ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 56170— 2014

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ

ДОМАШНЯЯ МУЛЬТИМЕДИЙНАЯ ПЛАТФОРМА

Класс 1.2. Основные параметры

ETSI TS 102 727 V1.1.1 (2010-01)
(NEQ)

Издание официальное

Москва
Стандартинформ
2015

gost.e-ecolog.ru

Предисловие

1 РАЗРАБОТАН Автономной некоммерческой организацией «Научно-технический центр информатики» (АНО «НТЦИ»)

2 ВНЕСЕН Техническим комитетом по стандартизации ТК 480 «Связь»

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 16 октября 2014 г. № 1338-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений стандарта Европейского института по стандартизации в области телекоммуникаций (ETSI) ЕТСИ ТС 102 727 V1.1.1 (2010-01) «Телевидение вещательное цифровое. Домашняя мультимедийная платформа (MHP) Спецификация 1.2.2» [ETSI TS 102 727 V1.1.1 (2010-01) «Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2)» , NEQ]

5 ВВЕДЕН ВПЕРВЫЕ

*Правила применения настоящего стандарта установлены в ГОСТ Р 1.0–2012 (раздел 8). Информация об изменениях к настоящему стандарту публикуется в ежегодном (по состоянию на 1 января текущего года) информационном указателе «Национальные стандарты», а официальный текст изменений и поправок – в ежемесячном информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ближайшем выпуске ежемесячного информационного указателя «Национальные стандарты». Соответствующая информация, уведомления и тексты размещаются также в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет (*gost.e-ecolog.ru

© Стандартинформ, 2015

Настоящий стандарт не может быть полностью или частично воспроизведен, тиражирован и распространен в качестве официального издания без разрешения Федерального агентства по техническому регулированию и метрологии

Содержание

1 Область применения 1
2 Нормативные ссылки 2
3 Термины, определения и сокращения 2
4 Классы домашней мультимедийной платформы 10
5 Базовая архитектура домашней мультимедийной платформы 11
5.1 Контекст применения MHP 11
5.2 Архитектура MHP 11
5.3 Интерфейсы между приложениями MHP и системой MHP 11
5.4 Плагины 11
5.5 Место профиля IPTV в предыдущих версиях MHP 11
6 Параметры транспортных протоколов 12
6.1 Параметры транспортных протоколов каналов вещания и интерактивных каналов 12
6.2 Транспортные протоколы загрузки приложений через интерактивный канал 13
6.3 Транспортные протоколы для загрузки приложений из интерактивного канала 15
6.4 Протоколы IPTV 16
7 Параметры форматов контента, поддерживаемых платформой MHP 17
7.1 Статические форматы контента 17
7.2 Форматы потокового вещания 18
7.3 Форматы резидентных шрифтов 18
7.4 Форматы загружаемых шрифтов 18
7.5 Форматы представления цвета 18
7.6 Форматы многоцелевых расширений почты Интернета 18
8 Параметры приложений DVB-HTML 19
8.1 Введение 19
8.2 Архитектура 19
8.3 Формат приложений 20
8.4 Расширяемый язык разметки XML 20
8.5 Язык разметки DVB (DVB-HTML) 20
8.6 Типы медиа 22
8.7 Синхронизация 27
8.8 Каскадные таблицы стилей 31
8.9 Интеграция Xlet 32
8.10 Создание (планирование) сценариев 32
8.11 Объектная модель документа (DOM) 33
8.12 Поддержка Cookie 38
8.13 Поддержка строки агента пользователя HTTP 39
8.14 Безопасность приложений DVB-HTML 39
8.15 Разрешения DVB-HTML 41
8.16 Нормирование параметров данных различного характера 42
9 Модель приложений MHP 43
9.1 Приложения MHP, связанные службой 43
9.2 Параметры модели приложений DVB-J 44
9.3 Модель DVB-HTML 46

9.4 Управление ресурсами, которые используются разными приложениями 49
9.5 Жизненный цикл интерфейсов Xlet, встроенных в приложения DVB-HTML 50
9.6 Службы и приложения, не связанные с традиционными службами DVB 50
9.7 Жизненный цикл приложений доступа в Интернет 51
9.8 Плагины 51
9.9 Хранение и кэширование приложений 51
9.10 Условия взаимодействия между приложениями MHP и резидентными приложениями 52
9.11 Провайдеры 52
9.12 Воздействие ограничений графики на модель приложения 52
9.13 Несвязанные приложения 53
10 Параметры сигнализации приложения MHP 54
10.1 Введение 54
10.2 Параметры сигнализации программно-зависимой информации 55
10.3 Система обозначений 56
10.4 Таблица AIT 56
10.5 Параметры идентификации приложений 59
10.6 Управление жизненным циклом приложения 60
10.7 Универсальные дескрипторы 61
10.8 Дескрипторы транспортного протокола 63
10.9 Дескрипторы DVB-J 65
10.10 Дескрипторы приложения DVB-HTML 65
10.11 Константы дескрипторов 66
10.12 Информация о службе 67
10.13 Сигнализация плагина 68
10.14 Сохраненные приложения 69
10.15 Сигнализация для провайдеров 70
10.16 Сигнализация для IPTV 70
10.17 XAIT для классических сетей DVB 70
10.18 Приложения, не связанные с резидентными устройствами 71
11 Платформа DVB-J 71
11.1 Виртуальная машина DVB-J 71
11.2 Общие вопросы 72
11.3 Основные программные интерфейсы DVB-J 74
11.4 Параметры программных интерфейсов приложений представления (воспроизведения) 74
11.5 API доступа к данным 78
11.6 Программные интерфейсы приложений информации о службе и выбора службы 78
11.7 Общая инфраструктура интерфейсов API 81
11.8 Безопасность 83
11.9 Другие программные интерфейсы приложений 84
11.10 Полномочия Java 86
11.11 Соответствие между объектами, локаторами контента и их текстовыми представлениями 87
11.12 Автономные приложения 88
11.13 Поддержка DVB-HTML 88
11.14 Доступ в Интернет 89

11.15 Интерфейсы API, определенные в OCAP 89
12 Безопасность 90
12.1 Введение 90
12.2 Аутентификация приложений 91
12.3 Параметры передачи сообщений безопасности 91
12.4 Детализация сообщений аутентификации приложения 91
12.5 Параметры профилирования приложений 92
12.6 Политика безопасности для приложений 92
12.7 Пример создания приложения, которое может быть аутентифицировано 92
12.8 Процесс управления сертификатом 92
12.9 Параметры безопасности обратного канала 92
12.10 Параметры профиля Интернет 93
12.11 Аппаратная реализация платформы MHP 93
12.12 Безопасность применения плагинов 93
12.13 Безопасность загрузки приложений из интерактивного канала 93
12.14 Хранение приложений 94
12.15 Безопасность внутренних приложений и контента, встроенного в другие приложения 94
12.16 Аутентификации несвязанных приложений 94
12.17 Аутентификация привилегированных приложений 94
13 Параметры эталонной модели графики платформы MHP 95
14 Системная интеграция платформы MHP 95
15 Детализированные определения профилей платформы MHP 95
15.1 Требования к формату PNG 95
15.2 Требования к составу форматов медиа, поддерживаемых API DVB-J 95
15.3 Требования к формату JPEG 95
15.4 Требования к поддержке локалей 95
15.5 Требования к растру видеоизображения 95
15.6 Отображение платформой MHP функциональных эквивалентов GEM 95
16 Требования к константам MHP 96
16.1 Требования к системным константам 96
16.2 Требования к константам DVB-J 96
17 Доступ Интернет-клиентов в сеть Интернет 96
17.1 Ссылки служб DVB и контента на контент WWW 96
17.2 Требования к Интернет-клиентам 96
17.3 Обработка потокового медиа из сети Интернет 97
17.4 Управление соединением Интернет 97
Библиография 98

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ТЕЛЕВИДЕНИЕ ВЕЩАТЕЛЬНОЕ ЦИФРОВОЕ

ДОМАШНЯЯ МУЛЬТИМЕДИЙНАЯ ПЛАТФОРМА

Класс 1.2. Основные параметры

Digital Video Broadcasting. Multimedia Home Platform. Classes 1.2. Basic parameters

Дата введения — 2015—09—01

1 Область применения

Настоящий стандарт распространяется на аппаратно-программный комплекс - домашнюю мультимедийную платформу (Multimedia Home Platform; MHP), позволяющую поддерживать предоставление интерактивных цифровых телевизионных служб, поставляемых на терминалы любых производителей, поддерживающих настоящий стандарт. Терминалы MHP принимают цифровые службы DVB при работе в каналах передачи различных стандартов, включая спутниковые, кабельные, эфирные и TCP/IP. Транспортный уровень терминалов MHP базируется на технологиях вещания в соответствии со стандартами DVB-T/T2, DVB-C/C2, DVB-S/S2 или IP-транспорта.

Настоящий стандарт обеспечивает доступ терминалов MHP к службам вещания применением прикладных программных интерфейсов (API) с гарантированной семантикой. Функциональные возможности приемника в составе терминала могут быть представлены форматом контента, имеющим семантические гарантии. Другим элементом стандарта MHP являются требования к аппаратным средствам терминала и к инфраструктуре сигнализации, что обеспечивает возможность работы терминала с любой совместимой сетью.

Настоящий стандарт предназначен для разработчиков программных и аппаратных платформ и для разработчиков приложений, использующих функциональные возможности MHP.

В настоящем стандарте MHP нормируются три области приложений:

- улучшенное вещание;
- интерактивное вещание;
- доступ к Интернету.

Улучшенное вещание сочетает цифровое вещание аудио-/видео служб с загруженными приложениями применением локальной интерактивности без использования интерактивного канала.

Интерактивное вещание обеспечивает несколько видов интерактивных служб, связанных со службой вещания или независимых от служб вещания. Интерактивное вещание использует интерактивный канал.

Приложение доступ к Интернету обеспечивает предоставление служб сети Интернет, включая ссылки между службами Интернет и службами вещания.

Определения профилей MHP, обратной совместимости профилей и перекрестной совместимости между профилями представлены в разделе 15 настоящего стандарта.

При работе с настоящим стандартом необходимо пользоваться следующими исходными правилами:

- <профиль> <n 1> должен быть надстройкой над <профилем> <n>;
- профиль интерактивного вещания 1 определяется как надстройка над расширенным профилем вещания 1.

Кроме того, настоящий стандарт определяет возможность MHP DVB поддержки доставки служб DVB по широкополосным сетям IP. Он определяет необходимые для доступа к широкополосным IP-сетям существующие интерфейсы API MHP и устанавливает расширения семантики для этих API. В необходимых случаях для доступа к функциям широкополосных сетей IP предусматривается использование интерфейсов и протоколов для доставки служб DVB в сетях, определенных в [1]. Кроме того, настоящий стандарт предусматривает (опционально) доступ к информации, передаваемой согласно [2].

Издание официальное

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 52210–2004 Телевидение вещательное цифровое. Термины и определения

ГОСТ Р 52591–2006 Система передачи данных пользователя в цифровом телевизионном формате. Основные параметры

ГОСТ Р 53528-2009 Телевидение вещательное цифровое. Требования к реализации протокола высокоскоростной передачи информации DSM-CC. Основные параметры

ГОСТ Р 54456-2011 Телевидение вещательное цифровое. Домашняя мультимедийная платформа. Класс 1.0. Основные параметры

Примечание – При пользовании настоящим стандартом целесообразно проверить действие ссылочных стандартов в информационной системе общего пользования – на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет или по ежегодному информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по выпускам ежемесячного информационного указателя «Национальные стандарты» за текущий год. Если заменен ссылочный стандарт, на который дана недатированная ссылка, то рекомендуется использовать действующую версию этого стандарта с учетом всех внесенных в данную версию изменений. Если заменен ссылочный стандарт, на который дана датированная ссылка, то рекомендуется использовать версию этого стандарта с указанным выше годом утверждения (принятия). Если после утверждения настоящего стандарта в ссылочный стандарт, на который дана датированная ссылка, внесено изменение, затрагивающее положение, на которое дана ссылка, то это положение рекомендуется применять без учета данного изменения. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, рекомендуется применять в части, не затрагивающей эту ссылку.

3 Термины, определения и сокращения

3.1 В настоящем стандарте применены термины по ГОСТ Р 52210, ГОСТ Р 52591, а также следующие термины с соответствующими определениями:

3.1.1 администратор приложений (application manager): Объект в MHP, который обеспечивает управление жизненным циклом приложений MHP, в том числе приложений DVB-J.

3.1.2 агент пользователя (user agent): Приложение, которое интерпретирует формат контента. Допускается реализация агента пользователя в форме плагина.

3.1.3 актор DVB-HTML (DVB-HTML actor): Область деятельности или процесса выполнения конкретного набора документов DVB-HTML для некоторого приложения DVB-HTML в среде MHP. Актор выполняется в агенте пользователя.

3.1.4 аплет (applet): Подпрограмма, встроенная в прикладную программу и загружаемая с сервера вместе с запрашиваемыми документами DVB-HTML как прикрепленный файл.

3.1.5 архитектура брокера общих объектных запросов (Common Object Request Broker Architecture; CORBA): Открытый стандарт для взаимодействия (интероперабельности) приложений.

3.1.6 атрибут (attribute): Характерный признак объекта.

3.1.7 байт-код (byte-code, Java byte-code): Машинно-независимый код, генерируемый Java-компилятором.

3.1.8 букет DVB (Bouquet DVB): Совокупность служб DVB, предлагаемых пользователю как единый продукт.

3.1.9 вещатель (broadcaster, Service Provider): Организация, которая собирает последовательность событий или программ для доставки.

3.1.10 видео «капли» (video «drips»): Форма медиа, когда на вход видеодекодера транспортный поток MPEG-2 подается блоками, содержащими кадры I и кадры P. Каждый блок должен содержать один кадр и определенное число синтаксических элементов в соответствии с [3].

3.1.11 виртуальная машина Java (Virtual Machine Java; JVM): Основная часть исполняющей системы Java (Java Runtime Environment; JRE). Виртуальная машина Java интерпретирует и исполняет Java байт-код, предварительно созданный из исходного текста Java-программы Java-компилятором. JVM может использоваться для выполнения программ, написанных на других языках программирования.

3.1.12 граница приложения (application boundary): Краткое общее описание элементов данных (документы языка разметки гипертекста (Hyper Text Mark-up Language; HTML), файлы кода, файлы изображения), сформированное в одно приложение, и логический локатор точки входа. Граница приложения описывается регулярным выражением на языке URL.

3.1.13 дескриптор (descriptor): Кодовое слово, служащее для описания типа передаваемых данных.

3.1.14 документ DVB-HTML (DVB-HTML document): Полный (завершенный) модуль элементов или форматов контента одного семейства HTML, определенного в настоящем стандарте.

3.1.15 домашняя мультимедийная платформа (Multimedia Home Platform; MHP): Аппаратно-программный комплекс, поддерживающий совокупность стандартов цифрового телевизионного вещания (DVB) и обеспечивающий доступ пользователя к интерактивным и вещательным службам.

3.1.16 домен (domain): Автономная часть сети или распределенной системы.

3.1.17 загрузка (download): Пересылка файлов по сети от пользователя или сервера пользователю или серверу.

3.1.18 идентификатор типа пакета (packet identifier; PID): Тринадцатибитовый указатель в заголовке транспортного пакета, определяющий принадлежность пакета тому или иному потоку данных.

3.1.19 Интернет-клиент (Приложение доступа в Интернет, Приложение Интернет клиент) (Internet client; Internet access application): Резидентное приложение терминала MHP, используемое для представления контента Интернет.

3.1.20 интероперабельность (функциональная совместимость) (interoperability): Нейтральная платформа, обеспечивающая прием и представление приложений для поставщика, автора и вещателя.

3.1.21 интерфейс API CA: Интерфейс между системой условного доступа (CA) и приложением MHP, является независимым интерфейсом высокого уровня, обеспечивающим доступ приложения к системам CA.

3.1.22 информация о службах (Service Information, SI): Совокупность таблиц, которые передаются в составе транспортных потоков MPEG-2, предназначенных для вещания. К основным таблицам информации о службах относятся таблицы, характеризующие параметры сети передачи, компоненты служб: таблица объединения букета программ (Bouquet Association Table; BAT), таблица информации о событиях (Event Information Table; EIT), таблица состояния событий (Running Status Table; RST), таблица описания служб (Service Description Table; SDT), таблица времени и даты (Time and Date Table; TDT), таблица смещения времени (Time Offset Table; TOT).

3.1.23 исполняющая система Java (Java Runtime Environment; JRE): Минимизированная реализация виртуальной машины, необходимая для исполнения Java-приложений (без Java-компилятора) и других средств разработки. Состоит из JVM и библиотеки Java-классов.

3.1.24 Карусель Данных (Data Carousel): Передача модулей данных с циклическим повторением.

3.1.25 Карусель Объектов (Object Carousel; OC): Передача в транспортном потоке циклически повторяющихся объектов (Файлов, Каталогов), связанных с потоками данных.

3.1.26 каскадные таблицы (Cascading Style Sheets; CSS): Таблицы, позволяющие хранить цвет, размеры текста и другие параметры в стилях. CSS являются методом просмотра контента HTML или XML.

3.1.27 **класс** (class): Разновидность абстрактного типа данных в объектно-ориентированном программировании (ООП). Содержит описание переменных и констант, характеризующих объект.

3.1.28 класс 1.0; класс 1.1: Классы MHP по видам предоставляемых услуг.

3.1.29 клиент (Client): Потребитель услуг одного или нескольких серверов.

3.1.30 конструктор класса (class constructor): Специальный блок инструкций, вызываемый при создании объекта.

3.1.31 конструктор по умолчанию (default constructor): Конструктор, создаваемый компилятором при отсутствии конструктора класса.

3.1.32 контекст (context): Состояние системы; окружение системы, среда исполнения программы; текущая ситуация.

3.1.33 контент (content): Содержание, мультимедийный продукт (например, телевизионная программа).

3.1.34 конфигурация (configuration): Совокупность аппаратных и программных средств и связей между ними.

3.1.35 конфигурирование: Установление конфигурации.

3.1.36 кэш (cache): Быстродействующая буферная память большой емкости, используемая для хранения копии областей оперативной памяти с наиболее частым доступом.

3.1.37 логическая служба MHP (MHP service): Служба, которая может быть выбрана через программный интерфейс приложения выбора службы или его функциональные эквиваленты. Логические службы MHP включают в себя службы DVB вещания и расширения, определяющие службы в будущих версиях этой спецификации.

3.1.38 лока́ль (locale): Набор параметров, включая набор символов, язык пользователя, страну, часовой пояс, а также другие предустановки, которые пользователь ожидает видеть в пользовательском интерфейсе.

3.1.39 локатор DVB-HTML (DVB-HTML locator): 1 Определитель местоположения. 2 В зависимости от формата приложения представляет собой ссылку, выраженную в синтаксисе [4], которая является однозначным указанием на документ DVB-HTML доступный для MHP в конкретном транспортном потоке. Определение этого термина действительно только в контексте DVB-HTML.

3.1.40 медиа (media): В контексте настоящего стандарта - информационные сообщения, передаваемые по каналам вещания и в сети Интернет (кадры звука MPEG, кадры изображения MPEG, кадры изображения JPEG, файлы текста, субтитры, загружаемые шрифты, графическая информация в формате PNG).

3.1.41 менеджер сеансов и ресурсов (Session and Resource Manager; SRM): Субсистема протокола системы команд и управления для средств цифровой записи (Digital Storage Media - Command and Control; DSM-CC), обеспечивающая централизованное управление сеансами DSM-CC и ресурсами одной или нескольких технологий сети.

3.1.42 метод (metod): Метод обработки информации в объектно-ориентированных языках.

3.1.43 методы класса (klass metods): Процедуры, описывающие поведение объектов.

3.1.44 многоцелевые расширения почты Интернет (Multipurpose Internet Mail Extensions; MIME): 1 Стандарт, описывающий передачу данных различных типов. 2 Спецификация для кодирования информации и форматирования сообщений.

3.1.45 навигатор (navigator): Резидентное приложение, обычно обеспечиваемое производителем, которое конечный пользователь может активировать в любое время. Навигатор может использоваться для выбора службы, приложения и инициирования взаимодействующих приложений.

3.1.46 настройка (tuning): Акт (событие) переключения между двумя транспортными потоками MPEG или мультиплексами.

Примечание – Переключение между двумя службами DVB, переносимыми в одном и том же транспортном потоке, не является настройкой.

3.1.47 обратный канал (return channel): Механизм передачи информации, который обеспечивает соединение между MHP и удаленным сервером.

3.1.48 объект (entity): Функциональный модуль в составе подсистемы (например, в состав подсистемы клиента входят объекты пользователь-сеть (П-С) и пользователь-пользователь (П-П)).

3.1.49 объектная модель документа (Document Object Mode; DOM): Программный Интерфейс, не зависящий от платформы и языка, позволяющий программам и скриптам получить доступ к содержимому HTML, XHTML и XML-документов, а также изменять содержимое, структуру и оформление таких документов.

3.1.50 объектно-ориентированное программирование (ООП): Принцип программирования, в котором программа рассматривается как совокупность дискретных объектов, представляющих собой самодостаточные совокупности структур данных и функций, взаимодействующих с другими объектами.

3.1.51 описание приложения (Application Description): Описание приложения содержит полную информацию о приложении, его параметрах, состоянии необходимом для активации и т.д. Описание приложения размещается в Таблице информации о приложениях (AIT).

3.1.52 переносимая сетевая графика (Portable Network Graphics, PNG): Формат файлов для растровых графических изображений.

3.1.53 плагин (plug-in): Совокупность функций, которые могут быть добавлены к универсальной платформе для того, чтобы обеспечить интерпретацию приложений и форматов контента.

3.1.54 пользователь (user): Оконечная система, которая может передавать или принимать информацию от других таких же оконечных систем с использованием сети и которая может функционировать как клиент, сервер или как клиент и сервер одновременно.

3.1.55 потоковый протокол реального времени (Real Time Streaming Protocol; RTSP): Прикладной протокол, предназначен для использования в системах, работающих с мультимедийными данными. Описан в [4].

3.1.56 приложение (application): 1 Программное обеспечение, предоставляющее клиенту возможность решения определенной задачи и реализуемое в среде клиента. 2 Функциональная реализация программного обеспечения, работающего в одном объекте или в нескольких взаимодействующих объектах. 3 Совокупность объектов, создающих среду для обработки информационных потоков на уровне приложений служб.

3.1.57 приложение DVB-HTML (DVB-HTML application): Совокупность документов, выбранных из семейства DVB-HTML элементов и форматов контента, определенных в. Границы множества документов определяются границами приложения.

3.1.58 приложение DVB-J (DVB-J application): Ряд (совокупность) классов DVB-J, которые функционируют совместно. Приложение DVB-J должно сообщать Администратору приложений о существовании этой копии приложения DVB-J для управления временем жизни копии через интерфейс жизненного цикла.

3.1.59 примитив (primitive): Программный модуль, выполняющий одну элементарную операцию, или блоки данных, передаваемые между разными уровнями системы для вызова каких-либо процедур.

3.1.60 программный интерфейс приложения (интерфейс прикладных программ, прикладной программный интерфейс) (Application Programming Interface; API): Интерфейс между приложением и отдельными функциями или ресурсами MHP, используется приложением для управления выполнением системных процедур.

3.1.61 программный поток данных (Program Stream; PS): Поток данных, образованный путем мультиплексирования элементарных потоков видеоданных и звукоданных цифрового вещательного телевидения, имеющих одну общую тактовую частоту, и сформированный из программных пакетов вещательного телевидения переменной длины.

3.1.62 протокол управления группами (пользователей) в сети Интернет (Internet Group Management Protocol; IGMP): Протокол многоадресной рассылки, управляет передачей пакетов между конечными пользователями и поддерживается протоколами IP в соответствии с [5].

3.1.63 профиль (profile): Описание группы минимальных конфигураций, определяющих часть спецификации, обеспечивающей возможности MHP. Профиль отображает функции, которые характеризуют контекст вариантов службы. Количество профилей ограничено. Отображение функций в ресурсы (возможности) и в аппаратные объекты не входит в контекст настоящего стандарта.

3.1.64 растровое изображение (bitmap image): Побитовое изображение, представляющее собой сетку пикселей или точек цветов (обычно прямоугольную) на компьютерном мониторе или на других отображающих устройствах.

3.1.65 регулярное выражение (regular expression): 1 Нотация для описания текстовых фрагментов (образов) в процедурах типа «найти» и «найти-и-заменить». 2 Система поиска текстовых фрагментов в электронных документах, основанная на специальной системе записи образцов для поиска. Правило поиска задает образец.

3.1.66 рекламные ссылки (Promotional links): Определяют механизм для ссылки на связанный материал в реальном времени. Предоставляют зрителю возможность заказывать просмотр контента, который в настоящее время просматривается.

3.1.67 ресурс (resource): Способность или качество системного объекта, которое может использоваться для создания вклада в реализацию службы (например, декодер стандарта MPEG, графическая система).

3.1.68 решение MHP (MHP solution): Решение, охватывающее набор технологий, необходимых, для реализации MHP, включая протоколы и программные интерфейсы приложений (API).

3.1.69 «сборщик мусора»: Специальный код, периодически освобождающий память. «Сборщик мусора» удаляет объекты, которые не будут востребованы приложением. Используется в объектно-ориентированном языке программирования Java, использующем механизм «сборки мусора».

3.1.70 сдвиг времени (timeshift): Режим одновременной записи и воспроизведения цифрового телевизионного контента, который позволяет приостановить процесс воспроизведения при продолжающейся записи и позволяет продолжить воспроизведение в более позднее время без потери содержания.

3.1.71 сеанс (session): Последовательность операций, при которой между пользователями сети устанавливается соединение, проводится обмен данными и завершается соединение.

3.1.72 секция (section): Синтаксическая структура, используемая для отображения всей сервисной информации в пакетах транспортного потока.

3.1.73 семантика (semantics): Система правил, предназначенная для определения смысловых значений отдельных конструкций алгоритмического языка.

3.1.74 сервер (server): Программный объект, экспортирующий ресурс имеющихся данных и устанавливаемый на физическое устройство, подключенное к сети и предоставляющее услуги другим устройствам, работающим в этой сети.

3.1.75 сервис (служба, услуга) (service): 1 Последовательность программ, которая под управлением вещателя может быть в режиме вещания передана как часть расписания. 2 Логический объект в системе предоставляемых функций и интерфейсов, поддерживающий одно или множество приложений, отличие которого от других объектов заключается в доступе конечного пользователя к управлению шлюзом сервисов.

3.1.76 сервис MHP (MHP service): Логическая служба MHP, которую можно выбрать с помощью API сервиса выбора или его функциональных эквивалентов. Логическая служба MHP включает в себя вещание служб DVB, службу хранения и службу приложений MHP, выполняемые в соответствии с файлом AIT, загруженные через интерактивный канал.

3.1.77 сеть (network): Совокупность элементов, поддерживающих связь, обеспечивающая соединение элементов, управление сеансом связи и/или управление подключением пользователя.

3.1.78 сеть DVB (DVB network): Набор мультиплексов транспортных потоков MPEG-2, переданных по единственной системе доставки (например, все цифровые каналы в конкретной кабельной системе).

3.1.79 служба (service) Последовательность программ управляемых вещателем, которую можно вещать в рамках графика.

3.1.80 синтаксис (syntax): Часть языка программирования, которая описывает структуру программ как набор символов.

3.1.81 слушатель события (event-listener): Интерфейс «слушатель», который «прослушивает» события, происходящие на объекте, для того чтобы отследить возникновение события и обработать его.

3.1.82 событие (event): Действие или ситуация, в ряде случаев возбуждаемые пользователем, на которые программа должна отреагировать.

3.1.83 совместимый плагин (interoperable plug-in): Плагин, который требует только стандартных интерфейсов API MHP.

3.1.84 состояния приложения DVB-HTML (DVB-HTML application states): Логические состояния, в которых может находиться агент DVB-HTML.

3.1.85 специфические протоколы службы (Service Specific): Протоколы, обеспечивающие регистрацию в MHP новых протоколов вещания.

3.1.86 стиль (style): Набор правил форматирования, который применяется к элементу документа, чтобы быстро изменить его внешний вид.

3.1.87 структура классов в ООП (Framework): Повторно используемая базовая структура, состоящая в общем случае из абстрактных и конкретных классов.

3.1.88 ссылка на программные часы (Program Clock Reference; PCR): Тридцати-трехбитовое число, оцениваемое в периодах частоты 90 кГц, вводимое на программном уровне индивидуально для каждой передаваемой телевизионной программы.

3.1.89 субсистема (subsystem): Единица логического «оборудования» в пределах DSM-CC системы (например, клиент, сервер или менеджер сеансов и ресурсов).

3.1.90 суффикс (suffix): Логический знак (символ, слово), обозначающий конец сообщения.

3.1.91 таблица информации приложений (Application Information Table; AIT): Таблица, обеспечивающая полную информацию о вещании данных и о необходимых операциях для активизации приложений.

3.1.92 таблица описания служб (Service Description Table; SDT): Таблица, описывающая службы, передаваемые в конкретном транспортном потоке.

3.1.93 тег (tag): Служебный элемент, который размещен в начале заголовка, хранится вместе с данными и не может быть использован как самостоятельный элемент.

3.1.94 тело (body): Набор операторов внутри некоторой структуры (например, тело цикла, тело процедуры).

3.1.95 терминал MHP (MHP terminal): Часть физического оборудования, соответствующего требованиям стандарта MHP, содержит виртуальную машину и экземпляр программного интерфейса приложений MHP.

3.1.96 транспорт (передача, транспортировка) (transport): Передача информации между различными объектами транспортного уровня, при котором гарантируется заданная степень надежности связи.

3.1.97 транспортный поток; **ТП** (transport stream; TS): Набор из нескольких программных потоков данных цифрового вещательного телевидения, сформированный из программных пакетов постоянной длины с коррекцией ошибок и независимым тактированием от своих источников синхронизации.

3.1.98 триггер (trigger): Событие, которое может вызвать изменение в поведении того приложения DVB-HTML, которое зарегистрировало интерес к такому событию. Триггеры могут приходить из потоков вещания, могут быть сгенерированы от других источников (таких, как системные часы) или могут быть сгенерированы в результате действий пользователя. Триггер может включать ссылку на всемирное координированное время (Universal Time Coordinated; UTC) относительно некоторого другого события, относительно нормального времени воспроизведения (Normal Play Time, NPT) потока медиа.

3.1.99 формат графического обмена (Graphics Interchange Format; GIF): Файловый формат 8-битной растровой графики, используется для передачи растровых графических изображений.

3.1.100 хэш-код (hash code): Битовая строка фиксированной длины, полученная из массива произвольной длины использованием хэш функции или функции свертки.

3.1.101 шлюз сервиса (Service Gateway): Интерфейс, предоставляющий клиенту каталог услуг и возможность подключаться к домену сервиса.

3.1.102 **экземпляр** (instance): Конкретный объект описанного класса.

3.1.103 экземпляр класса: Объект, типом которого является некоторый класс.

3.1.104 экземпляр приложения (application instance): Уникальный вызов приложения. Запуск того же самого приложения дважды дает два различных экземпляра приложений.

3.1.105 юникод (Unicode): Стандарт кодирования символов, представляющих знаки письменных языков.

3.1.106 язык разметки (Mark-up language): Набор кодов в текстовом файле, указывающий компьютеру правила формирования файла при выводе на принтер или дисплей или правила индексирования и связывания его компонентов.

3.1.107 язык спецификаций и описаний (Specification and Description Language; SDL): Язык, использующий графическое исполнение описания поведения системы. Применение SDL определено [6].

3.1.108 язык XML (расширяемый язык разметки) (Extensible Mark-up Language; XML): Язык прикладного программного обеспечения, используется для создания веб-страниц.

3.1.109 язык разметки гипертекста (язык HTML) (Hyper Text Mark-up Language; HTML): Стандартный язык описания гипертекстовых документов, используемых в сети Интернет. Язык HTML поддерживает функции форматирования текста, параметров шрифта, включения видео и звука, запуска программ поиска, вывода на экран документов и создания гиперссылок.

3.1.110 DVB-HTML (Digital Video Broadcast - HyperText Mark-up Language): Технология, обеспечивающая доступ телевизионных приемников DVB к контенту в сети Интернет с использованием языка HTML.

3.1.111 HTML-3.2: Версия языка HTML, обеспечивающая расширение свойств языка применением таблиц, аплетов, размещением текста вокруг изображения.

3.1.112 RGB (Red, Green, Blue; Красный, зеленый, синий): Модель цвета, основанная на аддитивном смешивании цветов.

3.1.113 STB (Set Top Box): Модуль, включающий в себя модули Set Top Unit (STU) и Network Interface Unit (NIU). STB может иметь интегрированное или модульное исполнение. Интегрированный STB предназначен для подключения к единственному интерфейсу DAVIC A1 или к эквивалентному интерфейсу. Модульный STB может иметь интерфейс DAVIC A0 в соответствии с [7] (приложение I, 2.2), [8] (приложение B, раздел 4) или его эквивалент и позволяет подключение нескольких NIU.

3.1.114 STU (Set Top Unit): Модуль, содержащий «сеть приставок независимых» функциональностей для STB. Типовой STU обеспечивает следующие функциональные возможности: обработка и функции памяти, демультиплексор MPEG-2 и аудио- видео декодеры, графический дисплей, модулятор ТВ сигнала, интерфейсы периферийных устройств.

3.1.115 TimeShift: Режим одновременной записи и воспроизведения цифрового контента телевидения, при котором процесс воспроизведения может быть приостановлен при продолжающейся записи, что позволяет продолжить воспроизведение в более позднее время при полном сохранении контента.

3.1.116 trick: Режим воспроизведения записи PVR.

3.1.117 Usenet: Всемирная сеть системы Unix, имеющая централизованное управление и используемая телеконференциями в качестве системы электронных досок.

3.1.118 UTF-8 (Unicode Transformation – 8 bits): Формат преобразования юникода, реализующий представление юникода, совместимое с 8-битным кодированием текста.

3.1.119 UTF-16 (Unicode Transformation – 16 bits): Формат преобразования юникода, реализующий представление юникода, совместимое с 16-битным кодированием текста.

3.1.120 URI (Uniform Resource Identifiers): Унифицированный идентификатор ресурса. Символьная строка, предназначенная для идентификации ресурса по его типу и расположению из любой точки Интернета. Включает унифицированные имена ресурсов (Uniform Resource Name, URN) и URL.

1.2 В настоящем стандарте применены следующие сокращения:

МСЭ – Международный союз электросвязи (International Telecommunication Union);

МЭК – Международная электротехническая комиссия (International Electrotechnical Commission);

ООП – объектно-ориентированное программирование;

ОС – операционная система (Operating System);

П-П (User-to-User) – пользователь-пользователь;

П-С (User-to-Network) – пользователь-сеть;

ТП – транспортный поток (цифрового вещательного телевидения) (transport stream);

AIT – таблица информации о приложении (Application Information Table);

API – программный интерфейс приложения (интерфейс прикладных программ, прикладной программный интерфейс) (Application Programming Interface);

AWT – оконная библиотека графического интерфейса (Abstract Windowing Toolkit);

BAT – таблица объединения букета (программ) (Bouquet Association Table);

BIOP – протокол взаимодействия брокеров (посредников) запросов к объектам вещания (Broadcast Inter-ORB Protocol);

CA – условный доступ (Conditional Access);

CORBA – архитектура брокера общих объектных запросов (Common Object Request Broker Architecture);

CRID – идентификатор ссылок контента (Content Reference IDentifier);

CRL – список отозванных сертификатов (Certificate Revocation Lists);

CSS – каскадные таблицы [9] (Cascading Style Sheets);

DAVIC – совет по аудио-визуальным проектам (DAVIC Digital Audio Visual Council);

DII – сообщение индикации информации загрузки DownloadInfoIndication в Карусели Объектов DSM-CC (DownloadInfoIndication);

DNS – система доменных имен (сервер доменных имен) (Domain Name System);

DOM – объектная модель документа (Document Object Mode);

DSM – цифровая запоминающая среда (Digital Storage Media);

DSM-CC – система команд и управления для средств цифровой записи (Digital Storage Media – Command and Control);

DSM-CC U-U – набор протоколов DSM-CC передачи от пользователя к пользователю (DSM-CC User to User);

DTD – описание типа документа (Document Type Definition);

DVB – цифровое телевизионное вещание (Digital Video Broadcasting);

DVB-IPTV – цифровое телевизионное вещание по каналам с IP протоколами;

DVB-J – платформа Java, являющаяся частью спецификации MHP;

DVB-J API – один из программных интерфейсов приложений Java, стандартизированных как часть спецификации MHP;

DVB-SI – информация о службах DVB (Digital Video Broadcasting - Service Information);

DVB-HTML – стандарт, предоставляющий цифровому телевидению доступ к веб контенту (Digital Video Broadcast - HyperText Mark-up Language);

DVB-T – стандарт наземного цифрового телевизионного вещания (Digital Video Broadcasting-Terrestrial);

DVB-T2 – стандарт наземного цифрового телевизионного вещания второго поколения (Digital Video Broadcasting-Terrestrial 2);

DVB-C – стандарт кабельного цифрового телевизионного вещания (Digital Video Broadcasting-Cable);

DVB-C2 – стандарт кабельного цифрового телевизионного вещания второго поколения (Digital Video Broadcasting-Cable 2);

DVB-S – стандарт спутникого цифрового телевизионного вещания (Digital Video Broadcasting-Satellite);

DVB-S2 – стандарт спутникого цифрового телевизионного вещания второго поколения (Digital Video Broadcasting-Satellite 2);

ECMAScript – объектно-ориентированный язык сценариев. Стандартизирован международной организацией ECMA [10];

EIT – таблица информации о событиях (Event Information Table);

EPG – электронный путеводитель (гид) по программам (Electronic Program Guide);

ETSI – Европейский Институт по стандартизации в области телекоммуникаций (European Telecommunications Standards Institute);

GIF – формат обмена графическими изображениями (Graphics Interchange Format);

GUI – графический интерфейс пользователя (Graphical User Interface);

HAVi – открытый стандарт аудио-, видеооборудования для дома (Home Audio Video Interoperability);

HTML – язык разметки гипертекста, язык HTML (Hyper Text Mark-up Language);

HTML-3.2 – версия языка HTML;

HTTP – протокол передачи гипертекстовых файлов (Hyper Text Transfer Protocol);

IDL – специальный язык для определения интерфейсов (Interface Definition Language);

IEC – Международная электротехническая комиссия (International Electrotechnical Commission/ Committee);

IETF – техническая комиссия Интернет, разрабатывающая документы RFC (Internet Engineering Tack Force);

IGMP – протокол управления группами (пользователей) в сети Интернет (Internet Group Management Protocol);

img – форма представления графических данных на устройствах вывода (Image);

IIOP – межброкерный протокол для Интернет, является составной частью архитектуры CORBA (Internet Inter-ORB Protocol);

INT – таблица нотификации (извещений) IP протокола (IP/MAC Notification Table);

IOR – ссылка на интероперабельный объект (Interoperable object reference);

IP – Интернет протокол (Internet Protocol);

ISO – Международная организация по стандартизации (International Organization for Standardization);

ITU – Международный союз электросвязи (International Telecommunications Union);

ITU-T – Сектор стандартизации электросвязи МСЭ (International Telecommunications Union – Telecommunication Standardization Sector);

JFIF – формат обмена файлами JPEG (JPEG File Interchange Format);

JDOM – Java-реализация DOM для XML, созданная с учетом особенностей языка и платформы Java (Java Document Object Model). JDOM интегрируется с Document Object Model (DOM) и Simple API for XML (SAX);

JMF – универсальная библиотека для работы с аудио- и видеоданными, является стандартным расширением платформы Java 2.JMF (Java Media Framework);

JPEG – группа экспертов по кодированию фотографических изображений (Joint Picture Expert Group);

JRE – исполняющая система Java (Java Runtime Environment);

JSR – дословно «запрос на спецификацию Java» (Java Specification Request);

JVM – виртуальная машина Java (Java Virtual Machine);

MHP – домашняя мультимедийная платформа (Multimedia Home Platform);

MIME – многоцелевые расширения почты Интернет (Multipurpose Internet Mail Extensions);

MMI – интерфейс человек-машина (Man Machine Interface);

MPEG – группа экспертов по движущимся изображениям (Motion Pictures Expert Group);

MPEG-2 – стандарт цифрового сжатия изображения и звука;

NIU – модуль интерфейса сети (Network Interface Unit);

NPT – нормальное время воспроизведения (Normal Play Time);

OC – Карусель Объектов (Object Carousel);

OCAP – спецификации платформы приложений OpenCable лабораторий кабельного телевидения (OpenCable Application Platform);

OS – операционная система (Operating System);

PCR – ссылка на программные часы (Program Clock Reference);

PDR – персональный цифровой магнитофон (см. PVR) (Personal Digital Recorder);

PFR0 – переносимый ресурс шрифта, версия 0 (Portable Font Resource version 0);

PID – идентификатор типа пакета (Packet Identifier);

PMT – таблицы состава программы (Program Map Table);

PNG – переносимая сетевая графика; формат файлов для растровых графических изображений (Portable Network Graphics);

PS – программный поток данных (Program Stream);

PSI – программно-зависимая информация (Program Specific Information);

PVR – персональный видео рекордер (магнитофон) (Personal Video Recorder);

RCMM – сообщения управления корневым сертификатом (Root Certificate Management Messages);

RFC – предложения для обсуждения, серия нормативных документов, стандартизирующих протоколы Интернет (Request for Comments);

RGB – красный, зеленый, синий (Red, Green, Blue);

RPC – вызов удаленных процедур (Remote Procedure Call);

RCE – расширение кода во время выполнения (Runtime Code Extension);

RST – таблица состояния событий (Running Status Table);

RTSP – потоковый протокол реального времени (Real Time Streaming Protocol);

SAX – способ последовательного чтения/записи XML-файлов (Simple API for XML);

SDL – язык спецификаций и описаний (Specification and Description Language);

SDT – таблица описания служб (Service Description Table);

SI – информация о службах (Service Information);

SRM – менеджер сеансов и ресурсов (Session and Resource Manager);

STB – модуль, включающий в себя модули Set Top Unit (STU) и NIU (Set Top Box);

STU – модуль, содержащий «сеть приставок независимых» функциональностей для STB (Set Top Unit);

SSL – уровень защищенных сокетов (Secure Sockets Layer);

TCP – протокол управления передачей (из стека протоколов TCP/IP) (Transmission Control Protocol);

TCP/IP – стек протоколов сетевого и транспортного уровня;

TDT – таблица времени и даты (Time and Date Table);

TLS – протокол безопасности транспортного уровня (Transport Layer Security);

TOT – таблица смещения времени (Time Offset Table);

TS – транспортный поток (цифрового вещательного телевидения) (Transport Stream);

UDP – протокол передачи дейтаграмм (User Datagram Protocol);

U-N – пользователь-сеть (User-to-Network);

UNO-CDR – универсальный объект сетевой структуры – общее представление данных (Universal Networked Object - Common Data Representation);

URI – унифицированный идентификатор ресурса (Uniform Resource Identifiers);

URL – унифицированный указатель (локатор) (определитель расположения) ресурса (Uniform Resource Locator);

Usenet – всемирная сеть системы Unix;

UTC – всеми́рное координи́рванное время (Universal Time Coordinated);

UTF – формат преобразования Юникода (Unicode Transformation Format);

U-U – пользователь-пользователь (User-to-User);

World Wide Web – глобальная распределенная система гипермедиа, размещенная в сети Интернет;

XAIT – кодирование XML для AIT (XML encoding for AIT);

Xlet – интерфейс, который используется для управления жизненным циклом приложения DVB-J;

XML – расширяемый язык разметки (язык XML) (Extensible Mark-up Language).

