ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

ГОСТ Р 54997— 2012

СИСТЕМА ЦИФРОВОГО ЗВУКОВОГО РАДИОВЕЩАНИЯ DRM. ЦИФРОВОЕ ЗВУКОВОЕ РАДИОВЕЩАНИЕ DAB

Требования транспортировки и бинарного кодирования для электронного справочника программ (EPG)

Издание официальное

Москва
Стандартинформ
2013

gost.e-ecolog.ru

Предисловие

Цели и принципы стандартизации в Российской Федерации установлены Федеральным законом от 27 декабря 2002 г. № 184-ФЗ «О техническом регулировании», а правила применения национальных стандартов Российской Федерации — ГОСТ Р 1.0—2004 «Стандартизация в Российской Федерации. Основные положения»

Сведения о стандарте

1 РАЗРАБОТАН Федеральным государственным унитарным предприятием Ордена Трудового Красного Знамени научно-исследовательским институтом радио, Самарский филиал «Самарское отделение научно-исследовательского института радио» (филиал ФГУП НИИР-СОНИИР)

2 ВНЕСЕН Управлением технического регулирования и стандартизации Федерального агентства по техническому регулированию и метрологии

3 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ Приказом Федерального агентства по техническому регулированию и метрологии от 19 сентября 2012 г. № 361-ст

4 Настоящий стандарт разработан с учетом основных нормативных положений Европейского института по стандартизации в области телекоммуникаций (ETSI) ЕТСИ «Цифровое вещание аудио; Всемирное цифровое радио. Требования транспортировки и бинарного кодирования для электронного справочника программ (EPG) (ETSI TS 102 371 V1.2.1 (2006-02) Technical Specification. Digital Audio Broadcasting (DAB); Digital Radio Mondiale (DRM); Transportation and Binary Encoding Specification for Electronic Programme Guide (EPG)

5 ВВЕДЕН ВПЕРВЫЕ

Информация об изменениях к настоящему стандарту публикуется в ежегодно издаваемом информационном указателе «Национальные стандарты», а текст изменений и поправок — в ежемесячно издаваемом информационном указателе «Национальные стандарты». В случае пересмотра (замены) или отмены настоящего стандарта соответствующее уведомление будет опубликовано в ежемесячно издаваемом информационном указателе «Национальные стандарты». Соответствующая информация, уведомление и тексты размещаются также в информационной системе общего пользования — на официальном сайте Федерального агентства по техническому регулированию и метрологии в сети Интернет

© Стандартинформ, 2013

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 . . . 1
3 Термины, определения и сокращения . . . 1
4 Кодирование . . . 3
4.1 Общие положения . . . 3
4.2 Требования к синтаксису . . . 3
4.3 Двоичные объекты . . . 3
4.4 Элементы . . . 3
4.5 Атрибуты . . . 5
4.6 CDATA и строки . . . 5
4.7 Перечисленные значения данных . . . 6
4.8 Типы общих данных . . . 6
4.9 Смешанные поля . . . 10
4.10 Маркерный табличный элемент . . . 11
4.11 Значения contentID по «умолчанию» . . . 11
5 Профилирование . . . 12
5.1 Профили . . . 12
5.2 Фрагментация данных профиля в объекты . . . 12
5.3 Схема объединения данных профиля . . . 13
5.4 Атрибуты, необходимые для объединения файлов . . . 13
6 Транспортировка данных EPG . . . 14
6.1 Транспортный механизм . . . 14
6.2 Максимальный размер объекта . . . 14
6.3 Максимальная пропускная способность канала . . . 14
6.4 Параметры MOT . . . 14
6.5 Транспортировка других объектов . . . 16
7 Сигнализация . . . 16
7.1 DAB . . . 16
7.2 DRM . . . 17
Приложение А (справочное) Пример двоичного кодирования . . . 18
Приложение Б (обязательное): Теги элемента . . . 20
Приложение В (обязательное): Теги атрибута . . . 22
Приложение Г (обязательное): Перечисленные типы . . . 24
Приложение Д (обязательное): Профилирование таблиц . . . 25
Библиография . . . 27

НАЦИОНАЛЬНЫЙ СТАНДАРТ РОССИЙСКОЙ ФЕДЕРАЦИИ

СИСТЕМА ЦИФРОВОГО ЗВУКОВОГО РАДИОВЕЩАНИЯ DRM. ЦИФРОВОЕ ЗВУКОВОЕ РАДИОВЕЩАНИЕ DAB

Требования транспортировки и бинарного кодирования для электронного справочника программ (EPG)

Digital radio mondiale (DRM). Digital audio broadcasting (DAB).
Transportation and binary encoding specification for electronic programme guide (EPG)

Дата введения — 2013—04—01

1 Область применения

Настоящий стандарт распространяется на параметры процессов кодирования, профилирования, передачи данных и сигнализации электронного справочника программ (Electronic Programme Guide; EPG) для систем цифрового звукового радиовещания (Digital Audio Broadcasting; DAB) и системы Всемирного цифрового радио (Digital Radio Mondiale; DRM).

Требования настоящего стандарта следует учитывать при разработке, изготовлении и сертификационных испытаниях кодирующих устройств и приемников пользователей систем DAB и DRM.

2 Нормативные ссылки

В настоящем стандарте использованы нормативные ссылки на следующие стандарты:

ГОСТ Р 54462—2011 Система цифрового радиовещания DRM. Требования и параметры.

МСЭ-Р Регламент радиосвязи (ITU-R Radio Regulations)

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 ежегодно издаваемому информационному указателю «Национальные стандарты», который опубликован по состоянию на 1 января текущего года, и по соответствующим ежемесячно издаваемым информационным указателям, опубликованным в текущем году. Если ссылочный стандарт заменен (изменен), то при пользовании настоящим стандартом следует руководствоваться заменяющим (измененным) стандартом. Если ссылочный стандарт отменен без замены, то положение, в котором дана ссылка на него, применяется в части, не затрагивающей эту ссылку.

3 Термины, определения и сокращения

3.1 В настоящем стандарте применены термины по ГОСТ Р 54462, а также следующие термины с соответствующими определениями:

3.1.1 **ансамбль** (ensemble): Переданный сигнал, включающий несколько регулярно и близко расположенных ортогональных несущих; ансамбль содержит службы аудио и службы данных.

3.1.2 **данные, связанные с программой** (Programme Associated Data; PAD): Информация, которая связана с аудиоданными с точки зрения содержания и синхронизации. Поле PAD расположено в конце аудио кадра DAB.

3.1.3 **заголовок MOT** (MOT header): Этот объект MOT содержит информацию о заголовке, которая описывает одно единственное тело MOT.

3.1.4 **идентификатор ансамбля** (Ensemble Identifier; EId): Уникальный 16-разрядный код, выделенный ансамблю и предназначенный для однозначной глобальной идентификации этого ансамбля.

Издание официальное

3.1.5 канал описания службы (Service Description Channel; SDC): Канал мультиплексированного потока данных, который переносит информацию, необходимую для декодирования служб, включенных в мультиплекс.

3.1.6 **объект MOT** (MOT entity): Единственное тело MOT или единственный каталог MOT или единственный заголовок MOT.

3.1.7 **приложение пользователя** (User application): Приложение данных, определенное в отдельном стандарте и загружаемое данными через DAB.

3.1.8 **расширенная часть PAD** (eXtended — Programme Associated Data; X-PAD): Расширенная часть данных, связанных с программой, которая переносится в конце аудио кадра DAB сразу перед CRC, ее длина является переменной.

3.1.9 **служба, сервис, услуга** (service): Составная часть службы (услуги).

3.1.10 **тело MOT** (MOT body): Тело MOT переносит любой вид данных конечной длины. Информация о теле MOT содержится в заголовке MOT.

