck

void startCheck (string image)

Иницирует проверку файла обновления.

Операция доступна только при состоянии объекта «Свободен». Операция считается выполненной, когда объект обратно перейдет в состояние «Свободен».

Параметры:

Имя параметра	Возможные значения	Описание
image	URL с указанием файла обновления доступного по протоколу http (например, http://test.com/imageupdate) или путь к файлу обновления расположенному на USB диске (например, /media/usbdisk/mag200/imageupdate)	Указанный файл будет проанализирован и информация о его компонентах будет прочитана

Возвращаемое значение:

Нет

stbUpdate.getImageVersionStr

string getImageVersionStr()

Возвращает версию образа.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string		Возвращается версию образа, которая задается при его сборке

stbUpdate.getImageDateStr

string getImageDateStr()

Возвращает дату создания образа.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string		Возвращается дата создания образа, которая задается при его сборке

stbUpdate.getImageDescStr

string getImageDescStr()

Возвращает описание образа.

Параметры:

Нет

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string		Возвращается описание образа, которое задается при его сборке

stbUpdate.startUpdate

void startUpdate(int bank, string image)

Иницирует обновление ПО устройства из указанного файла обновления.

Обновление будет произведено в указанный банк памяти.

Операция доступна только при состоянии объекта «Свободен». Операция считается выполненной, когда объект обратно перейдет в состояние «Свободен».

Параметры:

Имя параметра	Возможные значения	Описание
bank	0,1	«0» – обновление первого банка «1» - обновление второго банка
image	URL с указанием файла обновления доступного по протоколу http (например, http://test.com/imageupdate) или путь к файлу обновления расположенному на USB диске (например, /media/usbdisk/mag200 /imageupdate)	Будет произведено обновление ПО устройства с использованием указанного файла

Возвращаемое значение:

Нет

stbUpdate.startAutoUpdate

void startAutoUpdate(string image, bool checkVersion)

Иницирует автоматическое обновление ПО устройства из указанного файла обновления.

Обновление будет произведено в автоматически выбранный банк памяти.

В процессе обновления пользователю будет отображена форма, которая будет отображать ход выполнения операции.

Параметры:

Имя параметра	Возможные значения	Описание
image	URL с указанием файла обновления доступного по протоколу http (например, http://test.com/imageupdate) или путь к файлу обновления расположенному на USB диске (например, /media/usbdisk/mag200/imageupdate)	Будет произведено обновление ПО устройства с использованием указанного файла
checkVersion		true – обновление будет произведено только, если версия текущего ПО устройства старше false – обновление будет произведено в любом случае

Возвращаемое значение:

Нет

Пример использования API

Ниже приведен исходный код (HTML/JS) WEB страницы, демонстрирующей базовые основы работы с JS API.

Поле «Update file» содержит путь к файлу обновления (URL или путь к файлу).

Нажатие кнопки «Check» (или «0» на ПДУ) иницирует проверку файла, кнопка «Update» (или «1» на ПДУ) начинает процедуру обновления в банк 0. «2» на ПДУ начинает процедуру автоматического обновления.

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

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

```

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

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

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

```

Приложение 13. JavaScript API для подсистемы PVR

Операции с подсистемой производятся с помощью объекта «**pvrManager**».

Объект **pvrManager** не нуждается в дополнительной инициализации и всегда доступен из JavaScript контекста.

Объект предоставляет интерфейс к менеджеру записей каналов (PVR).

Менеджер записей позволяет назначать задачи на локальную запись необходимых каналов(поточков) в определённый период времени при наличии подключенного внешнего накопителя.

Внимание. Возможна запись только каналов, в которых вещается транспортный поток (mpeg-ts), как по мультикасту, так и с http сервера.

Описание объекта pvrManager

Таблица кодов ошибок

Возможные значения	Описание
0	Операция прошла успешно.
-1	Неправильный параметр.
-2	Недостаточно памяти для выполнения операции.
-3	Неправильный диапазон записи, например длительность закачки не должна превышать 24 часа.
-4	Задача с заданным идентификатором не найдена.
-5	Неправильное имя файла. Директория, в которую сохраняется запись должна начинаться с /media/USB- или /ram/media/USB-.
-6	Запись в заданный файл уже существует.
-7	Ошибка открытия URL канала.
-8	Ошибка открытия файла для сохранения.
-9	Превышено максимальное количество одновременных записей. Имеется ввиду не количество текущих задач, а количество одновременно сохраняющихся задач в определённый момент времени. См. также SetMaxRecordingCnt .
-10	В процессе записи получен конец потока (длительное отсутствие данных). Задача завершена с сохранением записи в заданный

	файл.
-11	Ошибка записи в файл. Например, закончилось место на носителе или его отключили в процессе записи.

Таблица состояний задач

Возможные значения	Описание
1	Ожидание начала записи.
2	Идёт запись.
3	Произошла ошибка, запись остановлена.
4	Запись завершена.

pvrManager.CreateRecordingTask

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

Запустить задачу записи канала.