4 Классы домашней мультимедийной платформы

Платформа MHP классифицируется по следующим видам представляемых сервисов:

- класс 1.0 – обеспечивается поддержка вещательных приложений и передачи данных через сети с IP протоколом;
- класс 1.1 – дополнительно к возможностям класса 1.0 обеспечивается обработка приложений с сохранением контента на устройстве пользователя, обработка приложения через каналы с IP протоколом, поддержка смарт-карт, поддержка доступа в Интернет.

В MHP класса 1.1 применяются три базовых профиля:

- Enhanced Broadcast Profile (расширенный профиль вещания) – вся информация поступает от провайдера цифрового телевизионного вещания;
- Interactive Broadcast Profile (интерактивный профиль вещания) – предполагает наличие обратного канала через IP-соединение, что дает возможность подключаться к удаленным серверам;
- Internet Access Profile (профиль доступа в Интернет).

В MHP класса 1.2 в дополнение к трем профилям MHP класса 1.1 применяется профиль IPTV.

5 Базовая архитектура домашней мультимедийной платформы

Базовая архитектура MHP в соответствии с ГОСТ Р 54456, 5.1 характеризуется:
- контекстом применения MHP;
- архитектурой MHP;
- интерфейсами MHP;
- использованием плагинов.

5.1 Контекст применения MHP

Контекст применения MHP соответствует требованиям ГОСТ Р 54456 (5.1.2) и [11] (5.1).

5.2 Архитектура MHP

Архитектура MHP соответствует требованиям ГОСТ Р 54456 (5.1.3) и [11] (5.2).

5.3 Интерфейсы между приложениями MHP и системой MHP

Интерфейсы между приложениями MHP и системой MHP должны быть в соответствии с ГОСТ Р 54456 (5.1.3) и [11] (5.3).

5.4 Плагины

Параметры плагинов должны быть в соответствии с ГОСТ Р 54456 (5.2.2) и [11] (5.4).

5.5 Место профиля IPTV в предыдущих версиях MHP

На рисунке 1 представлена структура профилей и опций MHP, позволяющая оценить связи между версиями MHP по отношению к профилю IPTV.

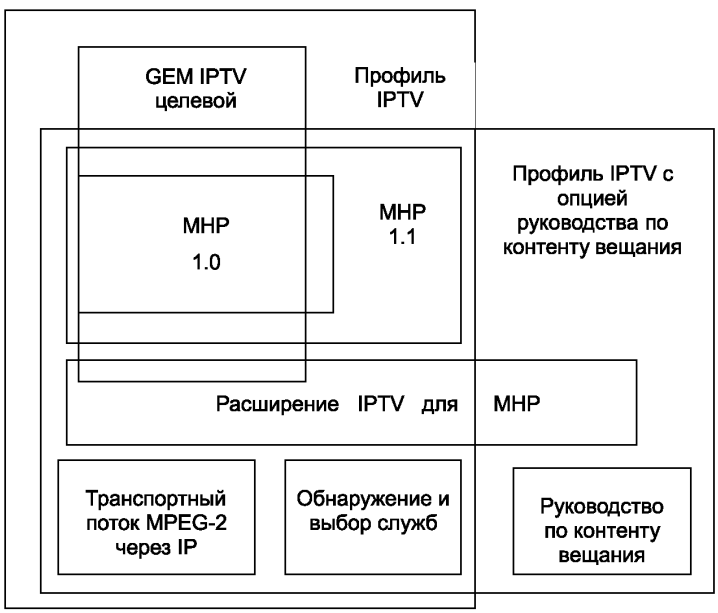


Рисунок 1 – Структура профилей и опций

Профиль IPTV определяет интерфейсы между терминалом MHP и соединенной с ним сетью. Протоколы, необходимые для IPTV, определяются [1].

Профиль IPTV может быть расширен. Одна из опций расширения предусматривает передачу руководства широкополосного контента в соответствии с [2]. Так как руководство широкополосного контента основано на технологии TV-Anytime, то оно включает в себя элементы API TV-Anytime и определяется как часть спецификации MHP/PVR в соответствии с [12].

Указанные протоколы DVB применимы не для всех рынков сбыта MHP. Некоторые рынки могут использовать собственные протоколы и, возможно, должны иметь приложения MHP/IPTV сосуществующие с этими протоколами. Некоторые рынки не могут и не будут использовать концепции DVB-SI, которые лежат в основе протоколов DVB IPTV. Целевое IPTV GEM определяет профиль IPTV, который может быть использован в этих обоих вариантах и определяется [11].

На рисунке 1 не показан вариант, являющийся привилегированным вариантом приложения. Он не связан с конкретными деталями IPTV. Этот вариант включает в себя функции, которые могут быть целесообразны в терминалах MHP, приобретенных при субсидировании провайдером служб. Такие терминалы MHP при необходимости должны содержать собственные реализации необходимых функций.

Поддержка привилегированных приложений терминалом MHP близка поддержке, смоделированной в [13]. Обзор возможностей поддержки представлен в [13]. Настоящий стандарт предусматривает некоторые (но не все) из этих возможностей и устанавливает новые возможности.

6 Параметры транспортных протоколов

MHP поддерживает следующие транспортные протоколы:

- транспортные протоколы канала вещания в соответствии с [15] (6.1), ГОСТ Р 54456 (6.1);
- транспортные протоколы интерактивных каналов в соответствии с [15] (6.3), ГОСТ Р 54456 (6.2);
- транспортные протоколы для загрузки приложений через интерактивный канал в соответствии с [15] (6.4).

6.1 Параметры транспортных протоколов каналов вещания и интерактивных каналов

6.1.1 Транспортный поток MPEG-2

Параметры транспортного потока MPEG-2 должны быть в соответствии с [16].

6.1.2 Секции транспортного потока MPEG-2

Параметры секций транспортного потока MPEG-2 должны быть в соответствии с [16] (2.4) и должны базироваться на протоколе транспортного потока, представленного в 6.1.1 настоящего стандарта.

6.1.3 Протокол передачи частных данных DSM-CC

Параметры протокола передачи частных данных DSM-CC должны быть в соответствии с [17] и ГОСТ Р 54456 (6.1.4).

6.1.4 Протокол Карусель Данных DSM-CC

Параметры протокола Карусель Данных DSM-CC должны быть в соответствии с [17] и ГОСТ Р 54456 (6.1.5).

6.1.5 Протокол Карусель Объектов DSM-CC U-U

Требования к протоколу Карусель Объектов U-U DSM-CC должны обеспечивать реализацию протоколов Карусели Объекта User-to-User DSM-CC и быть в соответствии с [17] и ГОСТ Р 54456 (6.1.6), с ограничениями и расширениями, определенными [14] (приложение В), [18], [19]. Должны учитываться требования [11] (6.2.5).

6.1.6 Параметры многопротокольной инкапсуляции

Протокол многоадресного доставки по каналу вещания в соответствии с [18] обеспечивает поддержку IP и основан на протоколе частных данных DSM-CC. Стандарт [18] определяет только параметры поддержки переноса многоадресного IP в MHP и не поддерживает перенос одноадресных IP.

6.1.7 Протокол Интернет (IP)

Требования к протоколу Интернет должны быть в соответствии с [11] (6.2.7) и [20].

6.1.8 Протокол дейтаграмм пользователя (UDP)

Требования к протоколу дейтаграмм пользователя должны быть в соответствии с [11] (6.2.8), ГОСТ Р 54456 (6.1.9) и [21].

6.1.9 Передача информации о службах

Требования к передаче информации о службах (SI) должны быть в соответствии с [22], [23] и ГОСТ Р 54456 (6.1.10).

6.1.10 Сигнализация IP

Требования к сигнализации IP должны быть в соответствии с требованиями к таблице INT (IP/MAC Notification Table) по [18] и ГОСТ Р 54456 (6.1.11).

6.2 Транспортные протоколы загрузки приложений через интерактивный канал

6.2.1 Вводная часть

Пункт содержит сведения о протоколах интерактивного канала. Протоколы интерактивных каналов относятся к типу протоколов, независимых от сети. Протоколы интерактивных каналов, доступные приложениям MHP, должны соответствовать перечню профилей MHP по [14] (раздел 15, таблица 65). Структура стека протоколов интерактивного канала показана на рисунке 2.

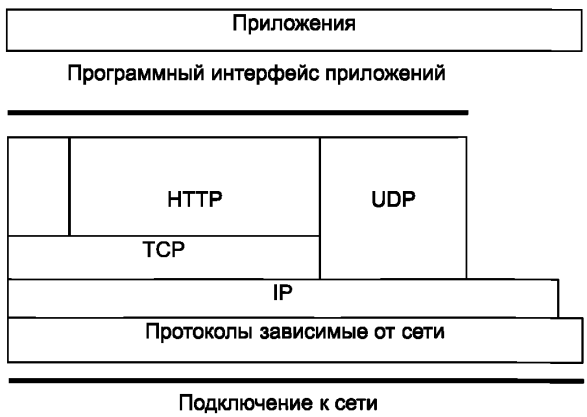


Рисунок 2 – Стек протоколов интерактивного канала

Состав и характеристики программных интерфейсов приложения (API), обеспечивающих эти протоколы, должны быть в соответствии с разделом 11 настоящего стандарта.

6.2.2 Протоколы, зависимые от сети

Состав протоколов, зависимых от сети, и их параметры должны быть в соответствии с ГОСТ Р 54456:
- для распределительных систем кабельного телевидения в соответствии с [24] (4.1; 5.2 - 5.5);
- для интерактивных каналов ТФОП и ЦСИС в соответствии с [25] (разделы 5, 6);
- для интерактивных каналов DECT в соответствии с [26] (разделы 4, 5);
- для интерактивных каналов GSM в соответствии с [27] (разделы 4, 5);
- для интерактивных каналов системы LMDS в соответствии с [28] (разделы 4 - 6);
- для интерактивных каналов распределительной системы SMATV в соответствии с [29] (разделы 4 - 6).
- для интерактивных каналов спутниковых распределительных систем в соответствии с [30] (разделы 4 - 9), [21].

6.2.3 Интернет протокол (IP)

Параметры Интернет протокола (IP) должны быть в соответствии с ГОСТ Р 54456 (6.2.3) и [20].

6.2.4 Протокол управления передачей (TCP)

Параметры протокола управления передачей (TCP) должны быть в соответствии с ГОСТ Р 54456 (6.2.4) и [31] (разделы 2, 3).

6.2.5 Протокол вызова удаленных процедур

Протокол вызова удаленных процедур (UNO-RPC) должен выполняться в соответствии с ГОСТ Р 54456 (6.2.5) при использовании протокола IIOP. Параметры протокола UNO-RPC должны быть в соответствии с [32] (2.1).

6.2.6 Протокол UNO-CDR

Параметры протокола UNO-CDR в соответствии с ГОСТ Р 5445 (6.2.6) должны быть в соответствии с [32].

6.2.7 Протокол U-U DSM-CC

Параметры протокола U-U DSM-CC (U-U) должны быть в соответствии с ГОСТ Р 53528 и [17]. Ограничения требований и расширения требований к протоколу DSM-CC (U-U) должны быть в соответствии с [18] (раздел 11) и [19] (4.7).

6.2.8 Протокол передачи гипертекстовых файлов (HTTP)

6.2.8.1 Протокол HTTP 1.1

Параметры протокола HTTP 1.1 должны быть в соответствии с ГОСТ Р 54456 (6.2.8) и [33]. Протокол HTTP 1.1 должен поддерживать все профили МНР.

6.2.8.2 Протокол HTTP 1.0

Терминалы МНР, поддерживающие протокол HTTP 1.0, определенный в соответствии с [34] должны обеспечивать устойчивые (постоянные) соединения. Допускается поддержка терминалами МНР других версий протокола HTTP, например HTTP 1.1, если они имеют обратную совместимость.

6.2.8.2.1 Постоянные соединения по протоколу HTTP 1.0

При подключении к серверу терминал МНР должен отправить маркер проверки работоспособности соединения Keep-Alive:

Connection: Keep-Alive.

Сервер HTTP (1.0 или 1.1) отвечает маркером соединения Keep-Alive и клиент может начать постоянное соединение по протоколу HTTP 1.0 (Keep-Alive).

6.2.8.2.2 Заголовок проверки работоспособности (Keep-Alive)

Заголовок проверки работоспособности (Keep-Alive) пересылается после передачи маркера запроса или ответа. Передача заголовка проверки работоспособности не обязательна, она выполняется при необходимости передачи параметра. Настоящий стандарт не устанавливает требований к передаваемым параметрам. Заголовок проверки работоспособности должен быть проигнорирован, если он получен при отсутствии маркера соединения. Формат заголовка проверки работоспособности должен быть в соответствии с [11] (6.3.7.2.2):

Keep-Alive-header = «Keep-Alive» «:» 0# keepalive-param

keepalive-param = param-name «=» value

Запись «0#» означает, что поле «keepalive-param» «может повторяться несколько раз. Если поле повторяется более одного раза, то эти поля разделяются запятой».

6.2.8.2.3 МНР и прокси-серверы

В соответствии с [11] (6.3.7.2.3) передача маркера соединения Keep-Alive на прокси-сервер HTTP 1.0 не допускается.

6.2.8.2.4 Совместимость версий HTTP 1.0 и HTTP 1.1

Совместимость версий протоколов HTTP 1.0 и HTTP 1.1 при установлении постоянных соединений должна обеспечиваться в соответствии с [11] (6.3.7.2.4).

6.2.8.3 Протокол HTTPS

Параметры протокола HTTPS должны быть в соответствии с [35].

6.2.9 Протокол дейтаграмм пользователя (UDP)

Параметры протокола дейтаграмм пользователя должны быть в соответствии с [21].

6.2.10 Протоколы системы доменных имен (DNS)

Терминалы МНР должны использовать протоколы DNS, обеспечивающие преобразование полных доменных имен в IP адреса в соответствии [36], [37] и уточнениями по [38], [39].

6.2.11 Дополнительные транспортные протоколы

Дополнительные транспортные протоколы, которые являются собственностью провайдеров служб или специфичны для провайдеров служб, могут поддерживаться путем регистрации провайдеров служб интерактивных каналов.

6.3 Транспортные протоколы для загрузки приложений из интерактивного канала

Загрузка приложений из интерактивного канала может выполняться по одному из трех сценариев:

- файловая система полностью реализована в канале вещания (классическая модель MHP 1,0);
- файловая система реализована через интерактивный канал;
- файловая система реализована в гибридном варианте потока вещания и интерактивного канала.

6.3.1 Файловая система, реализующаяся через интерактивный канал

В случае, когда интерактивный канал предоставляет единственную файловую систему, ID протокола должен быть 0x0003. Терминалы MHP могут использовать другие значения ID этого протокола.

6.3.1.1 Логическая структура файловой системы

Список элементов (URL), сигнализированных в дескрипторе (дескрипторах) транспортного протокола согласно [11]. (15.6.1.2.1) позволяет формировать единственное пространство имен.

Терминал MHP при установлении местоположения файла, определенного неполным (относительно неполным) именем файла, должен пытаться выбирать файл от каждого из элементов в этом списке в том порядке, в котором они найдены в списке до тех пор, пока файл не будет найден или пока не будет исчерпан список.

Элементы в списке согласно [11] (6.4.1.1) должны быть или ссылками на файлы с расширением zip в соответствии с [40] или на базовые URL, заканчивающиеся на «/», с которыми должен быть связан путь к требуемому файлу. Должны игнорироваться любые элементы в списке, не относящиеся к этим двум типам. Ошибки обнаружения конкретного файла через конкретный элемент списка также должны игнорироваться. Список должен содержать хотя бы один элемент.

Пример процедуры извлечения файла «dvb.fontindex» для приложения представлен в [11] (6.4.1.1).

6.3.1.2 Передача файлов

Файлы должны передаваться с URL HTTP использованием протокола HTTP 1.1 в соответствии с 6.2.8.1 настоящего стандарта или использованием протокола HTTP 1.0 в соответствии с 6.2.8.2 настоящего стандарта.

Файлы, передаваемые с URL HTTPS, используют протокол HTTPS в соответствии с 6.2.8.3 настоящего стандарта.

Для других схем URL, если зарегистрированный транспортный протокол интерактивного канала поставщика служб существует для соответствующей схемы URI, он должен использоваться этим поставщиком.

6.3.1.3 Кодирование класса

Классы приложений DVB-J могут поставляться в виде отдельных файлов класса или как файлы ZIP с [40].

Примечание – Этот случай отличается от случая вещания, где объектные файлы Карусели являются файлами класса.

Классы приложений DVB-J должны быть доставлены в виде отдельных файлов класса на уровне файловой системы.

Примечание – Это означает, что файлы класса DVB-J на сервере HTTP могут быть любыми отдельными файлами класса или файлами ZIP с [40], где файлы ZIP являются частью полной файловой системы.

6.3.1.4 Содержание каталога в файловой системе

Данная файловая система при решении задач аутентификации не считается поддерживающей каталоги в соответствии с правилами аутентификации по [11] (12.4.1.5).

В случае использования каталога в соответствии с [11] (14.7) для терминала MHP не гарантируется доступ к полному содержанию каталога.

6.3.2 Параметры гибридного варианта файловой системы вещательного потока и интерактивного канала

В гибридном варианте вся информация каталога предоставляется в вещательном потоке, но часть или все содержимое файла предоставляется через интерактивный канал.

6.3.2.1 Передача файлов

6.3.2.1.1 Передача файла по каналу вещания

В терминалах MHP, использующих Карусель Объектов U -U DSM-CC, содержание файла переносится при использовании протокола BIOP: File как нормальный случай Карусели Объектов U-U, определенной в ГОСТ Р 54456 (6.1.6) или [11] (6.2.5). Ссылка на интероперабельный объект (IOR) использует тело BIOPProfileBody или тело LiteOptionsProfileBody.

6.3.2.1.2 Доставка файла через интерактивный канал

В случае доставки файла через интерактивный канал ссылка IOR для идентификации расположения контента файла в интерактивном канале от привязки файла до наполнения файла использует тело HTTPProfileBody. Эта форма IOR должна использоваться только для объектов BIOP: File.

Для профиля HTTP 1.0 получение (поиск) содержания файла выполняется в соответствии с [11] (6.3.7.2). В таблице 1 представлен синтаксис тела профиля HTTP.

Таблица 1 – Синтаксис тела профиля HTTP

Синтаксис	Количество битов	Типы	Значение	Комментарий
HTTPProfileBody {				
profileId_tag	32	uimsbf	0x44564200	
profile_data_length	32	uimsbf	*	
profile_data_byte_order	8	uimsbf	0x00	
version.major	8	uimsbf	0x01	
version.minor	8	uimsbf	0x00	С прямым
host_data_length	8	uimsbf	N1	порядком битов
for (k=0; k <N1; k ++) {				Старшая версия
host_data	8	uimsbf	+	протокола 1
}				Младшая версия
port	16	uimsbf		протокола 0
objectKey_length	16	uimsbf	N2	
for (k=0; k <N2; k ++) {				
objectKey_data	8	uimsbf	+	
}				
}				

Параметры семантики полей:

version: Поле 8 битов указывает на версию протокола, который сервер будет использовать для доставки определенного файла. Значение версии 1.0 указывает, что транспортный протокол определяется в соответствии с [11] (6.3.7.2).

host_data: Эти байты переносят идентификатор Интернет-узла, которому могут быть отправлены сообщения. Это может быть полностью квалифицированное доменное имя или стандартная форма Интернет «десятичное представление с точками» (например, «192.231.79.52»).

port: Поле 16 битов указывает номер порта TCP/IP в указанном узле, где целевой агент прослушивает запросы.

objectKey_data: Эти байты формируют строку, которая переносит часть URL path_segments, которая уникально идентифицирует объект на сервере в соответствии с [4].

6.3.2.1.3 Тело профиля HTTPProfileBody

Тело профиля HTTP определяет узел, порт и path_segments. В запросах HTTP они связаны для формирования URL вида: gost.e-ecolog.ru

6.4 Протоколы IPTV

Терминалы MHP должны поддерживать протоколы, указанные в следующих подразделах.

Примечание – протоколы MHP-IPTV могут опираться на интерфейсы провайдера, определенные в 9.11 настоящего стандарта.

6.4.1 Транспортные протоколы

6.4.1.1 Служба обнаружения и селекции

Протоколы для одноадресной и многоадресной передачи открытия службы и отбора информации определены в [1] (5.4).

6.4.1.2 Протоколы для доставки руководства по контенту широкополосного доступа

Протоколы для доставки руководства по контенту однонаправленного и вещательного широкополосного доступа определяются в [2] (раздел 4).

6.4.1.3 Протокол RTP

Правила использования протокола RTP определяются в [1] (раздел 7), [11].

6.4.1.4 Протокол RTSP

Правила использования протокола RTSP определяются в [1] (раздел 6).

6.4.2 Информация о службах и протоколы метаданных

6.4.2.1 Обнаружение IP служб

Должны использоваться механизмы определения провайдеров служб и IP служб в соответствии с [1] (5.2) за исключением [1] (5.2.6.6).

Терминалы MHP должны выполнять мониторинг и выявление изменений в сигнализации информация о службах и протоколах метаданных один раз в день (не чаще).

6.4.2.2 Руководство по контенту широкополосного доступа

Параметры руководства по контенту широкополосного доступа определяется в [2] (5.2.6.6), [2] (разделы 5 – 7).

7 Параметры форматов контента, поддерживаемых платформой MHP

7.1 Статические форматы контента

7.1.1 MHP поддерживает статические форматы контента в соответствии с ГОСТ Р 54456 (7.1) и с уточнениями в соответствии с [11] (7.1).

Статические форматы контента включают в себя:

- I-кадры MPEG-2;
- формат обработки медиа «видео «капли»»;
- формат обработки аудиоклипов MPEG-1 аудио (Level I, II);
- формат текста мономедиа;
- формат встроенных наборов символов.

При обработке и формировании кадров растрового изображения должны поддерживаться следующие ограничения:

– один пиксель (элемент) изображения должен соответствовать одному графическому пикселю (элементу) в текущей графической конфигурации. MHP должна игнорировать любую индикацию в переданном изображении, касающуюся масштабирования пикселей, цветового пространства или цветовой гаммы. Ограничения кодирования цвета должны быть в соответствии с [11] (7.5).

– MHP должна использовать формат файла JPEG с кодированием, использующим последовательный режим DCT или прогрессивный режим, основанный на DCT в соответствии с [41]. Формат обмена файлами должен быть в соответствии с [42]. Ограничения параметров форматов PNG должны быть в соответствии с [11] (15.3).

– MHP должна поддерживать формат файлов для растровых графических изображений PNG. PNG определяется как Версия 1.0 в соответствии с [43]. Ограничения параметров воспроизведения цвета и форматов PNG должны быть в соответствии с [11] (15.1).

– MHP должна поддерживать формат обмена графическими изображениями согласно GIF [44] в соответствии с [11] (7.1.1.4).

7.1.2 MHP должна поддерживать обработку I-кадров MPEG-2, которые определяются в соответствии с [3] (6.1.1.11). Параметры полезной нагрузки файлов, представляющих I-кадры MPEG-2 должны быть в соответствии с [11] (7.1.2). Структура полезной нагрузки файла, предоставляющего I-кадры MPEG-2, должна быть:

sequence_header();

sequence_extension();

extension_and_user_data(0);
optional group_of_pictures_header() и extension_and_user_data(1);
picture_header (picture_coding_type = «I frame»);
picture_coding_extension (picture_structure = «frame picture»);
extension_and_user_data(2);
picture_data();
sequence_end_code().

7.1.3 В режиме обработки медиа, имеющего формат «видео «капли»», платформой МНР обрабатываются только I-кадры и P-кадры MPEG-2. Каждый блок должен содержать один кадр и определенное число синтаксических элементов (в соответствии с [3]), таких как sequence_header () или group_of_ picture_header ().

МНР должна поддерживать обработку блоков байтов контента, поступающих на декодер МНР, синтаксис которых соответствует [11] (7.1.3).

МНР должна поддерживать ограничения, накладываемые на P-кадры, в соответствии с [3] и [11] (7.1.3).

7.1.4 Платформа МНР должна выполнять обработку аудиоклипов в формате MPEG-1 аудио (Level I, II). Каждый файл аудио контента должен передаваться файлом двоичных данных элементарного потока. Каждый файл аудио контента должен содержать целое число аудио модулей доступа. Первый байт каждого файла аудио контента должен быть первым байтом аудио модуля доступа.

Параметры формата MPEG-1 аудио (Level I, II) должны быть в соответствии с [45] с ограничениями согласно [46].

7.1.5 Платформа МНР должна поддерживать правила кодирования текста в формате monomedia в соответствии с требованиями к модифицированному UTF-8 языку программирования Java спецификации Java Language Spec в соответствии с [11] (7.1.5).

7.1.6 МНР должна обеспечивать прием и вывод на экран терминала латинских символов, входящих в состав базового набора, в соответствии с [11] (приложение E).

Допускается ограничение состава обрабатываемых символов в соответствии с возможностями применяемых устройств ввода данных платформы МНР (например, клавиатурой).

7.2 Форматы потокового вещания

7.2.1 Аудио

Терминалы МНР должны поддерживать аудио MPEG с ограничениями и улучшениями в соответствии с [47].

7.2.2 Видео

Терминалы МНР для рынков формата вещания 625 строк/50 Гц должны поддерживать видео в соответствии с [47] (5.1).

Терминалы МНР для рынков формата вещания 525 строк/60 Гц должны поддерживать видео в соответствии с [47] (5.3).

Поддержка видео MPEG-2 высокой четкости должна обеспечиваться в соответствии с [47] (5.2 или 5.4).

Поддержка видео MPEG-4/AVC высокой четкости должна обеспечиваться в соответствии с [47] (5.7).

7.3 Форматы резидентных шрифтов

МНР поддерживает форматы резидентных шрифтов в соответствии с ГОСТ Р 54456 (7.3) и [11] (7.3).

7.4 Форматы загружаемых шрифтов

Форматы загружаемых шрифтов должны быть в соответствии с ГОСТ Р 54456 (7.4) и [11] (7.4).

7.5 Форматы представления цвета

Форматы представления цвета должны быть в соответствии с ГОСТ Р 54456 (7.5) и [11] (7.5).

7.6 Форматы многоцелевых расширений почты Интернета

Форматы многоцелевых расширений почты Интернета должны быть в соответствии с таблицей 2.

Таблица 2 – Форматы многоцелевых расширений почты Интернета

Типы MIME	Расширения Типов (Примечание)	Определение контента
«image/jpeg»	«.jpg»	В соответствии с [11] (7.1.1.2)
«image/png»	«.png»	В соответствии с [11] (7.1.1.3), допускаются ограничения в соответствии с 15.1 настоящего стандарта
«image/gif»	«.gif»	В соответствии с [11] (7.1.1.4)
«image/mpeg»	«.mpg»	В соответствии с [11] (7.1.2)
«video/mpeg»	«.mpg»	В соответствии с [11] (7.2.2)
«video/dvb.mpeg.drip»	«.drip»	В соответствии с [11] (7.1.3)
«audio/mpeg»	«.mp2»	В соответствии с [11] (7.1.4) или (7.2.1)
«text/dvb.utf8»	«.txt»	В соответствии с [11] (7.1.5)
«text/xml»	«.xml»	В настоящем стандарте не определено
«text/css»	«.css»	В настоящем стандарте не определено
«text/plain»	«.txt»	В соответствии с [11] (7.1.5)
«text/ecmascript»	«.js»	В настоящем стандарте не определено
«image/dvb.subtitle»	«.sub»	В соответствии с [11] (7.2.3)
«text/dvb.subtitle»		
«text/dvb.teletext»	«.tlx»	В соответствии с [11] (7.2.3.2)
«application/dvb.pfr»	«.pfr»	В соответствии с [11] (7.4)
«application/dvbj»	«.class»	В соответствии с [11] (6.2.5.1)
«multipart/dvb.service»	«.svc»	Применяется, если программа MPEG (Служба DVB) соответствует нормам DVB
«application/xml»	«.xml»	В соответствии с разделом 8 настоящего стандарта
Примечание – Вновь появляющиеся форматы могут иметь расширения с увеличенным количеством символов		

Типы MIME определены для резервирования пространства имен для поддержки (в будущем) загружаемых проигрывателей JMF. Расширения имен файлов должны включаться в передачах для того, чтобы приемники смогли определить тип контента. Для приложений DVB-J это описано в [11] (таблица 30).

Не все MIME и расширения файлов, определенные в таблице 2, используются в настоящем документе. Интерфейсы API DVB-J, использующие типы медиа описаны в 15.2 настоящего стандарта. В случае типов MIME для медиа, отсутствующих в списке для конкретного профиля, параметры доступа к этому типу контента из файлов не определены.

8 Параметры приложений DVB-HTML

8.1 Введение

8.1.1 Область применения

Одной из областей применения MHP является предоставление гипертекста (телетекста высшего качества) системы для представления информации вместе с передачей вещания.

Настоящий раздел содержит основные определения, необходимые для интеграции приложений DVB HTML на платформе MHP:

- определение термина приложение DVB HTML и его жизненного цикла представлено в 9.3.1 настоящего стандарта;
- параметры сигнализации приложений представлены в разделе 10 настоящего стандарта;
- определение контента для DVB HTML представлено в этом разделе.

8.1.2 Профили DVB HTML

Определение профиля DVB HTML представлено в разделе 15 настоящего стандарта.

8.2 Архитектура

8.2.1 Контекст

Приложение DVB-HTML в контексте MHP является набором ресурсов из семейства форматов контента DVB-HTML.

Ресурсы форматов приложений DVB-HTML определяются графом, ориентированным по ссылке в Точке Входа в Приложение (см. 10.10.2 настоящего стандарта). Допустимая область графа описывается дескриптором границы приложения. Интерпретация приложений DVB HTML обрабатывается приложением поддержки (агентом пользователя - в терминологии W3C). Жизненный цикл приложения DVB-HTML описан в 9.3 настоящего стандарта.

На интервале жизненного цикла приложения DVB HTML агент пользователя выполняет обработку ресурсов трех типов: декодирование; представление; взаимодействие. Описание указанных ресурсов в соответствии с [14] (8.2.1).

8.2.2 Аспекты интеграции

8.2.2.1 Доступ к DVB-J от ECMAScript

Все интерфейсы API, доступные для DVB-J, должны быть доступны ECMAScript (см. 8.10.2 настоящего стандарта).

8.2.2.2 Выполнение пользовательских агентов с помощью плагинов

В том случае, если агент пользователя реализован в виде совместимого плагина, то описание процедур должно выполняться при использовании следующих частей настоящего стандарта:

- раздел 5 «Базовая архитектура»;
- 9.8 «Плагины»;
- раздел 10 «Параметры сигнализации приложения MHP»;
- 10.13 «Сигнализация плагина»;
- раздел 12 «Безопасность»;
- 12.12 «Безопасность применения плагинов».

8.3 Формат приложений

Основные части настоящего стандарта основаны на рекомендациях консорциума World Wide Web Consortium (W3C). Рекомендации W3C являются основой блока стандартов WWW, которые используются как в стандартах DVB-HTML, так и для стандартов HTML для других устройств, таких как сотовые телефоны, карманные персональные компьютеры PDA (Personal Digital Assistant) и Интернет-устройства. На этих рекомендациях основаны инструменты разработки и программное обеспечение, используемые для вывода на экран контента для WWW. Настоящий стандарт в максимально возможной степени использует рекомендации W3C, позволяющие создавать общую основу контента и для приложений DVB-HTML, и для контента, распределенного на WWW.

В частности, используются следующие технологии:

- XML (см. 8.4 настоящего стандарта);
- XHTML (см. 8.5 настоящего стандарта);
- CSS (см. 8.8 настоящего стандарта);
- ECMAScript (см. 8.10 настоящего стандарта);
- DOM (см. 8.11 настоящего стандарта).

8.4 Расширяемый язык разметки XML

XML является безлицензионным независимым от платформы и хорошо поддерживаемым методом для структурирования данных в форме текста. Язык DVB HTML, как это определено в 8.5 настоящего стандарта, соответствует XML [48]. С момента своего создания XML стал основой нового интернет языка разметки указанной W3C.

Грамматика языка DVB-HTML представлена в определении типа документа (Document Type Definition, DTD) описана в [14] (приложение AA).

8.5 Язык разметки DVB (DVB-HTML)

Язык разметки DVB применяет язык XML. Набор символов документа XML для DVB-HTML должен быть UTF-8 или UTF-16 с форматом преобразования (Universal Multiple-Octet Coded Character Set, UCS) [49]. Набор отображаемых символов определен в [14] (приложение E).

8.5.1 Проблемы соответствия (совместимости)

8.5.1.1 Соответствие документов

Документ XML является совместимым с документом DVB- HTML, если он соответствует всем общим правилам, приведенным в 8.5.1.1.1 настоящего стандарта, и является или допустимым

документом по отношению к DTD DVB-HTML, или не допустимым документом, составленным по совместимым положениям, приведенным в 8.5.1.1.2 настоящего стандарта. Документы DVB HTML должны или не включать отдельную декларацию, или должны использовать значение «Нет».

8.5.1.1.1 Общие правила

DVB-совместимый HTML документ должен быть хорошо сформирован. Он должен включать декларации XML и должен соответствовать следующим условиям:

- корневой элемент должен быть <html>;

- корневой элемент документа определяет пространство имен XHTML использованием атрибута XMLNS, определенного в пространстве имен в XML [50]. Указатель пространства имен для XHTML находится по адресу «gost.e-ecolog.ru

- в документе до корневого элемента должна быть предусмотрена декларация DOCTYPE. Должен присутствовать общедоступный идентификатор, включенный в декларацию DOCTYPE, и должен ссылаться на DTD, описанному в [14] (приложение AA), используя свой формальный публичный идентификатор (FPI).

Идентификаторы FPI, которые ссылаются на DTD DVB-HTML, должны быть вида:

«- / / DVB / / DTD XHTML DVB-x.y HTML / / EN»

Данные x и y находятся [14] (таблица 1) для этой версии DTD DVB-HTML 1.0.

Следующие системные идентификаторы гарантированно определяют упомянутые выше DTD. Другие идентификаторы системы могут быть использованы при необходимости поиска DTD DVB. Реализация настоящего стандарта должна работать корректно без извлечения DTD из gost.e-ecolog.ru mhp/dtd/dvbhtml-x-y.dtd.

При отсутствии данных x и y находятся в таблице 3 для этой версии DVB-HTML 1.0 DTD.

Таблица 3 – Версии идентификации

	x	y
Значение	1	0

Ниже приведен рекомендуемый пример декларации DOCTYPE, используемый на практике в документах DVB-HTML:

<?xml version= «1.0»?>

<!DOCTYPE html PUBLIC «-//DVB//DTD XHTML DVB-HTML 1.0//EN»

«gost.e-ecolog.ru

8.5.1.1.2 Недопустимые, но совместимые документы

Условия совместимости недопустимых элементов должны быть в соответствии с [14] (8.5.1.1.2).

8.5.1.2 Совместимость агента пользователя DVB-HTML

Совместимость агента пользователя DVB-HTML с семейством агентов пользователя XHTML должна обеспечиваться в соответствии с [14] (8.5.1.2).

8.5.2 Набор необходимых модулей

Определения типа документа DVB-HTML состоят из абстрактных модулей, представленных в таблице 4. Определения абстрактных модулей должны быть в соответствии с [10]. Выполнение DTD определено в [14] (приложение AA).

Таблица 4 – Обязательные модули для DVB-HTML

Компоновка модулей	Наличие модуля
Structure	Yes
Text	Yes
Hypertext	Yes
List	Yes
Applet	
Presentation	Yes
Edit	
Bi-directional Text	Yes
Basic Forms	
Forms	Yes
Basic Tables	Yes

Окончание таблицы 4

Компоновка модулей	Наличие модуля
Tables	
Image	Yes
Client side Image map	Yes
Server side image map	
Object	Yes
Frames	Yes
Target	Yes
Iframe	Yes
Intrinsic events	
DVB Intrinsic events	Yes
Metainformation Module	Yes
Scripting	Yes
Style sheet	Yes
Style Attribute	Yes
Link	Yes
Base	Yes
Name Identification	
Legacy	

8.5.3 Семантика модулей

8.5.3.1 Модули XHTML

Семантика модулей XHTML должна быть в соответствии с [14] (8.5.3.1).

8.5.3.2 Атрибуты XHTML

Агенты пользователя DVB-HTML должны обеспечить механизм доступа к альтернативному тексту или к упоминаемому ресурсу и представить их пользователю. Конкретные средства зависят от реализации терминала.

Атрибут клавиши доступа Accesskey по [51] присваивает ключ доступа к элементу. Клавиша доступа является единственным символом из набора символов документа, который при нажатии дает фокус элементу. Семантика предоставления фокуса элементам является специфическим элементом ([51] (17.11.2)).

8.5.3.3 Модули DVB-HTML

8.5.3.3.1 Внутренние события DVB

Внутренние события DVB являются атрибутами, которые используются в сочетании с элементами, которые могут иметь определенные действия, происходящие в результате событий, выполняемых пользователем. Атрибуты, указанные в [14] (таблица 4), добавляются в набор атрибутов для соответствующих элементов. Пример использования этих атрибутов представлен в [14] (8.5.3.3.1).

8.6 Типы медиа

Агент пользователя DVB-HTML должен быть в состоянии распознавать и обрабатывать (совместимые с форматами раздела 7 настоящего стандарта) типы медиа в соответствии с таблицей 5.

Таблица 5 – Типы контента, доступные в DVB-HTML

Тип медиа MIME	Общее название
text/xml	XML
application/xml	
text/css	CSS
text/plain	Формат мономедиа для текста
text/dvb.utf8	
audio/mpeg	Формат мономедиа для аудиоклипов или Audio
image/jpeg	JPEG
image/png	PNG
image/gif	GIF

Окончание таблицы 5

Тип медиа MIME	Общее название
image/mpeg	MPEG-2 I-кадры
video/dvb.mpeg.drip	MPEG-2 видео капли
video/mpeg	MPEG видео
multipart/dvb.service	Многокомпонентная служба DVB
application/dvb.pfr	Загружаемые шрифты
application/dvbj	Файл класса DVB-J
text/ecmascript	ECMAScript

Ресурсы этих типов медиа MIME определяются при использовании локаторов, представленных в разделе 14 и в 8.16.5 настоящего стандарта с учетом условий эксплуатации.

Примечание – Агентам пользователей рекомендуется быть готовыми к обработке случаев, когда тип медиа MIME будет указан неправильно и возникнет необходимость использования альтернативных стратегий (например, на основе имени файла, проверки содержания).