3.2 В настоящем стандарте применены следующие сокращения:

CA (Conditional Access) — условный доступ;

CDATA (string data) — данные строки;

CRC (Cyclic Redundancy Check) — циклический контроль по четности;

CRID (Content Reference ID) — идентификатор ссылки контента;

CS (classification scheme) — схема классификации;

DAB (Digital Audio Broadcasting) — цифровое звуковое радиовещание;

Deflate — алгоритм сжатия без потерь, использующий комбинацию алгоритма LZ77 и алгоритма Хаффмана;

DRM (Digital Radio Mondiale) — Всемирное цифровое радио;

ECC (Extended Country Code) — расширенный код страны;

EId (Ensemble Identifier) — идентификатор ансамбля;

EPG (Electronic Programme Guide) — электронный справочник программ;

FIG (Fast Information Group) — группа быстрой информации;

GI (Group Information) — информация о группе;

GZIP (GNU Zip) — утилита (программа) компрессии и декомпрессии, использующая алгоритм DEFLATE;

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

LTO (Local Time Offset) — смещение местного времени;

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

MJD (Modified Julian Date) — измененная юлианская дата;

MOT (Multimedia Object Transfer) — передача мультимедийного объекта;

Mux (multiplex) — мультиплекс;

N/A (not available) — нет данных;

PAD (Programme Associated Data) — данные, связанные с программой;

PI (Programme Information) — информация о программе;

PNum (Programme Number) — количество программ;

Rfa (Reserved for future addition) — зарезервировано для будущего дополнения;

Rfu (Reserved for future use) — зарезервировано для будущего использования;

SCIdS (Service Component Identifier within the Service) — идентификатор компонента службы в службе (в рамках службы);

SDC (Service Description Channel) — канал описания службы;

SI Service Information — информация о службе;

SId (Service Identifier) — идентификатор службы;

UA (User application) — приложение пользователя;

URL (Uniform Resource Location) — унифицированный локатор (определитель местонахождения) ресурса;

UTC (Co-ordinated Universal Time) — всемирное координированное время;

XML (eXtensible Markup Language) — расширяемый язык разметки;

X-PAD (eXtended — Programme Associated Data) — расширенная часть PAD.

4 Кодирование

4.1 Общие положения

Раздел содержит описание двоичного кодирования совокупности tag-length-value. В этом случае каждый элемент или атрибут закодированы при использовании уникального значения тега, значения длины (указание на длину данных содержится в пределах этого элемента или атрибута) и фактического значения данных. Это позволяет приемникам пропускать ненужные или неидентифицированные элементы. На рисунке 1 показана схема кодирования совокупности tag-length-value.

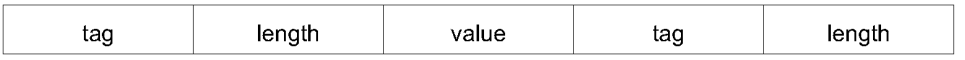

Рисунок 1 — Схема кодирования совокупности tag-length-value

В этих двоичных структурах закодированы элементы XML в соответствии с 4.3 настоящего стандарта. Атрибуты кодированы аналогичным образом в соответствии с 4.5 настоящего стандарта. В этих двоичных структурах иерархическая природа XML EPG обычно сохранена. Различным типам общих данных присваиваются эффективные двоичные кодировки в соответствии с 4.8 настоящего стандарта. Пример кодирования двоичного файла объекта XML приведен в приложении А.

4.2 Требования к синтаксису

Спецификации синтаксиса, представленные в настоящем стандарте, записаны при использовании формы псевдокода, подобного языку «С». В настоящем стандарте типы данных индивидуальных полей определяются при использовании мнемоники, приведенной в таблице 1.

Таблица 1 — Мнемоника типа данных для спецификации синтаксиса

Мнемоника	Описание
uimsbf	Целое число без знака, сначала старший значащий бит

4.3 Двоичные объекты

Структура двоичного объекта, определенная настоящим стандартом, приведена в таблице 2. Каждый двоичный объект переносит единственный элемент высокого уровня в пределах единственного объекта MOT.

Таблица 2 — Структура двоичного объекта

Синтаксис
binary_object() {
top_level_element()

top_level_element() — элемент высокого уровня, определенный в 4.4.2 настоящего стандарта.

4.4 Элементы

4.4.1 Кодирование элемента

Все элементы кодируются в соответствии с таблицей 3.

Таблица 3 — Структура элемента. Кодирование элемента

Синтаксис	Количество битов	Тип
element() {		
element_tag	8	uimsbf
element_length	8	uimsbf
if (element length == 0xFE) {		
extended_element_length	16	uimsbf
}		
if (element_length == 0xFF) {		
extended_element_length	24	uimsbf
}		
for (i=0; i< element_length or extended_element_length; i++) {		
element_data_byte	8	uimsbf
}		
}		

element_tag: Это поле идентифицирует элемент. Если в будущем возникнут потребности в новых элементах, то для них будут использоваться новые значения тега. Возможные значения определены в соответствии с приложением Б. Элементы с тегами, которые не определены, зарезервированы для использования в будущем. Приемники не должны обрабатывать эти теги и связанный с ними контент.

element_length: Это поле указывает на количество байтов данных, содержащихся в этом элементе. Диапазон значений этого поля от 0x00 до 0xFD (от 0 до 253). Если поле element_length принимает значения 0xFE или 0xFF, то длину элемента определяет дополнительное поле extended_element_length.

extended_element_length: Это поле указывает на количество байтов данных, содержащихся в этом элементе.

element_data_byte: Эти байты содержат атрибуты элемента, данные CDATA и дочерние элементы. Они кодируются в следующем порядке:

- атрибуты;
- дочерние элементы;
- контент CDATA.

4.4.2 Элементы высокого уровня

Настоящий стандарт определяет требования к двум элементам высокого уровня: epg и serviceInformation. Элемент высокого уровня переносится в границах двоичного объекта (согласно 4.3 настоящего стандарта), он должен быть единственным элементом в этом объекте (кроме вложенных в него дочерних элементов).

Возможные значения element_tag для элементов высокого уровня определены в таблице 4. Элементы высокого уровня с тегами, которые здесь не определены, зарезервированы для будущего использования. Эти теги и связанный с ними контент не должны обрабатываться приемниками.

Таблица 4 — Теги элементов высокого уровня

Элемент	Тег
Epg	0x02
Serviceinformation	0x03

Так же как и соответствующие элементы, определенные спецификацией EPG XML, элементы высокого уровня опционально могут содержать строковую маркерную таблицу (согласно 4.10 настоящего стандарта) и «по умолчанию» идентификатор contentID (согласно 4.11 настоящего стандарта). Эти элементы должны быть первыми в элементе высокого уровня после атрибутов.

Элемент высокого уровня должен кодироваться таким же образом, как и нормальный элемент (согласно 4.4 настоящего стандарта) за исключением того, что element_data_bytes должен быть закодирован в следующем порядке:

- атрибуты;
- строковая маркерная таблица (если есть);
- значение contentID «по умолчанию» (если есть);
- дочерние элементы;
- контент CDATA.

4.5 Атрибуты

4.5.1 Кодирование атрибута

Кодирование атрибутов выполняется в соответствии с данными, представленными в таблице 5.

Таблица 5 — Структура атрибута. Кодирование атрибута

Синтаксис	Количество битов	Тип
attribute() {		
attribute_tag	8	uimsbf
attribute_length	8	uimsbf
if (attribute_length == 0xFE) {		
extended_attribute_length	16	uimsbf
}		
if (attribute_length == 0xFF) {		
extended_attribute_length	24	uimsbf
}		
for (i=0; i<attribute_length or extended_attribute_length; i++) {		
attribute_data_byte	8	uimsbf
}		
}		

attribute_tag: Это поле однозначно определяет атрибут в родительском элементе. Возможные значения определены в соответствии с приложением В. Атрибуты с тегами, которые не определены здесь, зарезервированы для будущего использования и не должны обрабатываться приемниками.

attribute_length: Это поле указывает на количество байтов данных, содержащихся в этом атрибуте. Диапазон значений от 0x00 до 0xFD (от 0 до 253). Если в поле записано значение 0xFE или 0xFF, тогда длину атрибута определит дополнительное поле extended_attribute_length.

extended_attribute_length: Это поле указывает на количество байтов данных, содержавшихся в этом атрибуте.

attribute_data_byte: Эти байты содержат строки (согласно 4.6.1 настоящего стандарта), или перечисленное значение данных (согласно 4.7 настоящего стандарта), или тип общих данных (согласно 4.8 настоящего стандарта).

Примечание — Любые ссылки объекта должны быть расширены.

4.5.2 Атрибуты «по умолчанию».

В тех случаях, когда атрибут имеет значение «по умолчанию», он не должен кодироваться, так как приемник должен всегда автоматически использовать значение «по умолчанию».

4.6 CDATA и строки

4.6.1 Кодирование

Все CDATA или текстовые строки, кроме текстовых атрибутов, должны быть закодированы в соответствии с таблицей 6.

Таблица 6 — Структура элемента CDATA

Синтаксис	Количество битов	Тип
cdata () {		
cdata_tag	8	uimsbf
cdata_length	8	uimsbf
If (cdata_length == 0xFE) {		
extended_cdata_length	16	uimsbf
}		
If (cdata_length == 0xFF) {		
extended_cdata_length	24	uimsbf
}		
for (i=0; i<cdata_length or extended_cdata_length; i++) {		
cdata_data_byte	8	uimsbf
}		
}		

cdata_tag: В поле всегда должно быть записано 0x01.

cdata_length: Это поле указывает на количество байтов данных, содержащихся в этой строке. Диапазон этих значений от 0x00 до 0xFD (от 0 до 253). Если в поле записано значение 0xFE или 0xFF, тогда дополнительное поле extended_cdata_length определит атрибут длины.

extended_cdata_length: Поле указывает на количество байтов данных, содержащихся в этой строке.

cdata_data_byte: Поле содержит символы для элемента CDATA.