Параметры:

Имя параметра	Возможные значения	Описание
url	http://... rtp://... udp://...	URL канала, который необходимо записать.
fileName	/media/USB-... или /ram/media/USB-...	Полное имя файла, в который необходимо сохранить запись данного канала.
startTime	UTC время в формате "YYYYMMDDThhmmss" или количество секунд с начала эпохи (1970/01/01 UTC).	Время начала записи.
endTime	UTC время в формате "YYYYMMDDThhmmss" или количество секунд с начала эпохи (1970/01/01 UTC).	Время окончания записи.

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string	В случае успешного выполнения возвращается уникальный идентификатор задачи записи. В случае ошибки возвращается строка соответствующая коду ошибку (<0) см. Таблица кодов ошибок	Идентификатор задачи либо ошибка.

Замечание. Количество секунд с начала эпохи можно получить через объект Date, например:

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

pvrManager.GetAllTasks

string GetAllTasks()

Получить список всех задач.

Параметры:

Нет.

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string	[task_1,...,task_n], где task_n имеет следующий вид: {id:1,state:0,errorCode:0,filename:'/media/USB-1/1.ts', url:'http://192.168.1.1/mpegts' , startTime:3452344145,endTime:3452345345 }	Список всех задач в виде массива задач в JSON формате.

В этом объекте:
id – уникальный идентификатор задачи.
state – текущее состояние задачи. См. [Таблица состояний задач](#).
errorCode – код ошибки. См. [Таблица кодов ошибок](#).
fileName – заданное имя файла, в который происходит сохранение потока.
url – URL сохраняемого потока.
startTime и endTime – время начала и окончания записи. См. [CreateRecordingTask](#).

pvrManager.GetTasksByIds

string GetTasksByIds (string idList)

Получить список задач по заданному списку идентификаторов.

Параметры:

Имя параметра	Возможные значения	Описание
idList	строка вида: [id_1,...,id_n]	Список идентификаторов задач в формате JSON массива.

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string	<p>[task_1,...,task_n], где task_n имеет следующий вид:</p> <pre>{id:1,state:0,errorCode:0, filename:'/media/USB-1/1.ts', url:'http://192.168.1.1/mpegts', startTime:3452344145, endTime:3452345345}</pre> <p>В этом объекте: id – уникальный идентификатор задачи. state – текущее состояние задачи. См. Таблица состояний задач. errorCode – код ошибки. См. Таблица кодов ошибок. fileName – заданное имя файла, в который происходит сохранение потока. url – URL сохраняемого потока. startTime и endTime – время начала и окончания записи. См. CreateRecordingTask.</p>	Список задач в виде массива задач в JSON формате.

pvrManager.GetTaskById

string GetTaskById (string id)

Получить состояние задачи с заданным идентификатором.

Параметры:

Имя параметра	Возможные значения	Описание
id	строка	Идентификатор задачи.

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string	<p>Возвращает строку вида {id:1,state:0,errorCode:0, filename:'/media/USB-1/1.ts', url:'http://192.168.1.1/mpegts', startTime:3452344145, endTime:3452345345} Более детально см. GetAllTasks. В случае отсутствия задачи с таким идентификатором возвращается строка {}</p>	

pvrManager.RemoveTask

void RemoveTask (string id, int removeType)

Удалить задачу с заданным идентификатором.

Параметры:

Имя параметра	Возможные значения	Описание
id		Идентификатор задачи.
removeType	0-3	0 – не удалять ни каких файлов. 1 – если существует временный файл, то переименовать его в заданный при создании задачи. 2 – удалить только временный файл, если он существует. 3 – удалить и временный файл, и заданный при создании записи.

Возвращаемое значение:

Нет.

pvrManager.ChangeEndTime

int ChangeEndTime(string id, string endTime)

Изменить время окончания записи канала.

Параметры:

Имя параметра	Возможные значения	Описание
id		Идентификатор записи.
endTime	См. CreateRecordingTask	Новое время окончания записи.

Возвращаемое значение:

Тип данных	Возможные значения	Описание
int	См. Таблица кодов ошибок	Результат выполнения операции.

pvrManager.SetMaxRecordingCnt

void SetMaxRecordingCnt (int maxCnt)

Установить максимальное количество одновременных записей.

Параметры:

Имя параметра	Возможные значения	Описание
maxCnt		Максимальное количество одновременных записей.

Возвращаемое значение:

Нет.

Приложение 14. JavaScript API для управления закачками

Операции с подсистемой производятся с помощью объекта «**stbDownloadManager**».

Объект **stbDownloadManager** не нуждается в дополнительной инициализации и всегда доступен из JavaScript контекста.

Объект предоставляет интерфейс к менеджеру загрузки файлов.

Менеджер загрузки позволяет помещать в очередь задания для загрузки файлов и производить автоматическое управление заданиями в соответствии с расписанием.