8.6.1 Применяемые типы медиа MIME

%ContentType.datatype: описания DTD DVB HTML определяют объект «ContentType.datatype%;» для использования в атрибутах деклараций, где могут быть использованы типы медиа MIME. В таблице 6 перечислены эти атрибуты и модули XHTML.

Таблица 6 – Использование атрибутов элементов ContentType

Атрибуты элементов	Модули XHTML
a.type	Hypertext
link.type	Link
object.type	Object
object.codetype	Object
param.type	Object
form.enctype	Basic Forms, Forms
style.type	Stylesheet
script.type	Scripting

% URI.datatype: описания типа документа (DTD) определяют «% URI.datatype;», описание идентификатора (URI). Эти атрибуты перечислены в таблице 7 с соответствующими модулями XHTML для ссылки.

Таблица 7 – Перечень атрибутов URI

Атрибуты	Модули XHTML
a.href	гипертекст Hypertext
area.href	карта ссылок на стороне клиента (Client-side Image Map)
area.href	Основной (Base)
blockquote.cite	Основной текст (Basic Text)
form.action	Базовые формы, Формы (Basic Forms, Forms)
frame.longdesc	Кадры (Frames)
frame.src	Кадры (Frames)
head.profile	Структура (Structure)
iframe.longdes	I –кадр(Iframe)
iframe.src	I –кадр(Iframe)
img.src	I –кадр(Iframe)
img.longdesc	I –кадр(Iframe)
img.usemap	Карта ссылок на стороне клиента (Client-side Image Map)
input.src	Базовые формы, Формы (Basic Forms, Forms)
input.usemap	Базовые формы, Формы (Basic Forms, Forms)
link.href	Канал (Link)
object.archive	Объект (Object)

Окончание таблицы 7

Атрибуты	Модули XHTML
object.classid	Объект (Object)
object.codebase	Объект (Object)
object.data	Объект (Object)
object.usemap	Карта ссылок на стороне клиента (Client-side Image Map)
q.cite	Основной текст (Basic Text)
script.src	написание сценария (Scripting)

8.6.2 Ограничения использования типа медиа MIME

Ограничения использования типа медиа MIME должны быть в соответствии с таблицей 8. Режим работы терминала МНР указывается только там, где есть «X» на пересечении element.attribute и типа медиа MIME.

Таблица 8 – Ограничения использования типа медиа MIME

Атрибуты элементов	Типы медиа MIME														
	text/xml	application/xml	text/css	text/plain	text/dvb.utf8	audio/mpeg	image/jpeg	image/jpeg	image/gif	image/mpeg	video/mpeg	multipart/dvb.service	application/dvbj	video/dvb.mpeg.drip	text/ecmascript
Корневой элемент	x														
a.type	x					x						x	x		
link.type (Примечание)	x		x												
object.type	x					x	x	x	x	x	x	x		x	
object.codetype													x		
param.type	(Примечание)														
style.type			x												
script.type															x
Примечание – Ограничения типа медиа MIME применяются только к этому атрибуту, если атрибут param.valuetype имеет значение «ref». Не допускается поддержка типов объектов, которые не используют атрибут этого «type»															

В [14] (таблице 9) показано наличие ресурса, идентифицированного величиной URI каждой пары: element.attribute и медиа типа MIME в DVB-HTML.

8.6.3 Семантика типа медиа MIME

Семантика типа медиа MIME представлена в [14]:

- Локаторы выхода – согласно [14] (8.16.5.2);
- Аудио MPEG – согласно [14] (8.6.7);
- Видео MPEG – согласно [14] (8.6.8);
- Контент приложения – согласно [14] (8.6.5);
- Службы DVB – согласно [14] (8.6.9);
- Доступ к альтернативному тексту или к ресурсу – согласно [14] (8.5.3.2.1);
- Контент кадра – согласно [14] (8.6.4);
- Контент графики – согласно [14] (8.6.10);
- Контент сценария – согласно [14] (8.6.11);
- Контент таблицы стилей – согласно [14] (8.6.12).

8.6.4 Контент кадра

Контент кадра должен быть совместимым с документами DVB-HTML.

Примечание – К реализации не предъявляется требование предоставления механизма прокрутки кадров.

8.6.5 Контент приложения

8.6.5.1 Ссылки через локатор AIT

Локатор AIT имеет вид «DVB:// current.ait / ...», который применяется при использовании сигнализации AIT для обращения к контенту приложения. При использовании этой формы локатора приложения DVB-HTML могут ссылаться на доступные в настоящее время приложения (см. 11.7.2 настоящего стандарта).

Пример:

```
<a type=«application/dvbj» href=«dvb://current.ait/Orgid.Appid?arg_0=val»>
  Click here to launch application
</a>
```

Подробности процедры обращения к контенту должны быть в соответствии с [14] (8.6.5.1).

8.6.5.2 Ссылки, выполняемые при отсутствии локатора AIT

Ссылки на документы DVB-HTML или фрагменты контента при отсутствии локатора AIT могут произойти вследствии:

href: семантика определена в соответствии с [51].

usemap: ссылочный элемент, ссылки определяет отображение для наложения на изображение. Семантика определена в соответствии с [51].

object.data: ссылочный документ, отображается внутри объекта, как если бы это был кадр. Семантика определена в соответствии с [51].

src: ссылочный документ, представлен в кадре или I-кадре. Семантика определена в соответствии с [51].

8.6.6 Соединение, зависящее от значения атрибута link.rel

Типы медиа MIME, разрешенные для этого атрибута, зависят от значения атрибута link.rel.

Для link.rel=«stylesheet» определен только тип медиа MIME «text/css». Агент пользователя должен интерпретировать его в соответствии с [51] (14.3.2).

Для всех других значений link.rel разыскиваемый ресурс должен быть совместимым с документом DVB-HTML. Агент пользователя может использовать эту информацию в реализации конкретных путей (например, для предварительной выборки указанного документа).

8.6.7 Аудио MPEG

DVB HTML может использовать ссылку на тип медиа MIME (аудио MPEG в соответствии с [14] (таблица 9)). Тип аудио MPEG может быть либо неопределенной длительности (см. [14] (7.2.1)), либо определенной длительности (см. [14] (7.1.4)).

8.6.7.1 Ресурсы неопределенной продолжительности

Ссылки на элементарный поток MPEG неопределенной продолжительности в режиме вещания, как указано в [14] (7.2.1) должны выбирать компонент службы для представления в текущем контексте служб. Если в данный момент воспроизводится компонент аудио службы, то он заменяется в полном объеме в соответствии с указанным выбором.

Правила активизации ссылки на элементарный поток MPEG неопределенной продолжительности должны быть в соответствии с [11] (8.6.7.1).

В случае не срабатывания ссылок, должны применяться ограничения в соответствии с [11] (8.6.7.1.1).

8.6.7.2 Ресурсы определенной длительности

Ссылки на ресурс аудио файлов конечной продолжительности (например, переносимых в карусели DSMCC), который несет аудиоданные, как указано в [14] (7.1.4), аудио файлы должны создаваться в соответствии с ограничениями по [14] (приложение G, G.2).

Правила активизации ссылок на ресурс аудио файлов конечной продолжительности должны быть в соответствии с [11] (8.6.7.1).

8.6.7.2.1 Связь с событиями документа

В случае не срабатывания ссылок, должно применяться следующее ограничение: Событие DVBDOMStable и событие загрузки должны генерироваться до воспроизведения аудио.

8.6.8 Видео MPEG

Размещение видео MPEG или img возможно на всем экране или на четверти экрана. Если объект для отображения задан таким образом, что аппаратные средства его не поддерживают, то агент пользователя должен придерживаться обычных правил аварийного перехода на недоступные ресурсы в объекте. Решение задачи может быть получено предоставлением альтернативного контента с учетом необходимости предотвращения различий разметки экрана и img контента. Следует учитывать, что img и input теги этот альтернативный вариант не представляют и что следует учитывать возможности целевых STB для отображения заданных img.

8.6.8.1 Ресурсы видео неопределенной продолжительности

Ссылка на элементарный поток MPEG неопределенной продолжительности в режиме вещания выполняется в соответствии с [14] (7.2.2). Размещение видео в ограничительном прямоугольнике элемента, выполняемого после моделирования, должно выполняться в соответствии с [14] (8.6.8.1).

8.6.8.2 Ресурсы конечной длительности

Настоящий стандарт не определяет привязку сегментов видео конечной продолжительности.

8.6.9 Службы DVB

Использование локаторов URL, которые ссылаются на службу в контекстах, не требующих участия пользователя (т.е. object.data, object.archive) на объектах типа «multipart/dvb.service», определены только для служб, относящихся к форматам медиа (например, видео, аудио, субтитры), но компоненты медиа этой службы должны отображаться на экране в соответствии с 8.6.7, 8.6.8 настоящего стандарта.

Ссылки на службы DVB в контекстах, требующих действий пользователя (то есть ссылки a.href, area.href) должны вызвать выбор службы, на которую была выполнена ссылка, если приложение будет авторизовано согласно [14] (8.15.1.9). Если авторизация не удается, то ссылка должна обрабатываться, как недоступный ресурс.

Ссылки на службы DVB в форме «dvb» на вещательные службы и другие локаторы, ссылающиеся на сохраненные службы [11] (9.6.2) или службы по интерактивному каналу, должны быть разрешены (см. [11] (9.6.1)).

8.6.10 Контент графики

Должны поддерживаться форматы графики в соответствии с 7.1 настоящего стандарта. Такая графика должна выводиться на экран вместе с контентом в поле, образованном в результате обработки правил CSS.

Примечание – Допускаются ссылки на изображения других типов, отличающихся от минимальных требования для MHP, но следует учитывать, что реализация терминала может их не отобразить. Так как img и входные теги не обеспечивают возможность возвращения в исходное состояние, авторы должны быть осторожными в их использовании и должны убедиться, что целевой STB выведет контент на экран.

8.6.11 Контент сценария

Ресурс контента сценария должен быть в формате обычного текста, представленным в 7.1.5 настоящего стандарта. Контент должен быть представлен исходным кодом ECMAScript в соответствии с [10] и должен обрабатываться как встроенный.

8.6.12 Контент таблицы стилей

Ресурс контента таблицы стилей должен быть в формате обычного текста, представленного в 7.1.5 настоящего стандарта. Контент должен быть представлен исходным кодом CSS, как определено в [9].

8.6.13 Форматы локаторов URL протоколов HTTP и HTTPS

Для Form.Action действительны только URL HTTP (HTTPS).

Агент пользователя обязан только обрабатывать случай, когда возвращенный контент совместим с документом высокого уровня.

Форматы локаторов URL протоколов HTTP, HTTPS должны быть в соответствии с [14] (8.6.13).

8.6.14 Свойства CSS

8.6.14.1 Источники пункта назначения типа медиа MIME

<uri>: спецификация CSS [9] определяет тип значения <uri>. Места, где <uri> является допустимым значением в CSS, должны быть в соответствии с CSS [9].

Свойства CSS должны быть в соответствии с [14] (8.6.14).

8.6.14.2 Ограничения на использование типа медиа MIME

Настоящий стандарт не устанавливает правила поведения (реагирования) при ссылке на следующие типы медиа MIME для любого типа <uri> в CSS:

- text/xml;
- application/xml;
- text/css;
- audio/mpeg;
- text/dvb.subtitle;
- image/dvb.subtitle;
- application/dvbj;
- text/ecmascript.

8.6.15 Сгенерированный контент

Ресурс, на который выполняется ссылка, должен быть в формате текста, предусмотренного 7.1.5 настоящего стандарта.

Контент должен обрабатываться как встроенный в соответствии с CSS [9].

Примечание – В CSS [9] не допускается:
- создавать разметку (только XML CDATA);
- применять другие стили для создаваемого контента;
- выполнять доступ к контенту через DOM.

8.6.16 Стили графики

Графические форматы должны поддерживаться в соответствии с 7.1.1 настоящего стандарта. Такая графика должна обрабатываться в соответствии с CSS [9].

8.6.17 Стили видео

Видео и «видео «капли»» могут использоваться в качестве свойства стиля фонового изображения. Ресурс должен быть обработан как тип изображения в соответствии с CSS [9].

Видео отображается на экране в соответствии с требованиями 8.6.8.1 настоящего стандарта, как образец, который не требует участия пользователя.

Примечание – Видео MPEG или изображения могут визуализироваться только на полном экране или на четверти экрана. Если запрашивается вывод объекта на экран в форме, которая не поддерживается аппаратными средствами, то агент пользователя должен выполнять обычные правила нейтрализации запроса недоступных ресурсов объекта.

Видео может использоваться как свойство стиля фона окна просмотра «background-video-n». Ресурс должен использоваться для выбора потока видео, на который выполнена ссылка в контексте службы для отображения на экране в соответствующей плоскости фона видео.

«Видео «капли»» могут использоваться в качестве свойства окна просмотра» background:». Ресурс должен использоваться для заполнения соответствующей плоскости фона.

8.6.18 Стили служб DVB

Использование URL со ссылкой службы в качестве стиля фонового видео определено только для аспектов службы медиа (например, видео, аудио, субтитры) и не должна вызывать выбор службы, компоненты медиа этой службы должны отображаться в соответствии с [11] (8.8.5.3.2).

8.7 Синхронизация

8.7.1 Обзорная информация о триггерах

Триггеры обеспечивают возможность провайдеру приложения взаимодействовать с приложением, работающим на терминале конечного пользователя. Настоящий стандарт определяет параметры привязки к событиям потока DSM-CC. Триггеры принимаются агентом пользователя и используются для формирования события в работающем приложении DVB-HTML.

Триггеры представляют собой небольшие сообщения, отправленные в передаче вещания отдельно от основного контента, которые могут вызвать изменения в поведении приложений. Триггеры могут переносить время, в которое они должны быть доставлены с небольшим количеством данных полезной нагрузки, которые приложение может зарегистрировать.

Авторы приложения могут использовать традиционные базовые оценки времени медиа (например, временные коды SMPTE), которые представляют собой смещение в единицах кадр/с от начала медиа. Триггеры могут сигнализировать о достижении точки базового времени. Автор должен дать имена «name» этим события, чтобы обеспечить приложениям возможность подписки на эти события. Имя события может быть обозначено в соответствии с таблицей 9.

Таблица 9 – Примеры имен триггеров

Event name	Event time
Start	00:00:00.00
End of introduction	00:00:30.00
End of recipe	00:05:00.00
End of recipe	00:08:00.00
Start or roll out	00:11:00.00
End	00:11:30.00

Поведение объекта после события реализуется в коде приложения. Для модификации поведения объекта допускается передача данных с событием.

8.7.1.1 Транспортировка триггеров

Транспортировка триггеров выполняется в событиях потока DSM-CC.

8.7.1.2 Регистрация и прием триггеров

В целях интеграции с моделью событий W3C, триггеры доставляются приложениям DVB-HTML как события DOM. Регистрация выполняется использованием API DOM.

Доставка выполняется событием DOM. Параметры интерфейса события триггер DOM наследуются от интерфейса «Event» в DOM [52].

8.7.1.3 Привязка событий к потоку событий DSM-CC

Привязка триггеров к потоку событий DSM-CC выполняется в соответствии с механизмом по 8.7.3 настоящего стандарта.

По умолчанию приложение DVB-HTML связано со всеми Сообщениями Потока событий DSM-CC, которые расположены в корневом каталоге приложения, определенном в поле physical_root в дескрипторе dvb_html_application_location из AIT.

В этом случае сообщения потока событий DSMCC перечисляют события, используемые в контексте всего приложения (8.7.4 настоящего стандарта).

8.7.2 События триггера

8.7.2.1 Преобразование событий потока в события объектной модели документа (Document Object Mode, DOM)

В соответствии с [14] (8.7.1.3) процесс инициирования DVB-HTML содержит механизмы, которые преобразуют события потока в события DOM:

- механизм, переопределяющий, по умолчанию, ассоциации между приложением и сообщением потока событий;
- механизм, который разворачивает событие медиа в несколько событий DOM, включая отображение имени события медиа в тип DOM, извлечение любых полезных данных события медиа и преобразования их в свойства события DOM.

Обзорная диаграмма механизма преобразования событий представлена на рисунке 3.

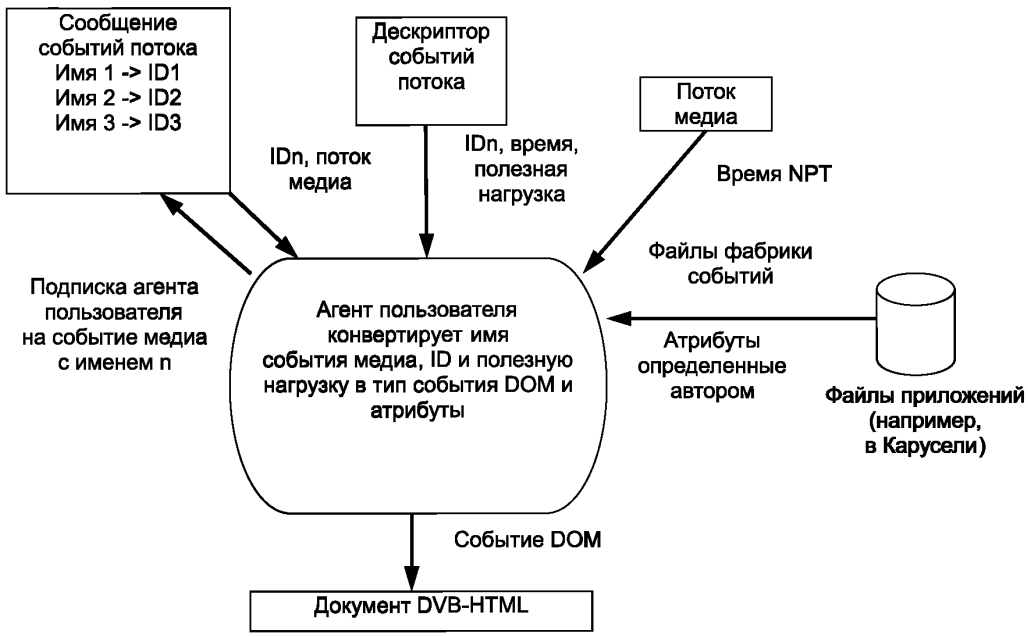


Рисунок 3 – Обзорная диаграмма механизма событий

Пример выполнения шагов (этапов) преобразования событий реализацией после того, как событие триггера, доставленное потоком событий DSM-CC, использует файл редактирования и файл фабрики, представлен в [11] (8.7.2.1). Возможны другие варианты выполнения этой процедуры, однако результат должен быть эквивалентен описанному.

8.7.2.2 Определения файла фабрики события

8.7.2.2.1 Синтаксис файла фабрики события

Синтаксис файла фабрики события должен быть в соответствии с [14] (8.7.2.2.1).

8.7.2.2.2 Семантика элементов

Семантика элементов файла фабрики события должна быть в соответствии с [14] (8.7.2.2.2).

8.7.2.2.3 Семантика атрибутов

Семантика атрибутов файла фабрики события должна быть в соответствии с [14] (8.7.2.2.3).

8.7.2.3 Элемент фабрики события по умолчанию

События медиа, которые не определены в файлах фабрики событий, связанные с документом DVB-HTML, и чьи имена «событие-имя» не начинаются с «DVB», рассматривают как связанные с элементом фабрики события, который определяется с синтаксисом в соответствии с [14] (8.7.2.2.3).

8.7.2.4 Файл фабрики события по умолчанию

При отсутствии файлов фабрики событий, связанных с документом DVB-HTML, они предполагаются по умолчанию.

Файл событий по умолчанию связанный с документом DVB-HTML содержит определения всех событий триггера.

Агрегация (объединение) элементов фабрики события по умолчанию представляет собой файл фабрики событий документа DVB-HTML.

8.7.2.5 Практический пример

Практический пример файла фабрики событий приложений представлен в [14] (8.7.2.2.5).

8.7.2.6 События системы

События системы используют тот же самый транспорт и механизмы связывания как события триггера. Но вместо того, чтобы быть преобразованными в события DOM и поставляться приложению DVB-HTML, они полностью обрабатываются агентом пользователя. Процесс отображения имен событий медиа в имена событий системы идентичен случаю событий триггера, которые являются элементами события системы и определяют (опционально) поток и имя события. Однако в этом случае атрибут типа определяет вид события системы. Определены следующие величины типа: «dvb.start» и «dvb.page».

Примечание – Все события, имена которых начинаются с «dvb» зарезервированы для использования этого имени, чтобы избежать столкновения с другими возможными определителеми имени.

Атрибуты, связанные с событиями системы, являются фиксированными, но используют тот же механизм извлечения информации из байтов полезной нагрузки.

8.7.2.6.1 События dvb.start и dvb.page

Описание процессов применения событий dvb.start и dvb.page должно быть в соответствии с [14] (8.7.2.6.1, 8.7.2.6.2).

8.7.3 Привязка файлов фабрики события к приложению

Приложение DVB-HTML обычно создается из нескольких документов (файлов) DVB-HTML, у каждого из них могут быть различные триггерные наборы событий. Каждый файл может быть связан с некоторым количеством (от 0 до n) файлов фабрики события. Если файлы фабрики события, связанные с документом DVB-HTML, отсутствуют, то по умолчанию наличие файла фабрики события предполагается (8.7.2.4 настоящего стандарта).

Если элементы, связанные с типом события во всех файлах фабрики событий, связанных с документом DVB-HTML отсутствуют, то по умолчанию наличие элементов фабрики события для этого события предполагается (8.7.2.3 настоящего стандарта).

Если с документом связано больше одного файла фабрики события, то тогда следует считать, что все элементы содержатся в единственном файле. Порядок файлов не является существенным.

Соответствие между файлами DVB и файлами HTML фабрики событий определяется отсутствием или наличием файла «событие соединения». Этот файл определяет связь между конкретными файлами фабрики событий и документами, которые к ним относятся. Если файл «событие соединения» отсутствует, то по умолчанию предполагается наличие механизма триггера (8.7.4 настоящего стандарта).

8.7.3.1 Синтаксис файла события соединения

Синтаксис файла события соединения должен быть в соответствии с [14] (8.7.3.1).

8.7.3.2 Семантика файла событие соединения

Семантика файла событие соединения должна быть в соответствии с [14] (8.7.3.2). Пример записи файла событие соединения представлен в [14] (8.7.3.3).

8.7.3.3 Наименование и расположение файла событие соединения

Файл событий соединения находится в том же каталоге, что и документ точки входа в приложение.

Имя файла событий соединения формируется заменой любого расширения имени файла строкой «lnk». Таким образом, для документа точки входа приложения:

foo.html .

Файл события соединения будет называться: foo.lnk.

8.7.4 Механизм триггера по умолчанию

Механизм триггера по умолчанию используется при отсутствии файла события соединения с приложением, связанным с DVB-HTML.

Режим по умолчанию обеспечивает простой механизм событий в случае, когда DVB-HTML приложение связано с системой событий или с триггером программ по простым данным полезной нагрузки.

Обзорная диаграмма работы триггера по умолчанию дана на рисунке 4.

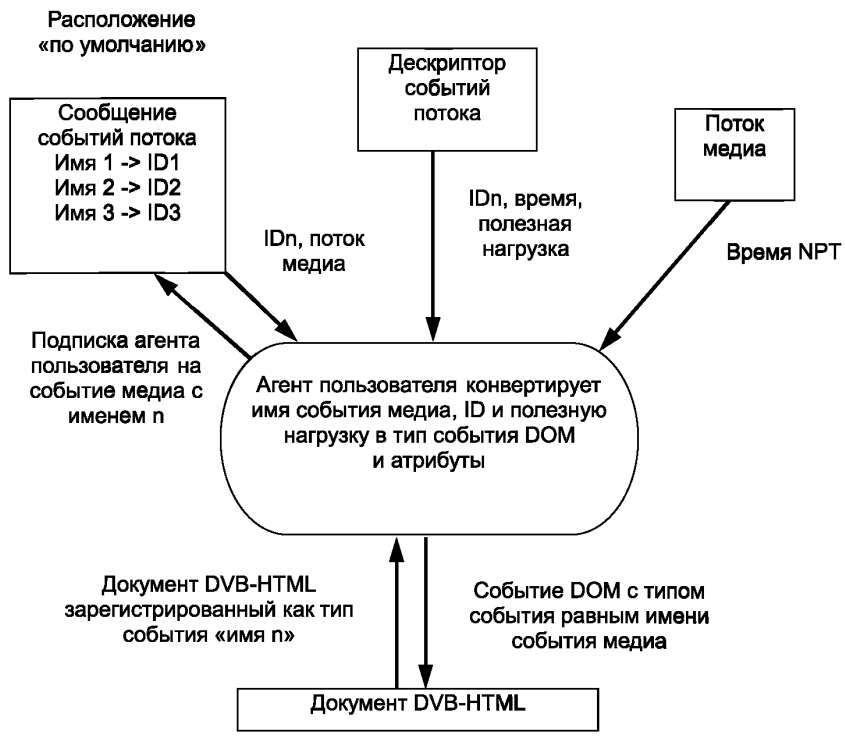


Рисунок 4 – Обзорная диаграмма работы триггера по умолчанию

Детализированное описание работы триггера по умолчанию представлено в [14] (8.7.4).

8.8 Каскадные таблицы стилей

Каскадные таблицы стилей (Cascading Style Sheets; CSS) являются методом просмотра контента HTML или XML.

8.8.1 Обзор профилирования CSS для MHP

DVB-HTML должна поддерживать CSS2, как определено в CSS [9], с уточнениями в соответствии [14] (8.8.2 – 8.8.7).

8.8.2 Поведение по умолчанию

При отсутствии любой другой информации о стиле для страницы агент пользователя по умолчанию должен предоставлять таблицу стилей в соответствии с [14] (приложение AB).

8.8.2.1 Правила шрифта таблицы стилей по умолчанию

Для агентов пользователя DVB HTML таблицу стилей используют по CSS (см. UA стили по умолчанию в CSS [9]). Эти таблицы представлены в [14] (приложение AB). Эта таблица для приложений.DVB HTML не является доступной непосредственно.

В дополнение к позициям стилей, перечисленных в [14] (приложение AB), этот стиль также должен предоставить правила @font-face для всех установленных шрифтов, которые должны включать минимальный установленный набор шрифтов (см. [11] (приложение G, G.4,)) и правила для общих семейств CSS [9] с засечками, без засечек, курсив, фантазия и с фиксированным межбуквенным расстоянием.

Конкретные реализации таблицы стилей должны:

- предоставлять описание шрифта «Tiresias»;
- обеспечивать нормальный вес шрифта;
- обеспечить нормальную ширину символа;
- обеспечивать требуемые размеры точки (см. [11] (таблица 85));
- обеспечить источник для поиска местоположения данных шрифта.

Конкретные реализации таблицы стилей не должны обеспечивать стиль шрифта (font-style) или объявления различных шрифтов (font-variant), если реализация обеспечивает данные шрифты, позволяющих визуально отличать представленный текст.

Ниже представлен пример минимальной реализации, использующей следующее правило:

```
@font-face {
 font-family: «Tiresias», serif, sans-serif, cursive, fantasy, monospace;
 font-weight: normal;
 font-stretch: normal;
 font-size: 36pt, 31pt, 26pt, 24pt;
 }
```

8.8.2.1.1 Расширение простого правила

Реализация может дополнительно включать unicode-range: данные для установленных шрифтов и других дескрипторов соответствия ([9] (15.3.6)), которые могут использоваться, например, для корректировки размера шрифта.

8.8.2.1.2 Возможности использования курсива, малых прописных и изменений ширины шрифта

Правила, приведенные выше, подразумевают, что доступные шрифты включают все варианты курсива, маленьких заглавных букв и ширины символов шрифта. Реализация, у которой были только минимальные установленные шрифты, должна обеспечивать те же самые (простые) глифы стиля для любой комбинации стилей, требуемых [9].

8.9 Интеграция Xlet

DVB-HTML определяет элемент <объект>, который может использоваться для встроенных Xlet. Для элементов <объект>, используемых для встроенных Xlet, тип должен быть установлен в «application/dvbj», элемент <объект> может содержать элементы <param>, содержащие параметры времени выполнения Xlet, определенные элементом <объект>. Последовательность этих параметров отображена на XletContext.ARGS и должна быть доступна Xlet в форме массива строк, при вызове:

javax.tv.xlet.XletContext.getXletProperty(XletContext.ARGS

Уточнения правил интеграции Xlet должны быть в соответствии с [14] (8.9).

8.9.1 Элемент объект

Семантика элемента <объект> должна быть в соответствии с [14] (8.9.1).

8.9.2 Элемент параметр

Определение элемента параметр и семантика атрибутов должны быть в соответствии с [14] (8.9.2).

8.9.3 Пример

Пример использования объекта и параметров элементов для включения Xlet внутреннего применения в документ DVB-HTML приведен в [14] (8.9.3).

8.10 Создание (планирование) сценариев

Для управления вычислительными объектами в среде узла используется простой объектно-ориентированный язык сценариев ECMAScript. Настоящий стандарт требует применения ECMAScript в соответствии с [10].

8.10.1 Связывание (образование связей) DOM 2

Настоящий стандарт требует использования обязательных библиотек DOM 2 в соответствующих ссылках для интерфейсов по 8.11 настоящего стандарта.

8.10.2 Интерфейс между ECMAScript и DVB-J

Используя интерфейс между ECMAScript и DVB-J, можно создавать приложения, которые являются гибридом DVB-J и DVB-HTML/ECMAScript. Например, основной механизм для конкретного приложения может быть реализован в DVB-J с интерфейсом пользователя с графическим контентом, реализованным в DVB-HTML/ECMAScript.

8.10.2.1 Интерфейсы API ECMAScript для доступа к DVB-J

В общедоступном приложении DVB-J пакеты, классы, методы и поля должны быть видимыми в ECMAScript, используя свойство глобального объекта, называемое Пакеты. Например, к классу идентификатора RMI доступ можно получить, используя Packages.java.rmi.Naming. К пакету java.lang доступ может быть получен, используя Packages.java.lang. В дополнение к классам DVB-J файлы класса, расположенные в базе кодов доступны, как определено тегом meta. Детализированные процедуры доступа к DVB-J представлены в [14] (8.10.2.1).

8.10.2.2 Соединение интерфейсов Inter-Xlet и Xlet-ECMAscript через интерфейс API org.dvb.ixc

Платформа MHP разрешает связь между интерфейсами Xlet для передачи объектов через API org.dvb.io.ixc, описанные в [14] (11.7.3, приложение). Детализация процедуры доступа к DVB-J представлена в [14] (8.10.2.2).

8.10.2.3 Безопасность

Модель обеспечения безопасности DVB-J применяется к приложению в целом, следовательно, Xlet, встроенный в документ DVB-HTML, имеет полномочия полного приложения DVB-HTML. ECMAScript может непосредственно вызвать DVB-J с теми же самыми полномочиями, как и полное приложение. См. 8.14 настоящего стандарта.

8.10.2.4 Неявный метод выбора

Неявный метод выбора применяется тогда, когда набору аргументов соответствует более одной сигнатуры (подписи) метода. Предпочтения аргументов сравниваются для каждого аргумента, начиная с первого до тех пор, пока одна из сигнатур не получит предпочтения. Предпочтение определяется по аргументам таблиц преобразования типов, перечисленных в порядке убывания предпочтения в соответствии с [14] (8.10.2.4).

8.10.2.5 Явный метод выбора

Методы могут быть выбраны явно при обращении к методу, как к элементу ассоциативного ECMAScript массива, индексированного подписью, например:

new Packages [«java.lang.String»] [«((char[])»] (c);

8.10.2.6 Статический метод вызова

Статические методы класса Java позволяют вызывать из CMAScript на JavaClass объект или на экземпляр класса. Статический метод вызова выполняется в соответствии с [14] (8.10.2.6).

8.10.2.7 Метод соответствия сигнатуры

В случае метода соответствия сигнатуры методы вызываются из ECMAScript при использовании таблиц, определяющих порядок выбора и преобразования типов при передаче аргументов в соответствии с [14] (8.10.2.7).

8.10.2.8 Новые типы объекта языка ECMAScript

Описание новых типов объекта языка ECMAScript: JavaArray, JavaObject и JavaClass представлено в [14] (8.10.2.8).

8.10.2.9 Преобразования типа (ECMAScript для DVB-J)

Допустимые преобразования типа (ECMAScript для DVB-J) представлены в [14] (8.10.2.9, таблицы 19 – 25). Преобразования, не указанные в таблицах, не допускаются.

8.10.2.10 Разделение на подклассы и создание экземпляра интерфейса

Экземпляры класса Java могут быть созданы при использовании «нового» оператора в соответствии с [14] (8.10.2.10).

8.10.2.11 Преобразования типа (DVB-J в ECMAScript)

Преобразование типа возвращаемых значений происходит в соответствии с [14] (8.10.2.11).

8.10.2.12 Захват исключений DVB-J в ECMAScript

Захват исключений DVB-J в ECMAScript выполняется в соответствии с [14] (8.10.2.12).

8.11 Объектная модель документа (DOM)

Программный интерфейс DOM является платформой и интерфейсом, обеспечивающих программный доступ к содержанию, структуре и стилю документов. Требуется поддержка привязки к языку для DOM и для языков Java и ECMAScript.

Поддерживаемые модули DOM представлены в [14] (8.11, таблица 27).

8.11.1 События DOM Уровень 2

8.11.1.1 Базовые интерфейсы

DVB-HTML должен поддерживать модуль событий как определено в [52], которая включает следующие интерфейсы: EventTarget, EventListener, DocumentEvent, Event и EventException.

8.11.1.2 Интерфейсы событий

Должны поддерживаться наборы событий объектной модели документа (DOM) согласно [52]:

- UIEvent;
- MutationEvent.

Не нужны для поддержки наборы событий:

- HTML;
- MouseEvent.

Рекомендуется, чтобы реализации, обеспечивающие поддержку мыши как устройства, предоставляли набор событий MouseEvent.

8.11.2 Модуль DOM события DVB

Приложение DOM может использовать метод hasFeature интерфейса DOM, чтобы установить, что модуль поддерживается. Строка функции для всех интерфейсов, перечисленных в этом модуле, является «DVBEvents» версия «2.0».

8.11.2.1 События клавиатуры

Доступ к событиям клавиатуры выполняется в соответствии с дополнительного модуля DVB DOM в 8.11.3 настоящего стандарта.

8.11.2.2 События жизненного цикла

Этот пункт отображает события жизненного цикла DVB-HTML к модели событий DOM.

8.11.2.2.1 Интерфейс DVBLifecycleEvent

Интерфейс DVBLifecycleEvent предоставляет приложению DVB-HTML конкретную контекстную информацию, связанную с его жизненным циклом.

Все события жизненного цикла доставлены в корневой элемент документа, находящегося в корневом кадре приложения и следуют стандартным правилам доставки события согласно (DOM).

Порядок предоставления информации изложен в [14] (8.11.1.2.2.1).

Определения IDL, семантика атрибутов и описание методов должны быть в соответствии с [14] (8.11.1.2.2.1.1 - 8.11.1.2.2.1.3).

8.11.2.2.2 Определения типов событий

Детализация типов событий, которые могут произойти: AppStarting, AppActive, AppPause, AppResume, AppDestroyed, AppKilled, AppTerminating, должна быть в соответствии с [14] (8.11.2.2.2.1 - 8.11.2.2.2.1.7).

8.11.2.2.3 Перечень изменений состояния

Таблица 10 содержит допустимые переходы в модели жизненного цикла (см. 9.3.3 настоящего стандарта).

Таблица 10 – Допустимые переходы в модели жизненного цикла

Изменение состояния	Загрузка	Активное	Пауза	Разрушено	Уничтожено
Загрузка к ...	Нет	Да (примечание 1)	Нет	Да (Примечание 2)	Да (Примечание 3)
Активное к...	Нет	Нет	Да (Примечание 4)	Да (Примечание 5)	Да (Примечание 6)
Пауза к ...	Нет	Да (Примечание 7)	Нет	Да (Примечание 8)	Да (Примечание 9)
Разрушено к...	Нет	Нет	Нет	Нет	Да (Примечание 10)
Уничтожено к...	Нет	Нет	Нет	Нет	Нет

Примечания:

1 В соответствии с перекрестной ссылкой 1 таблицы 11 настоящего стандарта.
2 В соответствии с перекрестной ссылкой 2 таблицы 11 настоящего стандарта.
3 В соответствии с перекрестной ссылкой 3 таблицы 11 настоящего стандарта.
4 В соответствии с перекрестной ссылкой 4 таблицы 11 настоящего стандарта.
5 В соответствии с перекрестной ссылкой 5 таблицы 11 настоящего стандарта.
6 В соответствии с перекрестной ссылкой 6 таблицы 11 настоящего стандарта.
7 В соответствии с перекрестной ссылкой 7 таблицы 11 настоящего стандарта.
8 В соответствии с перекрестной ссылкой 8 таблицы 11 настоящего стандарта.
9 В соответствии с перекрестной ссылкой 9 таблицы 11 настоящего стандарта.
10 В соответствии с перекрестной ссылкой 10 таблицы 11 настоящего стандарта.

Таблица 11 – Пояснения для примечаний к таблице 10 настоящего стандарта

Перекрестные ссылки	Изменение состояния	Событие доставлено
1	Загрузка к Активное	Событие AppActive на входе в активное состояние
2	Загрузка к Разрушено	Событие AppDestroyed на входе в состояние разрушено
3	Загрузка к Уничтожено	Событие AppKilled на входе в состояние уничтожено
4	Активное к Пауза	Событие AppPause на выходе из состояния активное
5	Активное к Разрушено	Событие AppDestroyed на входе в состояние разрушено
6	Активное к Уничтожено	Событие AppKilled на входе в состояние уничтожено
7	Пауза к Активное	Событие AppResume на входе в активное состояние
8	Пауза к Разрушенное	Событие AppResume на входе в состояние разрушено
9	Пауза к Уничтоженное	Событие AppKilled на входе в состояние уничтожено
10	Разрушенное к Уничтроженное	Событие AppKilled на входе в состояние уничтожено
Примечание – В случае ссылок 8 и 9 событие AppResume не принимается.		

8.11.2.3 Дополнительные события DVB

Дополнительные события DVB описаны в [11] (11.2.3).

8.11.2.3.1 События триггер

События триггер описаны в [11]. (11.2.3.1).

8.11.2.3.2 События DVBDOMStable

Событие DVBDOMStable сигнализирует о завершении формирования документа DOM. После получения этого события структура документа DOM будет завершена, однако в подкадрах или во встроенных кадрах структура DOM может быть не полной, и загрузка ресурсов, таких как контекст изображений может быть не законченной. До окончания генерации реализаций события структура DOM, на которую ссылается переменная документа, может быть неполной.

Это событие использует базовый интерфейс события для передачи контекстной информации.

Допустимый тип события представлен в [14] (8.11.2.3.2).

8.11.2.3.3 События DVB-HTML

События DVB-HTML используют базовый интерфей события для передачи контекстной информации. Основные типы событий описаны в [14] (8.11.2.3.3).

8.11.3 События клавиатуры DVB модуля DOM

Приложение DOM может использовать метод hasFeature (функция, версия) интерфейса DOMImplementation со значением параметра «DVBKeyEvents» и «2.0» (соответственно) для того, чтобы определить, поддерживается ли реализация модуля события. Строка функции используется также с методом createEvent. Это зависит от того как MHP получает набор поддерживаемых клавиатур.

События клавиатуры, сгенерированные реальной или виртуальной клавиатурой или от устройства дистанционного управления, должны быть сгенерированы агентом пользователя и могут быть зарегистрированы для приложений через DOM. Для регистрации событий используется тип строки «HKeyEvent» или «HRcEvent», которые могут разместить слушателя, который должен получить объект, реализующий интерфейс DVBKeyEvent.

8.11.3.1 Интерфейс события DVBKeyEvent

Интерфейс события DVBKeyEvent предоставляет конкретную контекстную информацию, связанную с событием дистанционного управления и клавиатурой. Интерфейс описывает такие представления событий клавиатуры как строка, цвет или символ (например, треугольник, «>», «игра»).

Описания определений IDL, атрибутов и методов представлены в [14] (8.11.3.1.1 - 8.11.3.1.3).

8.11.4 Модуль DOM DVB-HTML

8.11.4.1 Соответствие агента пользователя DVB-HTML реализации DOM

Агент пользователя DVB-HTML должен рассматриваться как реализация DOM только для HTML в соответствии с объектной моделью документа (DOM) согласно [53].

8.11.4.2 Отличия интерфейсов HTML от W3C DOM Уровень 1

Интерфейсы W3C DOM 1 HTML были разработаны для того, чтобы обеспечить удобство работы с документами в соответствии с [51]. Модуль DOM DVB-HTML был определен с целью управления документами DVB-HTML, но не для создания новых документов. В целом модуль DOM DVB-HTML повторяет дизайн интерфейсов DOM1 HTML W3C, однако могут иметь место отличия в соответствии с [14] (8.11.4.2).

8.11.4.3 Расширения

8.11.4.3.1 Перечисления

В DVB-HTML некоторые атрибуты элемента могут быть принимать ограниченное количество значений. Семантика обработки атрибутов с ограниченным количеством значений должна быть в соответствии с [14] (8.11.4.2).