Примечания:

1 Атрибуты с текстовыми значениями не должны быть закодированы в этом формате и вместо этого закодированы как атрибут (согласно 4.5 настоящего стандарта) с attribute_data_bytes.

2 Любые ссылки объекта должны быть расширены.

4.6.2 Наборы символов

Все строки CDATA и другие строки должны использовать кодирование UTF-8 в соответствии с ISO/IEC [1]. Символы от 0xE000 до 0xF8FF не должны включаться в закодированные строки двоичных файлов.

4.7 Перечисленные значения данных

Те атрибуты, которые относятся к типу перечисленных, закодированы единственным байтом, значение этого байта является определенным для этого отдельного атрибута в соответствии с таблицей Г.1 приложения Г.

4.8 Типы общих данных

4.8.1 Введение

Типы общих данных определены спецификацией EPG в соответствии с ETSI [2]. В этом подразделе определены типы общих данных, предназначенные для использования, идентификации и специфического кодирования.

4.8.2 Кодирование

Все элементы, определенные как timePointType, закодированы в соответствии с рисунками 2 и 3.

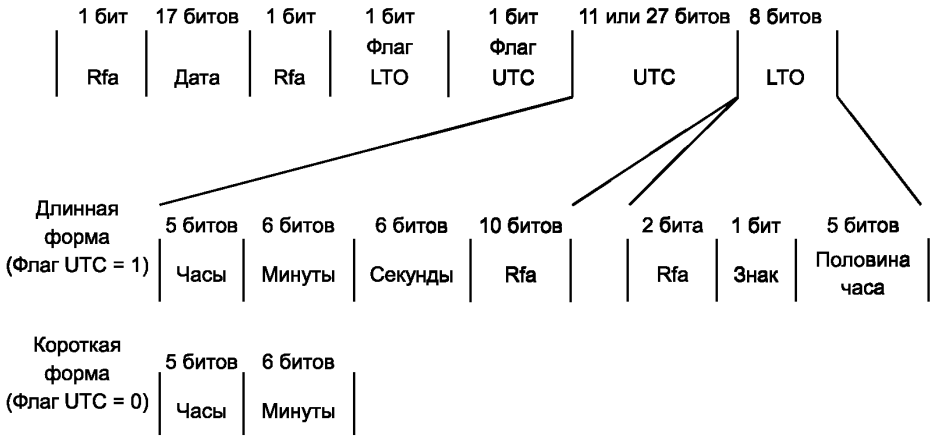

Рисунок 2 — Кодирование даты и времени (флаг LTO==1)



Рисунок 3 — Кодирование даты и времени (флаг LTO==0)

Rfa: Это 1-разрядное поле должно быть зарезервировано для будущих дополнений и до их определения должно быть установлено на 0.

Примечание — Приемники этот бит должны игнорировать.

Дата: Это 17-разрядное двоичное число без знака должно определить текущую дату согласно MJD в соответствии со стратегией кодирования ETSI [3]. Это число ежедневно постепенно увеличивается в 0000 UTC в диапазоне значений от 0 до 999. Например, значению MJD 50000 соответствует 10 октября 1995 года.

Rfa: Это 1-разрядное поле должно быть зарезервировано для будущих дополнений и до их определения должно быть установлено на 0.

Примечание — Приемники должны игнорировать этот бит.

Флаг LTO: Значение этого 1-разрядного поля указывает на представление поля LTO:

- 0: LTO не представлено, время определяется как UTC;
- 1: LTO представлено, местное время определяется как UTC плюс LTO.

Флаг UTC: Значение этого 1-разрядного поля указывает, какую форму использует UTC:

- 0: Краткая форма UTC;
- 1: Подробная форма UTC.

UTC: В зависимости от состояния флага UTC доступны две формы, определенные следующим образом:

- краткая форма: это 11-разрядное поле содержит два субполя, кодированные как двоичные числа без знака. Первое субполе — 5-разрядное (часовое), определяет часы. Второе субполе — 6-разрядное (минутное), определяет минуты;

- подробная форма: в дополнение к часовым и минутным субполям, определенным в краткой форме, это 27-разрядное поле должно содержать одно 6-разрядное субполе, которое должно быть закодировано как двоичное число без знака. Это поле должно определять секунды. Следующие 10 битов должны быть зарезервированы для будущих дополнений, и до их определения биты должны быть установлены на 0.

LTO: Это 8-разрядное поле должно дать смещение местного времени для данного времени. Это поле присутствует, если флаг LTO установлен в 1. Первые два бита зарезервированы для будущих дополнений, они должны быть обнулены до определения дополнений и должны быть проигнорированы приемниками. Следующий бит дает следующие значения LTO:

- 0: Положительное смещение;
- 1: Отрицательное смещение.

Заключительные 5 битов определяют смещение в получасах в диапазоне от минус 12 часов до плюс 12 часов.

Например, у программы вещания в 05:00 в Великобритании в течение летнего времени был бы UTC 04:00 и LTO плюс 1 час.

4.8.3 Продолжительность

Все атрибуты, определенные как durationType, закодированы как 16-разрядное целое число без знака, представляющее продолжительность интервалов времени в секундах от 0 до 65 535 (только для интервалов продолжительностью более 18 часов).

4.8.4 Идентификаторы ссылок контента CRID

Все атрибуты, определенные как CRIDType, закодированы как строковый атрибут (согласно 4.5 настоящего стандарта).

4.8.5 Короткие идентификаторы ссылок контента

Все атрибуты, определенные как shortCRIDType, закодированы как 24-разрядное целое число без знака.

4.8.6 Жанры

Все элементы, которые определены как genreType, должны быть закодированы следующим образом. Кодируется только атрибут href (схема классификации (CS) и уровни) вместе с типом атрибута (опционально). Имя и элементы определения не кодируются. Атрибут href элемента жанра должен быть закодирован в соответствии с рисунком 4.

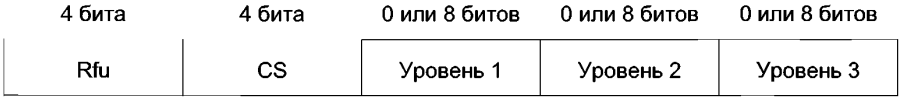

Рисунок 4 — Кодирование атрибута элемента жанра href

Rfu: Это 4-разрядное поле должно быть зарезервировано для будущего использования. Четыре бита этого поля должны быть обнулены.

CS: Это 4-разрядное поле должно указывать применяемую схему классификации (CS), например, 1 из «1.2.3.4», следующим образом:

- 0: CS не определен. Жанры с этим CS должны быть проигнорированы.
- 1: CS намерения;
- 2: CS формата;
- 3: CS контента;
- 4: CS целевой аудитории;
- 5: CS происхождения (производства);
- 6: CS предупреждения контента (Content alert);
- 7: CS типа медиа;
- 8: CS атмосферы (окружающей среды, обстановки);
- 9—15: схемы классификации не определены. Жанры с этими CS должны быть проигнорированы.

Семантика полей Уровень 1, Уровень 2, Уровень 3 должна быть в соответствии с ETSI [4] (4.7.5).

4.8.7 contentID

4.8.7.1 Для случая EPG DAB все элементы, определенные как contentIDType, кодируются в соответствии с рисунками 5, 6, 7.