Методы объекта « **stbDownloadManager** »

DeleteJob

void DeleteJob(double id, boolean deleteFile)

Указанная загрузка будет удалена

Параметры:

Имя параметра	Возможные значения	Описание
id	Numeric (-1, 0-4294967295)	Указывает, над какой загрузкой должно быть совершено действие
deleteFile	boolean	True – также удалить связанный файл False – удалить только загрузку, не удалять файл

Возвращаемое значение:

Нет

StartJob

void StartJob(double id)

Указанная загрузка будет переведена в состояние «ожидание очереди»

Параметры:

Имя	Возможные значения	Описание
-----	--------------------	----------

параметра		
Id	Numeric (-1, 0-4294967295)	Указывает, над какой закачкой должно быть совершено действие

Возвращаемое значение:

Нет

StopJob

void StopJob(double id)

Указанная закачка будет переведена в состояние «остановлена»

Параметры:

Имя параметра	Возможные значения	Описание
Id	Numeric (-1, 0-4294967295)	Указывает, над какой закачкой должно быть совершено действие

Возвращаемое значение:

Нет

AdjustJobPriority

void AdjustJobPriority(double id, Boolean rise)

Изменить приоритет указанной закачки

Параметры:

Имя параметра	Возможные значения	Описание
Id	Numeric (-1, 0-4294967295)	Указывает, над какой закачкой должно быть совершено действие
rise	boolean	True – повышает приоритет False – понижает приоритет данной закачки

Возвращаемое значение:

Нет

PlayDownloadedMedia

void PlayDownloadedMedia (double id)

Иницируется проигрывание ресурса в специальном окне портала (генерируется событие «stbEvent.onMediaAvailable(...)»)

Параметры:

Имя параметра	Возможные значения	Описание
Id	Numeric (-1, 0-4294967295)	Указывает, над какой загрузкой должно быть совершено действие

Возвращаемое значение:

Нет

AddJob

bool AddJob(String urlToDownload, String filePath)

Будет создано задание для загрузки файла по адресу «urlToDownload».

В случае успешного завершения загрузки файл будет размещен по адресу «filePath»

На момент создания задания файл «filePath» не должен существовать. В противном случае задание не будет создано

Параметры:

Имя параметра	Возможные значения	Описание
urlToDownload	URL в виде строки	Файл, находящийся по данному URL будет загружен
filePath	Путь к несуществующему файлу в пределах файловой системы приставки	После успешной загрузки файл будет размещен по адресу

Возвращаемое значение:

Тип данных	Возможные значения	Описание
Boolean	True, false	В случае успешного создания задания возвращается значение «true»

GetQueueInfo

string GetQueueInfo(string idList = «»)

Получить информацию об очереди закачек

Параметры:

Имя параметра	Возможные значения	Описание
idList	Должен содержать список ID объектов, выраженный в виде строки типа «1.0, 2.0, 3.0»	Будет передана информация только по заданной зачке В случае, если список пуст будет передана информация по всем объектам

Возвращаемое значение:

Тип данных	Возможные значения	Описание
string	Содержит Javascript массив объектов. Длина массива может быть произвольной (зависит от размера очереди) Объекты содержат поля: «id» - numeric (0-4294967295) «state» - numeric «stateStr» - string «url» - string «filePath» - string «progressPct» - numeric (0-100) «sizeDone» - numeric «sizeTotal» - numeric (значение -1 если неопределено)	«id» - идентификатор зачки «state» - номер состояния (0 – Stopped, 1 – WaitingQueue, 2 – Running, 3 – Completed, 4 – TemporaryError, 5 - PermanentError) «stateStr» - состояние строкой (поддерживается локализация) «url» - URL откуда закачивается файл «filePath» - путь по которому будет сохранен результирующий файл «progressPct» - прогресс зачки в процентах «sizeDone» - размер уже загруженных данных

	«prio» - numeric «attempt» - numeric	«sizeTotal» - общий размер файла (может быть неопределен) «prio» - приоритет загрузки «attempt» - попытка скачивания
--	---	--

История изменений документа.

Версия 1.19

Добавлено описание событий 7 и 0x22 в [Список используемых событий](#).

Добавлены функции:

[stb.GetVideoInfo](#),

[stb.SetWebProxy](#),

[stb.Play](#) с использованием прокси сервера.

Добавлено [Приложение 13. JavaScript API для подсистемы PVR](#).

Добавлено [Приложение 14. JavaScript API для управления загрузками](#).

Версия 1.18

Добавлено описание события DualMono в [Список используемых событий](#).

Добавлены функции:

[stb.SetBufferSize](#),

[stb.GetBufferLoad](#).

Добавлено описание работы с внешним CAS плагином в [stb.SetCASType](#) и [Настройка внешнего CAS плагина](#).

Версия 1.17

Добавлен раздел [Приложение 12. JavaScript API для подсистемы обновления ПО](#).

Добавлены функции:

[stb.GetMute](#),

[stb.StartLocalCfg](#),

[stb.ShowVirtualKeyboard](#),

[stb.HideVirtualKeyboard](#),

[stb.EnableServiceButton](#),

[stb.EnableVKButton](#),

[stb.EnableSpatialNavigation](#),

[stb.EnableSetCookieFrom](#).

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](#).