8.11.4.3.2 Исходные и текущие значения элементов управления формы

Элементы управления, которые принимают ввод от пользователя, например, фрагмент текста, опция, текст, пароль, флажок, переключатель, имеют как исходное, так и текущее значение. Имена и типы этих значений зависят от типа элемента управления в соответствии с [14] (8.11.4.3.2).

8.11.4.4 Аспекты системы

8.11.4.4.1 Доступ к документу

8.11.4.4.1.1 Java

Реализация DVB DOM выполняет интерфейс API org.dvb.dom как определено в 11.13.1 настоящего стандарта. Это обеспечит приложениям дескриптор DVB-HTML объектной модели документа.

8.11.4.4.1.2 ECMAScript

Реализация DVB DOM предоставляет приложениям объект документа, который является текущим документом DVB-HTML.

8.11.4.4.2 Модуль DOM DVB-HTML

Приложение DOM может использовать метод hasFeature интерфейса DOMImplementation, чтобы определить поддерживается модуль или нет. Строка функции для всех интерфейсов, перечисленных в этом модуле, является «DVBHTML» и величина версии «0».

8.11.4.4.3 Модификации (уточнения) DOM

Модификации DOM, включающие условия вызова DOM от модуля DOM DVB-HTML, синхронизацию с операциями рендеринга, изменения от ядра DOM, должны выполняться в соответствии с [14] (8.11.4.4.3.1, 8.11.4.4.3.2 и 8.11.4.4.3.3).

8.11.4.5 Служебные интерфейсы модуля DOM DVB-HTML

Этот пункт описывает различные служебные интерфейсы модуля DOM DVB-HTML, перечисленные ниже.

8.11.4.5.1 Интерфейс DVB-HTMLCollection

DVB HTMLCollection - список элементов DVB-HTML. К отдельному элементу может получить доступ любой порядковый (перечислимый) индекс или атрибут идентификатора элемента. Определение IDL должно быть в соответствии с [14] (8.11.4.5.1.1).

8.11.4.5.2 Интерфейс DVBHTMLDocument

DVBHTMLDocument является корнем иерархии документов DVB HTML и содержит весь контент. Помимо обеспечения доступа к иерархии, он также предоставляет несколько удобных способов доступа к определенным наборам информации из документа.

Описания атрибутов, открытых методов, закрытых методов, методов записи в соответствии с [14] (8.11.4.5.2.1 - 8.11.4.5.2.3).

8.11.4.6 Интерфейсы связанных элементов DVB-HTML

8.11.4.6.1 Интерфейс DVBHTMLElement

Все интерфейсы элемента DVB-HTML происходят из этого класса. Определение IDL и описания атрибутов должны быть в соответствии с [14] (8.11.4.6.1.1, 8.11.4.6.1.2).

8.11.4.6.2 Интерфейс DVBHTMLAnchorElement

Этот интерфейс модели элемент привязки соответствует тегу <a> DVB-HTML. Он предоставляет исходную привязку тега <a> с атрибутом href, или целевую привязку тега <a> с атрибутом идентификатора. Определение IDL и описания атрибутов должны быть в соответствии с [14] (8.11.4.6.2.1, 8.11.4.6.2.2).

8.11.4.6.3 Интерфейс DVBHTMLMapElement

Это интерфейс модели клиентской карты изображения. Определение элемента MAP в соответствии с [51] и [14]. Определение IDL и описания атрибутов должны быть в соответствии с [14] (8.11.4.6.3.1, 8.11.4.6.3.2).

8.11.4.6.4 Интерфейс DVBHTMLAreaElement

Это интерфейс модели участка изображения, которая может привести в действие триггеры, когда пользователь щелкает по участку. Определение IDL и описания атрибутов должны быть в соответствии с [14] (8.11.4.6.4.1, 8.11.4.6.4.2).

8.11.4.6.5 Интерфейс DVBHTMLButtonElement

Интерфейс DVBHTMLButtonElement представляет тег <button> DVB HTML. Определение IDL, описания атрибутов и методов должны быть в соответствии с [14] (8.11.4.6.5.1- 8.11.4.6.5.3).

8.11.4.6.6 DVBHTMLFormElement

Интерфейс DVBHTMLFORMElement охватывает режимы работы интерфейса совокупности (collection) и элемента (element). Это обеспечивает прямой доступ к входным элементам, а также атрибуты элемента формы. Определение FormElement в [51]. Определение IDL, описания атрибутов должны быть в соответствии с [14] (8.11.4.6.6.1, 8.11.4.6.6.2).

8.11.4.6.7 Интерфейс DVBHTMLFrameElement

Определение IDL, описания атрибутов интерфейса DVBHTMLFrameElement должны быть в соответствии с [14] (8.11.4.6.7.1, 8.11.4.6.7.2).

8.11.4.6.8 Интерфейс DVBHTMLFrameSetElement

Определение IDL, описания атрибутов интерфейса DVBHTMLFrameSetElement должны быть в соответствии с [14] (8.11.4.6.8.1, 8.11.4.6.8.2).

8.11.4.6.9 Интерфейс DVBHTMLIFrameElement

Определение IDL, описания атрибутов интерфейса DVBHTMLIFrameElement должны быть в соответствии с [14] (8.11.4.6.9.1, 8.11.4.6.9.2).

8.11.4.6.10 Интерфейс DVBHTMLImageElement

Определение IDL, описания атрибутов интерфейса DVBHTMLImageElement должны быть в соответствии с [14] (8.11.4.6.10.1, 8.11.4.6.10.2).

8.11.4.6.11 Интерфейс DVBHTMLObjectElement

Определение IDL, описания атрибутов интерфейса DVBHTMLObjectElement должны быть в соответствии с [14] (8.11.4.6.11.1, 8.11.4.6.11.2).

8.11.4.6.12 Интерфейс DVBHTMLInputElement

Определение IDL, описания атрибутов интерфейса DVBHTMLInputElement должны быть в соответствии с [14] (8.11.4.6.12.1, 8.11.4.6.12.2).

8.11.4.6.13 Интерфейс DVBHTMLOptionElement

Интерфейс DVBHTMLOptionElement должен иметь IDL и описания атрибутов в соответствии с [14] (8.11.4.6.13.1, 8.11.4.6.13.2).

8.11.4.6.14 Интерфейс DVBHTMLSelectElement

Этот интерфейс позволяет выполнять выбор одного или нескольких вариантов управления из списка. Некоторые графические агенты пользователя обеспечивают выбор единственного типа и нескольких типов виджетов. Опции могут быть сгруппированы в группу опций или непосредственно представлены для выбора. Определение IDL, описания атрибутов должны быть в соответствии с [14] (8.11.4.6.14.1, 8.11.4.6.14.2).

8.11.4.6.15 Интерфейс DVBHTMLTextAreaElement

Этот интерфейс представляет многострочную текстовую область. Он ссылается на тег DVB HTML <textarea>. Определение IDL, описания атрибутов должны быть в соответствии с [14] (8.11.4.6.15.1, 8.11.4.6.15.2).

8.11.5 Исключения DVB

Операции DOM в «исключительных» обстоятельствах могут вызывать исключения. В общем DVB-HTML вызывает исключения, определяемые согласно [53], однако некоторые методы в DVB-HTML при других обстоятельствах применяют другие исключения. В DVB-HTML определено, что DVBException в дополнение к DOMException из DOM 2 определяет дополнительные коды ошибок.

8.11.5.1 DVBException

Величина, возвращенная оператором **typeof** для DVBException, должна быть «**dvbexception**». Определения IDL и констант должны быть в соответствии с [14] (8.11.5.1.1, 8.11.5.1.2).

8.11.6 Привязки языка

8.11.6.1 Связывание ECMAScript

Связывание ECMAScript в соответствии с [14] (приложение AC).

8.11.6.2 Связывание Java

Связывание Java в соответствии с [14] (приложение AD).

8.11.7 Объектный модуль среды DVB

Приложение DOM может использовать метод hasFeature интерфейса реализации DOM, чтобы определить что этот модуль поддерживается.

8.11.7.1 Свободные переменные

DVB-HTML должен поддерживать доступ ECMAScript к определенным API через переменные, которые связаны в контексте каждой страницы:

- документ;
- навигатор;
- окна.

Функциональность свободных переменных описана в [14] (8.11.7.1).

8.11.7.2 Объекты среды хоста

8.11.7.2.1 Объект навигатор

Объект навигатор является непосредственно браузером. Он поддерживается в DVB-HTML атрибутами и методами в соответствии с [14] (8.11.7.2.1, 8.11.7.2.2).

8.11.7.2.2 Объект Window

Объект Window представляет окно верхнего уровня браузера или кадра. Существует только одно окно верхнего уровня в приложении DVB-HTML, родителем которого является NULL. Другие объекты в окне приложения представляют кадры в наборе кадров.

Объект Window поддерживается атрибутами DVB-HTML и методами в соответствии с [14] (8.11.7.2.2.1, 8.11.7.2.2.2, 8.11.7.2.2.3).

8.11.7.2.3 Объект расположения

Элемент, который анализирует uri, используется в поле расположения объекта окна и поддерживается в соответствии с [14] (8.11.7.2.3.1, 8.11.7.2.3.2).

8.11.8 Поддержка CSS

8.11.8.1 Модуль DOM CSS DVB

DVB-HTML не требуется поддержки модуля CSS DOM-2, однако требуется поддержка простого модуля CSS. Приложение может использовать метод DOM hasFeature реализации интерфейса DOM для определения поддержки этого модуля. Строкой функции для всех интерфейсов, перечисленных в этом модуле, является «DVBCSS» и «2.0».

8.11.8.1.1 Интерфейс DVBCSSInlineStyle

Интерфейс DVBInlineStyle должен быть реализован объектами, которые реализуют интерфейс Элемент (см. [53]), и которые представляют элементы, поддерживающие атрибут стиля. Интерфейс DVBCSSInlineStyle поддерживается атрибутами и методами в соответствии с [14] (8.11.8.1.1.1, 8.11.8.1.1.2).

8.11.8.1.2 Интерфейс DVBCSSStyle

Интерфейс **DVBInlineStyle** выполняется объектами, которые реализуют интерфейс элемента (см. [53]), который представляет собой корневой элемент документа. Интерфейс DVBCSSStyle поддерживается атрибутами и методами в соответствии со [14] (8.11.8.1.2.1, 8.11.8.1.2.2).

8.11.8.1.3 DVBCSSViewportRule

Интерфейс DVBCSSviewportRule представляет правило просмотра @viewport в таблице стилей CSS. Правило @viewport используется для определения на экране областей и отношения к видео приложению DVB-HTML. Интерфейс поддерживается атрибутами и методами в соответствии с [14] (8.11.8.1.3.1, 8.11.8.1.3.2).

8.11.8.1.4 Интерфейс DVBCSSViewportProperties

Интерфейс DVBCSSViewportProperties представляет механизм получения и настройки свойств в окне просмотра. Атрибуты этого интерфейса соответствуют всем указанным свойствам в 8.8.5.3.2. Интерфейс поддерживается атрибутами и методами в соответствии с [14] (8.11.8.1.4.1, 8.11.8.1.4.2).

8.12 Поддержка Cookie

8.12.1 Интерфейс DOM Cookie

Атрибут документа Cookie (см. 8.11.4.5.2 настоящего стандарта) предоставляет интерфейс для набора пар атрибут-значение, описанный в [54].

Атрибут документа Cookie для чтения/записи строки имеет следующие характеристики поведения:

- атрибуту может быть присвоено значение строки набора Cookie по правилам BNF в соответствии с [14] (8.12.1);

- если срок действия не указан, то время жизни cookie определяется временем жизни приложения DVB-HTML;

- после считывания атрибут возвращает строку get-cookie, состоящую из пар av для всех Cookie, срок действия которых не истек и они находятся в рамках текущего документа

get-cookie = av-pair* («;»av-pair)

Примечание – Если один и тот же атрибут определен для нескольких путей, которые применяются к текущему документу, то атрибут появляется в нескольких парах av.

8.12.2 Хранение и время жизни Cookie

8.12.2.1 Пределы хранения Cookie

Реализация терминала должна выполнять минимальные требования [54] к объему Cookie, включая требование хранения не менее 20 Cookie 4096 байтов.

8.12.2.2 Требования к персистентности Cookie

Если приложение запрашивает разрешение на доступ к постоянной памяти, то этот доступ предоставляется, реализации должны пытаться сохранить Cookie, пока они не устареют.

Если приложение не имеет разрешения для постоянного хранения или постоянная память недоступна, то Cookie должно сохраняться только в течение времени жизни приложения DVB-HTML.

8.12.2.3 Требования к конфендициальности Cookie

Допускается поддержка реализацией требований конфендициальности Cookie.

8.12.3 Определение контекста (область действия) Cookie

Правила определения области действия Cookie, доставки Cookie через Карусель Объектов DSM-CC и с использованием протокола HTTP должны быть в соответствии с [14] (8.12.3.1-8.12.3.3).

8.12.4 Поддержка Cookie протокола HTTP

Поддержка Cookie HTTP должна выполняться в соответствии с [14] (8.12.4.1, 8.12.4.2 и 8.12.4.3).

8.13 Поддержка строки агента пользователя HTTP

Запросы от агента пользователя DVB-HTML должны включать заголовок HTTP в виде:
User-Agent: <user-agent>
где <user-agent> является строкой агента пользователя DVB-HTML.

8.13.1 Строки агента пользователя

8.13.1.1 Текущие связанные строки агента пользователя

Примеры текущих строк агента пользователя приведены в [14] (таблице 60).

8.13.1.2 Строки агента пользователя БНФ

Строки агента пользователя BNF должны быть в соответствии с [14] (8.13.1.2).

8.14 Безопасность приложений DVB-HTML

8.14.1 Аутентификация файлов DVB-HTML

Безопасность приложений DVB-HTML обеспечивается в той же модели безопасности, как и приложения DVB-J, то есть права доступа к файлам связаны с оценкой подлинности (аутентификацией) приложений. Неаутентифицированные приложения будут функционировать в изолированной среде песочницы, в которой не будут работать конкретные формы локаторов и определенные API не будут доступны ECMAScript.

Прошедшим аутентификацию приложениям может быть предоставлено разрешение на использование локаторов ограниченного применения и API.

Агенты пользователя, интерпретируя подписанные приложения, должны делать доступным типы файла только для того приложения, для которого они будут иметь сигнатуру того же сертификата:

- DVB-HTML;
- ECMAScript;
- CSS;
- Внутренние файлы приложения (DVB-J файлы классов в соответствии с [11] (11.2.3)).

Локаторы, которые отключаются из-за ограничений по безопасности, будут вести себя так, как если бы ресурс был недоступен.

Вызовы API ECMAScript, которые не выполняются из-за ограничений безопасности, генерируют DVBException с кодом ошибки SECURITY_VIOLATION_ERR.

8.14.2 Выполнение кода расширения

8.14.2.1 Принципы безопасности

Принципы решения вопросов безопасности, возникающих во время выполнения кода расширения:

- платформа ECMAScript разрешает использование любого из методов ECMAScript выполнения кода расширения перечисленных ниже, для проверки подлинности не аутентифицированных приложений или аутентифицированных приложений, работающих в песочнице;
- платформа ECMAScript разрешает ограниченное использование методов расширения кода во время выполнения аутентифицируемого приложения, работающего вне песочницы, где ограниченное использование означает, что вход к коду расширения происходит только из соответственно аутентифицируемого источника;
- платформа ECMAScript не должна разрешать удаление предопределенных (встроенных) свойств объектов узла;
- платформа ECMAScript не должна разрешать замену предопределенного функционального свойства узла.

8.14.2.1.1 Использование расширения кода, выполненного в ECMAScript

Язык ECMAScript поддерживает выполнение кода расширения использованием следующих отдельных механизмов:

- функциональное свойство глобального объекта eval();
- объект Function.

Дополнительные механизмы, поддерживающие выполнение кода расширения, через объекты узла (платформы) должны быть в соответствии с [14] (8.14.2.1.1).

8.14.2.2 Расширения ECMAScript для надежного исполняемого кода

Этот пункт уточняет механизмы, определенные DVB, позволяющие использовать функцию Выполнения Расширения Кода (Runtime Code Extension; RCE), определенную спецификацией языка ECMAScript, и позволяющие использовать API DOM, не ставя под угрозу безопасность. Дополнительные механизмы в DVB-HTML отслеживают источник строк в системе и запрещают использование в RCE привилегированными приложениями строк из непроверенных внешних источников (как потенциально опасное).

8.14.2.2.1 Обеспечение безопасности распространением внутренних (безопасных) против внешних (небезопасных) строк

Использование ECMAScript поддерживает совместимость (в смысле безопасности) приложений, различая значения строки внутреннего типа, когда значение строки исходит изнутри языка ECMAScript или из текстовых строк, находящихся внутри приложения и строки внешнего типа, когда значение строки исходит из объектов узла (подозреваемых как потенциально опасных).

Этот подход имеет преимущество поддержания совместимости контента с контентом ECMAScript, записанным для существующих браузеров к W3C и спецификациям ECMA. Детализации преимуществ представлена в [14] (8.14.2.2.1).

8.14.2.2.2 Изменение ECMA [10] для поддержки внутренних и внешних строк

Нормативные изменения ECMA [10] для дифференцирования величин двух видов строк должны быть в соответствии с [14] (8.14.2.2.2). Используется соглашение о редактировании, дополнения к [14].

8.14.2.3 Источники опасных (внешних) строк

8.14.2.3.1 Источники опасных строк в ECMAScript

Следующие источники опасных строк в языке ECMAScript должны быть типизированы как источники внешних строк:

а) строки, полученные в результате операций на внешних строках;

б) строки, полученные в результате преобразования кодом String.fromCharCode символа в строку.

8.14.2.3.2 Источники объектов узла

Все строки, возвращаемые API Объекта Узла, определямые настоящим стандартом, должны быть типизированы как внешние строки.

Все возвращаемые строки должны быть типизированы как внешние строки.

Любые API Объекта Узла, предоставляемые платформой, помимо требуемых в настоящем стандарте, должны вернуть внешние строки, если нет сведений об их безопасности.

8.14.2.4 Использование строк в расширениях кода во время Выполнения Расширения Кода (Runtime Code Extension, RCE)

Использование внешних строк или объектов Строка в RCE должно выполняться в соответствии с [14] (8.14.2.4).

8.14.2.5 Защита от мутации объектов узла

Любая попытка удалить или заменить предопределенные свойства объекта узла влечет за собой DVBException с кодом ошибки SECURITY_VIOLATION_ERR. Любая попытка удалить или заменить предопределенные функции свойств объекта узла влечет за собой DVBException с кодом ошибки SECURITY_VIOLATION_ERR.

8.14.3 Безопасность использования документов из разных приложений

8.14.3.1 Ограничения на элементы DOM, введенные для безопасности

По условиям обеспечения безопасности доступ документам, предоставляемым из других приложений, не разрешен.

Аксессор (средство доступа) к свойствам на окна и объекты документа узла должен ограничивать доступ к свойствам и функциям этих объектов, когда контекст документа, в котором выполняется доступ, находится в другом приложении DVB HTML или в другом домене относительно документа, открытого в окне:

- в случае таких контекстов аксессор окна должен разрешать только присвоение строки к свойству расположения: установка любого другого свойства окна или получение любого свойства окна должны быть исключением;
- в случае таких контекстов аксессор документа не должен предоставлять доступ: установка или получение любого свойства документ должны быть исключением.

Правила обработки исключений и выполнения доступа к свойствам, методам и переменным объекта окна должны быть в соответствии с [14] (8.14.3.1).

8.15 Разрешения DVB-HTML

Описываются разрешения, определяющие песочницу и доступность для неподписанных приложений и разрешений, которые могут быть запрошены в файле запроса разрешения, отображаемого в модели приложения HTML DVB. Приложение ADVB HTML обладает следующими способами доступа за пределы песочницы:

а) загрузку новых ресурсов могут вызвать ссылки на странице, встроенные на следующих этапах;

б) сценарий может получить доступ к API по Мосту Java;

в) доступ к ресурсам могут получить свойства в объектах узла, доступные сценарию.

Атрибут расположения объекта Window.

Атрибут функции open() объекта Window.

г) ссылки URL () в атрибутах CSS.

В случае в), когда установка атрибутов через интерфейсы DOM осуществляется только при использовании атрибута, применяются ограничения по а). В некоторых случаях разрешение относится только к API, доступным по Мосту Java.

8.15.1 Разрешения для неподписанных приложений

Политика безопасности MHP включает набор ресурсов, которые всегда гарантированно предоставляются приложениям, если приложения выполняются. Неподписанные приложения имеют доступ только к этим ресурсам. В [14] (8.15.1) определено отображение этих ресурсов для приложений DVB-HTML. Приложениям DVB-HTML всегда предоставляется необходимое разрешение. Ресурсы, имеющие гарантированные полномочия (разрешения) для неподписанных приложений DVB-HTML, представлены в [14] (8.15.1.1 - 8.15.1.11).

8.15.2 Дополнительные полномочия для подписанных приложений

Подписанные приложения могут дополнительно запрашивать получение дополнительных полномочий. Эти полномочия запрашиваются при использовании файла запроса полномочия в соответствии с [14] (12.6). Определяется отображение элементов в файле запроса на полномочия, которые может предоставить терминал MHP в ответ на запрос. Кроме того, определяется влияние на приложение DVB-HTML.

В случае org.dvb.security только, подписанные приложения PrivilegedRCEPermission с файлом запроса разрешения могут запросить разрешение, которое является более строгим, чем предоставляемое приложениям без файла запроса разрешения.

Параметры запросов на получение полномочий для подписанных приложений DVB-HTML приведены в [14] (8.15.2.1-8.15.2.13).

8.16 Нормирование параметров данных различного характера

8.16.1 Значения данных

8.16.1.1 Синтаксис значений данных

Синтаксис абсолютных значений данных настоящим стандартом не определяется.

8.16.2 Значения времени суток (часов)

8.16.2.1 Синтаксис значений времени суток

Синтаксис значений времени суток должен быть в соответствии с [14] (8.16.2.1).

8.16.2.2 Смещения значений времени

Смещения значений времени используются для определения момента, когда элемент должен начинаться или заканчиваться по отношению к другому значению времени.

Значение смещения имеет следующий синтаксис:

offset-value ::= «+» (offset-value)

Значение смещения включает в себя знак значения тактовой частоты и указывает положительное смещение. Смещение измеряется от подразумеваемого начала времени.

8.16.3 Нереализуемые локаторы

Локаторы не могут быть реализуемы по нескольким причинам:

- локатору требуется коррекция, для которой полномочия или ресурсы недоступны;
- локатору требуется сеть или доступ к интерактивному каналу, для которых полномочия недоступны;
- локатору требуется доступ к данным, который заблокирован механизмом условного доступа;
- локатору требуется доступ к проверке подлинности файла, при которой аутентификация не удается;
- локатор требует монтажа новой файловой системы, которая не может быть установлена в то же время, в которое устанавливается файловая система.

8.16.3.1 Презентация локаторов в DVB HTML

Презентация локаторов в DVB HTML выполняется в соответствии с [14] (8.16.3.1).

8.16.4 Связь между HTTP и HTTPS

Строка агента пользователя (8.13.1 настоящего стандарта) отправляется в составе всех запросов в заголовке HTTP и HTTPS (6.3.7 настоящего стандарта).

8.16.5 Конкретные локаторы приложения DVB-HTML

Приложения DVB-HTML имеют два конкретных формата локаторов с семантикой, определенной в контексте DVB-HTML:

- расширенная форма DVB локатора (8.16.5.1 настоящего стандарта);
- локаторы выхода, используемые для самостоятельного завершения приложения (8.16.5.2 настоящего стандарта).

8.16.5.1 Локаторы DVB расширенной формы

Параметры локаторов DVB расширенной формы должны быть в соответствии с [14] (8.16.5.1.1 - 8.16.5.1.4).

8.16.5.2 Локаторы выхода

Параметры локаторов выхода должны быть в соответствии с [14] (8.16.5.2).

8.16.6 Домены

В соответствии с [14] (8.16.6) доменами могут быть:

- доменное имя сервера, который обслуживал документ, если оно может быть идентифицировано;
- пустая строка («») для страницы с dvb:locator;
- во всех других случаях Null.

9 Модель приложений МНР

9.1 Приложения МНР, связанные службой

9.1.1 Базовые элементы управления жизненным циклом

Основным элементом управления жизненным циклом приложения МНР является селекция (выбор) служб. Для приложений вещания и IPTV для выбора службы используется EPG. Службой является модуль, включающий операции представления и выполнения контента в МНР. Служба МНР может быть представлена группой частей контента, которые предназначены для совместного представления конечному пользователю. Служба может состоять из контента службы, включая потоки аудио/видео, потоки данных, приложение информации о службе и приложение сигнализации. Служба «целевой упакованной медиа» состоит из элементов, которые хранятся на носителе данных, в том числе потоки аудио/видео, потоки данных, приложения информации о службе и приложения сигнализации. Возможны другие формы службы, такие как сохраненная служба (см. [11] (9.6.2)).

Каждая служба, представленная платформой МНР, представлена контекстом службы.

Контекст службы в приложении DVB-J представлен экземпляром класса ServiceContext. Несколько приложений DVB-J представляются в одном контексте службы, количество объектов ServiceContext, представляющих контекст службы, зависит от реализации, но каждое приложение видит только один такой экземпляр. Детализация аспектов выбора служб представлена в [11] (9.1.1) с замечаниями и изменениями по [14] (9.1.1).

9.1.2 Запуск приложений

Операция запуска (старта) приложения должна выполняться платформой МНР после окончания операции выбора службы вещания в соответствии с [11] (9.1.2). Общие характеристики механизма управления жизненным циклом приложений МНР установлены в [11] (9.1.1).

9.1.3 Поддержка выполнения одновременно нескольких приложений МНР

Должна обеспечиваться поддержка одновременного представления и выполнения нескольких приложений МНР в границах, предусмотренных службой, в соответствии с [11] (9.1.3).

9.1.4 Остановка приложений

МНР должна поддерживать остановку приложений самими приложениями (при использовании API этих приложений) и при ручном управлении от терминала МНР. Примеры ситуаций, в которых допускается остановка приложений, приведены ниже:

- выбор новой службы для замены службы, выбранной ранее;
- остановка одного приложения другим приложением по решению администратора приложений из соображений безопасности;
- по решению вещателя об остановке приложения;
- из-за дефицита ресурсов терминала МНР.

Детализированные характеристики этих ситуаций должны быть в соответствии с [11] (9.1.4).

9.1.5 Персистентность приложений за границами службы

МНР должна поддерживать персистентность (сохраняемость) приложений за границами, предусмотренными службой, в соответствии с [11] (9.1.5). Условия сохраняемости приложений в случае потери режима Карусели должны быть в соответствии с [11] (6.2.5.3).

9.1.6 Управление автоматическим запуском

МНР должна поддерживать управление автоматическим запуском приложений вещания в условиях, определенных [11].

9.1.7 Настройка без выбора службы

В соответствии с [11] (9.1.7) МНР не должна поддерживать попытки пользователя терминала настройки МНР на транспортные потоки методами, которые не используют метод выбора службы вещания МНР.

9.1.8 Поддержка возможности выбора службы при активизации приложений

MHP должна поддерживать возможность выбора службы при активизации приложений использованием API выбора службы. Программный интерфейс приложений выбора службы включает класс ServiceContext, что дает доступ к контекстам, в которых могут быть представлены службы. Детализированное описание процедур выбора службы должно быть в соответствии с [11] (9.1.8).

9.1.9 Поддержка MHP кэшированных приложений

MHP поддерживает использование кэшированных (и сохраненных на терминале) приложений для ускорения загрузки приложений при условии, если приложение сигнализируется в описании приложения (таблица AIT) и содержит дескриптор application_storage_descriptor. Уточнения условий использования кэшированных и сохраненных на терминале приложений представлены в [11] (9.1.9).

9.2 Параметры модели приложений DVB-J

Примечание – Терминалы MHP, специализирующиеся на обработке пакетизированных профилей медиа или профилей IPTV, могут не поддерживать динамическое обновление описания приложения службы.

9.2.1 Поддержка запуска приложений

MHP должна поддерживать запуск приложений DVB-J любым из способов, установленных для приложений MHP [11] (9.2.1). Листинг API и процедуры активизации API должны быть в соответствии с [11] (приложение S).

9.2.2 Поддержка остановки приложений

MHP должна поддерживать остановку приложений DVB-J по любой из причин, перечисленных для приложений вещания MHP по 9.1.4 настоящего стандарта. Описание процедур остановки в соответствии с [11] (9.2.2).

9.2.3 Состояния жизненного цикла приложений DVB-J

9.2.3.1 Введение

Состояния жизненного цикла приложений DVB-J и параметры состояний платформы MHP должны поддерживаться виртуальной машиной Xlet. Этот подпункт содержит описание жизненного цикла модели Xlet для API DVB-J. Ниже описываются возможности Xlet в каждом состоянии и методы, которыми администратор приложений влияет на состояние жизненного цикла. Этот подпункт определяет жизненный цикл экземпляра приложения DVB-J. Жизненный цикл приложения DVB-J и экземпляра приложения одни и те же, за исключением того, что когда экземпляр приложения будет уничтожен, то приложение только временно разрушено, а затем становится не загруженным.

9.2.3.2 Состояние жизненного цикла машины для экземпляров приложений DVB-J

Машина Xlet обеспечивает функционирование программного интерфейса Xlet с учетом следующих условий:

- допускается короткая задержка запуска программного интерфейса Xlet;
- допускается возможность возникновения условий, в которых программный интерфейс приложений Xlet не предоставляет свою службу;
- обеспечивается возможность уничтожения программного интерфейса Xlet в любое время.

Диаграмма модели состояний машины Xlet представлена на рисунке 5. Состояния Xlet определены в [11] (таблица 8).

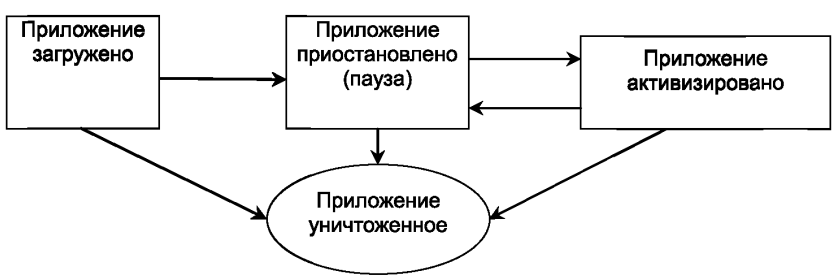


Рисунок 5 – Модель состояний виртуальной машиной Xlet

Детализированные параметры допустимых состояний жизненного цикла экземпляров приложения DVB-J платформы MHP должны быть в соответствии с [11] (9.2.3.2, таблица 8).

Каждый раз при запуске экземпляра приложения DVB-J (т.е. при вызове конструктора объекта, реализующего Xlet), оно должно выполняться в собственном новом экземпляре виртуальной машины в соответствии с 11.2.1 настоящего стандарта.

Только приложение DVB-J может определить возможность предоставления службы, для которой она была создана. Типичная последовательность выполнения приложения DVB-J показана в таблице 12.

Таблица 12 – Типичная последовательность выполнения приложения DVB-J

Администратор приложения	Приложение DVB-J
Администратор приложения создает новый экземпляр Xlet	По умолчанию конструктор вызывает Xlet (без аргументов), Приложение DVB-J в состоянии **Загрузка**
Администратор приложения создает необходимый объект контекст для работы приложения DVB-J и инициализирует Xlet	Приложение DVB-J использует объект контекст для собственной инициализации. В настоящее время находится в состоянии **Пауза**
Администратор приложений решил, что это подходящее время для приложения DVB-J для выполнения службы, поэтому он сигнализирует приложению DVB-J о входе в состояние Активное	Приложение DVB-J получает все необходимые ресурсы и начинает выполнять службу
Администратор приложений не нуждается в Приложении DVB-J для выполнения службы и сигнализирует Приложению DVB-J остановить выполнение службы	Приложение DVB-J прекращает выполнять свою службу и может освободить некоторые ресурсы, которые оно в настоящее время занимает
Администратор приложений установил, что приложение DVB-J больше не нужно, или нужно освободить место в памяти для приложения более высокого приоритета, поэтому он сигнализирует приложению DVB-J, что он является кандидатом на уничтожение	В зависимости от исполнения приложение DVB-J или сохраняет свое состояние, или, выполняя предпочтение пользователя, выполняет очистку

9.2.4 Интерфейс API Xlet

MHP использует программный интерфейс приложений Xlet для сигнализации жизненного цикла приложения. Сигнализация об изменении состояния жизненного цикла должна выполняться применением принципа обратного вызова, сигнализирующего об изменении состояния.

Параметры программного интерфейса приложений Xlet должны быть в соответствии с [11] (11.7.1).

9.2.4.1 Семантика изменений состояния Xlet

Состояние Xlet может изменяться при наличии вызова одного из методов интерфейса Xlet или при внутреннем переходе состояния и уведомления администратора приложений через XletContext Object.

Семантика важных изменений состояния Xlet:

– вызовы на Xlet: этот интерфейс указывает на успешное изменение состояния только при успешном возвращении вызова;

– вызовы на XletContext: методы notifyDestroyed () и notifyPaused () указывают на изменение состояния на входе.

Метод resumeRequest () указывает на отсутствие изменения состояния, а не только на запрос на изменение состояния.

Детализация семантики изменения состояния должна быть в соответствии с [11] (9.2.4.1).

9.2.4.2 Состояния Xlet, в которых допускаются запросы управления состоянием

Состояния Xlet, в которых допускаются запросы управления состоянием, должны быть в соответствии с таблицей 13.

Таблица 13 – Состояния Xlet, в которых допускаются запросы управления состоянием

Вид вызова	Состояния Xlet
уведомление о ликвидации	все состояния
уведомление о паузе	активное
запрос состояния	пауза

Вызовы этих методов должны игнорироваться, если Xlet находится в любом другом состоянии.

9.2.5 Поддержка одновременного выполнения нескольких приложений DVB-J

Платформа DVB-J в составе платформы MHP обеспечивает одновременное выполнение нескольких приложений DVB-J. При этом должно обеспечиваться совместное использование ресурсов MHP несколькими приложениями DVB-J, в том числе для совместного использования фокуса ввода и фокуса вывода.

9.2.5.1 Управление приложениями DVB-J другими приложениями DVB-J

Настоящий стандарт обеспечивает поддержку управления жизненным циклом приложения DVB-J другим приложением DVB-J. Эта функция позволяет поставщикам услуг создавать свои собственные «приложения запуска», которые обеспечивают пользователю презентацию приложений DVB-J, что позволяет пользователю запускать эти приложения DVB-J (см. 11.7.2 настоящего стандарта).

Примечание – Реальное управление временем жизни приложений DVB-J осуществляет только администратор приложений. В настоящем разделе определяются API, которые позволяют приложениям DVB-J запрашивать администратора приложений о запуске, остановке, приостановке и возобновлении приложений DVB-J.

9.2.5.2 Управление фокусом ввода

Фокус ввода определяется следующим образом:

- приложение, имеющее фокус ввода, способно принимать ввод событий пользователя;
- другие приложения, не имеющие фокуса ввода, могут запрашивать получение ввода подмножества событий пользователя через выделенные API (см. 11.3.2.2 «org.dvb.event»).

Приложение DVB-J имеет фокус ввода только в том случае, если java.awt.Component, имеющий фокус, принадлежит дереву компонентов этого приложения.

9.2.5.3 Другие ресурсы

Интерфейсы API, определенные, в настоящем пункте, оказывают поддержку выделению ресурсов/аннулированию ресурсов и уведомлению аннулирования ресурса. Семантика интерфейсов API не определяет условий, при которых доступ к ресурсу предоставляется или аннулируется.

Спецификация MHP описывает среду, в которой существут несколько приложений. Это означает, что несколько приложений могут конкурировать за доступ к одному и тому же ресурсу. API уведомления о ресурсе, описанные в 11.7.5 настоящего стандарта, позволяют терминалу информировать приложение, занимающее этот ресурс, о том, что другое приложение пытается получить доступ к этому ресурсу. Они также предоставляют владельцу ресурса и стороне, запрашивающей ресурс, возможность обмена данными, используя частные средства. Этот частный обмен данными отражен объектом request_data, который запрашивающая сторона может передать владельцу ресурса. Семантика этого объекта является частной и известной обоим приложениям. Дополнительные условия поддержки выделения ресурсов/аннулирования ресурсов описаны в [11] (9.2.5.3).

9.3 Модель DVB-HTML

9.3.1 Приложение DVB-HTML

9.3.1.1 Определение приложения DVB-HTML

Приложение DVB-HTML определяется как набор документов, выбранных из семейства элементов DVB-HTML и форматов контента в соответствии с настоящим стандартом. Величина множества описывается границами приложения.

9.3.1.2 Агент пользователя

Агентом пользователя является приложение, которое интерпретирует формат контента (в данном случае документов DVB-HTML).

Примечание – Он может быть реализован в виде плагина.

9.3.1.3 Актор DVB-HTML

Актор DVB-HTML определяется как область деятельности или процесса, участвующего в работе конкретного набора документов DVB-HTML для некоторых приложений DVB- HTML, а также любые экземпляры контекста для этих данных. Актор работает внутри агента пользователя (он может быть собственным, плагином или загруженным). Характер процесса явно не определяется, так как он зависит от природы самого агента пользователя. В любом агенте пользователя может присутствовать более одной такой области деятельности.

Для каждого запущенного приложения DVB-HTML существует один актор DVB-HTML, каждое приложение DVB-HTML может состоять из нескольких документов, некоторые из которых могут отображаться одновременно.

На рисунке 6 представлена схема взаимосвязей между акторами и приложениями

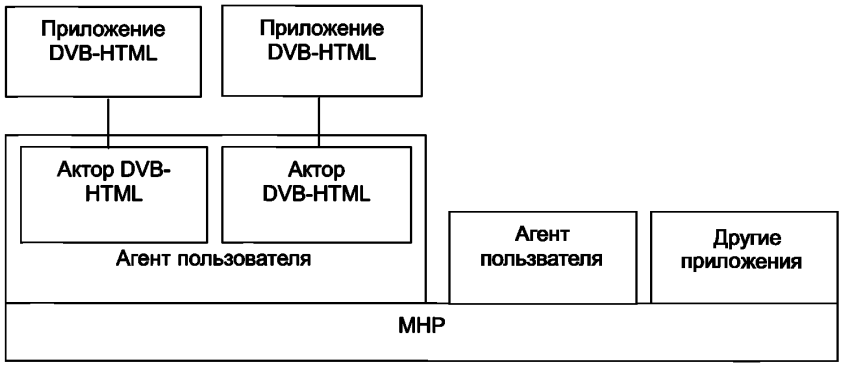


Рисунок 6 – Взаимосвязи между акторами и приложениями

9.3.1.4 Границы приложений

Границы приложения определяют размеры приложений DVB-HTML. Любой контент документа за пределами границы приложения не является частью приложения HTML-DVB и не должен быть доступным для ссылок приложения HTML- DVB. Терминал MHP может содержать конкретную реализацию механизма, позволяющего конечному пользователю принять решение о доступе к ресурсу за границей приложения, но эта возможность не находится в контексте исходного приложения HTML DVB. Это пространство имен может быть использовано вещателем для управления периметром навигации пользователя и для реализации MHP с более эффективным предварительным выбором приложений.

Комплект документов, составляющих приложение HTML-DVB платформы MHP, определяется регулярным выражением на языке локатора. Как правило, локатор состоит из текстовой строки в форме, определенной в соответствии с [55] и действует как связующее звено, скрепляющее приложение: scheme://host/dir1/dirn/file#subref.

Регулярное выражение может быть определено как шаблон строки, который позволяет проверить соответствие (или не соответствие) данной строки элементу текста. Пример регулярного выражения:

https? :// / www \. (dvb | etsi) \ org / [a-z0-9 /] + \ html?

9.3.1.4.1 Синтаксис регулярного выражения

Синтаксис регулярного выражения должен быть в соответствии с [11] (9.3.1.4.1).

9.3.2 Жизненный цикл приложения DVB-HTML

9.3.2.1 Сигнализация

Сигнализация приложения DVB-HTML выполняется в соответствии с разделом настоящего стандарта.

Администратору приложения может быть предложено запустить приложение DVB-HTML автоматическим запуском или через API запуска приложения. Приняв запрос на запуск приложения DVB-HTML (т.е. в AIT появляется AUTOSTART или PREFETCH или пользователь, создающий экземпляр). В случае отсутствии приложения с тем ApplicationID, который уже созает экземпляр класса, Администратор приложений должен пытаться найти подходящего агента пользователя. Он может в этот момент начать предварительную выборку.

Если Администратор приложения не может создать экземпляр агента пользователя из-за отсутствия ресурсов или из-за отсутствия подходящего и доступного агента пользователя, то любая предварительная выборка может быть прервана, и любой сигнал запуска может быть проигнорирован. Это зависит от времени начала автозапуска приложений DVB-HTML. Для выбранного с упреждением приложения DVB-HTML требуется триггер, который переносит время, в которое приложение должно начать предоставление службы.