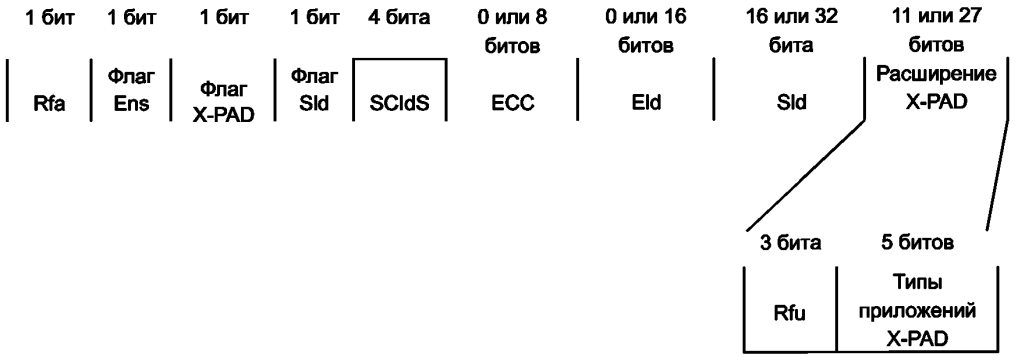

Рисунок 5 — Кодирование contentID для DAB



Рисунок 6 — Пример кодирования contentID (флаг Ens == 1 и флаг X-PAD == 0)

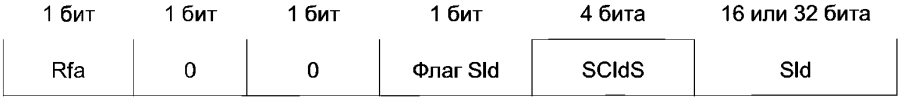

Рисунок 7 — Пример кодирования contentID (флаг Ens == 0 и флаг X-PAD == 0)

Rfa: Это 1-разрядное поле должно быть зарезервировано для будущих дополнений. При кодировании в настоящее время этот бит должен быть обнулен.

Примечание — Приемники этот бит должны игнорировать.

Флаг Ens: Этот 1-разрядный флаг должен указать о наличии или отсутствии ECC и Eld в contentID следующим образом:

- 0: ECC и Eld не присутствуют. Служба, на которую ссылаются в contentID, передана в том же самом ансамбле как эта служба EPG;
- 1: ECC и Eld присутствуют.

Флаг X-PAD: Этот 1-разрядный флаг указывает, переносят или не переносят адресуемый компонент в канале X-PAD следующим образом:

- 0: адресуемый компонент в канале X-PAD не переносят, расширение X-PAD не присутствует;
- 1: адресуемый компонент переносят в канале X-PAD, расширение X-PAD присутствует.

Флаг Sld: Этот 1-разрядный флаг указывает, как закодировано поле Sld следующим образом:

- 0: Sld закодирован как 16-разрядный идентификатор службы (аудио служба);
- 1: Sld закодирован как 32-разрядный идентификатор службы (информационная служба).

SCIdS: Это 4-разрядное поле определяет ID компонента службы в рамках службы (SCIdS).

ECC: Это 8-разрядное поле (опционально) определяет код ECC ансамбля, на котором выполняется служба вещания. Поле ECC присутствует, если флаг Ens установлен в 1.

Eld: Это опциональное 16-разрядное поле определяет идентификатор ансамбля (Eld), на котором выполняется служба вещания. Поле Eld присутствует, если флаг Ens установлен в 1.

Sld: Это 16-разрядное или 32-разрядное поле (разрядность определяется в соответствии с флагом Sld) определяет идентификатор службы.

Расширение X-PAD: Это поле данных расширения X-PAD (опционально) присутствует, если флаг X-PAD установлен в 1.

Rfu: Это 3-разрядное поле должно быть зарезервировано для будущего использования поля типов приложений (X-PAD App Type) (обозначенного флагом XPAD). Три бита этого поля должны быть обнулены.

Примечание — Приемники должны проверить наличие в этих битах нулей, для того чтобы определить статус поля X-PAD App Type. Если какой-либо из битов поля Rfu не установлен в нуль, тогда приемники не должны декодировать флаг X-PAD App Type.

Типы приложений X-PAD: Это 5-разрядное поле (обозначенное как флаг XPAD) определяет первый X-PAD App Type (используемое только для приложений данных X-PAD).

4.8.7.2 Для случая EPG DRM все элементы, определенные как contentIDType, закодированы в соответствии с рисунком 8.

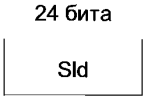

Рисунок 8 — Кодирование contentiD для DRM

SId: Это 24-разрядное поле определяет идентификатор службы.

4.8.8 Идентификаторы ансамбля ensembleID

4.8.8.1 Для случая EPG DAB все элементы, определенные как ensembleIDType, закодированы в соответствии с рисунком 9.

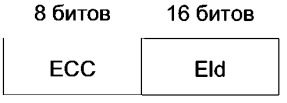

Рисунок 9 — Кодирование ensembleID для DAB

ECC: Это 8-разрядное поле определяет расширенный код страны (ECC) ансамбля.

EId: Это 16-разрядное поле определяет идентификатор ансамбля (EId).

4.8.8.2 Для случая EPG DRM все элементы, определенные как ensembleIDType, закодированы в соответствии с рисунком 10.

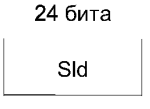

Рисунок 10 — Кодирование ensembleID для DRM

SId: Это 24-разрядное поле определяет идентификатор службы.

4.8.9 triggerType/Pnum

Детализированные параметры кодирования элементов поля triggerType (4 байта) должны быть в соответствии с ETSI [2].

4.8.10 URL

Все элементы, определенные как urlType, должны быть закодированы как строки (согласно 4.6 настоящего стандарта).

4.8.11 Тип MIME

Все элементы, определенные как mimeType, должны быть закодированы как строки (согласно 4.6 настоящего стандарта).

4.9 Смешанные поля

4.9.1 Все атрибуты, определенные как xml:lang, должны быть закодированы как атрибут строки (согласно 4.5 настоящего стандарта).

4.9.2 Поле index, используемое в <memberOf> , кодируется как 16-разрядное целое число без знака.

4.9.3 Поле version, используемое в <programme>, <programmeEvent>, <serviceinformation>, <ensemble>, <service>, <programmeGroups>, <programmeGroup> and <schedule>, кодируется как 16-разрядное целое число без знака.

4.9.4 Поле «скорость передачи» (bitrate), используемое в <service> и <programme>, кодируется как 16-разрядное целое число без знака, которое при умножении на 0,1 дает скорость передачи, близкую к ожидаемой средней скорости передачи в кбит/с.

4.9.5 Поле кГц, используемое в <frequency>, кодируется как 24-разрядное целое число без знака, определяющее частоту в кГц.

4.9.6 Поле numOfItems используется в <programmeGroup>. Кодируется как 16-разрядное целое число без знака.

4.9.7 Поле «ширина и высота» (width and height) используется в <multimedia>. Кодируется как 16-разрядное целое число без знака.

4.10 Маркерный табличный элемент

4.10.1 Маркерный табличный элемент не определен спецификацией XML. Часто повторяющиеся строки в символьных данных EPG («маркеры») могут быть закодированы при использовании таблицы маркеров. Таблица может содержать не более 16 маркеров. Эта таблица определяет теги (их байты могут быть идентифицированы в символьном потоке данных) и эквивалентные им строки. Когда декодер находит тег маркера в символьном потоке, он должен заменить тег эквивалентной строкой. В том случае, если маркерный табличный элемент встречается в двух верхних уровнях элементов (epg и serviceInformation), он должен встречаться перед любыми другими элементами. Этот элемент применяется ко всем символьным данным в пределах родительского элемента высокого уровня (т. е. epg или serviceInformation) и всех дочерних элементах родительского элемента. Этот элемент должен быть закодирован, как определено в 4.4 настоящего стандарта, со следующими условиями:

- element_tag: Значение всегда должно быть 0x04;
- element_data_byte: Эти байты содержат последовательность одного или более маркеров (согласно 4.10.2 настоящего стандарта).

4.10.2 Маркеры

Записи в маркерной таблице должны быть закодированы как уникальный тег и связанная с ним строка. Маркеры строк не должны включать ссылки на другие маркеры. Структура маркера должна быть в соответствии с таблицей 7.

Таблица 7 — Структура маркера

Синтаксис	Количество битов	Тип
token() {		
token_tag	8	uimsbf
token_length	8	uimsbf
for (i=0; i<token_ length; i++) {		
token_data_byte	8	uimsbf
}		
}		

token_tag: Этот байт идентифицирует маркер. Есть 16 возможных значений тега (непечатаемые символы): 0x01, 0x02, 0x03, 0x04, 0x05, 0x06, 0x07, 0x08, 0x0B, 0x0C, 0x0E, 0x0F, 0x10, 0x11, 0x12, 0x13.

Примечание — За исключением значений 0x00 (нуль), 0x09 (вкладка), 0x0A (перевод строки) и 0x0D (возврат каретки). Каждый тег может встречаться в таблице маркеров не более одного раза.

token_length: Это поле указывает на количество байтов данных в маркерной строке. Диапазон допустимых значений от 0x00 до 0xFF (от 0 до 255).

token_data_byte: Маркерная строка.

4.11 Значения contentID по «умолчанию»

Этот элемент не определен в спецификации XML. Он может встретиться только в пределах элемента высокого уровня (epg), и если он встретился, то должен появляться после таблицы маркеров строкой (если она присутствует) и перед любыми другими дочерними элементами. Этот элемент

применяется ко всем элементам, расположенным в пределах родительского элемента высокого уровня (т. е. epg) и перед всеми дочерними элементами родительского элемента.