Модель DVB-HTML MHP должна предусматривать (после поступления заявки на запуск приложения DVB-HTML) поиск агента пользователя и поиск актора.

MHP в соответствии с [11] (9.3.2.2) обеспечивает поддержку запуска приложений DVB HTML на интервале их жизненного цикла одним из следующих способов:

- по требованию администратора приложений;
- по условиям автоматического запуска;
- посредством запуска приложения, запускающего программный интерфейс приложений.

Актор DVB-HTML может находиться в одном из следующих состояний приложения DVB-HTML:

- «загрузка»;
- «активное»;
- «приостановленное»;
- «уничтоженное» (Destroyed);
- «уничтоженное - стертое» (Killed).

Переходы актора DVB-HTML из одного состояния в другое состояние могут выполняться при появлении следующих событий:

- триггера – запроса на переход в новое состояние;
- триггера – запроса о переходе в новый документ;
- при изменении данных AIT.

Агент пользователя может выступать как несколько акторов, акторы могут находиться в некоторых из перечисленных состояний в одно время, акторы помечены уникальными идентификаторами приложения.

Приложение DVB-HTML перемещается между документами, а документы остаются в пределах границ приложения DVB-HTM.

В приложении DVB-HTML существующий документ обычно заменяет ссылки, но атрибуты на ссылки могут присутствовать, что позволяет новому и старому документу быть видимыми в том же самом временном отрезке (time.y.)

Состояние модели управления жизненным циклом приложения DVB-HTML, описанной в этом пункте, отражается сигнализацией по 10.6 настоящего стандарта, абстрактным представлением работы приложения DVB-HTML и учитывает виды ресурсов, в которых нуждается агент пользователя для функционирования ресурсов, связанных с выходом (визуализация), входом (события захвата) и связи (наличие контента). Модель в значительной степени абстрактна и иллюстративна, она не подразумевает применения любой стратегии управления ресурсами и не предназначена для чрезмерного ограничения реализаций агента пользователя.

На рисунке 7 приведена диаграмма состояний и переходов между ними.

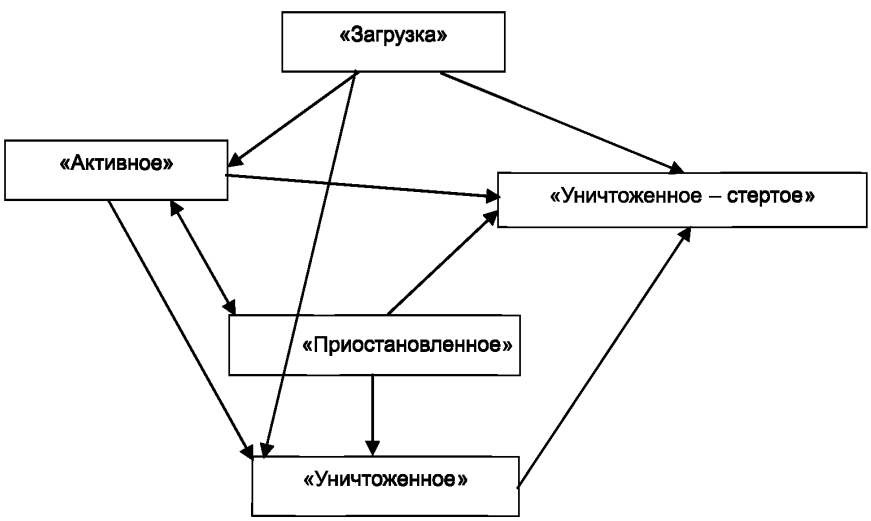


Рисунок 7 – Диаграмма состояний и переходов между ними

9.3.3 Модель состояний

Модель состояний приложения DVB-HTML соответствует состояниям виртуальной машины. В каждом из состояний должно обеспечиваться:

- в состоянии «загрузка» (Loading) – доступ к ресурсам контента и ресурсам сигнализации. Контент зрителю не представляется;

- в состоянии «активное» (aktive) – полный доступ актора к контенту действующего документа и всем ресурсам MHP в соответствии с правилами управления ресурсом и условиями обеспечения безопасности;

- в состоянии «приостановленное» (Paused) – переход в рабочее состояние с минимизированными возможностями. Если администратор приложений или агент пользователя нуждаются в ресурсах в других целях, актор может быть перемещен в приостановленное состояние, в котором он не имеет полного доступа к ресурсам. При активизации актора он возвращается в свое предыдущее состояние;

- в состоянии «уничтоженное» (Destroyed) – потеря ресурса контента и сохранение возможности для актора запуска приложения DVB-HTML;

- в состоянии «уничтоженное - стертое» (Killed) – потеря всех ресурсов. Переход в состояние «уничтоженное» (Killed) инициирует сброс актора и возвращение платформе MHP ресурсов.

9.3.3.1 Состояние «загрузка»

Состояние «загрузка» вводится в «машину состояний» только один раз на интервале жизни актора DVB-HTML. Процедура загрузки содержит перечень фаз этого состояния, каждую из которых можно рассматривать как часть этого состояния в соответствии с [14] (9.3.3.1.1 - 9.3.3.1.6).

9.3.3.2 Состояние «активное»

Состояние «активное» является стационарным (устойчивым) состоянием. Перечень фаз состояния в соответствии с [14] (9.3.3.2.1 - 9.3.3.2.6).

9.3.3.3 Состояние «приостановленное»

Перечень фаз состояния «приостановленное» и их характеристики в соответствии с [14] (9.3.3.3.1 - 9.3.3.3.5).

9.3.3.4 Состояние «уничтоженное» (Destroyed)

Перечень фаз состояния «уничтоженное» и их характеристики в соответствии с [14] (9.3.3.4.1 - 9.3.3.4.5).

9.3.3.5 Состояние «уничтоженное - стертое» (Killed)

Перечень фаз состояния «уничтоженное - стертое» в соответствии с [14] (9.3.3.5.1 - 9.3.3.4.6).

9.3.4 События приложений в активном состоянии

Каждая страница в приложении должна получать загрузку и разгружать события с ограничениями в соответствии с [14] (9.3.4). Иллюстрация диаграммы состояний представлена в [14] (рисунок 11).

9.3.4.1 Очередь обработки событий

Обработка очереди событий должна выполняться в соответствии с [14] (9.3.4.1).

9.4 Управление ресурсами, которые используются разными приложениями

Некоторые загруженные резидентные приложения, являющиеся расширениями существующего документа, могут выполнить часть функций навигатора, например, приложение монитор, определенное в [13]. Применение такого загруженного программного обеспечения не должно противоречить требованиям настоящего стандарта.

9.4.1 Экземпляры приложения, работающие в одном и том же контексте службы

Конфликт ресурсов между двумя приложениями сигнализации выполняется как часть службы и работает в контексте той же самой службы. Конфликт должен разрешаться посредством сигнализации приоритета в поле aplication_priority в описании приложения для каждого приложения. При сравнении приоритетов двух приложений, приложение с более высокой величиной application_priority нужно считать более важным для сохранения.

По умолчанию, только приложение, которому принадлежит ресурс, имеет право изменить состояние/настройки/конфигурацию того ресурса. Если резидентное приложение (например, навигатор) производит изменения в состоянии/настройках/конфигурации ресурса, терминал MHP должен сообщить приложению MHP, которому прежде принадлежал этот ресурс, что приложение MHP потеряло владение ресурсом.

Примечание – Единственным исключением из этого правила является случай, когда приложения, работающие внутри контекста службы, совместно владеют обработчиками компонентов службы в этом контексте услуг в соответствии с 11.6.2 настоящего стандарта.

Подробности взаимодействия экземпляров приложения представлены в [14] (10.4.).

9.4.2 Экземпляры приложения, не работающие в одном и том же контексте службы

Настоящий стандарт не устанавливает параметры политики разрешения конфликтов для экземпляров приложения, не работающих в одном и том же контексте службы.

9.5 Жизненный цикл интерфейсов Xlet, встроенных в приложения DVB-HTML

Состояние интерфейсов Xlet зависит от состояния страницы, которая его содержит, а так же от состояния содержимого приложения.

Жизненный цикл Xlet связан с жизненным циклом приложений и страницы HTML, но Xlet в результате своих собственных действий может находиться в менее активном состоянии, чем содержание страницы HTML. Например, Xlet может уничтожить себя даже если страница, в которой он содержится, активна.

9.5.1 Запуск встроенных Xlet

Для загрузки своих классов для каждого интерфейса Xlet с указанным тегом объекта создается новый загрузчик классов. Для загрузчиков классов Xlet применяется стандартная семантика MHP. Например, не должны использоваться загрузчики классов, доступные другим Xlet. Подробности запуска встроенных интерфейсов Xlet представлены в [14] (9.5.1).

9.5.2 Завершение встроенных Xlet

Когда страница становится невидимой в процессе навигации или когда содержание приложения HTML DVB устанавливается в состояние Пауза, все встроенные Xlet, которые находятся в активном состоянии, необходимо ввести в состояние Пауза. Детализация завершения встроенных интерфейсов Xlet представлена в [14] (9.5.2).

9.5.3 Общие вопросы обработки встроенных Xlet

Разрешается существование нескольких Xlet на документ XHTML и нескольких Xlet в службе DVB.

К экземплярам встроенных Xlet предъявляются следующие требования:

- они не должны быть перечислены в AIT;
- они не должны быть представлены в листинге приложения с перечислением API запуска (через приложения DVB HTML);
- они должны наследовать транспортные протоколы, содержащие приложения HTML.

Если встроенный Xlet находится в активном состоянии, то он должен иметь доступ к DOM при условии, что реализация будет иметь доступа к DOM в других состояниях.

9.6 Службы и приложения, не связанные с традиционными службами DVB

В дополнение к 9.1 настоящего стандарта, описывающему приложения, жизненный цикл которых управляется через службы, в этом подразделе указаны приложения, жизненный цикл которых не связан со службами. К таким приложениям относятся:

- приложения, сигнализизация которых поступает через интерактивный канал (согласно 9.6.1 настоящего стандарта);
- приложения, полученные из памяти терминала MHP (согласно 9.6.2 настоящего стандарта).

9.6.1 Загрузка приложений из интерактивного канала

При загрузке приложений из интерактивного канала сигнализация приложения поступает из интерактивного канала, а не из канала вещания. Подробности загрузки в соответствии с [11] (9.6.1).

9.6.2 Сохраненные службы

Сохраненные службы являются инкапсуляцией одного или более приложений. Жизненный цикл сохраненной службы должен соблюдать правила, определенные для службы в [11] (9.1).

Правила обработки сохраненных служб представлены в [11] (9.6.2).

В соответствии с [14] (9.6) термины «описание приложения» и «дескриптор приложения» должны интерпретироваться относящимися к таблице информации о приложении (AIT), определенной в [14] (10.4.).

9.7 Жизненный цикл приложений доступа в Интернет

9.7.1 Общие вопросы

Терминалы MHP, поддерживающие приложения доступа в Интернет, в каждый момент времени должны поддерживать выполнение одного приложения доступа в Интернет. Требования поддержки выполнения больше чем одного приложения доступа в Интернет в данный момент времени не предъявляются.

Не предъявляются требования поддержки одновременного выполнения приложений доступа в Интернет и приложений MHP.

Допускаются случаи, когда в терминалах MHP приложения доступа в Интернет и приложения MHP могут быть в памяти терминала и могут обрабатываться одновременно, но не могут совместно использовать доступ к экрану. Такие терминалы MHP нужно рассматривать как поддерживающие приложения доступа в Интернет и приложения MHP одновременно в остальной стандарта части, не относящейся к доступу к экрану. В этих терминалах MHP любые приложения MHP в активном состоянии могут быть помещены в приостановленное состояние (Пауза), когда они теряют доступ к экрану. Настоящий стандарт не определяет приоритеты доступа к ресурсам памяти в случае конфликта между приложениями MHP и приложениями доступа в Интернет.

9.7.2 Запуск приложений доступа в Интернет из приложений MHP

В тех случаях, когда терминалы MHP поддерживают одновременное выполнение Интернет приложений и приложений MHP в одно и то же время, Интернет приложение запускается по запросу в соответствии с [11] (9.7.2).

9.7.3 Выбор служб DVB из приложений доступа в Интернет

В MHP приложения доступа в Интернет должны поддерживать запуск конечным пользователем служб DVB (то есть использованием URL 'dvb:') и приложений MHP, сообщенных по интерактивному каналу (использованием файла описания приложения, на который ссылаются URL HTTP). Эти процедуры должны быть обработаны стандартным механизмом выбора службы (например, реализацией API выбора службы) в соответствии с [11] (9.7.3).

9.8 Плагины

Уточнения правил применения плагинов, определенных в 5.4 настоящего стандарта, должны быть в соответствии с [11] (9.8).

9.9 Хранение и кэширование приложений

Описываются общие правила хранения, которые применяются ко всем сохраняемым приложениям, версии управления и правила удаления сохраненных приложений. (см. также 9.1.9, 9.6.2 настоящего стандарта).

9.9.1 Хранение файлов

Перечень метаданных, которые должны содержаться в сохраняемом файле (когда файл является частью приложения) должен быть в соответствии с [11] (9.9.1).

9.9.2 Управление версией

Чтобы иметь право на любой вид хранения приложения, приложение должно быть сообщено с дескриптором application_storage_descriptor, который содержит номер версии приложения. Сохраненное приложение однозначно определено тройным номером версии organisation_id, application_id,. Терминалы должны поддерживать несколько хранящихся версий одного и того же приложения. Например, поставщики служб могут использовать одно и то же приложение, но в обновленных версиях в разное время.

Авторы приложения ответственны за обеспечение вещания приложения с корректным номером версии в соответствии с [11] (9.9.2).

9.4.3 Удаление сохраненных приложений

В любом состоянии, кроме Уничтоженный или Незагруженный (NOT_LOADED), терминал не должен удалять приложение из хранения или кэша в соответствии с [11] (9.9.3).

Приложение может быть удалено из хранения или кэша при завершении приложения.

9.9.4 Прерванная загрузка

При прерывании загрузки приложения для хранения или кэширования в соответствии с [11] (9.9.4) прежде, чем все критические файлы будут загружены, терминалам рекомендуется сохранять частично загруженные приложения в специфичном кэше и при повторной попытке загрузки в терминал того же приложения той же самой версии возобновить загрузку в том месте, где она была прервана.

9.9.5 Динамический режим работы

Если приложение, которое ранее не было сохранено, но должно быть сохранено, должны выполняться операции в соответствии с [11] (9.9.5).

9.10 Условия взаимодействия между приложениями MHP и резидентными приложениями

Работающие приложения MHP не должны уничтожаться вследствие представления интерфейсом пользователя (UI) резидентного приложения (например, навигатора MHP). Они могут уничтожаться из-за дефицита ресурсов приемника или как следствие применения конечным пользователем некоторого администратора приложений, предоставляемым одним из резидентных приложений, который должен уничтожить приложение MHP из-за переключения каналов при выборе службы.

Дополнительные условия взаимодействия между приложениями MHP и резидентными приложениями (ресурсами MHP) на интервале жизненного цикла рассмотрены в [14] (9.10).

9.11 Провайдеры

9.11.1 Введение

Провайдеры (термин, использованный в Java SE) применяются для расширения средств платформы, представленных через стандартизированные API. Провайдеры часто представляются адаптерами между реализациями стандартных API и совокупностью конкретных протоколов. Они используют стандартные API, независимы от конкретной реализации и могут быть развернуты операторами непосредственно при отсутствии системы обновления программного обеспечения или подобного процесса. Ниже даны определения провайдеров как платформы. Так, провайдерами, разрешенными этой платформой, могут быть:

- провайдер для протокола PKCS #11 соединения смарт-карт для обеспечения безопасности обратного канала;
- провайдеры для соединения стандартных API информации о службах (SI) MHP с SI и протоколами собственных форматов;
- провайдеры для соединения стандартных API MHP представления медиа с различными протоколами для управления представлением медиа в IPTV.

Провайдеры могут быть представлены в контексте единственным приложением MHP или могут применяться ко всем приложениям MHP в терминале MHP. Первые из них (называются связанные Xlet) доступны нормальным подписанным приложениям MHP. Вторые из них (называются связанная Система) доступны только привилегированным приложениям.

Поддерживаются три модели для распределения класса поставщика. Описание моделей провайдера представлено в [11] (9.11.1).

9.11.2 Параметры жизненного цикла провайдеров Xlet

Параметры жизненного цикла провайдеров Xlet должны быть в соответствии с [11] (9.11.2).

9.11.3 Параметры жизненного цикла системы связанных провайдеров

Параметры жизненного цикла системы связанных провайдеров должны быть в соответствии с [11] (9.11.3).

9.12 Воздействие ограничений графики на модель приложения

9.12.1 Воздействие на универсальные приложения

Дескриптор ограничения графики (см. 10.4.3.3 настоящего стандарта) должен использоваться одним из следующих случаев:

- выбором конфигурации графики «по умолчанию» для приложения при запуске экземпляра приложения или в случае решения не запускать приложение;

- определением, может ли приложение продолжать показывать пользовательский интерфейс (или продолжать работать) в случае нескольких доступных изменений конфигураций графики.

Детализация применения дескриптора ограничений представлена в [11] (9.12.1).

9.12.2 Ограничения графики приложения DVB-J

Конфигурация графики приложения «по умолчанию» должна использоваться для возвращаемой HScene следующими методами:

- org.havi.ui.HSceneFactory.getDefaultHScene();
- org.havi.ui.HSceneFactory.getDefaultHScene(HScreen);
- javax.tv.graphics.TVContainer.getRootContainer();
- javax.microedition.xlet.XletContext.getContainer().

Примечание – в соответствии с [11] (9.12.2) конфигурация графики HGraphicsConfigurations может быть открыта в пакете org.havi.ui и зарезервирована в виде соответствующего ресурса.

Правила обработки приложений DVB-J без видимого интерфейса пользователя должны быть в соответствии с [11] (9.12.2).

9.13 Несвязанные приложения

Подраздел не является обязательным к применению для всех профилей MHP. Спецификации терминала MHP могут включать в себя всю или часть требований этого подраздела или могут не включать их.

9.13.1 Определение несвязанных приложений

9.13.1.1 Версии несвязанных приложений

Несвязанными приложениями являются приложения, чьи жизненные циклы не связаны с представлением конкретного текста, службы или контента, предоставляемого по требованию, в отличие от случая вещания приложений MHP (см. 9.1 настоящего стандарта), которые связаны с сигнализацией одного или нескольких ТВ каналов и только во время представления телеканала. Действие несвязанных приложений могут быть получены в MHP 1,0 с сигнализацией их в составе всех телеканалов в сети, что позволяет им работать независимо от выбора ТВ канала или службы.

Спецификация [13] предусматривает поддержку 3-х версий несвязанных приложений:

- приложения производителя хоста (узла) с установленной реализацией OCAP;
- приложения с сигнализацией через XAIT;
- приложения, зарегистрированного через приложение монитора (Monitor Application).

Параметры второй и третьей версии определяются настоящим стандартом. Параметры первой версии настоящим стандартом не нормируются.

9.13.1.2 Отклонения от решений OCAP

Проект MHP основан на максимальной близости к решениям Спецификации [13]. Однако некоторые изменения были признаны необходимыми для решения различных требований рынка. К ним относятся изменения в соответствии с [11] (9.13.1.2).

9.13.1.3 Обзор контекстов служб, определяющих среду применяемых контекстов настоящего стандарта

Настоящий стандарт определяет единственную среду, которая может включать следующие контексты, представляющие:

- вещательные службы;
- сохраненные службы приложений;
- службы сигнализации по обратному каналу;
- Интернет-службы для клиентов (в терминалах MHP, поддерживающих профиль доступа в Интернет);
- записанные службы (в терминалах MHP, поддерживающих расширение PVR/PDR);
- абстрактные службы (содержащие несвязанные приложения).

Детализация возможностей применения абстрактных служб представлена в [11] (9.13.1.3).

9.13.2 Модель службы OCAP

Требования Спецификации [13] (10.2.2.2) являются опциональными.

9.13.3 Жизненный цикл приложений

Требования Спецификации [13] (10.2.2.3) являются опциональными.

9.13.4 Инициализация MHP

При инициализации MHP должны выполняться следующие действия:

- должен быть загружен и подвергнут синтаксическому анализу процесс кодирования XML для AIT (включая обновленные списки служб и базы данных приложений);
- должно быть идентифицировано самое последнее приложение Монитор (любое сигнализированное в XAIT и существующее в сети или сохраненное в терминале MHP). После выполнения идентификации он должен быть запущен и ожидать сигнализации, посылая пакет org.ocap.OcapSystem. monitorConfiguredSignal, (см. 11.15.5 настоящего стандарта);
- для каждой абстрактной службы из списка служб, которые сигнализируются как «auto_select», должен быть создан контекст служб, в котором должна быть выбрана абстрактная служба. Если версия процесса XAIT сигнализирует сохраненное приложение, совпадающее с определенным в XAIT, то сохраненная версия может быть запущена.

10 Параметры сигнализации приложения MHP

10.1 Введение

В настоящем разделе определены следующие функции сигнализации приложения платформы MHP:

- идентификация приемником платформы MHP приложений, связанных со службой и определение местоположения этих приложений;
- управление жизненными циклами приложений в соответствии с данными сигнализации, передаваемыми в потоках вещания;
- идентификация приемником платформы MHP источников вещания данных, необходимых для приложений службы.

10.1.1 Общие требования к сигнализации

Сигнализация для любых приложений MHP должна обеспечивать:

- формирование таблицы PMT с дескриптором сигнализации приложения (с целью идентификации компонента службы, переносимого в таблице AIT);
- формирование таблицы AIT с информацией в общем цикле дескриптора (common_descriptor_ loop) таблицы AIT: дескриптор транспортного протокола (все дескрипторы приложений должны быть (по крайней мере) в области действия одного дескриптора транспортного протокола);
- формирование таблицы AIT с информацией в цикле дескриптора информации приложения, включающей в себя дескриптор приложения и дескриптор имени приложения.

10.1.2 Общие требования к дополнительной сигнализации для приложений DVB-J

Для приложений DVB-J платформа MHP должна обеспечивать сигнализацию при передаче в таблице AIT в цикле дескриптора информации приложения:

- дескриптора приложения DVB-J;
- дескриптора расположения приложения DVB-J.

10.1.3 Общие требования к дополнительной сигнализации для приложений DVB-HTML

Для приложений DVB-HTML платформа MHP должна обеспечивать сигнализацию в таблице AIT в цикле дескриптора информации приложения:

- дескриптора приложения DVB-HTML;
- дескриптора расположения приложения DVB-HTML.

10.1.4 Общие требования к дополнительной сигнализации для приложений, переносимых через Карусель Объектов

Для приложений, переносимых через Карусель Объектов, платформа MHP должна обеспечивать передачу дескриптора транспортного протокола с селекторными байтами, содержащими конкретную информацию в соответствии с 10.8.1.1 настоящего стандарта в общем (первом) цикле дескриптора или в цикле приложения (внутреннем) дескриптора.

10.1.5 Общие требования к дополнительной сигнализации для приложений переносимых через IP

Для приложений, передаваемых через каналы с протоколом IP, платформа MHP должна обеспечивать дополнительно сигнализацию в таблице AIT:

- дескриптор IP сигнализации в общем (первом) цикле дескриптора;
- дескриптор транспортного протокола в общем (первом) цикле дескриптора или в цикле дескриптора приложений с селекторными байтами, содержащими конкретную информацию IP. Синтаксис селекторных байтов в соответствии с [14] (10.8.1.2, таблица 79).

10.1.6 Расширения сигнализации приложений платформы MHP

Настоящий стандарт предусматривает возможность расширения сигнализации приложений платформы MHP. Расширение может выполняться:

1) Добавлением:
- новых транспортных протоколов с расширением в соответствии с [14] (10.8.1, таблица 77);
- добавлением дескрипторов IP сигнализации, которые будут определены дополнительно;

2) Дополнением конкретных представлений приложений:
- дополнительных конкретных дескрипторов DVB-J в формах, предусмотренных [14] (10.9);
- дополнительных конкретных кодов управления жизненным циклом приложения в границах, предусмотренных [14] (10.6).

Значения констант регистрируются в форме, предусмотренной [14] (10.11, таблица 90).

10.1.7 Информация о службах

MHP должна выполнять обработку таблиц SDT в соответствии с [14] (10.12).

10.2 Параметры сигнализации программно-зависимой информации

Служба, переносящая приложения MHP, должна содержать информацию, достаточную, чтобы определить расположения:

- описания приложения согласно [11] (10.4) для каждого приложения в службе;
- источника кода программы и данных.

Цикл (внутренний) элементарного потока PMT для службы DVB должен поддерживать потоки ссылок не менее одного приложения MHP с целью определения:

- расположения потока, транспортирующего AIT;
- расположения потоков, транспортирующих коды и данные приложений.

10.2.1 Поток сигнализации приложений

Информация элементарного потока для PMT описывает элементарный поток, переносящий AIT сигнализации приложений, и имеет следующие характеристики:

- поле stream_type имеет значение 0x05 в соответствии с [16] (частные секции);
- Дескриптор Сигнализации Приложений.

Количество элементарных потоков сигнализации приложений может быть больше одного.

10.2.2 Потоки вещания данных

Минимально необходимый объем сигнализации, связанной с компонентами вещания данных, обеспечивается значением поля stream_type PMT в соответствии с [18]. Детализированные данные протокола вещания представлены в AIT в соответствии с [14] (10.4) или 10.4 настоящего стандарта.

Опционально PMT включает дескрипторы id вещания данных.

Примечание – Включение идентификатора дескрипторов вещания данных позволяет приемникам начать монтаж файловой системы, которая обеспечивает приложения одновременно с получением AIT, идентифицирующей необходимые приложения. Предоставление возможности выполнения этих параллельных операций может позволить приемникам ускорить активацию интерактивных приложений согласно [14] (приложение B, B.2.8).

Дескриптор идентификации вещания данных идентифицирует «основной» компонент вещательной передачи данных. Детализированная семантика этой дополнительной сигнализации отображает транспортный протокол. Например, в случае Каруселей Объектов DVB идентифицируется компонент, переносящий DownloadServerInitiate (DSI), в соответствии с [14] (приложение B, B.4).

10.3 Система обозначений

10.3.1 Термин «зарезервировано»

Термин «зарезервировано» при использовании для определения кодированного потока битов указывает, что величина может использоваться в будущем для расширений определенных ISO. Если иные определения в конкретном пункте отсутствуют, то все биты, обозначенные «зарезервировано», должны быть установлены в «1».

10.3.2 Термин reserved_future_use

Термин «reserved_future_use» при использовании для определения кодированного битового потока указывает, что значение может быть использовано в будущем для расширений определенных ETSI. Если не указано иное применение, то все биты «reserved_future_use» должны быть установлены в «1».

10.4 Таблица AIT

Таблица AIT предоставляет информацию о передаче данных, необходимом состоянии активации приложений, которые переносят эту информацию. AIT включает набор субтаблиц AIT (см. 10.4.5 настоящего стандарта) выбранной службы, у которой есть application_type, которую может декодировать приемник.

Данные в AIT позволяют вещателю обращаться к приемнику с требованием об изменении состояния активации приложения.

10.4.1 Ошибки данных таблицы AIT

Ошибка в дескрипторе должна привести к его отбрасыванию. После чего в дескрипторном цикле должен обрабатываться следующий дескриптор (если он существует). Область, в которой выполняется обнаружение ошибок дескриптора, должна быть ограничена секцией информации приложения, в котором дескрипторы переносятся.

Ошибка, обнаруженная в цикле приложения за границами дескриптора, должна приводить к отбрасыванию пораженного элемента в цикле приложений. Обработка этого цикла приложений должна продолжаться со следующего элемента записи (если он существует).

Примечание – В соответствии с изложенным выше, следствием появления ошибки в обязательном дескрипторе может быть отсутствие этого дескриптора в цикле приложений. В этом случае цикл приложений должен игнорироваться.

Ошибка в секции информации приложения за пределами цикла приложения должна привести к отбрасыванию секция информации приложения. Обработка AIT должна продолжаться следующей секцией информации приложения (если она существует).

10.4.2 Передача AIT и мониторинг AIT

Терминал MHP выполняет контроль изменений в PMT, присутствующих в элементарных потоках AIT. Изменения в PMT должны обнаруживаться в течение не более 1 с. Терминалы MHP должны контролировать все элементарные потоки AIT в выбранной службе, как описано более подробно ниже.

Частота повторения субтаблиц должна быть не менее 10 с. При условии, что AIT поддерживает не менее 3-х элементарных потоков, интервал времени между обновлением AIT может составлять 30 с.

Примечание – В случае, если количество элементарных потоков в потоке вещания более 3-х, то параметры отклика приемника на изменение интервала времени между обновлением AIT и нормируется.

Терминал MHP должен контролировать секции AIT тех типов приложений, которые он может декодировать. К этим типам приложений относятся те, для которых у терминала есть интероперабельные плагины. Доступные интероперабельные плагины должны включать те приложения MHP, приложения типов которых уже сейчас можно декодировать, которые сигнализированы с плагинами дескрипторов (см. 10.13.4 настоящего стандарта).

Набор типов приложений, перечисленных в базе данных приложения, отражает набор контролируемых секций AIT. Так, это может быть подмножеством типов приложения, передаваемых при вещании, и содержащего расширенный набор возможностей терминала.

Если субтаблица AIT сигнализации приложения полностью исчезнет из сети, то приложение должно продолжать работать. Терминал MHP осуществляет мониторинг появления субтаблицы AIT как определено выше.

10.4.3 Оптимизированная сигнализация AIT

Опциональное поле AIT_version_number, переносящее дескриптор сигнализации приложения, позволяет оптимизировать нагрузку приемника, поскольку это позволяет приемникам получать AIT только после того, как они увидят изменения в версии AIT, объявленные в PMT (См. 10.7.1 настоящего стандарта).

10.4.4 Доступность AIT

Если приложение настраивается в тех случаях, когда данные его сигнализации не доступны и нет возможности выбора новой службы, то оно будет продолжать работать в условиях недоступности AIT.

В терминалах MHP с несколькими сетевыми интерфейсами, если AIT выбранной службы доступна через любой из этих интерфейсов, то сигнализация AIT используется в обычном режиме.

10.4.5 Определение субтаблиц AIT

Все секции, имеющие AIT с одинаковыми table_id и одинаковым значениями application_type, являются элементами одной субтаблицы.

10.4.6 Синтаксис AIT

Секция информации приложений описывает приложения и связанную с ними информацию. Каждая секция информации приложения включает в себя один **Общий** дескрипторный цикл на верхнем уровне для дескрипторов, который совместно используется приложениями этой субтаблицы и циклом приложений. Каждое приложение в цикле приложений содержит **Прикладной** цикл дескрипторов, содержащий дескрипторы, связанные с этим приложением.

Как и в случае таблиц SI DVB, дескрипторный цикл **Общий** содержит субтаблицу. Это означает, что любые дескрипторы, присутствующие в дескрипторном цикле **Общий**, применяются ко всем секциям субтаблицы.

Как и в других таблицах SI DVB, любые строки, содержащиеся в этих таблицах, не должны оканчиваться нулем.

Синтаксис секций информации приложения представлен в таблице 14.

Таблица 14 – Синтаксис секций информации приложения

	Количество битов	Идентификатор
application_information_section() {		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
test_application_flag	1	bslbf
application_type	15	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
common_descriptors_length	12	uimsbf
for(i=0;i<N;i++){	4	bslbf
descriptor()	12	uimsbf
}	8	uimsbf
reserved_future_use	4	bslbf
application_loop_length	12	uimsbf
for(i=0;i<N;i++){	32	rpchof
application_identifier()		
application_control_code		
reserved_future_use		
application_descriptors_loop_length		
for(j=0;j<N;j++){		
descriptor()		
}		
}		
CRC_32		
}		

table_id: поле 8 битов имеет значение 0x74, идентифицирующее эту таблицу.

section_syntax_indicator: индикатор 1 бит с установленным значением «1».

section_length: поле 12 битов, в первых двух битах должно быть установлено «00». Остальные 10 битов указывают количество байтов в секции, начиная сразу после поля **section_length** и включая поле CRC_32. Значение в этом поле не должно превышать 1021 (0x3FD).

test_application_flag: поле 1 бит указывает на приложение, которое было передано в целях тестирования приемника и которое не должно быть запущено или перечислено в любом API или выведено на экран в любом интерфейсе пользователя приемника MHP при нормальных условиях эксплуатации.

application_type: поле 15 битов идентифицирует тип приложений, описанных в субтаблице этой AIT. Типы приложений в соответствии с таблицей 15.

Таблица 15 – Типы приложений

Типы приложений	Описание
0x0000	Зарезервировано для применения в будущем (reserved_future_use)
0x0001	Приложение DVB-J
0x0002	Приложение DVB-HTML
0x0003 до 0x7FFF	подлежат регистрации в DVB

version_number: поле 5 битов идентифицирует номер версии субтаблицы. Значение version_number должно увеличиваться на 1 при изменении информации содержащейся в субтаблице. Когда значение достигает значения 31, то следующим значение станет «0».

current_next_indicator: поле 1 бит, должно иметь значение «1».

section_number: поле 8 битов содержит номер секции. section_number первой секции в субтаблице должен быть «0x00». Значение section_number должно увеличиваться на 1 с каждой дополнительной секцией, имеющей те же самые table_id и application_type.

last_section_number: поле 8 битов указывает номер последней секции (т.е. секции с самым высоким значением section_number, являющейся частью субтаблицы).

common_descriptors_length: поле 12 битов указывает общую длину в байтах дескрипторов следующих далее. Дескрипторы в этом цикле дескрипторов применяются для всех приложений, содержащихся в этой субтаблице AIT.

application_control_code: поле 8 битов управляет состоянием приложения. Семантика этого поля зависит от типа приложения в соответствии с 10.6 настоящего стандарта.

application_loop_length: поле 12 битов содержит общую длину в байтах следующего цикла, содержащего информацию приложения.

application_identifier(): поле 48 битов идентифицирует приложение. Структура этого поля определяется в 10.5 настоящего стандарта.

application_descriptors_loop_length: поле 12 битов дает общую длину в байтах следующих дескрипторов. Дескрипторы в этом цикле применяются для конкретных приложений.

CRC_32: поле 32 бита, содержит значение CRC, которое дает нулевой выход регистра в декодере, определенного в [22] (приложение B) после обработки всей секции.

10.4.7 Применение в AIT частных дескрипторов

Частные дескрипторы допускается включать в AIT при условии их соответствия требованиям DVB-SI [22]. К общим требованиям к дескриптору частных данных относятся следующие:

- если этот дескриптор находится в любом цикле дескрипторов AIT, то любой спецификатор, определенный в этом цикле дескрипторов действует на все дескрипторы и определяемые пользователем значения в цикле дескрипторов частных данных до конца цикла дескриптора, или до появления другого дескриптора спецификатора частных данных;
- необходимость использования дескриптора в **Общем** (первом) цикле дескрипторов не распространяется на значения дескрипторов в **Прикладном** (втором) цикле дескриптора или значения дескрипторов, определяемые пользователем.

10.4.8 Кодирование текста в AIT

Текстовые строки в AIT должны кодироваться при использовании формата UTF-8 в соответствии с [14] (7.1.5, 14.5) и настоящего стандарта (если не указаны иные требования).

10.4.9 Файл AIT

10.4.9.1 Синтаксис

Кодирование файла интерактивного канала должно выполняться при следующих условиях:

- все данные должен содержать единственный файл;
- файл должен содержать сцепленные секции информации приложения (определенные в 10.4.6 настоящего стандарта).
- допускается упорядочивать несколько секций следующим образом:
- в порядке размещения по возрастанию значения application_type;
- в случае единственного application_type устанавливается в порядке возрастания величины section_number;
- во всех секциях индикатор current_next_indicator должен быть установлен в «1».

10.4.9.2 Синтаксические ограничения, семантика, применяемые типы MIME

Синтаксические ограничения, семантика, применяемые типы MIME должны быть в соответствии с [14] (10.4.9.2).

10.5 Параметры идентификации приложений

10.5.1 Параметры кодирования приложения

Каждое приложение связано с идентификатором приложения, который представляет собой поле 6 байтов со структурой в соответствии с таблицей 16.

Таблица 16 – Синтаксис идентификации приложений

	Количество битов	Идентификатор
application_identifier {		
organization_id	32	bslbf
application_id	16	bslbf
}		

organization_id: поле 32 бита несет глобальное уникальное значение идентификации организации, которая несет ответственность за Приложение. Эти значения являются зарегистрированными в [56]. Значения нуля не должны кодироваться.

Это поле приводится в поле organisationName от имени субъекта в «листе» сертификата о подлинности приложений

Примечание – Включение этой области в листе сертификата обеспечивает аутентификацию значения.

application_id: в соответствии с [11] (10.4.3) поле 16 битов определяет функцию приложения, которая определяется организацией, зарегистрированной в organisation_id и которая устанавливает политику распределения внутри организации. Значения нуля не должны кодироваться. Из соображений безопасности значения идентификатора приложения делятся на два диапазона: один для неподписанных приложений и один для подписанных приложений.

В таблице 17 представлены диапазоны значений **application_id** согласно [11] (10.4.3, таблица 13)

Таблица 17 – Диапазоны значений **application_id**

Значения **application_id**	Область применения
0x0000	Не применяется
0x0001 до 0x3fff	Application_id для неподписанных («без знака») приложений
0x0001 до 0x3fff	Application_id для подписанных («со знаком») приложений
0x4000 до 0x7fff	Зарезервировано для применения для DVB в будущем
0x8000 to 0xfffd	Application_id для привелигированных приложений
0xa000 to 0xfffd	Зарезервировано для применения для DVB в будущем
0xfffe	Символ для подписанных приложений от организации
0xffff	Символ для всех приложений от организации

Тот же application_identifier () должен указываться только один раз в составе комплекта субтаблиц AIT с таким же application_type в службе.

10.5.2 Воздействия на жизненный цикл

Основные положения о воздействиях на жизненный цикл представлены в [14] (10.5.2).

10.5.3 Аутентификация идентификации приложения

Аутентификация идентификации приложения должна быть в соответствии с [11] (12.5.6).

10.6 Управление жизненным циклом приложения

Передача сигнализации обеспечивает вещателям механизм управления жизненным циклом приложений стандартных типов (см. 9.1 настоящего стандарта).

10.6.1 Вход и выход из домена (области) приложения

Домен приложения определен как совокупность служб, когда приложения перечислены в AIT. Приложения могут быть перечислены во внутреннем (**Прикладном**) цикле AIT или во внешнем (**Общем**) дескрипторе авторизации приложения. Службы, перечисленные иными способами, находятся за пределами домена приложения.

10.6.2 Динамическое управление жизненным циклом приложения

Динамическое управление жизненным циклом приложения платформы MHP должно обеспечиваться передачей в AIT кода управления application_control_code. Этим кодом вещатель сигнализирует приемнику о необходимых операциях, которые должны выполняться с учетом фазы его жизненного цикла. Если приемник получает код, который он не может опознать, то выполнение приложения должно продолжаться.

В том случае, когда изменение в кодах управления вызывает изменение состояния запущенного (работающего) приложения MHP, должно генерироваться сообщение в AppStateChangeEvent для всех приложений DVB-J, которые зарегистрированы для получения (формирования) события соответствующего приложения.

10.6.2.1 Коды управления жизненным циклом DVB-J

Коды управления жизненным циклом DVB-J должны быть в соответствии с [11] (10.4.3.1) и перечислены в таблице 18.

Таблица 18 – Значения кодов управления жизненным циклом DVB-J

Код	Идентификатор	Семантика
0x00		Резервируется для применения в будущем
0x01	AUTOSTART	Загружаются элементы файловой системы (например, модуль Карусели Объектов), содержащие класс, реализующий интерфейс Xlet, загружается класс, реализующий Xlet в VM и объект Xlet экземпляра и запускается приложение в соответствии с обычными ограничениями
0x02	PRESENT	Указывает, что приложение предствлено в службе, но автоматически не запускается
0x03	DESTROY	При установке кода изменяется с AUTOSTART или PRESENT (НАСТОЯЩЕЕ) на УНИЧТОЖИТЬ, метод Xlet DESTROY вызывается (с безусловным набором параметра ЛОЖЬ) менеджером приложений и приложению разрешается уничтожить себя корректно
0x04	KILL	При установке кода изменяется с AUTOSTART или PRESENT (НАСТОЯЩЕЕ) на KILL, метод Xlet DESTROY вызывается (с безусловным набором параметра ИСТИННО) менеджером приложений
0x05		Резервируется для применения в будущем
0x06	REMOTE	Идентифицирует удаленное приложение, которое запускается только после выбора услуги
0x07 до 0xFF		Резервируется для применения в будущем
0x07	DISABLED	Приложение не включается и попытки запустить его должны блокироваться. Такие приложения с дескрипторами dvb_j_application_descriptor и initial_class_byte не должны сигнализироваться в dvb_j_application_location_descriptor и никогда не используются
0x08 to 0xFF		Резервируется для применения в будущем

Параметры и состояния жизненного цикла приложений DVB-J платформы MHP должны быть в соответствии с 9.2.3 настоящего стандарта и [14] (9.2.3).

10.6.2.2 Коды управления жизненным циклом приложения DVB-HTML

Коды управления жизненным циклом приложения DVB-HTML платформы MHP должны быть в соответствии с представленными в [14] (10.6.2.2, таблица 68) и перечислены в таблице 19.

Таблица 19 – Значения кодов управления жизненным циклом DVB- HTML

Код	Идентификатор	Семантика
0x00		Резервируется для применения в будущем
0x01	AUTOSTART	Загружена Точка входа Приложения приложения DVB-HTML. Загрузка выполнена в агента пользователя, создается DVB-HTML актор (в загружающемся состоянии) и запущено приложение DVB-HTML. При выполнении этих шагов актор DVB-HTML находится в Активном состоянии
0x02	PRESENT	Указывает, что приложение предствлено в службе, но автоматически не запускается
0x03	DESTROY	При установке кода изменяется с AUTOSTART или PRESENT (НАСТОЯЩЕЕ) на УНИЧТОЖИТЬ, актор DVB-HTML переходит к состоянию Уничтожено
0x04	KILL	При установке кода изменяется с AUTOSTART или PRESENT (НАСТОЯЩЕЕ) на KILL, актор DVB-HTML переходит к состоянию Уничтожен
0x05	PREFETCH	Актор на входе находится в активном состоянии и ожидает триггер
0x06	REMOTE	Идентифицирует удаленное приложение, которое запускается только после выбора услуги
0x07 до 0xFF		Резервируется для применения в будущем

Параметры и состояния жизненного цикла приложений DVB-HTML платформы MHP должны быть в соответствии с 9.3.2 настоящего стандарта и [14] (9.3.2).

10.7 Универсальные дескрипторы

10.7.1 Дескриптор сигнализации приложений

Дескриптор сигнализации приложений предназначен для использования в цикле элементарного потока PMT, когда значение поля **stream_type** равно 0x05. Дескриптор идентифицирует элементарный поток, переносящий таблицу AIT.

Опционально дескриптор сигнализации приложений переносит цикл пары **application_type** и **version_number**. Это позволяет приемнику быть информированным о версии AIT в качестве побочного эффекта контроля PMT (контроль в нормальных условиях будет достаточно детализированным). (См. 10.4.3 настоящего стандарта). Синтаксис дескриптора сигнализации приложения приведен в таблице 20.

Таблица 20 – Синтаксис дескриптора сигнализации приложения

	Количество битов	Идентификатор
application_signalling_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
for(i=0; i<N; i++){	1	uimsbf
reserved_future_use	15	bslbf
application_type	3	uimsbf
reserved_future_use	5	
AIT_version_number		
}		
}		

descriptor_tag: поле со значением 0x6F идентифицирует этот дескриптор.

descriptor_length: поле индицирует количество байтов следующего поля descriptor_ length.

application_type: поле индицирует тип субтаблицы таблицы AIT в этом элементарном потоке.

AIT_version_number: поле представляет «текущий» номер версии субтаблицы таблицы AIT, идентифицированной полем типа приложения.

10.7.2 Дескриптор ID вещания данных

Дескриптор идентификатора вещания данных определен для использования в информации элементарного потока PMT. Дескриптор определяет:

- транспортный формат вещания данных «главный компонент», которого переносится в этом же элементарном потоке;
- семантику «главных компонент» конкретного транспортного протокола;
- набор типов приложений для любых приложений автозапуска, поставляемых оператором вещания данных.

Одиночный элементарный поток может иметь несколько дескрипторов идентификатора (id) вещания данных для указания его соответствия нескольким спецификациям вещания данных. Кроме того, дескриптор id нескольких спецификаций вещания данных можно использовать для отображения в списке дополнительных типов приложений в рамках, определенных дескриптором идентификации данных.

Дескриптор id вещания данных указывает, что автоматический запуск приложений осуществляется несколькими механизмами доставки (например, единственная служба может иметь более одной Карусели Объектов доставки приложений автоматического запуска).

10.7.2.1 Универсальный дескриптор

Дескриптор идентификатора вещания данных определен в универсальной форме по [22], которая показана в [14] (таблица 70).

-

10.7.2.2 Дескриптор ID вещания данных MHP

В том случае, когда ID вещания данных равен 0x00F0 или 0x00F1 [14] (таблица 90), то синтаксис дескриптора идентификатора вещания данных должен быть в соответствии с [14] (таблица 71). Это расширяет (опционально) универсальный дескриптор списком типов приложений, для которых приложения автозапуска могут существовать в вещании данных. Этот список обеспечивает подсказку, позволяющую терминалу MHP при предоставлении нескольких служб соединение с источником вещания данных располагать по приоритетам. Синтаксис дескриптора ID вещания данных MHP в соответствии с таблицей 21.

Таблица 21 – Синтаксис дескриптора идентификатора вещания данных

	Количество битов	Идентификатор
data_broadcast_id_descriptor() {		
descriptor_tag	8	uimsbf
descriptor_length	8	uimsbf
data_broadcast_id	16	uimsbf
for (i=0; i<N; i++) {	1	uimsbf
reserved_future_use	15	
application_type		
}		
}		

descriptor_tag: поле 8 битов при значении 0x66 идентифицирует этот дескриптор.

data_broadcast_id: поле 16 битов идентифицирует формат протокола вещания данных. Значения форматов протокола указаны в [56].

application_type: поле 15 битов индицирует тип приложения (механизм или плагин, на которую заявка может быть выполнена). См. [14] (таблица 66).

10.7.3 Дескриптор приложения

В каждом цикле «**Прикладной**» (внутренний) дескриптора AIT должен находиться один дескриптор приложения. Синтаксис и семантика дескриптора приложения и синтаксис дескриптора приложения должны быть в соответствии с [14] (10.7.3).

10.7.4 Дескриптор информации пользователя

Дескриптор информации пользователя дополняет дескриптор приложения и предоставляет пользователю необходимую информацию (когда дескриптор приложения обеспечивает информацию для автоматического использования приемником). Эти дескрипторы определены для использования в цикле **Прикладной**.

10.7.4.1 Дескриптор имени приложения

Дескриптор имени приложения должен позволять отличать одно приложение от другого и должен быть информативным для пользователя. Один экземпляр дескриптора должен быть включен в информацию по применению приложения. Синтаксис и семантика этого дескриптора должны быть в соответствии с [14] (10.7.4.1).

10.7.4.2 Дескриптор иконок приложения

Экземпляр дескриптора иконок приложения может включаться в информацию по применению приложения. Это позволяет иконке быть связанной с приложением. Формат контента для иконок должен быть ограничен PNG [43], как указано в 15.1 настоящего стандарта. Синтаксис и семантика иконок должны быть в соответствии с [11] (10.4.3.2).

10.7.5 Дескрипторы авторизации внешних приложений

Дескрипторы авторизации внешних приложений (external_application_authorisation_descriptors) могут размещаться в цикле **Общий** (первый) дескриптора таблицы AIT. Каждый такой дескриптор содержит информацию о внешних приложениях, которые могут работать с приложениями, перечисленными в субтаблице таблицы AIT. Внешняя авторизация применяется к приложениям с идентифицированным полем application_identifier (), которые имеют поле application_type, идентифицированное субтаблицей AIT, где содержится этот дескриптор.

Синтаксис и семантика дескрипторов авторизации внешних приложений должны быть в соответствии с [14] (10.7.5).

10.7.6 Дескриптор ограничения графики

Дескриптор ограничения графики должен быть в соответствии с [11] (10.4.3.3) со следующими изменениями.

Дескриптор ограничения графики может присутствовать во внутреннем (приложений) цикле AIT (в этом случае он относится только к этому приложению) или в **Общем** (внешнем) цикле AIT (в этом случае он применяется ко всем приложениям, сигнализированным в субтаблице AIT).

10.8 Дескрипторы транспортного протокола

10.8.1 Дескриптор транспортного протокола

Дескриптор транспортного протокола идентифицирует транспортный протокол, связанный с компонентом службы, и может представлять информацию, зависимую от протокола.

Дескриптор транспортного протокола может применяться в **Общем** цикле дескрипторов для транспортных протоколов всех приложений, перечисленных в субтаблице, или в **Прикладном** цикле дескрипторов для дополнительных транспортных протоколов, доступных конкретному приложению. Каждому приложению, описанному в этой секции, должен соответствовать хотя бы один дескриптор транспортного протокола. Синтаксис и семантика дескриптора транспортного протокола представлен в таблице 22.

Таблица 22 – Синтаксис дескриптора транспортного протокола

	Количество битов	Мнемоника	Величина
transport_protocol_descriptor() {			
descriptor_tag	8	uimsbf	N1
descriptor_length	8	uimsbf	
protocol_id	16	uimsbf	
transport_protocol_label	8	uimsbf	
for(i=0; i<N; i++) {	8	uimsbf	
selector_byte			
}			
}			

descriptor_tag: поле 8 битов со значением 0x02 идентифицирует этот дескриптор.

protocol_id: идентификатор протокола, используемого для переноса приложений. Значения protocol_id, представленные в таблице 23, зарегистрированы в [56].

Таблица 23 – Значения protocol_id

Protocol_id	Описание
0x0000	Зарезервировано для применения в будущем.
0x0001	Карусель Объектов MHP в соответствии с [14] (приложение B).
0x0002	IP при многопротокольной инкапсуляции DVB в соответствии с [18], [19].
0x0003	Транспорт через HTTP по интерактивному каналу в соответствии с 10.8.1.3 настоящего стандарта.
0x0004 до 0x00FF	Зарезервировано для DVB.
0x0100 до 0xFFFF	При условии регистрации в [56].

Детализированные синтаксис и семантика дескриптора транспортного потокола должны быть в соответствии с [14] (10.8.1).

10.8.1.1 Транспорт через Карусель Объектов

В том случае, когда источником транспортного протокола является Карусель Объектов, идентификатор транспортного протокола имеет значение 0x0001, селектор байтов должен быть в соответствии с [14] (таблица 78). Семантика полей должна быть в соответствии с [14] (10.8.1.1, таблица 78).

10.8.1.2 Транспорт через IP

В том случае, когда источником транспортного протокола является IP, идентификатор транспортного протокола имеет значение 0x0002, синтаксис селекторных байтов IP транспорта должен быть в соответствии с [14] (таблица 79). Эта структура включает компоненты data_broadcast_descriptor, определенные в соответствии с [18] и включающие всю информацию, необходимую для получения приложения и компонент приложения, доставленных протоколами IP, включая детализированные профили (обязательные или опциональные), перечисленные в [14] (раздел 15).

10.8.1.3 Транспорт через интерактивный канал

Если ID протокола будет равен 0x0003, то синтаксис селекторных байтов дескриптора транспортного протокола должен быть в соответствии с [14] (10.8.1.3, таблица 80). Это позволяет выполнять кодирование нескольких URL. Для повышения эффективности кодирования адреса URL множество адресов разделяются на общую базовую часть и набор расширения URL. Набор URL может идентифицировать ZIP (ZIP [57]) файлы или базовые URL, заканчивающиеся символом «/», которые инкапсулируют части файловой системы. (См. [11] (6.4.1.1) для описания этой системы файлов).

Могут быть предусмотрены дескрипторы нескольких транспортных протоколов со значением идентификатора протокола 0x0003 и той же самой меткой транспортного протокола, чтобы установить более широкий набор URL для описания файловой системы.

Семантика селекторных байтов должна быть в соответствии с [14] (10.8.1.3).

10.8.2 Дескриптор сигнализации IP

Дескриптор сигнализации IP позволяет идентифицировать провайдеров многоадресных потоков IP, используемых всеми приложениями (при передаче дескриптора в **Общем** цикле дескрипторов или используемых одиночным приложением). Этот дескриптор и INT с action_type 0x01 должны использоваться для приложений в случае наличия многоадресных потоков IP в канале вещания.

Идентификация, предусмотренная в дескрипторе сигнализации IP, позволяет восстановить таблицу уведомлений IP (INT) с полем action_type, имеющим значение 0x01, и установить соответствие между IP-адресом, портом и местоположением потока.

Синтаксис и семантика идентификатора дескриптора сигнализации IP должны быть в соответствии с [14] (10.8.2).

10.8.3 Сигнализация упреждающей выборки

10.8.3.1 Введение

Сигнализация упреждающей выборки предусматривается для выборки файлов, которые могут быть использованы на интервале жизни приложений. После запуска приложения эти выборки файлов могут запрашиваться приложением с помощью механизма API.

Для одного приложения дескрипторы предварительной выборки и дескриптор расположения DII, присутствующие в AIT должны указывать на одно и то же транспортное соединение. Если метки транспортного протокола, существующие в этих дескрипторах, будут различны, то дескриптор транспортного протокола, на который сделана ссылка, должен указать на то же самое транспортное соединение (Карусель). Эта сигнализация является опциональной.

10.8.3.2 Дескриптор упреждающей сигнализации

Дескрипторы «0» или «1» упреждающей выборки могут входить в цикл дескрипторов «приложение» таблицы AIT.

Синтаксис дескрипторов упреждающей выборки должен быть в соответствии с [14] (таблица 82).

10.8.3.3 Дескриптор расположения DII

Модули, которые являются частью Карусели Объектов DSM-CC, передаются в сообщении DownloadInfoIndication (DII).

Для выявления всех модулей, которые имеют метку упреждающей выборки в соответствии с [14] (10.8.3.2), платформы MHP должны обеспечивать поиск всех сообщений DII. Поиск сообщений DII выполняется с помощью дескриптора расположения DII, который перечисляет расположения DII. Платформы MHP должны исследовать сообщения DII на наличие меток в том порядке, в котором они перечислены в дескрипторе DII.

Синтаксис дескрипторов расположения меток DII должен быть в соответствии с [14] (таблица 83).

10.9 Дескрипторы DVB-J

В состав дескрипторов DVB-J платформы MHP входят дескриптор приложения DVB-J и дескриптор локации приложения DVB-J.

10.9.1 Дескриптор приложения DVB-J

Дескриптор приложения DVB-J несет информацию о параметрах запуска этого приложения. Один экземпляр дескриптора приложения DVB-J должен содержаться в **Прикладном** цикле дескрипторов AIT для каждого приложения DVB-J. Синтаксис и семантика дескриптора приложения DVB-J должны быть в соответствии с [14] (10.9.1, таблица 84).

10.9.2 Дескриптор расположения приложения DVB-J

Дескриптор расположения приложения DVB-J содержит информацию о маршруте приложения, что обеспечивает поиск приложения и управление приложением DVB-J. Один экземпляр дескриптора должен содержаться в **Прикладном** цикле дескрипторов AIT для каждого приложения DVB-J. Синтаксис и семантика дескриптора расположения приложения DVB-J должны быть в соответствии с [14] (10.9.2, таблица 85).

10.10 Дескрипторы приложения DVB-HTML

В состав дескрипторов приложения DVB-HTML входят:

- дескриптор приложения DVB-HTML;
- дескриптор расположения приложения DVB-HTML;
- дескриптор границ приложения.

10.10.1 Дескриптор приложения DVB-HTML

Дескриптор приложения DVB-HTML содержит значения параметров приложения и сигнализации управления, примененные вещателем приложения.

Один экземпляр дескриптора приложения DVB-HTML и дескриптора расположения DVB-HTML должен содержаться в **Прикладном** цикле дескрипторов AIT для каждого приложения DVB-HTML.

Синтаксис и семантика дескриптора приложения DVB-HTML должны быть в соответствии с [14] (10.10.1, таблица 86).

10.10.2 Дескриптор расположения приложения DVB-HTML

Один экземпляр дескриптора расположения приложения DVB-HTML должен содержаться в прикладном цикле дескрипторов AIT для каждого приложения DVB-HTML. Синтаксис и семантика дескриптора расположения приложения DVB-HTML должны быть в соответствии с [14] (10.10.2, таблица 87).

10.10.2.1 Точка входа приложения

Точка входа приложения определяется путем (маршрутом) «main/index.htm». Этот маршрут сохранен и в строке дескриптора расположения initial_path_bytes.

10.10.3 Дескриптор границы приложения DVB-HTML

Дескриптор границы приложения DVB-HTML является опциональным, он устанавливается в **Прикладном** цикле дескрипторов AIT и позволяет создавать регулярное выражение, описывающее

элементы данных, которые формируют приложение. Синтаксис и семантика дескриптора границы приложения DVB-HTML должны быть в соответствии с [14] (10.10.3, таблица 89).

10.11 Константы дескрипторов

Список значений констант дескрипторов представлен в таблице 24.

Таблица 24 – Список значений констант дескрипторов

Область применения	Тип	Значение	Область определения	Границы применения
Дескриптор спецификатора частных данных	Тэг дескриптора	0x5F	PSI и SI таблицы	SI
Дескриптор id вещания данных		0x66	PMT	
Дескриптор сигнализации приложений		0x6F	PMT	
Дескриптор идентификации службы		0x71	SDT	
Дескриптор метки		0x70	DII информация о модуле, информация о пользователе	SI-DAT
Дескриптор пиоритета кэширования		0x71		
Дескриптор типа кониента		0x72	BIOP информация об объекте	
Зарезервировано MHP для будущих дескрипторов Карусели Объектов		От 0x73 до 0x7F	Карусели объектов	
Зарезервировано MHP для применения в будущем	таблица ID на AIT PID	От 0x00 до 0x73		MHP
Таблица информации приложения		0x74		
Зарезервировано MHP для применения в будущем		От 0x75 до 0x7F		
Зарезервировано для частного применения		От 0x80 до 0xFF		
Дескриптор приложения	Тэг дескриптора	0x00	AIT	MHP
Дескриптор имени приложения		0x01		
Дескриптор транспортного протокола		0x02		
Дескриптор приложения DVB-J		0x03		
Дескриптор локации приложения DVB-J		0x04		
Дескриптор авторизации внешнего приложения		0x05		
Зарезервировано MHP для применения в будущем		0x06 и 0x07		
Дескриптор приложения DVB-HTML		0x08		
Дескриптор локации приложения DVB-HTML		0x09		
Дескриптор границ приложения DVB-HTML		0x0A		
Дескриптор иконок приложения		0x0B		
Дескриптор приложения предварительной выборки		0x0C		
Дескриптор локации DLL		0x0D		
Зарезервировано MHP для применения в будущем		От 0x0E до 0x10		
Дескриптор сигнализации IP		0x11		
Дескриптор экспорта провайдера		0x12		
Дескриптор применения провайдера		0x13		
Дескриптор ограничений графики		0x14		
Зарезервировано для MHP для применения в будущем		От 0x15 до 0x5E		
Зарезервировано MHP для применения в будущем		От 0x12 до 0x5E		
Дескриптор делегированного приложения		0x0E		
Дескриптор плагина		0x0F		
Дескриптор хранения приложения		0x10		
Зарезервировано MHP для применения в будущем		От 0x12 до 0x5E		
Дескриптор конкретных частных данных (Примечание 2)		0x5F		
При условии регистрации в [56]		От 0x60 до 0x7F		
Определяет пользователь (Примечание 3)		От 0x70 до 0xFE		

Окончание таблицы 24

Область применения	Тип	Значение	Область определения	Границы применения
Карусель Объектов MHP	Данные ID вещателя	0x00F0	PMT, AIT	SI
Зарезервировано для многопротокольной инкапсуляции MHP		0x00F1		
Наличие приложений MHP		0x00F2	EIT, SDT	SI
Зарезервировано MHP для применения в будущем		От 0x00F3 до 0x00FE	PMT, AIT	SI
Служба приложения MHP	Тип службы поддержки	0x10	SDT	SI

Примечания:

1 В соответствии с [14] (таблица 90).

2 Спецификатор дескриптора частных данных DVB SI определен для использования в приложении (информационная таблица) для применения частных (конфиденциальных) дескрипторов.

3 Все дескрипторы, определенные пользователем, могут работать в пределе спецификатора дескриптора частных данных.

10.11.1 Служба приложений MHP

В соответствии с [14] (10.11.1) служба приложений MHP должна использоваться для служб, которые предусматривают не менее одного автоматического запуска приложения MHP, если это приложение является основным компонентом службы.

10.12 Информация о службе

10.12.1 Дескриптор идентификатора службы

Дескриптор идентификатора службы может быть включен в таблицу SDT, содержащую описание службы. Каждый такой дескриптор определяет единственный текстовый идентификатор службы. Синтаксис, семантика и детализированные условия применения дескриптора идентификатора службы должны быть в соответствии с [14] (10.12.1, таблица 91).

10.12.2 Дескриптор вещания данных анонсирования приложения MHP

Универсальный дескриптор вещания данных определен в [22]. Синтаксис и семантика селекторных байтов, когда данные содержат идентификатор, 0x00F2 ([14] (таблица 92)).

В этом случае селекторные байты обеспечивают список приложений MHP и информацию о каждом приложении. Эти дескрипторы могут быть перечислены (в списке) в SDT или EIT для идентификации приложений MHP, связанных со службой или событием. Этот дескриптор только указывает на связь между службой или событием и приложениями. Расположение каждого перечисленного приложения определяется через AIT. Этот дескриптор не должен перечислять приложения, если в соответствующей записи в AIT установлен или будет установлен флаг test_application_flag.

10.12.2.1 Семантика дескриптора вещания данных

Семантика для следующих элементов синтаксиса определена в [22]:

- descriptor_tag;
- descriptor_length;
- data_broadcast_id;
- component_tag;
- selector_length;
- ISO_639_language_code;
- text_length;
- text_char.

В [14] (10.12.2.1) определена семантика для следующих элементов:

- **organization_id:** в соответствии с 10.5.1 настоящего стандарта;
- **application_id:** в соответствии с 10.5.1 настоящего стандарта;
- **application_type:** в соответствии с 10.4.6 настоящего стандарта;
- **application_profile_length:** поле 8 битов указывает длину цикла профиля приложения в байтах;

- **application_profile:** в соответствии с [14] (10.7.3);
- **version.major:** в соответствии с [14] (10.7.3);
- **version.minor:** в соответствии с [14] (10.7.3);
- **version.micro:** в соответствии с [14] (10.7.3);
- **application_names_length:** поле 8 битов указывает количество байтов в последующих многоязычных именах приложений;
- **ISO_639_language_code:** в соответствии с [14] (10.7.4);
- **application_name_length:** в соответствии с [14] (10.7.4);
- **application_name_char:** в соответствии с [14] (10.7.4);
- **reserved_length:** поле 8 битов определяет количество зарезервированных байтов;
- **reserved_future_use:** поле 8 битов;
- **private_data_length:** поле 8 битов определяет количество частных данных;
- **private_data_byte:** поле 8 битов.

10.13 Сигнализация плагина

Для делегированных (переданных) приложений и плагинов применяется два сценария сигнализации:
- собственный сценарий сигнализации;
- сценарий сигнализации MHP.

Замена спецификации терминала MHP функционально эквивалентной сигнализацией не допускается.

10.13.1 Собственный сценарий сигнализации

В случае собственного сценария сигнализация выполняется так же как для обычного приложения MHP.

При необходимости сигнализация может быть также передана при использовании дескриптора плагина (см. 10.13.4 настоящего стандарта).

10.13.2 Сценарий сигнализации MHP

В случае сценария MHP собственную сигнализацию плагина частично или полностью может заменить сигнализация делегированного приложения.

Сигнализация MHP позволяет терминалу MHP:
- определить доступность делегированных приложений;
- определить необходимый плагин;
- выполнить запуск плагина;
- выполнить представление делегированных приложений плагину.

Допускается использование плагином собственной сигнализации для определения расположения, загрузки и выполнения делегированного приложения. Технология этого процесса в настоящем стандарте не определяется.

Каждое делегированное приложение связано с дескриптором приложения в соответствии с [14] (10.7.3). Семантика этого дескриптора поддерживается со следующими комментариями:

– профиль версии конкретного формата приложения и информация о версии конкретного формата приложения не зарегистрированы платформой MHP. Тем не менее, правила MHP сравнения значений профиля и версии сохраняются;

– поле метки транспортного протокола может быть не заполненным, если делегированное приложение не транспортируется транспортным протоколом, поддерживаемым MHP.

Для поддержки плагинов предоставляется дополнительная сигнализация MHP:

– дескриптор для делегированного приложения – опционально, в соответствии с 10.13.3 настоящего стандарта;

– обязательный дескриптор плагинов для использования сигнализацией MHP в соответствии с 10.13.4 настоящего стандарта.

10.13.3 Дескриптор делегированного приложения (опционально)

Несколько экземпляров этого дескриптора приложения могут быть размещены в Дескрипторе Приложения делегированного приложения.

Это позволяет переопределить идентификацию плагина по умолчанию, основанную на типе приложения, сообщенном в Дескрипторе Приложения для делегированного приложения. Если этот дескриптор отсутствует, то отсутствуют и идентифицированные конкретные плагины.

Если идентифицировано больше одного приложения, то они перечислены в порядке, начиная от наиболее привилегированного и кончая наименее привилегированным. Если ни одно из привилегированных приложений не будет доступно терминалу МНР, то он должен использовать механизм по умолчанию, чтобы идентифицировать плагин для этого делегированного приложения.

Кроме того, в соответствии с [14] (10.13.3):

- экземпляры этого дескриптора могут быть размещены в **Общем** цикле или в **Прикладном** цикле таблицы AIT делегированного приложения.
- синтаксис и семантика дескриптора плагина должны быть в соответствии с [11] (10.7.3).

10.13.4 Дескриптор плагина

Не менее одного дескриптора приложения плагина должно быть помещено в Описание Приложения каждого плагина.

Этот дескриптор определяет для приложения, реализующего плагин, набор делегированных форматов приложения, которые он поддерживает. Дескриптор идентифицирует тип делегированного поддерживаемого приложения и поддерживаемый набор профилей и версий. Для согласования приложений на совместимой платформе применяются обычные правила сопоставления делегированного приложения с соответствующим плагином.

Приложения DVB-J, сигнализированные с этим дескриптором, должны реализовать API, определенные в 11.7.8 настоящего стандарта.

Плагин нужно считать подходящим для рабочего контента, если application_type соответствует совместимым номерам версии.

Кроме того, в соответствии с [14] (10.13.4) необходимо учитывать следующее:

- дескрипторы приложений плагинов должны размещаться в цикле приложения каждого плагина;
- плагин считается подходящим для рабочего контента, если будет соответствовать application_type и номеру версии, совместимому с алгоритмом, определенному в [14] (10.7.3).

Синтаксис и семантика дескриптора плагина должны быть в соответствии с [11] (10.7.4).

10.14 Сохраненные приложения

10.14.1 Использование сигнализации сохраненных приложений

Терминал МНР, который не обеспечивает хранение приложений или не распознает эти расширения, связанные с хранением приложений, будет воспринимать сохраненные приложения как жизнеспособные, которые потенциально могут быть запущены из соединения «ограниченной службы».

Сигнализация сохраненных приложений должна использоваться в соответствии с [11] (10.8.1).

10.14.2 Дескрипторы сохраненного приложения

Дескриптор сохраненного приложения рекламирует возможность хранения приложения и дает представление о его свойствах. Наличие дескриптора хранения приложения указывает, что обеспечивается Файл Описания Приложения (см. [11] (10.8.3)). Для сохраняемых приложений дескриптор хранения приложения помещается в Описание Приложения.

Этот дескриптор и соответствующие Файлы Описания Приложение поддерживаются приемниками, осуществляющими спекулятивное кэширование.

Синтаксис и семантика дескриптора хранения приложения должны быть в соответствии с [11] (10.8.2, таблица 81).

Если свойство хранения связано с вещанием, то жизненным циклом приложения управляет вещатель. Такие приложения пригодны для кэширования, но не пригодны для обработки в качестве автономного приложения.

Если свойство хранения является «автономным», то приложение может быть успешно запущено пользователем независимо от службы вещания. Приложения с этим свойством могут быть запущены, как связанные с вещанием, если выбранная служба перечислена в Описании Приложения.

Представленные выше характеристики дескрипторов хранения приложения должны интерпретироваться в соответствии с 10.14.3 настоящего стандарта.

Для сохраняемого приложения один дескриптор хранения приложения должен быть установлен в любом **Общем** цикле дескрипторов или **Прикладного** цикла дескриптора AIT. Уточнения семантики полей **storage_property, launchable_completely_from_cache, priority** дескриптора хранения приложения должны быть в соответствии с [14] (10.14.2).

10.14.3 Файл Описания Приложения

10.14.3.1 Описание

Файл Описания Приложения обеспечивает список файлов, которые должны быть установлены так же, как и другая необходимая связанная информация. Для нотации используется синтаксис, основанный на XML.

Для тех приложений, которые могут быть сохранены, файл Файла Описания Приложения должен быть помещен как приложение в ту Карусель, что и приложение.

10.14.3.2 Имя и расположение Файла Описания Приложения

Файл Описания Приложения (ADF) должен быть расположен в основном каталоге приложения DVB-J (как сигнализировано в дескрипторе расположение приложения DVB-J). По соглашению, именем ADF считается:

'dvb.storage.oooooooo.aaaa'

где:

oooooooo - идентификатор организации приложения в виде восьми символьных шестнадцатеричных строк;

aaaa - идентификатор приложения в виде четырех символьных шестнадцатеричных строк.

Идентификатор организации и идентификатор приложения должны быть дополнены начальными нулями к указанной длине.

Строчные шестнадцатеричные цифры используются для кодирования идентификатора организации и идентификатора приложения согласно [11] (10.8.2).

В соответствии с [14] (10.14.3.2) описание файла приложения должно быть расположено в корневом (основном) каталоге приложения DVB-J (как сигнализируется дескриптором расположения приложения DVB-J) или в физическом корневом каталоге приложения DVB-HTML (как сигнализируется дескриптором расположения приложения DVB-HTML).

10.14.3.3 Синтаксис и семантика Файла Описания Приложения

Синтаксис и семантика Файла Описания Приложения должны быть в соответствии с [11] (10.8.3.3, 10.8.3.4) с изменениями семантики в соответствии с [14] (10.14.3.4).

10.15 Сигнализация для провайдеров

Дескриптор, экспортируемый провайдером, используется в **Прикладном** (внутреннем) цикле AIT для приложения, которое экспортирует (содержит) XletBoundProvider. Он декларирует, что классы XletBoundProvider находятся в этом приложении. Этот дескриптор не имеет смысла, если он используется в **Общем** (внешнем) цикле AIT.

Синтаксис и семантика дескриптора экспорта должны быть в соответствии с [14] (10.15).

10.16 Сигнализация для IPTV

10.16.1 Сигнализация приложений связанной службы

Приложения связанных служб должны сигнализироваться включением ApplicationList или ExternalapplicationAuthorisedList в IPService или в SD&S элементах Пакета в соответствии с [1]. Параметры сигнализации связанных приложений определены в [14] (приложение AR).

10.16.2 XML для AIT (XAIT)

Приложения несвязанных служб должны сигнализироваться включением ApplicationList в запись SD&S обнаружения провайдера субсидированных служб. Параметры сигнализации несвязанных приложений определены в [14] (приложение AR). Любые элементы ApplicationList в записях открытия провайдера служб для других провайдеров служб должны игнорироваться.

Мониторинг изменений в XAIT должен выполняться в соответствии с [1] (5.4.3).

Примечание – См. org.ocap.application.AppSignalHander для уведомления об обновлении XAIT приложения.

10.17 XAIT для классических сетей DVB

Метод сигнализации XAIT для классических сетей DVB предназначен для операторов и терминалов, которые для сигнализации таблицы XAIT полагаются только на вещательную среду DVB. Терминал не может полагаться на обратный канал TCP/IP, являющийся доступным и поэтому не может использовать метод сигнализации XML XAIT, определенный для IPTV в [14] (10.16).

В настоящем разделе установлены два варианта сигнализации XAIT. В пунктах от 10.17.1 до 10.17.4 определен рекомендуемый вариант сигнализации XAIT. В пунктах 10.17.5 и 10.17.6 определен совместимый, но не рекомендуемый вариант сигнализации XAIT.

10.17.1 Определение XAIT

Как и в [13] XAIT сообщен на выделенном фиксированном PID. Значением по умолчанию этого PID является 0x1FFC, соответствующее установленному в [13]. Однако это значение PID может быть определено оператором через дескриптор xait_pid_descriptor. Если xait_pid_descriptor сигнализируется в сети оператора, терминал должен использовать значение PID для XAIT, который определен в этом xait_pid_descriptor. Если дескриптор xait_pid_descriptor сигнализации сети отсутствует или если он не будет найден терминалом во время сканирования каналов, терминал должен использовать значение PID по умолчанию 0x1FFC.

Детализированное описание процедур определения XAIT должно быть в соответствии с [14] (10.17.1).

10.17.2 Сигнализация транспорта с помощью Карусели Объектов

Сигнализация расположения несвязанных приложений в XAIT должна использовать дескриптор transport_protocol_descriptor, определенный в [14] (10.8.1), с характеристиками, определенными в соответствии с [14] (10.17.2).

10.17.3 Дескриптор xait_pid_descriptor

Дескриптор xait_pid_descriptor используется для указания значения PID таблицы XAIT для этой сети. Правила кодирования дескриптора xait_pid_descriptor представлены в [14] (таблица 95).

Максимальная величина PID составляет 0x1FFF, для OCAP величина PID XAIT составляет 0x1FFC.

10.17.4 Параметры поддержки «привилегированное приложение» (registerUnboundApp)

Когда терминалом опционально поддерживается «привилегированное приложение», загрузка и хранение несвязанного приложения выполняется через приложение монитора, используя метод AppManagerProxy.registerUnboundApp («XAIT») как определено в [13]. Приложения по [14] (10.2.2.3.2), зарегистрированные как «привилегированные приложения», должны поддерживаться и обрабатываться в соответствии с правилами OCAP.

10.17.5 Определение XAIT (реализация возможна, но не рекомендуется)

В классических сетях DVB XAIT является фактически AIT специальных служб. Если сигнализация приложений XAIT использует дескриптор транспортного протокола protocol_id 1, то это является случаем, который относится к приложениям, переносимым в Каруселях специальных служб.

Требования [13] (11.2.2.3) должно поддерживаться за исключением по [13] (11.2.2.3.9 , 11.2.2.3.12, 11.2.2.3.13 11.2.2.3.17).

10.17.6 Открытие XAIT (реализация возможна, но не рекомендуется)

Дескриптор xait_location_descriptor() применяется для определения расположения службы, переносящей информацию XAIT. Синтаксис и семантика дескриптора расположения XAIT должны быть в соответствии с [14] (10.17.6).

10.18 Приложения, не связанные с резидентными устройствами

Субсидированная служба, может потребовать обеспечения в терминале средствами, которые не рассматриваются в рамках контекста настоящего стандарта. При необходимости реализация представления субсидированной службы выполняется в соответствии с [11] (11.2.3).

11 Платформа DVB-J

11.1 Виртуальная машина DVB-J

Параметры виртуальной машины DVB-J должны быть в соответствии с JSR [58].

11.2 Общие вопросы

11.2.1 Основные соображения

В тех случаях, когда класс, соответствующий настоящему стандарту, является реализацией конкретного интерфейса, и интерфейс требует, чтобы класс обеспечил не рекомендуемый стандартом метод, интерфейс должен переопределять эту метку и применять метод в соответствии с настоящим стандартом.

Каждый экземпляр приложения DVB-J логически работает (выполняется) в его собственном экземпляре виртуальной машины. Процедура завершения объекта, представляющего виртуальную машину, в которой запущено приложение, должна выполняться в соответствии с [11] (11.2.1).

Другие ограничения и уточнения общего характера, возникающие при реализации платформы DVB-J, должны быть в соответствии с [11] (11.2.1).

11.2.2 Зависимость от классов, не включенных в реализуемый профиль

В тех случаях, когда класс, предусмотренный настоящим стандартом, имеет методы, поля или конструкторы с подписью, указывающей на зависимость от классов, не включенных в реализуемый профиль, эти методы, поля или конструкторы не должны присутствовать в реализации. Выбор совместимой реализации может потребовать присутствия этих методов и классов. Настоящий стандарт допускает, но не требует включения полного пакета множества.

11.2.3 Загрузка класса

11.2.3.1 Основные принципы

Среда приложений DVB-J должна быть описана таким образом, чтобы каждое приложение исполнялось своим собственным загрузчиком или иерархическим загрузчиком для всех классов, которые не являются частью платформы. Вследствие этого два приложения никогда не смогут получить доступ к одной и той же копии любого приложения, определенного статической переменной.

В подписанном приложении все классы или файлы, которые будут загружены через путь класса, должны быть подписаны, по крайней мере, набором сертификатов, использующих для подписи начальный класс Xlet приложения. Это относится, например, к классу файлов, содержащих приложения и изображения и другие данные, загруженные с помощью механизма java.lang.Class.getResource (). При аутентификации подписанного приложения терминал может выбрать любой из этих сертификатов и использовать для аутентификации все последующие загруженные классы или файлы. Механизм этого выбора определяется реализацией.

Когда файл JAR включен в путь поиска (например, для DVB ClassLoader) и подписывается в соответствии с моделью безопасности MHP, то все классы или файлы, которые будут загружены с этого JAR файла, считаются так же подписанными сертификатами с подписью файла JAR модели безопасности MHP.

Примечания:

1 MHP не накладывает никаких требований на использование информации, подписанной в файле JAR.

2 Если терминал MHP доверяет нескольким сертификатам, используемым для подписания первоначального Xlet, то он должен попытаться выбрать наиболее надежный из них.

В случае появления неуместных, бессмысленных или незаконных локаторов они должны игнорироваться, поиск должен продолжаться.

11.2.3.2 Загрузка класса и провайдеры

Для спецификаций терминалов MHP, которые включают дополнительные API провайдера (см. 11.7.9 настоящего стандарта), указанное требование для классов, которые подписываются определенным сертификатом, изменяется следующим образом: классы провайдера должны быть подписаны сертификатом, где organisation_id в поле субъекта сертификата соответствует organisation_id данного провайдера.

11.2.4 Разгрузка

Класс разгрузки должен быть в соответствии с [58].

11.2.5 Слушатели события

Слушатели события должны быть зарегистрированы в org.dvb и org.davic.

Примечание – Количество потоков, используемых для отправки событий слушателям событий, определяется реализацией. Приложения не должны блокировать слушателей событий, поскольку это может помешать доставке других событий.

11.2.6 Модель событий в интерфейсах API DAVIC

События, определенные в интерфейсах API DAVIC согласно [7] расширяют java.lang. Объект должен расшириться java.util. EventObject. Слушатели события определены в [7] с расширением java.util. EventListener.

11.2.7 Модель событий в DAVIC и API DVB

Каждый класс в org.dvb и org.davic, наследовавшийся от java.util. EventObject, является только контейнером для этих полей, конструктор которого не выполняет проверку достоверности. Экземпляры этих классов предназначены для реализации платформы, а не приложений. Реализация платформы только создает эти события с соответствующей переданной информацией.

В org.dvb и org.davic, если нет других указаний о добавлении слушателей события, добавляют каждого слушателя только однажды, если метод дополнения с теми же самыми параметрами вызывается несколько раз. Это означает, что событие поставляется каждому слушателю только один раз, даже если оно вызывалось дважды.

11.2.8 Побочный эффект настройки

Это требование применяется только к терминалам MHP, содержащим блок настройки.

Интерфейсы API MHP не должны вызывать процедуры настройки, если в реализации терминала блок настройки не предусмотрен.

11.2.9 Управление ограниченными ресурсами медиа для нескольких приложений

В случае, когда приложение обращается с запросами на ограниченные ресурсы декодирования медиа, возникает конфликт запросов. Это относится к I-кадрам MPEG-2, формату «видео «капли»» MPEG-2 и потоковому видео. Такой же конфликт запросов возникает между потоковой передачей аудио и сохраненным аудио.

В том случае, когда презентация не вещательной передачи медиа (сохраненные аудио или видео или неподвижное изображения) прерывается из-за потери ресурса в том же самом приложении, эта презентация отменяется и не будет восстановлена.

Если прерывается презентация вещательной передачи медиа из-за потери ресурса в том же самом приложении, то презентация вещательной передачи восстанавливается после окончания прерывания презентации.

11.2.10 Приоритет потока взаимосвязанных приложений

Потоки взаимосвязанных приложений ThreadGroup должны иметь приоритет MaxPriority из java. lang.Thread.NORM_PRIORITY.

Примечание – Как следствие приложения не будут в состоянии создавать приоритеты потоков более высокие, чем java.lang.Thread. NORM_PRIORITY, так как они не имеют java.lang.RuntimePermission («modifyThread»). Приложения могут выполнять вычислительные ресурсоемкие задачи в рамках взаимосвязанных приложений, не учитывая возможность возникновения конфликтов.

11.2.11 Кодирование текста

По умолчанию кодирование символов API Java должно быть UTF-8 в соответствии с [11] (7.1.5). Должно поддерживаться кодирование символов «latin1» в соответствии с [59].

В случае присутствия в строке Java разметки кодов, установленных в [11] (таблица 80), разметка должна быть закодирована в символах Java, в которых наиболее значимый байт равен нулю, а величина младшего байта определена в [11] (таблица 80). Кодирование «DVBMarkupUTF8» должно поддерживаться и определяться аналогично случаю «UTF8» с исключениями в соответствии с [11] (11.2.11).