Правила и параметры кодирования элемента contentID «по умолчанию» должны быть в соответствии с ETSI [4] (4.10).

5 Профилирование

5.1 Профили

Для каждого из трех типов данных EPG (информация о программе, информация о группе и информация о службе) предусматривается два профиля Базовый и Усовершенствованный.

5.1.1 Базовый профиль

Таргет-приемники для профиля Базовый могут быть простыми и встроенными. Эти приемники обычно имеют доступную память для кода декодера EPG и для хранения данных в размере 25 кбайт. С целью максимизации доступной пропускной способности вещания и емкости хранения используется простой двоичный механизм кодирования в соответствии с разделом 4 настоящего стандарта.

Для профиля Базовый не разрешается компрессия при использовании GZIP (алгоритм DEFLATE) любых объектов этого профиля или каталога передаваемых мультимедийных объектов карусели, в которой выполняется вещание. Применение атрибутов и элементов спецификации XML, которые могут использоваться для профиля Базовый, ограничено. Список разрешенных атрибутов и элементов в соответствии с приложением Д.

5.1.2 Профиль Усовершенствованный

Любые атрибуты или элементы спецификации EPG XML, находящиеся вне профиля Базовый, передаются в рамках профиля Усовершенствованный. Кроме того, этот профиль предусматривает применение GZIP (алгоритм DEFLATE) для сжатия закодированных двоичных данных (опционально).

5.2 Фрагментация данных профиля в объекты

Данные EPG фрагментированы в объекты, базирующиеся на типе данных и профиле. EPG может описать службы в тех случаях, когда EPG не передается как служба EPG в том же самом ансамбле/канале. Примеры фрагментации данных приведены в ETSI [4] (приложение В).

Необходимо отметить:

- данные профиля Усовершенствованный нужно переносить как объекты профиля Усовершенствованный в канале информации профиля Базовый;

- данные профилей Базовый и Усовершенствованный для конкретной службы должны содержаться в единственной карусели. Соответствующий декодер должен объединить наборы информации от различных профилей для того, чтобы сформировать совместимый EPG. Следовательно, служба EPG профиля Усовершенствованный должна также быть связана со службой профиля Базовый. Декодеры служб профиля Усовершенствованный должны быть способны декодировать службы профиля Базовый. Для получения дополнительной информации о схеме объединения данных профилей необходимо руководствоваться 5.3 настоящего стандарта.

5.2.1 Информация о службе

Информация о службе профиля Базовый для всех служб единственного ансамбля/канала, описанная этой службой EPG, должна содержаться в одном объекте.

Дополнительная информация о службе должна переноситься в дополнительных объектах информации о службе профиля Усовершенствованный и может быть дополнительно сжата с GZIP.

5.2.2 Информация о программе (PI)

Информация о программе в профиле Базовый должна представляться одним объектом для единственной службы на интервале одних суток. Общее количество объектов информации о программе профиля Базовый должно быть равно количеству описанных служб, умноженных на количество суток, которые поддерживаются EPG.

Одни сутки определяются как время, в течение которого все программы передают одну или более служб (которые тарифицируются), и которые начинаются в интервале времени между 0:00:00 (местного времени) и 23:59:59 (местного времени) конкретной даты.

Все программы в пределах объекта PI профиля Базовый должны быть рассортированы хронологически, начиная от времени старта.

Примечание — Программа содержит множество факторов времени. Эти индивидуальные факторы времени должны быть хронологически рассортированы. Индивидуальная программа должна быть рассортирована в рамках базовых объектов информации о программе по его первому фактору времени — времени старта.

Информация с расширенными подробностями о программе (переносится в профиле Усовершенствованный) выполняется группированием ансамблем/каналом за время продолжительности службы или за сутки. Приемники должны быть способны к поддержке всех методов группирования данных PI профиля Усовершенствованный.

Примечание — Объекты информации о программе не должны удаляться из канала вещания до истечения тарифицированного времени окончания всех программ, содержащихся в пределах объектов, содержавшихся в данных информации о программе.

5.2.3 Информация о группе (GI)

Информация о группе для всех служб профиля Базовый о единственном ансамбле/канале, описанная этим EPG, должна содержаться в одном объекте.

Если служба EPG содержит данные о двух и более ансамблях/каналах, то на каждый ансамбль/канал должен создаваться объект информации GI профиля Базовый.

Дополнительная информация GI для какой-либо из служб должна переноситься в дополнительных объектах GI профиля Усовершенствованный и может быть дополнительно компрессирована с применением GZIP.

5.3 Схема объединения данных профиля

Данные профилей Базовый и Усовершенствованный получены из основного XML-файла, который соответствует схемам EPG, определенным в ETSI [2]. Эти файлы являются хорошо согласованными XML-документами, которые закодированы отдельно при использовании двоичного формата кодирования, упомянутого в разделе 4 настоящего стандарта.

Формат XML профиля Базовый должен соответствовать форме основного XML-документа (файла), но с включенными элементами и атрибутами профиля Базовый. Каждый элемент должен иметь вложение и порядок, соответствующие основному документу.

Данные профиля Усовершенствованный должны соответствовать форме исходного XML со следующими перемещенными данными:

- атрибуты и элементы, которые включены в профиль Базовый;
- text/CDATA, которые включены в профиль Базовый;
- элементы, которые стали пустыми в результате этих перемещений.

Детализированные правила применения данных профилей Базовый и Усовершенствованный должны быть в соответствии с ETSI [4] (5.3).

5.4 Атрибуты, необходимые для объединения файлов

Для обеспечения возможности приемнику профиля Усовершенствованный выполнять объединение файлов профилей Базовый и Усовершенствованный некоторые атрибуты и теги в этих файлах должны быть дублированы. В обоих файлах должны присутствовать следующие элементы и атрибуты:

- атрибуты объединенной информации о службе в соответствии с таблицей 8.

Таблица 8 — Атрибуты объединенной информации о службе

Элемент	Атрибут
Serviceinformation	версия
Serviceinformation.ensemble	Id
Serviceinformation. ensemble.service.serviceID	Id

- атрибуты объединенной информации о программе в соответствии с таблицей 9.

Таблица 9 — Атрибуты объединенной информации о программе

Элемент	Атрибут
Epg.C	версия
Programme	shortId

- атрибуты объединенной информации о группе должны быть в соответствии с таблицей 10.

Таблица 10 — Атрибуты информации о группе

Элемент	Атрибут
Epg.Programmegroups	версия
Serviceinformation	shortId

Приемник не должен пытаться объединять данные, если ID и версии данных в профилях Базовый и Усовершенствованный не совпадают. В этом случае должны использоваться только данные профиля Базовый.

6 Транспортировка данных EPG

6.1 Транспортный механизм

В качестве метода передачи данных EPG используется MOT в режиме каталога (ETSI [5]). Отображение данных EPG на объекты MOT описано в 5.2 настоящего стандарта.

Каталог MOT не должен компрессироваться, информация о заголовке в пределах каталога MOT должна сортироваться в порядке возрастания ContentName, сообщенного параметром расширения каталога MOT SortedHeaderInformation в соответствии с детализацией спецификации MOT (ETSI [5]).

6.2 Максимальный размер объекта

Размер каждого объекта MOT профиля Базовый не должен превышать 16 кбайт (16 384 байт) и размер каталога MOT не должен превышать 8 кбайт (8 192 байт). Размер каждого из объектов профиля Усовершенствованный не ограничен.

Если размер объекта каталога MOT превысит 8 кбайт, то индивидуальные службы должны пересылаться в альтернативную карусель с размером объекта каталога не более максимального. Если размер любого объекта EPG профиля Базовый превышает 16 кбайт (16 384 байт), то уровень детализации в пределах объекта должен быть сокращен до размера объекта, не превышающего максимальное значение.

Данные профилей Базовый и Усовершенствованный для определенной службы должны содержаться в пределах единственной карусели.

6.3 Максимальная пропускная способность канала

Для любых каруселей, содержащих объекты EPG, скорость передачи данных для MOT, транспортируемой или в пакетном режиме, или в формате PAD, не должна превышать 64 кбит/с. Для режима передачи пакета полная скорость подканала, включая EPG, ограничена величиной 128 кбит/с.

6.4 Параметры MOT

6.4.1 Использование параметров MOT

Параметры MOT используются для предоставления информации об отслеживании (трекинге) приемников с целью эффективной загрузки и кэширования данных.