11.2.11.1 Кодирование текста в информации о службе

Для терминалов MHP, спецификации которых включают функциональный эквивалент с именем «SI», кодирование текста в информации о службе выполняется в соответствии с [11] (11.2.11.1).

В тех случаях, когда методы доступа строк интерфейса API SI DVB или интерфейсов API Java TV закодированы в таблицах SI и возвращаются в приложения как объекты строки, должны поддерживаться кодировки символов в соответствии с [22] (приложение А):

- [60] (по умолчанию);
- [59] через [59] (SI первый байт строки кода 0x01...0x05);
- [59] через [59] (SI первые строки байт-код 0x10);
- 16-бит [49] UCS-2 (SI первые строки байт-код 0x11).

11.3 Основные программные интерфейсы DVB-J

Настоящий стандарт не требует применения для локаторов конкретных правил кодирования текста, однако в терминалов требования к кодированию текста устанавливаются. Объекты, для которых требуется такое кодирование текста, определены в [11] (14.8). Там, где необходимо кодирование текста локатора, локатор может быть создан из текстового представления, используя метод производителя, определенный в классе javax.tv.locator. LocatorFactory.

Примечание – Приложения переносимых MHP не должны содержать жестко закодированных представлений текста для локаторов, так как, скорее всего локаторы в разных сетях будут разными. Если приложению необходимо выполнить сигнализацию величин для локаторов, то они могут быть переданы в качестве аргументов Xlet или помещены в небольшой текстовый файл, который считывается из Карусели.

11.3.1 Интерфейсы API платформы Java

Примечание – Следующие пакеты определены в JSR [58]: java.lang, java.void, java.io, javax.microedition. io, java.net. Детализация особенностей поддержки перечисленных пакетов представлена в [11] (11.3.1.1-11.3.1.5).

11.4 Параметры программных интерфейсов приложений представления (воспроизведения)

11.4.1 Графические API пользователя

В состав графических API пользователя должны входить следующие интерфейсы:

- базовый программный интерфейс приложений GUI;
- телевизионный интерфейс пользователя;
- интерфейс расширенной графики;
- интерфейс обработки входных событий;
- интерфейс компоновки шрифта;
- интерфейс PFR0.

11.4.1.1 Базовый API графического интерфейса пользователя (GUI)

Наименования свойств для использования с методом getProperty java.awt.Image и его подклассы, начинающиеся с «DVB», зарезервированы для использования в будущем.

Приложения должны быть в состоянии использовать Toolkit.beep без отрицательных воздействиях на приложение. Вывод конкретного API не должен быть слышимым для нормальных конечных пользователей и не должен конфликтовать с другими API.

Методы getScreenResolution и getScreenSize должны поддерживаться с дополнительной семантикой, описанной в [61].

Параметры кодирования типов контента изображения для использования java.awt.image определены в [11] (7.1.1). Набор поддерживаемых форматов является зависимым профилем.

При использовании класса java.awt.FontMetrics ширина набора символов или строки, возвращаемых методом charsWidth или stringWidth, должна быть корректной с учетом любого кернинга (регулирования межзнакого интервала) применяемого шрифта при визуализации изображения.

Детализация вопросов использования базового программного интерфейса приложений GUI представлена в [11] (11.4.1.1).

11.4.1.2 Телевизионный интерфейс пользователя

Должны поддерживаться пакеты org.havi.ui и org.havi.ui.event, определенные в [61]. Не должны поддерживаться платформой MHP следующие виджеты UI HAVi, классы поддержки и интерфейсы:

- org.havi.ui.HAnimateEffect;
- org.havi.ui.HAnimateLook;
- org.havi.ui.HAnimation;
- org.havi.ui.HDefaultTextLayoutManager;

- org.havi.ui.HEventMulticaster;
- org.havi.ui.HFlatEffectMatte;
- org.havi.ui.HFlatMatte;
- org.havi.ui.HGraphicButton;
- org.havi.ui.HGraphicLook;
- org.havi.ui.HIcon;
- org.havi.ui.HImageEffectMatte;
- org.havi.ui.HImageMatte;
- org.havi.ui.HListElement;
- org.havi.ui.HListGroup;
- org.havi.ui.HListGroupLook;
- org.havi.ui.HMultilineEntry;
- org.havi.ui.HMultilineEntryLook;
- org.havi.ui.HRange;
- org.havi.ui.HRangeLook;
- org.havi.ui.HRangeValue;
- org.havi.ui.HScreenDimension;
- org.havi.ui.HSinglelineEntry;
- org.havi.ui.HSinglelineEntryLook;
- org.havi.ui.HStaticAnimation;
- org.havi.ui.HStaticIcon;
- org.havi.ui.HStaticRange;
- org.havi.ui.HStaticText;
- org.havi.ui.HSwitchable;
- org.havi.ui.HText;
- org.havi.ui.HTextButton;
- org.havi.ui.HTextLook;
- org.havi.ui.HToggleButton;
- org.havi.ui.HtoggleGroup.

Если поддержка I-кадров MPEG-2 (опционально) не предусмотрена, то все попытки запроса экземпляра org.havi.ui.HStillImageBackgroundConfiguration будут игнорированы. Для выполнения запроса указанного экземпляра необходимо присутствие самого класса.

Другие отклонения от требований [61] должны быть в соответствии с [11] (11.4.1.2).

11.4.1.3 Интерфейс расширенной графики

Параметры интерфейса расширенной графики представлены в [11] (приложение U). Класс org.dvb.ui.DVBTextLayoutManager в соответствии с настоящим стандартом не применяется.

11.4.1.4 Режим просмотра телевидения

MHP поддерживает два способа приема приложением входных событий: использованием метода AWT в java.awt.Component и пакета org.dvb.event в соответствии с 11.3.2.2 настоящего стандарта. При использовании метода AWT приложение обычно получает все входные события, когда компонент находится в фокусе. Минимальный набор входных поддерживаемых событий определен в [11] (приложение G, G.5).

При использовании метода AWT детализированные правила приема входных событий должны быть в соответствии с [11] (11.4.1.4).

11.4.1.5 Компоновка шрифта

11.4.1.5.1 Шрифт формата PFR0

Для шрифтов, имеющих формат PFR0, параметры привязки программных интерфейсов приложений Java, касающиеся метрик шрифта и формата самого шрифта, должны быть в соответствии с [11] (11.4.1.5.1).

11.4.1.5.2 Шрифт формата OpenType

Для шрифтов в формате OpenType, описанных в [11] (7.4.2), взаимосвязь между Java API, относящаяся к метрике шрифта, и формату самого шрифта, а также возвращаемые значения методов должны быть в соответствии с [11] (11.4.1.5.2).

11.4.2 Интерфейс API потоковых медиа

11.4.2.1 Граничные условия решения интерфейса API потоковых медиа при работе в составе MHP

Пакеты javax. media и javax. media.protocol из [62] должны быть реализованы с разъяснениями, расширениями и ограничениями, установленными в соответствующих пунктах ниже.

11.4.2.2 Пояснения к условиям воспроизведения контента

Условия воспроизведения контента проигрывателем JMF, создаваемого локаторами и URL, должны быть в соответствии с [11] (11.4.2.2).

11.4.2.3 Поведение проигрывателя медиа по умолчанию

Правила представления по умолчанию службы проигрывателем медиа JMF, когда в службе будет присутствовать несколько компонентов аудио, видео или титров должны выполняться в следующем порядке:

- для аудио и субтитров на различных языках для выбора потока используется предпочтение пользователя;

- в соответствии с сетевой информацией сигнализации (таблица состава программ, PMT), содержащейся в транспортных потоках, и в соответствии со спецификациями терминалов MHP могут быть определены дополнительные механизмы, которые переопределяют значения по умолчанию. Если какой-либо из компонентов медиа, включающих службу, изменится, то реализация в максимально возможной степени заменит измененные компоненты с подходящими заменами в предпочтении пользователя и способом, зависящем от реализации терминала.

11.4.2.4 Необходимые средства управления для формата «видео «капли»»

Терминал MHP поддерживает следующие средства управления в формате «видео «капли»» в соответствии с [11] (7.1.3):

- javax.tv. media.AWTVideoSizeControl,
- org.dvb.media.BackgroundVideoPresentationControl

11.4.2.5 Расширения платформы

11.4.2.5.1 Расширения специфицированные DVB

Расширения классов и интерфейсов, специфицированных DVB, представлены в [11] (приложение N), исключая в соответствии с [14] (11.4.2),следующие:

- org.dvb.media.SubtitlingEventControl;
- org.dvb.media.SubtitleAvailableEvent;
- org.dvb.media.SubtitleListener;
- org.dvb.media.SubtitleNotAvailableEvent;
- org.dvb.media.SubtitleNotSelectedEvent;
- org.dvb.media.SubtitleSelectedEvent;
- org.dvb.media.CAStopEvent;
- org.dvb.media.CAException.

Если плеер, связанный с DripFeedDataSource, получает ResourceWithdrawnEvent и вслед за ним ResourceReturnedEvent, то видеовыход видеодекодера, с которым связан проигрыватель, зависит от реализации. Приложение MHP, использующее плеер, после получения ResourceReturnedEvent должно немедленно вызвать метод DripFeedDataSource.feed с I-кадром.

Примечание – Разработчики приложений должны принимать во внимание, что при создании последовательности «видео «капли»» избыточное количество P-кадров между I-кадрами может привести к существенной задержке обновления видимого изображения из-за ограничений на частоту, с которой новые данные могут подаваться на декодер.

11.4.2.5.2 Расширения в org.davic

Следующие классы и интерфейсы включаются в пакет org.davic.media в соответствии с [7] (приложение L):

- MediaPresentedEvent;
- MediaLocator;
- MediaTimePositionControl;
- ResourceWithdrawnEvent;
- FreezeControl;
- ResourceReturnedEvent;
- MediaTimePositionChangedEven;

- LanguageNotAvailableException;
- NotAuthorizedException;
- MediaFreezeException;
- LanguageControl;
- AudioLanguageControl.

Для пакетизированных целевых медиа следующие классы не обязательны:

- org.davic. media.FreezeControl
- org.davic. media.MediaFreezeException

Для спецификаций терминалов, включающих API условного доступа, следует включать класс org. davic.media.NotAuthorizedMediaException.

Классы из пакета org.davic. медиа, определенные в [7] (приложение L), включаются в спецификацию терминала MHP с изменениями семантики в соответствии с [11] (11.4.2.5.2).

11.4.2.5.3 Расширения в javax.tv

Состав и характеристики расширений javax.tv должны быть в соответствии с [11] (11.4.2.5.3).

11.4.2.5.3.1 Типы, установленные ТВ Java

ТВ Java устанавливает следующие типы, перечисленные ниже:

- javax.tv.service.navigation.DeliverySystemType;
- javax.tv.service.guide.ProgramScheduleChangeType;
- javax.tv.service.ServiceInformationType;
- javax.tv.service.ServiceType;
- javax.tv.service.SIChangeType;
- javax.tv.service.SIRequestFailureType;
- javax.tv.service.navigation.StreamType.

Рекомендации, позволяющие определить дополнительные значения этих типов, представлены в [11] (11.4.2.5.3.1).

11.4.2.5.4 Необходимые средства управления для профилей вещания и профилей пакетизированных медиа

Следующие средства управления поддерживаются для потокового вещания форматов, определенных в 7.2 настоящего стандарта:

- org.davic.media.LanguageControl;
- org.davic.media.AudioLanguageControl;
- org.davic.media.SubtitlingLanguageControl;
- org.davic.media.FreezeControl;
- javax.tv.media.MediaSelectControl;
- javax.tv.media.AWTVideoSizeControl;
- org.dvb.media.VideoPresentationControl;
- org.dvb.media.BackgroundVideoPresentationControl;
- org.dvb.media.SubtitlingEventControl;
- org.dvb.media.VideoFormatControl;
- org.dvb.media.DVBMediaSelectControl.

Для медиа декодеров (7.1.4 настоящего стандарта) поддерживается элементы управления из org. davic.media.MediaTimePositionControl.

Применение некоторых классов, указанных в 11.4.2.5.4 и 11.4.2.5.5 настоящего стандарта, не обязательно, как указано в других разделах настоящего стандарта.

11.4.2.5.5 Разъяснения к применению средств управления

Разъяснения к применению средств управления вещанием профилей и пакетизированных профилей должны быть в соответствии с [11] (11.4.2.5.4.5).

11.4.2.5.6 Основные компоненты плейеров JMF

Основные компонент плейеров JMF должны быть в соответствии с [11] (11.4.2.5.6).

11.4.2.6 Ограничения структуры классов вещания

Элементы управления, поддерживаемые MHP, могут не иметь соответствующих компонентов графического интерфейса пользователя (GUI). Любые вызовы Control.getControlComponent () могут возвращаться в форме нулевых значений.

Разработчики приложений MHP не должны рассчитывать на наличие следующих классов или интерфейсов:

- javax.media.CachingControl (если не возвращается вызовом метода JMF);
- javax.media.CachingControlEvent;
- javax.media.GainControl (если не возвращается вызовом метода JMF).

Детализация ограничений платформы вещания должна быть в соответствии с [11] (11.4.2.6).

11.4.2.7 Пересечение методов MediaSelectControl и SubtitlingLanguageControl/AudioLanguageControl

Метод org.davic.media.LanguageControl.listAvailableLanguages () должен перечислить все доступные языки в затронутой службе.

Правила работы пересекающихся методов MediaSelectControl и SubtitlingLanguageControl/AudioLanguageControl должны быть в соответствии с [11] (11.4.2.7).

11.4.2.8 Правила взаимодействия интерфейсов API потоковых медиа и интерфейсами API пользователя ТВ

Основные принципы взаимодействия интерфейсов API потоковых медиа и интерфейсов API пользователя ТВ, управления режимами работы плейера, управления поведением приложения ТВ, правила динамического режим работы плеера JMF должны быть в соответствии с [11] (11.4.2.8.1 – 11.4.2.8.4).

11.4.2.9 Детализация управления ресурсами

Детализация параметров управления ресурсами плеера JMF описана в [11] (11.4.2.8.5).

11.4.2.10 Интеграция с поставщиками

В соответствии с [14] (11.4.2), [11] (11.4.2.9) в MHP используется API провайдера DVBTextLayoutManager, специфицированный в [11] (приложение U).

11.4.2.11 Дополнения и изменения семантики для IPTV

Для служб и элементов контента, доступных через протокол RTSP, вызовы метода Player.setRate должны привести к запросу изменения масштаба в соответствии с [4] (12.34). Возвращаемое значение метода должно соответствовать реальному значению масштаба, выбранному сервером.

11.5 API доступа к данным

11.5.1 API доступа к вещательному транспортному протоколу

Доступ к протоколу вещания определен в [11] (приложение P).

Относительные имена файлов, используемые для доступа к объектам в файловой системе приложений, должны приниматься по базовому каталогу, указанному в 10.5, настоящего стандарта.

Примечание – Абсолютные и относительные пути могут быть использованы в соответствии с API, связанных, например, с java.net.URL (при использовании с « file:/ /» строки префикса) и с java.io.File.

Правила кэширования, указанные в [11] (приложение B, B.5), должны оцениваться в момент загрузки информации с помощью DSMCCObject.synchronousLoad () или DSMCCObject.asyncronousLoad () или неявной загрузки в ответ на другие действия, как это определено в 11.5.1.1 настоящего стандарта. После того, как объект DVB-J был загружен, любые возможные изменения в Карусели Объектов в содержании объектов представляются невидимыми.

11.5.1.1 Ограничения на java.io. File методы файла для Каруселей Вещания

Приложение должно быть в состоянии использовать стандартный класс java.io.File для доступа к Карусели Вещания. В этом случае должны применяться определения в соответствии с [11] (11.5.1.1) и с учетом изменений [14] (11.5.1.1).

11.6 Программные интерфейсы приложений информации о службе и выбора службы

11.6.1 Программные интерфейсы приложений информации о службе DVB

Конкретные API информации о службе DVB определены в [14] (приложение M).

11.6.2 API выбора службы

API выбора службы определены пакетом javax.tv.service.selection от JSR [62] и пакетом org.dvb. service.selection, определенным в [11] (приложение AK).

Все экземпляры ServiceContext, возвращенные терминалом MHP, должны быть экземплярами DvbServiceContext.

Детализация условий применения API для обоих случаев представлена в [11] (11.6.2).

11.6.3 API настройки

Спецификации терминалов MHP для пакетизированных медиа необязаны поддерживать настройку. Определения и требования настоящего пункта применяются только к спецификациям терминалов, поддерживающим настройку.

11.6.3.1 Общее описание

Параметры программного интерфейса приложений настройки определены в [7] (приложение H.4) за исключением [7] (приложение H, H.4). Уточнения условий применения программного интерфейса приложений настройки представлены в [11] (11.6.3.1).

11.6.3.2 Настройка на IPTV

В режиме IPTV функция настройки API выполняется управлением входом демультиплексора MPEG-2 присоединением транспортного потока ко входу демультиплексора MPEG-2.

11.6.4 API условного доступа

11.6.4.1 Общие замечания

Интерфейс API CA обеспечивает приложения системы условного доступа независимым интерфейсом для доступа к функциям, связанным с CA, например, к электронным путеводителям по программам (Electronic Program Guide; EPG). Операции низкого уровня функциональности (такие, как обработка сообщений EMM и ECM) выполняются системой CA автономно. Они остаются невидимыми для приложений MHP и в настоящем стандарте не рассматриваются.

11.6.4.2 Общие параметры

Интерфейс API CA должен поддерживать реализацию приложений пользовательских диалогов высокого уровня MMI при условии обеспечения необходимой безопасности. Решения интерфейса API CA должны быть совместимыми с требованиями настоящего стандарта и требованиями к интерфейсу условного доступа технической спецификации DAVIC [7] (приложение I).

11.6.4.3 Общие требования к системе CA и к интерфейсу API CA

Интерфейс API CA должен поддерживать модули CA, которые могут быть модулями типа DAVIC CA0 в соответствии с технической спецификацией DAVIC [7] (приложение I, 2.2) или совместимыми модулями типа DAVIC CA1. Должны поддерживаться конфигурации модулей смешанных типов (CA0 и CA1 одновременно).

Примечание – В настоящем стандарте устройство CA упоминается как модуль, даже если в случае CA1 он представляет собой смарт-карту.

Интерфейс API CA должен обеспечивать поддержку собственной системы CA с расширениями DAVIC, относящимися к системе CA.

Допускается включение в интерфейс API CA систем CA с независимыми функциями, которые могут не поддерживаться всеми интерфейсами CA, совместимыми с DAVIC.

Интерфейс API CA должен обеспечивать передачу приложениям уведомлений об удалении или установке модулей CA во время выполнения приложений.

Интерфейс API CA должен обеспечивать приложению получение списка идентификаторов систем условного доступа, поддерживаемых STB.

Интерфейс API CA должен обеспечивать приложению возможность определения физического расположения модулей безопасности, если в MHP предусматриваются несколько слотов для таких модулей.

В случае установки в MHP нескольких систем CA интерфейс API CA должен обеспечивать приложению возможность выбора системы CA для дескремблирования службы.

Интерфейс API CA должен обеспечивать для взаимодействующих приложений возможность передачи сообщения модулю безопасности и позволять модулю безопасности передавать сообщения к взаимодействующим приложениям. Интерфейс API CA должен обрабатывать и учитывать ответы на исходные сообщения.

11.6.4.4 Требования масштабируемости API CA

Интерфейс API CA должен поддерживать MHP с несколькими модулями сетевого интерфейса и несколькими входящими транспортными потоками.

Интерфейс API CA должен поддерживать не менее одного модуля CA.

Интерфейс API CA должен обеспечивать возможность нескольким приложениям инициировать дескремблирование нескольких служб одновременно, если они поддерживаются модулем.

11.6.4.5 Требования к дескремблированию

Интерфейс API CA должен поддерживать запуск и остановку процесса дескремблирования служб или компонентов служб, которые получают доступ к низкоуровневым API, таким как API фильтра секции в соответствии с требованиями технической спецификации DAVIC [7].

Интерфейс API CA должен обеспечивать формирование отчета о возможности (или невозможности) дескремблирования транспортного потока в случаях отсутствия доступа модуля NIU к модулю безопасности или в случае несовпадения скремблирующих алгоритмов.

Интерфейс API CA должен поддерживать процедуры проверки возможности дескремблирования определенной службы или события в случаях, когда модуль имеет для этой цели необходимые ресурсы. Служба или событие могут находиться или могут не находиться в процессе обработки. Интерфейс API CA должен позволить системе CA накладывать ограничения на доступ к этим данным.

Интерфейс API CA должен обеспечивать поиск списка всех прав пользователя. Права могут быть перечислены отдельно для каждого сочетания системы CA и модуля безопасности. API может применять ограничения доступа, налагаемые системой CA, на эти данные.

11.6.4.6 Модель объекта API CA и состояния API CA

11.6.4.6.1 Модель объекта

Модель объекта включает следующие основные части:

- класс CAModule, который является абстракцией физического модуля CA. Он обеспечивает функции запроса о правах и получения соответствующей информации;
- класс DescramblerProxy, реализующий интерфейс ResourceProxy для управления ресурсами;
- объект DescramblerProxy, обеспечивающий функцию дескремблирования одной службы;
- классы TransportStream, Service (служба) и ElementaryStream от API компонентов MPEG, использующиеся в качестве параметров в методах классов CAModule и DescramblerProxy;
- объект CAModuleManager, управляющий модулями CA и отслеживающий доступные модули в STU. При реализации объекта CAModuleManager автоматически создает экземпляр CAModule и регистрирует его, когда модуль вставляется в STU. Класс CAModule не имеет общедоступного конструктора. CAModule автоматически удаляется из объекта CAModuleManager при удалении модуля из STU.

Параметры модели объекта API CA должны быть в соответствии с [7] (приложение I, 3.1).

11.6.4.6.2 Состояния интерфейса API CA

На рисунке 8 показаны три состояния, в которых может находиться система CA:

- «дескремблирование не введено» (no_descrambling);
- «готовность интерфейса MMI» (MMI_done);
- «дескремблирование введено» (descrambling).

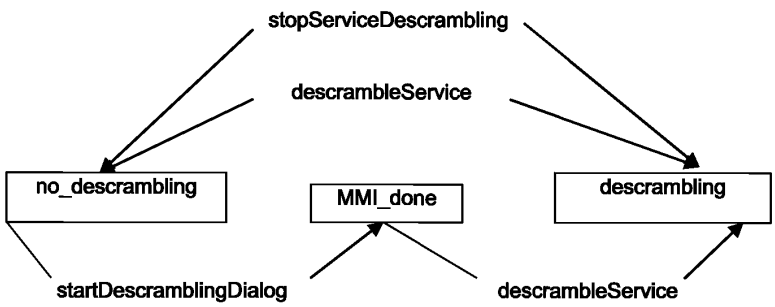


Рисунок 8 – Изменения состояния системы

По умолчанию система (класс CAModule) устанавливается в состояние «дескремблирование не введено». В этом состоянии дескремблирование не выполняется. Переход в состояние «дескремблирование введено» может быть выполнен двумя способами. Первый способ реализуется прямым вызовом метода startDescrambling. После вызова этой функции начинается процесс дескремблирования. В этом случае представление информации может включать диалоги пользователя по запросам системы CA. Если дескремблирование при использовании первого способа невозможно, то применяется второй способ, который заключается в вызове метода startDescramblingDialog. Это заставляет систему CA в случае необходимости начинать диалог с пользователем и ввести состояние «готовность интерфейса MMI» MMI_done. После этого состояние «дескремблирование введено» может быть достигнуто вызовом метода startDescrambling.

11.6.4.7 Определения интерфейса API CA, параметры интерфейса API CA

В API CA входят следующие составные части:

- Исключения (Exceptions);
- Модель слушателя событий (Event-Listener model);
- Модель слушателя событий MMI;
- Модель слушателя событий, пересылка сообщений (Event-Listener model, Message passing);
- CAModule-Manager;
- CAModule;
- DescramblerProxy;
- CA0Module;
- CA1Module;
- MMI;
- Передачи сообщений (Message Passing).

Перечень составных частей API CA и их определений должны быть в соответствии с [7] (приложение I, раздел 4).

11.6.5 API служебной информации, независимой от протокола

11.6.5.1 Обобщенное описание

API служебной информации, независимой от протокола, определены следующими пакетами [62]:

- javax.tv.service;
- javax.tv.service.guide;
- javax.tv.service.navigation;
- javax.tv.service.transport.

Отображение этих пакетов на протокол DVB-SI определено [11] (11.6.5.1 и приложение O).

11.6.6 Открытие службы и выбор службы для IPTV

Открытие службы и выбор службы для IPTV выполняются применением пакета org.dvb.service.sds. Класс org.dvb.service.DvbSIManager должен реализовать org.dvb.service.sds. SDSRecordAccess. Пакет org.dvb.service.sds определен в [11] (приложение AW).

11.6.7 Сопряжение API SI, независимой от протокола, и TV-Anytime

Сопряжение API SI, независимой от протокола, и TV-Anytime выполняет пакет org.dvb.tvanytime.javatv. Класс org.dvb.service должен реализовать интерфейс DvbSIManager org.dvb.tvanytime.javatv. CRIDAccess.

Экземпляры javax.tv.service.guide.ProgramEvent, данные которые были получены через протоколы TV-Anytime, используемые в соответствии с [2], должны реализовать org.dvb.tvanytime.javatv. TVAProgramEvent.

Пакет org.dvb.tvanytime.javatv определен в [11] (приложение AY).

11.7 Общая инфраструктура интерфейсов API

11.7.1 API, поддерживающие приложения жизненного цикла DVB-J

Этот API сформирован из классов Java и интерфейсов, входящих в состав пакета javax.tv.xlet, определенного в JSR [62]. Кроме того, подобные классы содержит пакет javax.microedition.xlet. В любом случае XletContext, который передают к Xlet, должен реализовать javax.microedition.xlet.XletContext и javax.tv.xlet.XletContext.

Терминалы MHP должны определять, какой интерфейс Xlet реализован исходным классом приложения и использовать соответствующий API управления приложениями. В случае, когда начальный класс приложения MHP реализует javax.tv.xlet. Xlet и javax.microedition.xlet.Xlet, терминал MHP должен использовать API управления приложениями javax.tv.xlet.

11.7.1.1 Свойства Xlet

Терминалы MHP должны поддерживать следующие свойства Xlet:

- dvb.org.id;
- dvb.app.id;
- dvb.caller.parameters.

Они могут быть получены от Xlet XletContext, вызывая getXletProperty с именем строки, определенным выше.

Все ключи для XletContext.getXletProperty, начинающиеся dvb, зарезервированы для использования в спецификациях проекта DVB. Подробности применения Xlet в соответствии с [11] (11.7.1.1) с уточнениями по [14] (11.7.1.1).

11.7.2 API открытия приложения и запуска приложения

Этот API сформирован из пакета org.dvb.application, определенного в [14] (приложение S).

Свойства API для использования с методом AppAttributes.getProperty определены [14] (таблица 98).

В [14] (таблица 99) определены источники информации, которая должна использоваться для метода возвращения информации из записи в базе данных приложений для приложения, сигнализированного в AIT. Эти данные должны применяться к обоим AIT, поставленным через вещание MPEG-2 и поставленным как файл AIT ([14] (10.4.9)).

11.7.3 API взаимодействия приложений

Этот API сформирован из пакетов java.rmi и javax.microedition.xlet.ixc как определено в JSR [58], а также пакета org.dvb.io.ixc. Механизм формирования API взаимодействия приложений представлен в [11] (11.7.3).

11.7.4 Основные концепции MPEG

Примечание – Интерфейс API от DAVIC, определенный в этом пункте, является опциональным для задач потокового медиа в разделе 15, потому что он имеет отношение к транспортным потокам.

Этот API сформирован из классов Java, определенных в [7] (приложение G):

- ApplicationOrigin;
- ElementaryStream;
- Service;
- TransportStream.

Методы, возвращающие экземпляры службы элементарного потока или транспортного потока, должны возвратить экземпляры определенного подкласса DVB (DvbElementaryStream и т.д.), если спецификация терминала MHP включает эти классы.

В соответствии с [14] (11.7.4) перечень классов API должен быть дополнен классами:

- DvbElementaryStream;
- DvbService;
- DvbTransportStream.

11.7.5 Уведомления о ресурсах

Уведомления о ресурсах определены в [7] (приложение F). В соответствии с [11] они обеспечиваются применением:

- Метода notifyRelease ();
- Параметра requestData метода requestRelease, реализующего интерфейс java.rmi.Remote;
- Метода resourceProxy.getClient ().

11.7.6 API ссылок на контент

Этот API состоит из классов, находящихся в [7] (приложение H, H.4), и классов Locator и DvbLocator. Он также включает пакет javax.tv.locator. Он также включает org.dvb.locator.package, определенный в [14] (приложение AL). Сигнатура org.davic.net.Locator класса будет расширена: «реализации javax. tv.locator.Locator».

Детализация применения методов и локаторов представлена в [14] (11.7.6).

11.7.7 API отчетов об ошибках

Этот API состоит из интерфейса и исключений, определенных из [7] (приложение G):

- NotAuthorizedInterface;
- NotAuthorizedException;
- ObjectUnavailableException;
- ResourceException;

- TuningException.

NotAuthorizedInterface и NotAuthorizedException должны присутствовать в терминала MHP, которые включают в себя API фильтра секции. Если терминал MHP не включает API условного доступа, то NotAuthorizedException не должен генерироваться платформой.

11.7.8 API плагинов

API плагинов (пакет org.dvb.application.plugins) определены в [11] (приложения AF, AE).

Особенности применения интероперабельных плагинов семантики метода org.dvb.application. AppAttributes.getProperty() в соответствии с [11] (11.7.8).

11.7.9 API провайдера

API провайдера (пакет org.dvb.spi.) определены в [11].

Объектная структура API определяется пакетами org.dvb.spi и org.dvb.spi.util. В [11] (11.7.10) представлены соответствия между API провайдера и соответствующими API, вызванными приложениями MHP для пакетов:

- выбора провайдера - org.dvb.spi.selection;
- информации служб (SI) провайдера - org.dvb.spi.si.simple и org.dvb.spi.si.full;
- взаимодействия объекта DSM-CC с реализацией провайдера служб транспорта интерактивных каналов (ICT).

11.7.10 API ссылок на контент для IPTV

API ссылок на контент для IPTV (для пакетов org.dvb.locator.ip) определены в [11] (приложение AU).

11.7.11 API ссылок на контент и метаданные для TV-Anytime

API ссылок на контент и метаданные для TV-Anytime представлены пакетами org.dvb.tvanytime. metadata и org.dvb.tvanytime.resolution согласно [12] вместе с пакетом org.dvb.tvanytime.metadata.ip.

К экземплярам javax.tv.locator.Locator (ссылка ProgramEvents), чьи метаданные доступны только через протоколы TV-Anytime, в соответствии с [2] применяются действия в соответствии с [11] (11.7.11):

- если экземпляр локатора был получен явно или неявно по разрешению CRID, то реализация должна помнить соответствующий CRID. Попытки получить экземпляр ProgramEvent, используя такой локатор, должны привести к попытке получить метаданные, связанные с запомненным CRID;
- если внешней форме локатора соответствует опциональный элемент ProgramURL в TV-Anytime <ScheduleEvent> или фрагмент <BroadcastEvent>, доступный для терминала MHP, терминал MHP должен получить CRID из этого фрагмента, а затем пытаться получить метаданные для этого CRID;
- если внешняя форма локатора включает время и продолжительность и терминал MHP имеет информацию о расписании, соответствующую этой продолжительности, тогда он должен получать и возвращать метаданные для CRID с самым большим перекрытием по времени и продолжительности локатора;
- попытки получить экземпляр ProgramEvent для экземпляра локатора должны перестать работать вместе cSIRequestFailureType из DATA_UNAVAILABLE.

Пакет org.dvb.tvanytime.metadata.ip определен в [11] (приложение AX).

11.8 Безопасность

11.8.1 Основы безопасности

Платформа MHP должна устанавливать SecurityManager перед началом обработки приложений DVB-J. В терминалах MHP, где свойство (атрибут объекта) mhp.smartcard.reader известно и имеет значение «Поддерживается» («SUPPORTED»), приложения MHP должны быть в состоянии использовать Provider, установленный приложением MHP, как определено в [11] (приложение AJ) в тех методах в пакете java.security, у которых есть в качестве входного параметр «Провайдер» (Provider).

11.8.2 API для безопасности обратного канала

Этот API определен в [63], которая включает пакет javax.microedition.io.

Примечания и дополнения должны быть в соответствии с [11] (11.8.2) и [14] (12.10).

11.8.3 Классы дополнительных полномочий

Пакеты, представляющие классы дополнительных полномочий, представлены в [11] (приложение T).

Для спецификаций терминала MHP, которые не поддерживают функциональный эквивалент условного доступа, пакет org.dvb.net.ca не требуется.

Для спецификаций терминала MHP, которые не поддерживают функциональный эквивалент SI, класс org.dvb.net.tuning.DvbNetworkInterfaceSIUtil не требуется.

Для спецификаций терминала MHP, которые не поддерживают API по 11.6.3 настоящего стандарта, класс org.dvb.net.tuning. TunerPermission) не требуется.

11.8.4 Общие вопросы безопасности

При отсутствии специальных требований подкласса java.security.Permission, проверка входных параметров с их конструкторами не нужна. Когда платформа MHP создает конструкторы, она отвечает за их создание с допустимыми значениями. Поведение платформы MHP в случае, когда приложение создает конструкторы с недопустимыми значениями стандарт, не устанавливает.

11.8.5 API шифрования

API шифрования определен в [63].

Примечание – Опциональный пакет JCE является необязательным в [58].

В терминалах MHP, где свойство mhp.smartcard.reader известно и имеет значение «Поддерживается» («SUPPORTED»), приложения MHP должны включать провайдера, установленного приложением MHP, как это определено в [11] (приложение AJ).

Терминалы MHP по умолчанию для класса javax.crypto.Cipher должны включать провайдера следующим образом:

- терминалы MHP должны поддерживать алгоритм RSA;
- дополнительно допускается поддержка терминалами MHP требований и алгоритмов 12.10.2 настоящего стандарта.

11.8.6 Расширения DVB для шифрования

11.8.6.1 Введение

«Java2 Standard Edition» включает API, обеспечивающие возможность использовать маркеры PKCS11 (такие как смарт-карты) в приложениях Java. Эти API упрощают обработку приложениями Java съемных смарт-карт. Они также должны входить в систему маркера PKCS11 для связанных операций шифрования и дешифрования.

MHP базируется на Персональном Базисном Профиле J2ME ([58]), который не предоставляет эти новые API. Поэтому следующие новые пакеты определены в настоящем стандарте в соответствии с [11] (приложение AI):

- org.dvb.security;
- org.dvb.auth.callback;
- org.dvb.net.ssl;
- org.dvb.security.pkcs11.

Пояснения к перечисленным пакетам представлены в [11] (11.8.6.1.2 - 11.8.6.1.4).

Установка провайдеров служб шифрования должна поддерживаться в соответствии с [11] (приложение AJ).

11.9 Другие программные интерфейсы приложений

11.9.1 Программный интерфейс приложений поддержки таймера

Программный интерфейс приложений поддержки таймера определяется параметрами API таймера в пакете javax.tv.util в Java TV версия 1.0 в соответствии с [11] (11.9.1).

Реализации API таймера должны обеспечивать:

- минимальный интервал повторения не более 40 мс;
- дискретность формирования интервала повторения не более 10 мс.

TVTimer.scheduleTimerSpec могут формировать TVTimerScheduleFailedException, если терминал MHP не может обеспечить больше ни одного таймера согласно [11] (приложение G, таблица 88).

Примечание – минимальный интервал повторения 40 мс был мотивирован стандартной частотой кадров 25 Гц. Однако это не подразумевает, что таймер может быть использован для покадровой мультипликации.

11.9.2 Требования к программному интерфейсу приложений установок и предпочтений пользователя

Требования к программному интерфейсу приложений установок и предпочтений пользователя должны быть в соответствии с [11] (приложение L).

Для всех API должны быть доступны установки, предпочитаемые пользователями, перечисленные ниже:

- язык пользователя;
- мнение родителей;
- код страны;
- размер шрифта (по умолчанию).

Другие установки, предпочитаемые пользователями, должны быть доступны в том случае, когда UserPreferencePermission предоставлено для «чтения» в соответствии с [11] (11.10.2.8).

11.9.3 Профили и свойства версий

Спецификации базовых терминалов MHP соответствующих профилей должны поддерживать все системные свойства, определенные в этом пункте. Для спецификаций терминалов базовой MHP свойства с указанием профиля («mhp.profile.enhanced_broadcast», mhp.profile.interactive_broadcast» и «mhp. profile.internet_access») должны быть интерпретированы как относящиеся к описаниям профиля в разделе 15 настоящего стандарта. Свойства со ссылкой на нумерацию версий должны быть интерпретированы относящимися к соответствующему номеру версии MHP.

Приложения могут обнаружить поддерживаемый профиль и версию профиля (и таким образом, обеспечивать поддержку функциональности) путем извлечения профиля и свойств версии. Если конкретный профиль не поддерживается, то соответствующие свойства версия возвращаются со значением «0». Свойства, приведенные в таблице 25, должны быть включены в набор свойств в класса java.lang. System. Из этого следует, что эти свойства можно получить с помощью java.lang.System.getProperty(). Так как этот API возвращает строку, то возвращаемые числовые значения должны быть закодированы, как это определено java.lang.Integer.toString (INT).

Таблица 25 – Свойства системы для профиля и версии опроса

Свойства	Семантика профиля	Допустимые значения	Пример
mhp.profile.enhanced_broadcast	поддержка расширенного профиля вещания	да, «0»	да
mhp.profile.interactive_broadcast	поддержка интерактивного профиля вещания	да, н«0»	«0»
mhp.profile.internet_access	поддержка доступа в Интернет	положительное целое число	«0»
mhp.eb.version.major	поддержка основной версии профиля расширенного вещания	положительное целое число	«1»
mhp.eb.version.minor	поддержка дополнительной версии профиля расширенного вещания	положительное целое число	«0»
mhp.eb.version.micro	поддержка малой версии профиля расширенного вещания	положительное целое число	«0»
mhp.ib.version.major	поддержка основной версии интерактивного профиля вещания	положительное целое число	«1»
mhp.ib.version.minor	поддержка дополнительной версии интерактивного профиля вещания	положительное целое число	«0»
mhp.ib.version.micro	поддержка минимальной версии интерактивного профиля вещания	положительное целое число	«0»
mhp.ia.version.major	поддержка основной версии доступа в Интернет	положительное целое число	«1»
mhp.ia.version.minor	поддержка дополнительной версии доступа в Интернет	положительное целое число	«0»
mhp.ia.version.micro	поддержка минимальной версии доступа в Интернет	положительное целое число	«0»
Примечание – В стандартах ГОСТ Р 54456, ГОСТ Р 55712 для профилей, не поддерживаемых терминалом, допускается вместо значения «0» возвращение значения «НЕТ».			

Свойства запрашиваемой версии профиля, которая не поддерживается терминалом MHP, не должны поддерживаться. Поэтому на вызов System.getProperty() этих свойств будет возвращен «0».

11.9.3.1 Информация об опциях

Определения обязательных и опциональных профилей представлены в разделе 15 настоящего стандарта.

Реализация MHP поддерживает опцию только, если соответствующее свойство известно и имеет значение «Поддерживается». Свойства являются частью набора свойств класса java.lang.System. Общий синтаксис свойств, которые указывают, поддерживается ли определенная функция или нет:

mhp.option.<the optional feature>.

В таблице 26 перечислены опции, определенные к настоящему моменту.

Таблица 26 – Свойства системы для опциональной функции опроса

Опция	Свойство
Многоадесный IP через канал вещания	mhp.option.ip.multicast
HTTP 1.1 через интерактивный канал	mhp.option.http.1.1
DSMCC U-U через интерактивный канал	mhp.option.dsmcc.uu
DVB-HTML	mhp.option.dvb.html
Имеет ли приемник MHP считыватель смарт-карт, доступный для приложений через API считывателя смарт-карт?	mhp.smartcard.reader
Память для сохраняемых служб (см. 9.6.2 настоящего стандарта) больше нуля (Примечание)	mhp.stored.services
Доступа к карте памяти (см. 11.5.7 настоящего стандарта)	mhp.option.memory.card
OpenType (см. 7.4.2 и 11.4.1.5.2 настоящего стандарта)	mhp.option.opentype
Высокое разрешение (см. 7.2.2 настоящего стандарта и [11] (приложение G, G.1.1.3)	mhp.option.highdef
Буферизированная анимация (см. org.dvb.ui.BufferedAnimation)	mhp.option.buffered.animation
Путеводитель по контенту вещания	mhp.option.broadband.content.guide
Привелегированные приложения	mhp.option.privileged.applications
Примечание – доступная память может быть недостаточной для хранения конкретного приложения и, следовательно, запрашивает о добавлении приложения сохраненных служб.	

11.9.4 Интерфейс API доступа к некондиционным смарт-картам

API для доступа к некондиционным смарт-картам должен включать:

- дополнительный пакет «SATSA-APDU», определенный [64];
- пакет classjavax.microedition.apdu.APDUPermission, определенный [11] (приложение B, B.1.2.1);
- пакет org.dvb.smartcard, определенный в [11] (приложение AM).

Примечания о параметрах API для доступа к некондиционным смарт-картам должны быть в соответствии с [11] (11.9.4).

11.9.5 XML синтаксичекого анализа API

11.9.5.1 API последовательного синтаксического анализа файлов XML

API последовательного синтаксического анализа файлов XML определен в [65] (раздел 2).

11.9.5.2 Java-реализация DOM для XML (JDOM)

Java-реализация объектной модели документа (DOM) использует пакет org.dvb.xml.jdom, который определен [12].

11.9.6 Аппаратный API терминала MHP

Аппаратный API терминала MHP выполняется использованием пакета org.dvb.hardware, определенного [11] (приложение AZ).

11.10 Полномочия Java

В соответствии с моделью безопасности для приложений DVB-J права доступа к файлам связаны с проверкой подлинности приложений. Неаутентифицированные приложения работают в среде «песочницы». Это означает, что некоторые формы локаторов и некоторые API будут недоступны. Приложениям, прошедшим проверку, может быть предоставлено разрешение на использование регламентированных локаторов и API. Предусмотрен механизм проверки подлинности приложений запросом «подписанного» приложения.

В [11] (12.1.3) предусмотрены и другие механизмы для создания подлинных приложений. В любом случае, в этом пункте термин «подписанное» приложение должен быть интерпретирован как приложение, которое имеет право на получение полномочий вне «песочницы» DVB-J. Приложение «неподписанное» должно интерпретироваться как состояние приложения, которое не было упаковано должным способом, не имеет подтверждения подлинности и размещается в «песочнице».

Ниже описываются полномочия, которые определены для включения в «песочницу» неподписанных приложений, и полномочия, которые могут запрашиваться в файле запроса разрешения и отображаться на объектах разрешения платформы Java.

11.10.1 Полномочия для неподписанных приложений

Политика безопасности MHP включает ряд ресурсов, которые гарантированно предоставляются приложениям при выполнении приложения. Неподписанные приложения имеют доступ к только этим ресурсам. Ниже представлен перечень пунктов согласно [11], содержащих характеристики отображений этих ресурсов к объектам Полномочий Java:

- 11.10.1.1 java.awt.AWTPermission;
- 11.10.1.2 java.net.SocketPermission;
- 11.10.1.3 java.util.PropertyPermission;
- 11.10.1.4 java.lang.RuntimePermission;
- 11.10.1.5 java.io.SerializablePermission;
- 11.10.1.6 java.io.FilePermission;
- 11.10.1.7 javax.tv.media.MediaSelectPermission;
- 11.10.1.8 javax.tv.service.ReadPermission;
- 11.10.1.9 javax.tv.service.selection.ServiceContextPermission;
- 11.10.1.10 java.util.Locale.setDefault;
- 11.10.1.11 Applications signalled in AIT File;
- 11.10.1.12 javax.microedition.xlet.ixc.IxcPermission;
- 11.10.1.13 MonitorAppPermission and ServiceTypePermission.

11.10.2 Дополнительные полномочия для подписанных приложений

Подписанные приложения могут запрашивать дополнительные полномочия. При запросе полномочий используется файл запроса разрешения ([11] (12.6). Отображение элементов в файле запроса разрешения к Полномочиям Java, которые могут быть предоставлены терминалу MHP в ответ на запрос, представлено в [11] (11.10.2.1 - 11.10.2.16).

11.11 Соответствие между объектами, локаторами контента и их текстовыми представлениями

В этом пункте отображается соответствие между объектами, типами локатора и их текстовыми представлениями и методами DVB-J. В пункте перечислены методы и конструкторы, которые принимают или возвращают (в соответствии с сигнатурой метода) экземпляры org.davic.net.Locator Java, javax.tv.locator.Locator, javax.media.MediaLocator или их подклассов. Внешней формой этих локаторов является текстовое представление, определенное в [11] (таблица 103). В тех случаях, когда указан метод, принимающий несколько форм локатора, то должен обеспечиваться прием всех форм, перечисленных в этом пункте.

В состав методов приема и возврата экземпляров объектов, соответствующих [11] входят:

- методы, обрабатывающие экземпляры объектов, описывающих транспортный поток MPEG, должны быть в соответствии с [11] (11.11.1);
- методы, обрабатывающие экземпляры объектов, описывающих сеть DVB, должны быть в соответствии с [11] (11.11.2);
- методы, обрабатывающие экземпляры объектов, описывающих букет DVB, должны быть в соответствии с [14] (11.11.3):
- javax.tv.service.transport.BouquetCollection.retrieveBouquet();
- javax.tv.service.transport.Bouquet.getLocator().
- методы, обрабатывающие экземпляры объектов, описывающих службы, должны быть в соответствии с [14] (11.11.4);
- методы, обрабатывающие экземпляры объектов, относящихся к специфическим службам MPEG/DVB, должны быть в соответствии с [11] (11.11.4.1) с примечаниями и изменениями в соответствии с [14] (11.11.4.1);

- методы, обрабатывающие экземпляры объектов со ссылкой на универсальную службу, должны быть в соответствии с [11] (11.11.4.2) при интерпретации термина «Специфическая служба MHP» как «Служба DVB» в соответствии с [14] (11.11.4.2);
- методы, обрабатывающие экземпляры объектов со ссылкой на контент для IPTV, должны быть в соответствии с [11] (11.11.4.3), должны включать все методы, перечисленные в [11] (11.11.1 и 11.11.2);
- методы, обрабатывающие экземпляры объектов со ссылкой на службы хранения, должны быть в соответствии с [11] (11.11.4.4);
- методы, обрабатывающие экземпляры объектов со ссылкой на службы клиентов Интернет, должны быть в соответствии с [14] (11.11.4.5):
- org.dvb.application.storage.StoredApplicationService.getLocator;
- javax.tv.service.Service.getLocator – служба представлена StoredApplicationService;
- методы, обрабатывающие экземпляры объектов, описывающих события DVB, должны быть в соответствии с [11] (11.11.5) с примечаниями и изменениями в соответствии с [14] (11.11.5);
- методы, обрабатывающие экземпляры объектов, описывающих элементарный поток MPEG, должны быть в соответствии с [11] (11.11.6) с примечаниями и изменениями в соответствии с [14] (11.11.6);
- методы, обрабатывающие ссылочные файлы должны быть в соответствии с [11] (11.11.7) с примечаниями и изменениями в соответствии с [11] (11.11.7);
- методы, возвращающие экземпляр локатора, ссылающегося на каталог, должны быть в соответствии с [11] (11.11.8) с примечаниями и изменениями в соответствии с [14] (11.11.8);
- методы обработки локаторов со ссылкой на декодер drip feed должны быть в соответствии с [11] (11.11.9);
- «несущественные» методы должны быть в соответствии с [11] (11.11.10);
- методы, обрабатывающие множество типов локаторов, должны быть в соответствии с [14] (11.11.11);
- методы и классы, поддерживающие протокол HTTP в DVB-J, должны быть в соответствии с [11] (11.11.12);
- методы и классы, поддерживающие приложения MHP, должны быть в соответствии с [11] (11.11.13).

11.12 Автономные приложения

11.12.1 Общее поведение

Расширенная модель MHP позволяет выполнять несколько форм приложений в автономном режиме независимо от любых служб вещания. Окружение этих автономных приложений должно быть в соответствии с [11] (11.12.1) с примечаниями и изменениями в соответствии с [14] (11.12.1).

11.12.2 Службы хранения

Спецификации MHP поддерживают две формы хранения приложений, определенных в 9.1.9 и в 9.6.2 настоящего стандарта.

11.12.2.1 API хранения приложений

API хранения приложений определяются в [11] (приложение AG).

11.12.2.2 Приложение API MHP 1.0 с измененным поведением

API, определенные в MHP 1.0, применяют расширенную семантику для поддержки листинга и запуска сохраненных служб в соответствии с [11] (11.12.2.2).

11.13 Поддержка DVB-HTML

11.13.1 API объектной модели документа

11.13.1.1 Платформа

Необходимые API объектной модели документа определены из объектной модели документа (DOM) уровень 2 в соответствии с [14] (11.13.1.1).

11.13.1.2 Расширения определенные DVB

Расширения DOM, определенные DVB, представлены в [14] (приложение AD).

11.13.1.3 Доступ к DOM (только для чтения)

В тех случаях, когда приложение DVB-J только имеет доступ к DOM или запрашивает доступ к DOM только для чтения, поддержка документа выполняется в соответствии с [14] (11.13.1.3).

11.13.2 Встроенные интерфейсы Xlet в странице DVB-HTML

Список семантических различий в API MHP между приложениями DVB-J и Xlet, встроенным в страницу DVB-HTML приложения DVB-HTML, представлен [14] (11.13.2).

11.14 Доступ в Интернет

11.14.1 Прикладные программные интерфейсы управления Интернет- клиентами

Параметры прикладных программных интерфейсов приложений управления Интернет-клиентам должны быть в соответствии с [11] (приложение АН).

Интернет-клиенты, доступные приложениям MHP, должны появляться в списке служб, обслуживаемых методом SIManager, которые возвращаются при вызове метода SIManager.filterServices. Детализированные правила поддержки приложений Интернет- клиентов должны быть в соответствии с [11] (11.14.1).

11.14.2 Поддержка Интернет-апплета

Поддержка Интернет-апплета определена в 11.14.2.1-11.14.2.5 настоящего стандарта.

11.14.2.1 HTML теги

Интернет-апплеты должны поддерживаться с помощью меток «<applet>» и «<object>» с семантикой, определенной в [51].

11.14.2.2 Платформа Java

Платформа Java для Интернет-апплетов должна быть в соответствии с JSR [66].

11.14.2.3 Безопасность

Настоящий стандарт не требует поддержки апплетов со знаком или механизма подписывания кода и API, определенных в [66].

Если апплеты могут получить доступ к интерфейсам API Java, которые определеляются в раздела 11 настоящего стандарта, за исключением требований, перечисленных в 11.14.2 настоящего стандарта, то они должны иметь разрешения на полномочия доступа приложений MHP без знака. Из полномочий, доступных приложениям MHP со знаком, доступны:

- java.util. PropertyPermission;
- java.net. SocketPermission.

11.15 Интерфейсы API, определенные в OCAP

Этот подраздел не является обязательным для всех профилей MHP. Спецификации терминала MHP могут включать в себя всю или часть спецификации элементов этого подраздела или могут не учитывать эту спецификацию.

11.15.1 Введение

Настоящий стандарт включает отдельные API, определенные в [13]. Многие классы в OCAP полностью или частично определены для среды кабельного телевидения США и не применимы в других регионах. В [11] (таблица 42) приведены классы, включенные от OCAP, и их применимость на соответствие настоящему стандарту.

11.15.2 Пакет org.ocap.application

Поддержка пакета org.ocap.application ([13] (приложение G)) должна выполняться с изменениями в соответствии с [11] (11.15.2).

11.15.3 Пакет org.ocap.service

Пакет org.ocap.service ([13] (приложение P)) должен поддерживаться за исключением ServiceContextResourceUsage.

11.15.4 Пакет org.ocap.system

Из пакета org.ocap.system ([13] (приложение Q)) должен поддерживаться MonitorAppPermission. Должны поддерживаться следующие разрешения имен:

- «регистратор»;
- «servicemanager»;
- «служба»;

- «handler.appFilter»;
- «безопасность»;
- «перезагрузка»;
- «systemevent».

11.15.5 Пакет org.ocap

Должен поддерживаться пакет org.ocap ([13] (приложение O)). Требования инициализации API для проверки соответствия не применяются, поскольку эти интерфейсы в соответствии с настоящим стандартом не используются.

11.15.6 Пакет org.ocap system.event

Должен поддерживаться пакет org.ocap.system ([13] (приложение U)).

12 Безопасность

12.1 Введение

Этот раздел нормирует следующие области параметров, определяющих безопасность платформы МНР:

- аутентификации приложений;
- политики безопасности приложений;
- аутентификации и конфиденциальности обратного канала;
- управления сертификатом.

12.1.1 Платформы безопасности приложений

Платформа безопасности позволяет приемнику аутентифицировать источник кода приложения или других файлов. В случае аутентификации файлов кода приложения приемник должен предоставить приложению права доступа для уязвимых по безопасности ресурсов в соответствии с [11] (12.6).

В системе безопасности используется 3 типа сообщений аутентификации: хэш-код шифрования, подписи, сертификаты.

Хэш-код шифрования представляет собой битовую строку фиксированной длины, полученную из массива произвольной длины использованием хэш функции (функции свертки).

Подписи предоставляют основной хэш-код, вычисленный по соответствующим данным, «подписанный» санкционирующей организацией. Процесс подписания надежно связывает основной хэш-код с подписавшей стороной и показывает, что данные не были фальсифицированы, так как они подписаны.

Сертификаты обеспечивают «цепочку доверия» от организации авторизации до доверенной третьей стороны, которой можно доверять (центра сертификации), который известен приемнику.

Сообщения аутентификации передаются в файлах файловой системы. Это позволяет применять схему аутентификации к любой иерархической файловой системе, работающей в каналах вещания или по обратным каналам.

Приложения, которые имеют право считаться доверенными, должны быть идентифицированы application_id в диапазоне величин, выделенных для приложения «со знаком» в соответствии с [11] (10.4.3, таблица 13) и таблицей 17 настоящего стандарта.

Приложения, которые не имеют права считаться доверенными, будут идентифицированы application_id в диапазоне величин, выделенных для приложений «без знака». Для приложения с application_id из области величин «со знаком» в процессе аутентификации возникает сбой, если это приложение неподписано. Приложения с application_id из области величин «без знака» не имеют права считаться доверенными, даже если файлы были переданы с подписями.

12.1.2 Безопасность обратного канала

Настоящий стандарт предусматривает использование протоколов общего назначения и комплекта шифрования, соответствующих [11] (12.1.2).

12.1.3 Платформа МНР – подписанное приложение

В платформе МНР доверие к приложению устанавливается только к приложению, имеющему подпись.

12.2 Аутентификация приложений

Название профилей или задач «аутентификация приложения», не требующих соответствия с функциональным эквивалентом, определено в [11] (15.6). Функциональный эквивалент должен обеспечить механизм установления кода, которому можно доверять, чтобы позволить предоставление любых полномочий по запросу приложения МНР. Платформа безопасности должна содержать механизм подписывания отдельных файлов или наборов файлов с сертификатом, который может быть отличающимся от любого, который используется, чтобы определить доверительный уровень приложения.

Примечание – Файлы с данными, которые подписаны, могут иметь свои сертификаты для приложений, используя методы getSigners, описанные в [11] (приложение P, P.2.1).

12.2.1 Обзор сообщений аутентификации

Используются три различных типов сообщений: хэш-коды, подписи и сертификаты. Каждое сообщение помещается в файл. Размещение файлов зависит от их функций и определяется под соответствующими заголовками в соответствии с [11] (12.4).

12.2.1.1 Хэш-коды

В настоящем стандарте описывается применение хэш-кодов на следующие виды информации:

- файлы;
- директории.

При вычислении хэш рассматриваются содержание и атрибуты объектов, а не конкретная транспортная информация. Аутентификация не зависит от базового транспортного протокола.

В случае директории значение хэш зависит от значения хэш объектов, связанных с ним, и обеспечивает хэш всех объектов, которые должны быть аутентифицированы в «дереве» под ним.

12.2.1.2 Подписи

Аутентифицируемые данные являются иерархической файловой системой (например, ОС DSM-CC). Корень аутентифицируемого «дерева» переносит одну или более подписей. Это позволяет одной или более организациям подписывать набор информации.

Корень аутентифицируемого «дерева» может быть корневым каталогом файловой системы или «главным» каталогом «поддерева».

12.2.1.3 Сертификаты

Сертификат содержит открытый ключ, который может использоваться для декодирования хэш-кода, содержащегося в подписи, и обеспечить проверку «дерева»/«поддерева». Сам сертификат подписан более высоким органом сертификации. Для правильной аутентификации «поддерево» должно соответствовать «цепочке» сертификатов от подписи до корневого сертификата.

12.2.1.4 Аутентификация иерархических файловых систем

Решение аутентификации иерархических файловых систем основано на аутентификации иерархической структуры объектов. Хэш-коды вычисляются систематически и с накоплением через отдельные или через все объекты в иерархии.

Подпись наверху иерархии идентифицирует источник объектов. Платформа обеспечивает гибкий и нетрудоемкий метод, разрешающий аутентификацию «поддеревьев» файловой системы с единственной подписью. Начиная с проверки подписи, процесс становится более трудоемким, чем проверка величин хэш-кода. Этот механизм более эффективен, чем подписание каждого объекта «поддерева».

Проверка хэш-кода в реальном времени необходима только загружаемым объектам. Данный механизм не требует аутентификации целого поддерева.

12.3 Параметры передачи сообщений безопасности

Сообщения аутентификации транспортируются в файлах. Службы передачи не должны получать доступ к контенту файла. В том случае, если файловой системой является Карусель Объектов, то IOR для файлов сообщения аутентификации должен использовать только тело профиля BIOP.

12.4 Детализация сообщений аутентификации приложения

Три структуры данных определены для передачи информации аутентификации:

- HashFile;
- SignatureFile;
- CertificateFile.

Они размещаются в файлах файловой системы. Расположение файла зависит от его функции.

Подробности формирования файлов сообщений HashFile, SignatureFile и CertificateFile представлены в [11] (12.4.1, 12.4.2, 12.4.3).

Последовательность выполнения аутентификации файла представлена в [11] (12.4.4).

12.5 Параметры профилирования приложений

Параметры профилирования приложений, соответствующих [67], определенных в соответствии с [68], для аутентификации приложений вещания платформы MHP должны быть в соответствии с [11] (12.5).

12.6 Политика безопасности для приложений

Политика безопасности для приложений определяет правила доступа к загруженным приложениям. Политика доступа к загруженным приложениям определяется совокупностью двух факторов:

- права доступа по требованию вещателя с использованием сигнализации;
- права доступа, предоставляемые пользователю.

Детализированные параметры политики безопасности должны быть в соответствии с [11] (12.6).

12.7 Пример создания приложения, которое может быть аутентифицировано

Пример организации файловой системы, переносящей приложение, приведен в [11] (12.7.1). Файловая система выполняет одно подписанное приложение Xlet1.

Основным классом приложения является файл root/Xlet1/classes/Xlet1.class. *Другие классы содержатся в следующих файлах:*

- root/Xlet1/classes/foo1.class;
- root/Xlet1/classes/subclasses/sub1.class;
- root/Xlet1/classes/subclasses/sub2.class.

Xlet1 использует данные, расположенные в файле root/Xlet1/data/Xlet1.dat. Этот файл не прошел проверку подлинности.

Каждая субдиректория, которая содержит подписанные файлы, содержит hashfile.

Детализированное описание примера создания приложения приведено в [11] (12.7.2 -12.7.2.4).

12.8 Процесс управления сертификатом

Процесс управления сертификатом включает операции управления списком отозванных сертификатов (Certificate Revocation Lists; CRL) и операции управления корневым сертификатом с использованием механизма сообщений управления корневым сертификатом (Root Certificate Management Messages; RCMM).

Параметры управления списком отозванных сертификатов должны быть в соответствии с [11] (12.9.1).

Параметры управления корневым сертификатом должны быть в соответствии с [11] (12.9.2).

12.9 Параметры безопасности обратного канала

Основной вклад в безопасность обратного канала обеспечивается применением протокола безопасности транспортного уровня (Transport Layer Security; TLS) в соответствии с [11] (12.10) и [69].

12.9.1 Функциональные возможности приложений MHP (при обеспечении безопасности)

Безопасность обратного канала MHP обеспечивается использованием комплектов шифров, представленных в 12.10.2 настоящего стандарта.

К MHP не следует предъявлять требования:

- выполнения функций сервера протокола TLS;
- соответствия требованиям протокола SSL 3.0.

Терминалы MHP должны поддерживать аутентификацию клиента для TLS, где источник клавиши поступает на класс javax.net.ssl.SSLContext через javax.net.ssl.KeyManager.

12.9.2 Комплекты шифров TLS

Минимально-необходимый комплект шифров, которые должны использовать реализации профиля TLS платформы MHP, включает в себя следующие шифры:

- RSA ();
- MD5;

- SHA-1;
- DES;
- AES.

Детализация методов шифрования для реализаций MHP представлена в [11] (12.10.2, таблица 56). Определения терминов приведены в [70].

12.9.3 Загрузка сертификатов протокола TLS

Перед установлением соединения по протоколу TLS платформа MHP должна гарантировать наличие в списке сертификатов, отправленных сервером, не менее одного надежного (которому можно доверять) сертификата. Эта операция выполняется проверкой списка сертификатов относительно резидентного сертификата платформы MHP. В среде MHP загружаемое приложение может устанавливать сеансы TLS, которые могут быть использованы, например, для передачи конфиденциальных сообщений.

Интерфейс API, используемый загружаемым приложением, описан в 11.8.2 настоящего стандарта и в [11] (11.8.2).

Процесс проверки подлинности сервера включает в себя проверку цепочки сертификатов, отправленных сервером TLS.

Правила использования сертификата TLS для случаев, когда корневые сертификаты передаются вместе с заявкой, должны быть в соответствии с [11] (12.10.3.2.1).

Правила использования сертификата TLS для случая, когда передача сертификатов не предусмотрена, должны быть в соответствии с [11] (12.10.3.2.2).

Реализации MHP могут игнорировать поле распределения списка отзыва сертификатов во время аутентификации TLS сервера через обратный канал.

12.10 Параметры профиля Интернет

Параметры профиля Интернет на основе [67] по версии [68], описывающего основную часть сертификатов и стандартные расширения сертификатов, приведены в [11] (12.11).

12.11 Аппаратная реализация платформы MHP

Аппаратные средства платформы MHP должны поддерживать следующие параметры, чтобы обеспечивать работоспособность модели обеспечения безопасности MHP:

- количество подписей N ([11] (12.9.2.2)) - не менее 2;
- глубина цепочки сертификата - не менее 5 уровней;
- не менее 8 списков CRL;
- не менее 10 записей на CRL;
- не менее 3 корневых сертификатов (независимо от числа базовых корневых центров сертификации);
- не менее 3 корневых сертификатов для тестирования (см. [11] (12.9.3)).

Применение функциональных эквивалентов механизмов аутентификации настоящим стандартом не допускается [14] (12.10).

12.12 Безопасность применения плагинов

Безопасность применения плагинов должна обеспечиваться аналогично безопасности применения любого приложения DVB-J. Аспекты безопасности делегированного приложения не могут ограничивать полномочия, доступные плагину. Плагин запрашивает набор полномочий, необходимых для поддержки делегированного приложения.

Для делегированного приложения, соответствующего модели обеспечения безопасности, определенной в настоящем стандарте, плагин должен реализовать систему обеспечения безопасности для предоставления подмножества полномочий к делегированному приложению в соответствии с [11] (приложение AF). Для делегированного приложения модель обеспечения безопасности которого не соответствует настоящему стандарту, плагин может дополнительно реализовать систему собственной безопасности для управления подмножеством этих полномочий для делегированных приложений.

12.13 Безопасность загрузки приложений из интерактивного канала

Безопасность загрузки приложений из интерактивного канала должна обеспечиваться аналогично случаю загрузки приложений из вещательного канала, но со своими особенностями.

Особенности загрузки приложений из интерактивного канала включают в себя: инициацию начала загрузки, загрузку приложения, динамическую загрузку на интервале жизни приложения, загрузку кода приложения и поддержку загруженных данных. Детализация правил загрузки должна быть в соответствии с [11] (12.14.1).

12.14 Хранение приложений

При выполнении приложений все файлы сохраненных приложений должны быть полностью аутентифицированы как определено в [11] (12.4.4).

Объемы аутентификаций, выполняемых во время хранения и выполнения приложений, отличаются в соответствии с [11] (12.15.1, 12.15.2).

Во всех случаях используемые сертификаты должны быть допустимыми на время выполнения (например, не истекли сроки применения или не отменены, и не были удалены корневые сертификаты, которые используют сертификаты).

Нормальные ограничения на аутентификацию, включая 11.2.3 настоящего стандарта, будут соблюдаться при выполнении приложения. Это относится к случаю, когда часть файлов класса хранится, а часть не хранится. Отклонения от этого правила могут привести к сбою процесса аутентификации, даже если все сертификаты допустимы.

12.14.1 Автономные сохраненные приложения

На этапе хранения приложения сохраненные файлы должны аутентифицироваться, так как будто они загружались приложением. На этом этапе терминал не должен специально обрабатывать файлы класса.

В соответствии с [11] (12.4.3.6) в случае, если сертификат, который был проверен, но во время процесса хранения он был отменен или истек срок его действия, то любые дополнительные цепочки сертификата, которые не были проверены во время хранения, должны быть полностью проверены при использовании тех же правил, какие применялись бы к файлу вещательной передачи.

Если автономные приложения хранятся с доступом к файлам вещательных передач (например, использующие API org.dvb.dsmcc), должны примениться к этим файлам требования [11] (12.4.4).

Во время выполнения должна проверяться пригодность с не истекшим сроком годности и не отозванных любых сертификатов, используемых для аутентификации учетных данных.

12.14.2 Кэшируемые приложения

Аутентификация кэшируемых приложений определяется реализацией аутентификации, выполняемой во время хранения, с примечаниями в соответствии с [11] (12.15.2).

12.15 Безопасность внутренних приложений и контента, встроенного в другие приложения

Безопасность внутренних приложений и контента, встроенного в другие приложения, обеспечивается наследованием набора полномочий, предоставленных приложению, в которое они встроены.

12.16 Аутентификации несвязанных приложений

Все несвязанные приложения должны быть подписаны и аутентифицированы. Приложения без подписи или приложения подписанные, создающие сбой аутентификации, не должны запускаться как несвязанные приложения, даже если они были определены как несвязанные.

12.17 Аутентификация привилегированных приложений

Привилегированные приложения должны подвергаться дополнительной аутентификации в дополнение к нормальным механизмам аутентификации.

MHP к привилегированным приложениям не устанавливает никаких конкретных дополнительных механизмов аутентификации. Рекомендации по формированию дополнительных механизмов аутентификации представлены в [11] (12.18).

Приложения MHP, которые не прошли дополнительную аутентификацию, не должны получать экземпляры MonitorAppPermission. Для получения любых экземпляров MonitorAppPermission приложение MHP может быть успешно аутентифицируемым дополнительной системой аутентификации, сообщенной с application_id в диапазоне значений. Это требование является в настоящем стандарте обязательным в соответствии с [11] (12.1.8).

13 Параметры эталонной модели графики платформы MHP

MHP предоставляет средства для управления позиционированием: видео на устройстве вывода, компоненты интерфейса в диалоговом окне, такие, как кнопки и списки, а также необработанные графические примитивы.

Каждый экран, подключенный к MHP, имеет три плоскости, которые являются плоскостью фона, плоскостью видео и плоскостью графики.

Обязательные требования к графическим возможностям терминала платформы MHP должны быть в соответствии с [11] (приложение G, G.1).

14 Системная интеграция платформы MHP

Системная интеграция платформы MHP обеспечивается совокупностью параметров следующих форматов:

- отображения пространства имен объектов и файлов;
- имен файлов с учетом зарезервированных имен;
- нотации XML;
- сетевой сигнализации;
- кодирования текста идентификаторов приложений;
- имен файлов с учетом обеспечения сохраняемости контента;
- файлов и имен файлов, обеспечивающих доступ MHP к контенту, сохраненному в файлах;
- типов объектов, локаторов и их текстовых представлений;
- механизмов, обеспечивающих идентификацию службы;
- механизмов, обеспечивающих функционирование MHP с системой условного доступа.

Требования к перечисленным частям должны быть в соответствии с ГОСТ Р 54456 (13.1–13.10).

15 Детализированные определения профилей платформы MHP

Детализированные определения профилей платформы MHP должны быть в соответствии с [14] (таблица 104).

15.1 Требования к формату PNG

Требования к формату PNG должны быть в соответствии с ГОСТ Р 54456 (14.1).

15.2 Требования к составу форматов медиа, поддерживаемых API DVB-J

Требования к составу форматов медиа, поддерживаемых API DVB-J, должны быть в соответствии с ГОСТ Р 54456 (14.2).

15.3 Требования к формату JPEG

Требования к формату JPEG должны быть в соответствии с ГОСТ Р 54456 (7.1.1.2, 14.3).

15.4 Требования к поддержке локалей

Требования к поддержке локалей должны быть в соответствии с ГОСТ Р 54456 (14.4).

15.5 Требования к растру видеоизображения

Требования к растру видеоизображения в зависимости от формата видеоизображения должны быть в соответствии с ГОСТ Р 54456 (14.5).

15.6 Отображение платформой MHP функциональных эквивалентов GEM

Большинство функциональных эквивалентов определены в [11] как абстракции возможностей, ранее определенных в MHP. Применение функциональных эквивалентов в настоящем стандарте не разрешено, однако конструкторам MHP целесообразно учитывать возможные отображения функциональных эквивалентов GEM и соответствующих функций MHP. Это отображение представляется в [14] (таблица 105).

16 Требования к константам MHP

16.1 Требования к системным константам

Требования к системным константам MHP должны быть в соответствии с [11] (16.1, таблица 65).

16.2 Требования к константам DVB-J

Подраздел не содержит констант для класса, который не требуется в соответствии с настоящим стандартом (например, класс в пакете org.dvb.si). Этот пункт содержит величины общедоступных статических финальных статических символов различных API Java.

16.2.1 Общедоступные и защищенные заключительные статические примитивные поля от пакетов DVB

Список значений, присвоенных общедоступным и защищенным заключительным статическим примитивным полям, определенным в пакетах DVB-J, должен быть в соответствии с [11] (16.2.1).

17 Доступ Интернет-клиентов в сеть Интернет

17.1 Ссылки служб DVB и контента на контент WWW

Клиенты доступа к Интернету платформы MHP должны поддерживать следующие механизмы ссылки на службы DVB и на контент, содержащийся в сети WWW:

- использование гипертекстовой ссылки «href» должно запускать службу DVB так же, как это делает приложение MHP, используя API выбор службы MHP или конечный пользователь терминала MHP с помощью навигатора выбора службы DVB;
- Интернет-клиенты доступа MHP в Интернет, не охваченные моделью безопасности MHP, должны быть в состоянии выполнять выбор служб и настройку без учета разрешений;
- Интернет-клиенты доступа MHP в Интернет должны поддерживать атрибуты element.attribute, определенные в [14] (таблица 6) при использовании элементов типов медиа с атрибутами элементов в соответствии с [14] (таблица 8). Дополнительным условием является использование только следующих типов MIME:
  - audio/mpeg, video/mpeg;
  - multipart/dvb.service.

17.2 Требования к Интернет-клиентам

В таблице 27 представлены протоколы, которые должны поддерживаться приложениями Интернет-клиентов, и которые являются частью профиля MHP доступа в Интернет.

Таблица 27 – Минимально необходимые протоколы

Приложение Интернет-клиент	Минимально необходимые протоколы
Браузер WWW	HTTP в соответствии с [11] (6.3.7.2). HTTPS в соответствии с [11] (16.3.7.3)
Клиент email	Протокол SMTP для передачи email в соответствии с [71] или HTTP для сервера WebMail. Протоколы приема email зависят от реализации
Новости Usenet	NNTP в соответствии с [72] или HTTP для сервера WebNews

Интернет-клиенты браузера WWW должны поддерживать URL «mailto», определенный [35], [73].

Интернет-клиенты электронной почты должны поддерживать использование URL «http» и «https» в соответствии с [33] (3.2.2).

Для случая Интернет-клиентов новостей Usenet Интернет-клиенты, представленные браузером WWW и email, должны поддерживать использование URL «news» в соответствии с [74].

Интернет-клиенты новостей Usenet должны поддерживать использование URL «http», «https» и «mailto».

17.3 Обработка потокового медиа из сети Интернет

Настоящий стандарт не предъявляет требований к обработке терминалами MHP контента потоков медиа из сети Интернет. Однако в тех случаях, когда такая обработка поддерживается и терминалам MHP доступны приложения DVB-J, то обработку рекомендуется выполнять при использовании медиа платформы Java в соответствии с [62].

17.4 Управление соединением Интернет

Приложения Интернет-клиента должны по умолчанию использовать ISP сконфигурированный в приложение MHP. Детали управления соединением Интернет должны быть в соответствии с [11] (17.4).

Библиография

[1]	ETSI TS 102 034	Digital Video Broadcasting (DVB); Transport of MPEG-2 Based DVB Services over IP Based Networks
[2]	ETSI TS 102 539	Digital Video Broadcasting (DVB); Carriage of BCG information over IP
[3]	ISO/IEC 13818-2 1996	Information technology—Generic coding of moving pictures and associated audio information - Part 2: Video (MPEG-2 Video)
[4]	IETF RFC 2326 April 1998	Real Time Streaming Protocol (RTSP)
[5]	IETF RFC 1112	Host Extensions for IP Multicasting
[6]	ITU-T Recommendation Z.100 (11/2007)	SERIES Z: Languages and general software aspects for telecommunication systems. Formal description techniques (FDT) – Specification and Description Language (SDL). Specification and Description Language (SDL)
[7]	DAVIC 1.4.1, p9 June 1999	DAVIC 1.4.1 Specification Part 9. Complete DAVIC. Specifications, DAVIC
[8]	EUROPEAN STANDARD EN 50221 February 1997	Common Interface Specification for Conditional Access and other Digital Video Broadcasting Decoder Applications
[9]	CSS 2, 1998	Cascading Style Sheets, level 2 CSS2 Specification, Including the errata published in REC-CSS2-19980512
[10]	ECMA-262	ECMAScript Language Specification
[11]	ETSI TS 102 728 (V1.1.1)	Digital Video Broadcasting (DVB); Globally Executable MHP (GEM) Specification 1.2.2 (including IPTV)
[12]	ETSI TS 102 816	Digital Video Broadcasting (DVB); PVR/PDR Extension to the Multimedia Home Platform
[13]	OCAP 1.1	OpenCable Application Platform Specification OCAP 1.1 Profile
[14]	ETSI TS 102 727 (V1.1.1)	Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.2.2
[15]	ETSI TS 102 812 V1.2.1 (2003-06)	Digital Video Broadcasting (DVB); Multimedia Home Platform (MHP) Specification 1.1.1
[16]	ISO/IEC 13818-1	Information technology - Generic coding of moving pictures and associated audio information: Systems
[17]	ISO/IEC 13818-6 (1998)	Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extensions for DSM-CC
[18]	ETSI EN 301 192 (V1.3.1)	Digital Video Broadcasting (DVB); DVB specification for data broadcasting
[19]	ETSI TR 101 202 (V1.1.1)	Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting
[20]	IETF RFC 791 (1981)	Internet protocol
[21]	IETF RFC 768 (1980)	User Datagram Protocol
[22]	ETSI EN 300 468	Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems
[23]	ETSI ETR 211 (1997)	Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)
[24]	ETSI ES 200 800	Digital Video Broadcasting (DVB); Interaction channel for Cable TV distribution systems (CATV)
[25]	ETSI ETS 300 801	Digital Video Broadcasting (DVB); Interaction channel through Public Switched Telecommunications Network (PSTN)/ntegrated Services Digital Networks (ISDN)

[26]	ETSI EN 301 193 (V1.1.1)	Digital Video Broadcasting (DVB); Interaction channel through the Digital Enhanced Cordless Telecommunications (DECT)
[27]	ETSI EN 301 195 (V1.1.1)	Digital Video Broadcasting (DVB); Interaction channel through the Global System for Mobile communications (GSM)
[28]	ETSI EN 301 199 (V1.2.1)	Digital Video Broadcasting (DVB); Interaction channel for Local Multi-point Distribution Systems (LMDS)
[29]	ETSI TR 101 201 (V1.1.1)	Digital Video Broadcasting (DVB); Implementation guidelines for Data Broadcasting
[30]	ETSI EN 301 790 (V1.2.2)	Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems
[31]	IETF RFC 793 (1981)	Transmission control protocol
[32]		The Common Object Request Broker: Architecture and Specification», Object Management Group
[33]	IETF RFC 2616 (1999)	Hypertext Transfer Protocol -- HTTP/1.1
[34]	IETF RFC 1945 (1996)	Hypertext Transfer Protocol -- HTTP/1.0
[35]	IETF RFC 2818 (2000)	HTTP over TLS
[36]	IETF RFC 1034 (1987)	Domain names - concepts and facilities
[37]	IETF RFC 1035 (1987)	Domain names - implementation and specification
[38]	IETF RFC 1982 (1996)	Serial Number Arithmetic
[39]	IETF RFC 2181 (1997)	Clarifications to the DNS Specification
[40]	ATSC A/100-5 (2003)	DTV Application Software Environment Level 1 (DASE-1) - Part 5: ZIP Archive Resource Format
[41]	ISO/IEC 10918-1	Information technology - Digital compression and coding of continuous-tone still images: Requirements and guidelines
[42]	JFIF	JPEG File Interchange Format». Eric Hamilton, C-Cube Microsystems
[43]	W3C Recommendation 1 October 1996, Version 1.0	PNG (Portable Network Graphics) Specification
[44]	GIF 89a	Graphics Interchange Format (sm)», version 89a
[45]	ISO/IEC 11172-3 (1993)	Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s - Part 3: Audio
[46]	ETSI TS 101 154 (V1.8.1)	Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream
[47]	ETSI TR 101 154 (V1.6.1)	Digital Video Broadcasting (DVB); Implementation guidelines for the use of MPEG-2 Systems, Video and Audio in satellite, cable and terrestrial broadcasting applications
[48]	XML 1.0	Second Edition: Extensible Markup Language (XML) 1.0
[49]	ISO 10646-1	Information technology - Universal multiple-octet coded character set (UCS), Part 1: Architecture and Basic Multilingual Plane
[50]	Namespaces in XML, 1999	World Wide Web Consortium 14-January-1999
[51]	HTML 4, 4.01	HTML 4.01 Specification, W3C Recommendation 24 December 1999
[52]	W3C Recommendation, 13 November, 2000	Document Object Model (DOM) Level 2 Events Specification, V1.0
[53]	W3C Recommendation, 13 November, 2000	Document Object Model (DOM) Level 2 Core Specification, V1.0
[54]	IETF RFC 2109 (1997)	HTTP State Management Mechanism
[55]	IETF RFC 2396 (1998)	Uniform Resource Identifiers (URI): Generic Syntax

[56]	ETSI TR 101 162	Digital broadcasting systems for television, sound and data services; Allocation of Service Information (SI) codes for Digital Video Broadcasting (DVB) systems
[57]	ZIP, 2003, ATSC A/100-5	DASE-1 ZIP Archive Resource Format
[58]	JSR 217	Personal Basis Profile (PBP) 1.1
[59]	ISO 8859 (все части)	Information technology -- 8-bit single-byte coded graphic character sets
[60]	ISO 6937	Information technology -- Coded graphic character set for text communication – Latin alphabet
[61]	HAVi, v1.1	HAVi v1.1 Chapter 8, 15-May-2001 HAVi v1.1 Java L2 APIs, 15-May-2001 HAVi v1.1 Chapter 7,15-May-2001
[62]	JSR 927	Java TVTM API 1.1
[63]	JSR 219: FP 1.1	Foundation Profile 1.1 for Java 2 Platform, Micro Edition
[64]	JSR 177 - SATSA, v1.0	Security and Trust Services API (SATSA) for Java 2 Platform, Micro Edition
[65]	JSR 172, v1.0	J2ME Web Services Specification
[66]	JSR 216	Personal Profile 1.1
[67]	ITU-T Recommendation X.509 (1997)	Information technology - Open Systems Interconnection -The Directory: Authentication framework
[68]	IETF RFC 2459 (1999)	Internet X.509 Public Key Infrastructure Certificate and CRL Profile
[69]	IETF RFC 2246 (1999)	The TLS Protocol Version 1.0
[70]	IETF RFC 3268 (2002)	Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)
[71]	IETF RFC 821 (1982)	Simple Mail Transfer Protocol
[72]	IETF RFC 977 (1982)	Network News Transfer Protocol
[73]	IETF RFC 2368 (1998)	The mailto URL scheme
[74]	IETF RFC 1738 (1994)	Uniform Resource Locators (URL)

УДК 621.397:681.327.8:006.354 ОКС 33.170 ОКП 657400

Ключевые слова: телевидение вещательное цифровое, домашняя мультимедийная платформа, протокол DSM-CC, профиль, приложение, вещание, доступ в Интернет, безопасность

Редактор *М.М. Меламед*
Технический редактор *А.Б. Заварзина*
Корректор *В.Г. Смолин*
Компьютерная верстка *Д.Е. Першин*

Сдано в набор 24.09.2015. Подписано в печать 8.10.2015. Формат 60x841/8. Гарнитура Ариал.
Усл. печ. л. 12,09. Уч.-изд. л. 10,90. Тираж 30 экз. Зак. 3399.

Набрано в ООО «Академиздат».

gost.e-ecolog.ru gost.e-ecolog.ru

Издано и отпечатано во
ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.

gost.e-ecolog.ru gost.e-ecolog.ru

gost.e-ecolog.ru