Параметры MOT, которые должны быть применены к индивидуальным объектам MOT, переносятся с информацией в заголовке MOT, каждая запись в каталоге входит в каталог MOT. Сводка параметров MOT для индивидуальных объектов, которые применяются для спецификации EPG, дана в таблице 11 и определена подробно ниже в следующих подразделах.

Функция MOT кэширования опциональна и для провайдера UA и для приемника (ETSI [5]).

Если используются параметры MOT ProfileSubset, CAInfo, ContentName и/или UniqueBodyVersion, то эти параметры должны быть рассортированы в этом порядке и помещены в начале списка параметров MOT в заголовке MOT.

Примечание — В пределах контекста, определенного спецификой профиля, могут быть заданы другие параметры. Однако параметры, обработка которых не предусмотрена приемником данного профиля, будут проигнорированы.

Перечень параметров MOT, используемых для идентификации контента индивидуальных одиночных объектов, представлен в таблице 11.

Таблица 11 — Перечень параметров MOT, используемых для идентификации контента индивидуальных одиночных объектов

Наименование параметра	Величина идентификатора	Определено в документе	Обязательность применения для провайдера UA	Обязательность применения для приемника
ProfileSubset	0x21	ETSI [5]	Не обязателен (но параметр должен использоваться для всех объектов профиля Усовершенствованный)	Обязателен
ContentName	0x0C	ETSI [5]	Обязателен	Обязателен
CompressionType	0x11	ETSI [5]	Не обязателен (но параметр должен использоваться для всех объектов, компрессированных по уровню MOTtransport)	Обязателен для приемников профиля Усовершенствованный (объекты профиля Базовый не должны компрессироваться)
CAInfo	0x23	ETSI [5]	Не обязателен (но параметр должен использоваться для всех параметров компрессированных по уровню MOT)	Обязателен (приемники, не поддерживающие CA, должны отказаться от зашифрованных объектов)
ScopeStart	0x25	ETSI [4]	ETSI [4] (6.4.6)	Не обязателен
ScopeEnd	0x26	ETSI [4]	ETSI [4] (6.4.7)	Не обязателен
ScopeID	0x27	ETSI [4]	Обязателен	Не обязателен

6.4.2 Ядро заголовка MOT

Параметр ContentType указывает на основную категорию контента тел MOT. Параметр ContentSubType указывает на точный тип контента тел MOT в зависимости от значения поля в соответствии с ETSI [5].

Параметры ContentType/ContentSubType идентифицируют типы данных в зависимости от того, являются ли данные расписанием EPG, службой или информацией группы. Величины разрешенных значений для ContentType и ContentSubType для конкретных данных EPG перечислены в таблице 12. ContentType всех конкретных данных приложения EPG имеет значение «7».

Значения ContentSubType уникальны только в пределах приложения (EPG). Другие приложения могут использовать те же самые значения для других типов контента.

Таблица 12 — Величины разрешенных значений для ContentType и ContentSubType для конкретных данных EPG

Значения ContentType/ContentSubType	Описание контента
7/0	Объект содержит служебную информацию
7/1	Объект содержит информацию о программах
7/2	Объект содержит информацию о группе

6.4.3 Параметр ProfileSubset

В тех случаях, когда карусель содержит больше объектов, чем необходимо для одного профиля, декодером MOT может быть применена дополнительная обработка, если он «знает», с каким профилем данный объект используется. Параметр ProfileSubset идентифицирует профиль, для которого объект релевантен (соответствует запрашиваемому). Определены возможные значения ProfileSubset для ID профилей, представленных в ETSI [4] (таблица 13). Детализированные правила применения параметра ProfileSubset должны быть в соответствии со спецификацией MOT (ETSI [5] и ETSI [4] (6.4.2).

6.4.4 Параметр ContentName

Этот обязательный параметр MOT однозначно определяет объект в рамках карусели MOT и в рамках службы EPG. Параметр ContentName используется в соответствии со спецификацией MOT (EN [5]).

6.4.5 Параметр CompressionType

Параметр CompressionType используется в соответствии со спецификацией MOT (ETSI [5]).

Объекты, содержащие данные профиля Базовый, не должны компрессироваться.

Объекты, содержащие информацию профиля Усовершенствованный, могут включать необработанный текст. Вследствие этого к закодированным объектам двоичного файла может быть применено компрессирование GZIP. Эти данные будут использоваться только приемниками профиля Усовершенствованный, у которых должна быть возможность распаковывать эти данные. Максимальный размер окна, поддерживаемый декодером EPG для GZIP, должен быть не более 32 кбайт.

6.4.6 Параметр CAInfo

Этот параметр используется, если условный доступ на уровне MOT применен к данным MOT. Параметр CAInfo используется в соответствии со спецификацией MOT (ETSI [5]). В случае присутствия этого параметра приемник, не совместимый с CA, должен игнорировать этот объект MOT.

6.4.7 Параметр ScopeStart

Параметр ScopeStart должен применяться для индикации тарифицированной даты вещания, времени старта вещания первой программы (местное время), которые соответствуют данным, находящимся в объекте информации о программе (PI) EPG.

Этот параметр используется только для объектов информации о программе EPG, он не должен использоваться для объектов информации о службе или для объектов информации о группе.

Для объектов информации о программе параметр ScopeStart является обязательным. Он должен кодироваться в соответствии с 4.8.2 настоящего стандарта со следующим ограничением: должна использоваться только краткая форма UTC (LTO допускается опционально). Время старта должно округляться в меньшую сторону до ближайшей минуты.

6.4.8 Параметр ScopeEnd

Этот параметр используется для индикации тарифицированной даты и времени окончания последней программы (местное время), которые соответствуют данным, находящимся в объекте информации о программе (PI) EPG.

Параметр ScopeEnd не должен использоваться для объектов информации о службе или для объектов информации о группе.

Для объектов информации о программе этот параметр является обязательным. Он должен кодироваться в соответствии с 4.8.2 настоящего стандарта со следующим ограничением: должна использоваться только краткая форма UTC с опциональным LTO. Время окончания должно быть округлено в меньшую сторону до ближайшей минуты. Детализированные правила применения параметра ScopeEnd должны быть в соответствии с ETSI [4] (6.4.7).

6.4.9 Параметр ScopeID

Этот параметр указывает на ensembleID ансамбля/канала, для которого объект содержит данные (кодированные в соответствии с 4.8.8 настоящего стандарта).

Правила применения параметра ScopeEnd должны быть в соответствии с ETSI [4] (6.4.8).

6.5 Транспортировка других объектов

Карусель MOT может содержать дополнительные объекты, которые должны транспортироваться в рамках MOT в соответствии с ETSI [5] и при использовании сигнализации, детализированной в рамках этого же стандарта.

Эти дополнительные объекты могут транспортироваться в той же самой карусели MOT, как и данные профиля Базовый при условии, что размер каталога MOT не будет превышать максимального размера, разрешенного для данных профиля Базовый (согласно 6.2 настоящего стандарта). В том случае, если каталог MOT превышает разрешенный размер, эти дополнительные объекты должны транспортироваться в дополнительной карусели. Специфичные параметры EPG (ScopeStart, ScopeEnd и ScopeID) для этих объектов не должны использоваться.

7 Сигнализация

7.1 DAB

7.1.1 Сигнализация типа приложения при использовании приложения EPG в канале данных DAB должна выполняться при помощи FIG0/13, значение UserApplicationType должно быть «EPG» в соответствии с ETSI [6]. Данные профилируются в объекты EPG профилей Базовый и Усовершенствованный в соответствии с разделом 5 настоящего стандарта. Поле данных приложения EPG представляет собой последовательность ProfileID.

Если количество идентификаторов ProfileID больше одного, то список должен быть рассортирован в порядке возрастания, начиная от самой низкой величины ProfileID. Список значений идентификаторов ProfileIDs представлен в ETSI [4] (таблица 13). Значения ProfileID от 0x03 до 0xFF зарезервированы для использования в будущем и не должны обрабатываться приемниками.

7.1.2 Сигнализация времени вещания предоставлением данных о времени вещания в FIG 0/10 и смещения местного времени в FIG 0/9 является обязательным требованием приложения пользователя. Детализированные требования к этим параметрам должны быть в соответствии с ETSI [3].

7.1.3 Сигнализация о количестве программ в FIG 0/16 необходима для использования атрибута программы «trigger» в рамках спецификации EPG DAB. Детализированная информация об этих параметрах должна быть в соответствии с ETSI [3].

7.2 DRM

7.2.1 Сигнализация объекта данных типа 5 (приложение данных)

Использование приложения EPG в канале DRM должно быть обозначено применением объекта данных типа 5 (в соответствии с ETSI [7]) со значением домена приложения «DAB» (в соответствии с ETSI [8]) и с UserApplicationType, принимающим значение «EPG» (в соответствии с ETSI [6]).

Параметры сигнализация данных профиля должны быть в соответствии с описанием для DAB согласно 7.1.1 настоящего стандарта.

7.2.2 Сигнализация времени вещания

Корректное предоставление справочного времени вещания в пределах объекта данных SDC типа 8 является обязательным требованием приложения пользователя. Детализация параметров сигнализации должна быть в соответствии с ETSI [7].

Приложение А
(справочное)

Пример двоичного кодирования

```
<epg xmlns:epg=gost.e-ecolog.ru org/schemas/epg xmlns:xsi="gost.e-ecolog.ru
xsi:schemaLocation=gost.e-ecolog.ru org/schemas/epg epgScheule_13.xsd system="DAB">
    <schedule version="1" creationTime="2001-02-28T00:00:00" originator="BBC">
        <scope startTime="2003-12-18T17:00:00" stopTime="2003-12-18T18:00:00">
            <serviceScope id="e1.ce15.c224.0"/>
        </scope>
        <programme shortId="16442449">
            <epg:mediumName>PM</epg:mediumName>
            <epg:location>
                <epg:time time="2003-12-18T17:00:00" duration="PT1H0M0S"/>
                <epg:bearer id="e1.ce15.c224.0"/>
            </epg:location>
        </programme>
    </schedule>
</epg>
```

В таблице А.1 показаны байты двоичного кодирования объекта.

Таблица А.1 — Пример двоичного кодирования

Байты	Описание
02	**<epg>**
3F	Длина = 63 байта
21	**<schedule>**
3D	Длина = 61 байт
24	**<scope>**
16	Длина = 22 байта
80	**<startTime>**
04	Длина = 4 байта
33 BF C4 40	2003-12-18T17:00:00 (краткая форма записи местного времени)
81	**<stopTime>**
04	Длина = 4 байта
33 BF C4 80	2003-12-18T18:00:00 (краткая форма записи местного времени)
25	**<serviceScope>**
08	Длина = 8 байтов
80	ID атрибута
06	Длина = 6 байтов
40 E1 CE 15 C2 24	E1.ce15.c224.0
1C	**<programme>**
23	Длина = 35 байтов
81	Shortid атрибут
03	Длина = 3 байта
FA E4 51	16442449
11	**<mediumName>**
04	Длина = 4 байта
01	CDATA
02	Длина = 2 байта
50 4D	PM
19	**<location>**
16	Длина = 22 байта
2C	**<time>**
0A	Длина = 10 байтов
80	атрибут времени
04	Длина = 4 байта
33 BF C4 40	2003-12-18T17:00:00 (краткая форма записи местного времени)
81	атрибут продолжительности

Окончание таблицы А.1

Байты	Описание
02 0E 10 2D 08 80 06 40 E1 CE 15 C2 24	Длина = 2 байта 3600 **<bearer>** Длина = 8 байтов ID атрибута Длина = 6 байтов E1.ce15.c224.0

Приложение Б
(обязательное)

Теги элемента

Все теги элемента, кроме элементов верхнего уровня и других специальных элементов, находятся в диапазоне значений от 0x10 до 0x7E. Все теги атрибута находятся в диапазоне значений от 0x80 до 0xFF. Тег 0x7F зарезервирован и никогда не будет использоваться.

В таблице Б.1 представлен перечень тегов элементов.

Таблица Б.1 — Теги элементов

Дочерние элементы	Возможные исходные элементы	Тег	Определено в пункте настоящего стандарта
epg	Top-level (верхний уровень)	0x02	4.4.2
serviceInformation	Top-level (верхний уровень)	0x03	4.4.2
tokenTableElement	epg, serviceInformation	0x04	4.10
defaultcontenIDElement	epg	0x05	4.11
shortName	programmeGroup, ensemble, service, programme, programmeEvent	0x10	4.4
mediumName	programmeGroup, ensemble, service, programme, programmeEvent	0x11	4.4
longName	programmeGroup, ensemble, service, programme, programmeEvent	0x12	4.4
mediaDescription	programmeGroup, ensemble, service, programme, programmeEvent	0x13	4.4
genre	programmeGroup, service, programme, programmeEvent	0x14	4.4
CA	ensemble, service, programme, programmeEvent	0x15	4.4
keywords	programmeGroup, ensemble, service, programme, programmeEvent	0x16	4.4
memberOf	programmeGroup, programme, programmeEvent	0x17	4.4
link	programmeGroup, ensemble, service, programme, programmeEvent	0x18	4.4
location	programme, programmeEvent	0x19	4.4
shortDescription	mediaDescription	0x1A	4.4
longDescription	mediaDescription	0x1B	4.4
programme	epg, schedule	0x1C	4.4
programmeGroups	epg	0x20	4.4
schedule	epg	0x21	4.4
alternateSource	epg	0x22	4.4
programmeGroup	programmeGroups	0x23	4.4
scope	schedule	0x24	4.4
serviceScope	scope	0x25	4.4
ensemble	serviceInformation	0x26	4.4
frequency	ensemble	0x27	4.4
service	ensemble	0x28	4.4

Окончание таблицы Б.1

Дочерние элементы	Возможные исходные элементы	Тег	Определено в пункте настоящего стандарта
serviceID	service	0x29	4.4
epgLanguage	service	0x2A	4.4
multimedia	mediaDescription	0x2B	4.4
time	location	0x2C	4.4
bearer	location	0x2D	4.4
programmeEvent	programme	0x2E	4.4
relativeTime	location	0x2F	4.4
simulcast	service	0x30	4.4

Приложение В
(обязательное)

Теги атрибута

Правила кодирования атрибутов определены в 4.5 настоящего стандарта. В таблицах В.1, В.2 и В.3 представлены дополнительные правила кодирования тегов атрибутов epgSchedule, epgSI, epgDataTypes соответственно.

Таблица В.1 — Теги атрибута epgSchedule

Элемент	Атрибут	Тег	Определено в пункте настоящего стандарта
epg	system	0x80	4.7
programmeGroups	version creationTime originator	0x80 0x81 0x82	4.9.3 4.8.2 4.6
programmeGroups	id shortId version type numOfItems	0x80 0x81 0x82 0x83 0x84	4.8.8 4.8.5 4.9.3 4.7 4.9.6
schedule	version creationTime originator	0x80 0x81 0x82	4.9.3 4.8.2 4.6
scope	startTime stopTime	0x80 0x81	4.8.2 4.8.2
serviceScope	id	0x80	4.8.7
alternateSource	protocol type url	0x80 0x81 0x82	4.7 4.7 4.8.10

Таблица В.2 — Теги атрибута epgSI

Элемент	Атрибут	Тег	Определено в пункте настоящего стандарта
serviceInformation	version creationTime originator serviceProvider system	0x80 0x81 0x82 0x83 0x84	4.9.3 4.8.2 4.6 4.6 4.7
Ensemble	id version	0x80 0x81	4.8.8 4.9.3
Frequency	type kHz	0x80 0x81	4.7 4.9.5
Service	version format не используется bitrate	0x80 0x81 0x82 0x83	4.9.3 4.7 N/A 4.9.4
Simulcast	system	0x80	4.7
	id	0x81	4.8.7
serviceID	id type	0x80 0x81	4.8.7 4.7
Примечание — Маркер 0x82 не должен использоваться в качестве служебного элемента из-за обратной совместимости.			

Т а б л и ц а В.3 — Теги атрибутов epgDataTypes

Элемент	Атрибут	Тег	Определено в пункте настоящего стандарта
CA	type	0x80	4.7
keywords	xml:lang	0x80	4.9.1
multimedia	mimeValue xml:lang url type width height	0x80 0x81 0x82 0x83 0x84 0x85	4.8.11 4.9.1 4.8.10 4.7 4.9.7 4.9.7
time	time duration actualTime actualDuration	0x80 0x81 0x82 0x83	4.8.2 4.8.3 4.8.2 4.8.3
relativeTime	time duration actualTime actualDuration	0x80 0x81 0x82 0x83	4.8.2 4.8.3 4.8.2 4.8.3
bearer	id trigger	0x80 0x81	4.8.7 4.8.9
memberOf	id shortId index	0x80 0x81 0x82	4.8.4 4.8.5 4.9.2
epgLanguage	xml:lang	0x80	4.9.1
link	url mimeValue xml:lang description expiryTime	0x80 0x81 0x82 0x83 0x84	4.8.10 4.8.11 4.9.1 4.6 4.8.2
programme	id shortId version recommendation broadcast не используется xml:lang bitrate	0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87	4.8.4 4.8.5 4.9.3 4.7 4.7 N/A 4.9.1 4.9.4
programmeEvent	id shortId version recommendation broadcast	0x80 0x81 0x82 0x83 0x84	4.8.4 4.8.5 4.9.3 4.7 4.7
shortName	xml:lang	0x80	4.9.1
mediumName	xml:lang	0x80	4.9.1
longName	xml:lang	0x80	4.9.1
shortDescription	xml:lang	0x80	4.9.1
longDescription	xml:lang	0x80	4.9.1
genre	href type	0x80 0x81	4.8.6 4.7
П р и м е ч а н и е — Маркер 0x85 не должен использоваться программным элементом из-за обратной совместимости.			

Приложение Г
(обязательное)

Перечисленные типы

В таблице Г.1 представлены значения перечисленного типа. Атрибут «по умолчанию», показанный курсивом в таблице Г.1, всегда имеет значение 0x01 и не должен кодироваться.

Таблица Г.1 — Значения перечисленного типа

Элемент	Атрибут	Значение	Тег
epg	system	*DAB*	0x01
		DRM	0x02
programmeGroup	type	series show programConcept magazine programCompilation otherCollection otherChoice topic	0x02 0x03 0x04 0x05 0x06 0x07 0x08 0x09
alternateSource	protocol	*URL* DAB DRM	0x01 0x02 0x03
alternateSource	type	*identical* more less similar	0x01 0x02 0x03 0x04
Frequency	type	*primary* alternative	0x01 0x02
service	format	*audio* DLS MOTSlideshow MOTBWS	0x01 0x02 0x03 0x04
		TPEG DGPS	0x05 0x06
		proprietary	0x07
serviceID	type	*primary* secondary	0x01 0x02
CA	type	*none* unspecified	0x01 0x02
programme, programmeEvent	broadcast	*on-air* off-air	0x01 0x02
programme, programmeEvent	recommendation	*no* yes	0x01 0x02
multimedia	type	logo_unrestricted logo_mono_square logo_colour_square logo_mono_rectangle logo_mono_rectangle	0x02 0x03 0x04 0x05 0x06
genre	type	*main* secondary other	0x01 0x02 0x03
Примечание — Курсивом показан атрибут «по умолчанию».			

Приложение Д
(обязательное)

Профилирование таблиц

Д.1 Элементы и атрибуты, передаваемые в профиле Базовый

В таблицах Д.1—Д.3 представлены элементы и атрибуты, формирующие профиль Базовый.

Т а б л и ц а Д.1 — Информация о службе (SI) профиля Базовый

Элемент	Атрибут	Ключ применения
serviceInformation	 version system	R O R2
serviceInformation.ensemble		R
	id	R
serviceInformation.ensemble.shortName	 xml:lang	R R2
serviceInformation.ensemble.shortName	 xml:lang	R R2
serviceInformation.ensemble.frequency	 type кHz	R R R
serviceInformation.ensemble.mediaDescription		O
serviceInformation.ensemble.mediaDescription.multimedia	 type mimeValue xml:lang URL width height	O O O R2 R1 O O
serviceInformation.ensemble.service	 format bitrate	R R2 O
serviceInformation.ensemble.service.serviceID	 id type	R R R2
serviceInformation.ensemble.service.simulcast		O
	system	R2
	id	R1
serviceInformation.ensemble.service.shortName	 xml:lang	R R2
serviceInformation.ensemble.service.mediumName	 xml:lang	R R2
serviceInformation.ensemble.service.mediaDescription		O
serviceInformation.ensemble.service.mediaDescription.multimedia	 type mimeValue xml:lang URL width height	O O O R2 R1 O O

П р и м е ч а н и е — Элементы и атрибуты должны применяться в соответствии с ключом, определяющим условия применения:
O: Применение опционально;
R: Применение необходимо;
R1: Применение необходимо, если исходный элемент заполнен;
R2: Применение необходимо, если фактическое значение отличается от значения «по умолчанию».

Таблица Д.2 — Информация о программе (PI) профиля Базовый

Элемент	Атрибут	Ключ применения
epg		R
epg.schedule		R
	version	O
epg.schedule.scope		O
	startTime	R1
	stopTime	R1
epg.schedule.scope.serviceScope		O
	id	R1
epg.schedule.programme		R
	short id	R
	recommendation	R2
	broadcast	R2
	bitrate	R2
epg.schedule.programme.mediumName		R
	xml:lang	R2
epg.schedule.programme.longName		O
	xml:lang	R2
epg.schedule.programme.location		R
epg.schedule.programme.location.time		R
	time	R
	duration	R
epg.schedule.programme. location.bearer		R2
	id	R2
	trigger	O
epg.schedule.programme.mediaDescription		R
epg.schedule.programme.mediaDescription ShortDescription		O
	xml:lang	R2
epg.schedule.programme.genre		O
	href	R1
	type	R2
epg.schedule.programme.memberof		O
	short id	R1
	index	O

Примечание — Элементы и атрибуты должны применяться в соответствии с ключом, определяющим условия применения:
O: Применение опционально;
R: Применение необходимо;
R1: Применение необходимо, если исходный элемент заполнен;
R2: Применение необходимо, если фактическое значение отличается от значения «по умолчанию».

Таблица Д.3 — Информация о группе (GI) профиля Базовый

Элемент	Атрибут	Ключ применения
epg		R
epg.programmeGroups		R
	version	O
epg.programmeGroups.programmeGroup		R
	short id	R
	type	O
	numofItems	O

Окончание таблицы Д.3

Элемент	Атрибут	Ключ применения
epg.programmeGroups.programmeGroup.mediumName		R
	xml:lang	R2
epg.programmeGroups.programmeGroup.longName		O
	xml:lang	R2
epg.programmeGroups.programmeGroup.genre		O
	href	R1
	type	R2
epg.programmeGroups.programmeGroup.memberof		O
	short id	R1
	Index	O

Примечание — Элементы и атрибуты должны применяться в соответствии с ключом, определяющим условия применения:
O: Применение опционально;
R: Применение необходимо;
R1: Применение необходимо, если исходный элемент заполнен;
R2: Применение необходимо, если фактическое значение отличается от значения «по умолчанию».

Д.2 Элементы и атрибуты, передаваемые в профиле Усовершенствованный

Профиль Усовершенствованный может включать любые элементы и атрибуты, предусмотренные спецификацией EPG XML.

Вещатель для конкретной службы EPG может перемещать элементы и атрибуты профиля Базовый в элементы и атрибуты профиля Усовершенствованный.

Библиография

[1] ISO/IEC 10646 Information technology — Universal Multiple-Octet Coded Character Set (UCS)

[2] ETSI TS 102 818 Digital Audio Broadcasting (DAB); XML specification for DAB Electronic Programme Guide (EPG)

[3] ETSI EN 300 401 Radio broadcasting systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers

[4] ETSI TS 102 371 V1.2.1 (2006-02) Technical Specification. Digital Audio Broadcasting (DAB); Digital Radio Mondiale (DRM); Transportation and Binary Encoding Specification for Electronic Programme Guide (EPG)

[5] ETSI EN 301 234 Digital Audio Broadcasting (DAB); Multimedia Object Transfer (MOT) Protocol

[6] ETSI TS 101 756 Digital Audio Broadcasting (DAB); Registered Tables

[7] ETSI ES 201 980 Digital Radio Mondiale (DRM); System specification

[8] ETSI TS 101 968 Digital Radio Mondiale (DRM); Data applications directory

УДК 621.396.97: 681.327.8:006.354 ОКС 33.170 ОКПО 657300

Ключевые слова: цифровое звуковое радиовещание (DAB), Всемирное цифровое радио (DRM), электронный справочник программ (EPG), параметры кодирования данных EPG

Редактор *Е.В. Вахрушева*
Технический редактор *Н.С. Гришанова*
Корректор *М.И. Першина*
Компьютерная верстка *И.А. Налейкиной*

Сдано в набор 06.05.2013. Подписано в печать 24.05.2013. Формат 60 × 84 1/8. Гарнитура Ариал.
Усл. печ. л. 3,72. Уч.-изд. л. 3,40. Тираж 81 экз. Зак. 562.

ФГУП «СТАНДАРТИНФОРМ», 123995 Москва, Гранатный пер., 4.
gost.e-ecolog.ru gost.e-ecolog.ru
Набрано во ФГУП «СТАНДАРТИНФОРМ» на ПЭВМ.
Отпечатано в филиале ФГУП «СТАНДАРТИНФОРМ» — тип. «Московский печатник», 105062 Москва, Лялин пер., 6.

gost.e-ecolog.ru