Виды mp3-тегов:

[ID3v1.0](#v10)

[ID3v1.1](#v11)

[Расширенный ID3v1 (неофициальный формат)](#extd)

[ID3v2.2](#v22)

Фреймы ID3v2.2

[ID3v2.3 (наиболее популярный)](#v23)

Фреймы ID3v2.3

[ID3v2.4](#v24)

Фреймы ID3v2.4

Обозначения

$xx – обозначение байта в шестнадцатеричном виде

%xxxxxxxx – обозначение байта в двоичном виде

Десинхронизация (Unsynchronisation)

Аудиоданные в формате mp3 состоят из mp3-фреймов, каждый из которых содержит 4-байтный заголовок и собственно закодированный звук. Заголовок начинается с 11 бит, установленных в единицы – это маркер, по которому плееры распознают аудиоданные. В ID3v2-тегах может также встретиться последовательность из 11 единиц, из-за этого некоторые плееры могут начать проигрывать содержимое тега. Чтобы такого не происходило, теги целиком или отдельные фреймы можно “десинхронизировать” – заменять каждую последовательность %11111111 %111xxxxx на %11111111 %00000000 %111xxxxx, то есть вставлять нулевой байт после первого байта каждого ложного маркера. Также, чтобы облегчить декодирование, последовательность $FF $00 заменяется на $FF $00 $00 – тогда при декодировании достаточно заменять $FF $00 на $FF. Стандарты ID3v2.2 и v2.3 позволяют десинхронизировать только тег целиком, стандарт ID3v2.4 – тег целиком или отдельные фреймы. Это помечается флагом в заголовке тега или фрейма. Чтобы десинхронизация не затронула байты, в которых хранится размер тега или фрейма, размер хранится в synchsafe байтах – их старший бит всегда равен 0. При их чтении старший бит каждого байта нужно игнорировать. В v2.2 и v2.3 размер всего тега хранится в synchsafe байтах, размер фреймов – в обычных. В v2.4 размер тега и фреймов хранится в synchsafe байтах. В ID3v2 встречаются 4- и 5-байтные synchsafe числа. Пример 4-байтных synchsafe чисел:

|  |  |  |
| --- | --- | --- |
| Десятичное число | Число в обычном двоичном виде | Число в виде 4 byte synchsafe integer |
| 255 | 00000000 00000000 00000000 11111111 | 00000000 00000000 00000001 01111111 |
| 384 | 00000000 00000000 00000001 10000000 | 00000000 00000000 00000011 00000000 |

Все целые числа в ID3v2 беззнаковые, поэтому 4-байтный synchsafe integer может хранить числа от 0 до (228 – 1), 5-байтный – от 0 до (235 – 1).

Synchsafe байт с произвольным значением обозначается как %0xxxxxxx

[[к оглавлению]](#head)

ID3v1.0

|  |  |
| --- | --- |
| Количество байт | Поле |
| 3 | Заголовок “TAG” |
| 30 | Название трека – 30 символов |
| 30 | Исполнитель – 30 символов |
| 30 | Название альбома – 30 символов |
| 4 | Год – YYYY |
| 30 | Комментарий |
| 1 | Жанр |

Итого 128 байт. Добавляется в конец mp3-файла. Десинхронизация не производится.

Список жанров: <https://en.wikipedia.org/wiki/List_of_ID3v1_Genres>

Строки хранятся в формате [ISO-8859-1](https://ru.wikipedia.org/wiki/ISO_8859-1)

[[к оглавлению]](#head)

ID3v1.1

|  |  |
| --- | --- |
| Количество байт | Поле |
| 3 | Заголовок “TAG” |
| 30 | Название трека – 30 символов |
| 30 | Исполнитель – 30 символов |
| 30 | Название альбома – 30 символов |
| 4 | Год – YYYY |
| 28 | Комментарий |
| 1 | Нулевой байт |
| 1 | Номер трека в альбоме |
| 1 | Жанр |

Размер, расположение, список жанров и формат строк – как в v1.0. Десинхронизация не производится.

[[к оглавлению]](#head)

Расширенный ID3v1

В процессе

[[к оглавлению]](#head)

ID3v2.2

В процессе; принципиально не отличается от [ID3v2.4](#v24)

ID3v2.3

В процессе; принципиально не отличается от [ID3v2.4](#v24)

ID3v2.4

ID3v2.4 может располагаться перед аудиоданными и после них; при расположении после аудиоданных он должен предшествовать тегам других видов – например, ID3v1. Один файл может содержать сколько угодно ID3v2.4-тегов.

Файл с тегом ID3v2.4 может выглядеть так:

<ID3v2.4> <аудиоданные> <ID3v2.4> <ID3v1>

Можно предположить, что чаще всего файлы выглядят так:

<ID3v2.4> <аудиоданные>

Структура тега

|  |  |  |
| --- | --- | --- |
| Количество байт | Часть | Особенности |
| 10 | [Заголовок (header)](#header) | Обязателен |
| От 6 до 15 | [Расширенный заголовок (extended header)](#extdheader) | Необязателен |
| Не менее 11 | [Фреймы (frames)](#frames) | Как минимум 1 фрейм |
| Любое | [Пэддинг (padding)](#padding) | Необязателен |
| 10 | [Футер (footer)](#footer) | Необязателен |

Padding и footer взаимоисключающи – либо есть только padding, либо только footer, либо нет их обоих.

[[к оглавлению]](#head)

Заголовок

|  |  |  |
| --- | --- | --- |
| Количество байт | Поле | Значение |
| 3 | Идентификатор ID3v2 | “ID3” |
| 1 | Версия ID3v2 | $04 |
| 1 | Субверсия[[1]](#fn1) (revision number) | Меньше $FF |
| 1 | Флаги | %[a](#fla)[b](#flb)[c](#flc)[d](#fld)0000 |
| 4 | Размер тега в байтах[[2]](#fn2) | 4 \* %0xxxxxxx |

Итого 10 байт.

[1] – субверсии одной версии обладают обратной совместимостью, поэтому их можно игнорировать

[2] – synchsafe integer. В данном поле хранится размер всего тега, кроме хедера и футера, то есть сумма размеров расширенного хедера, фреймов и пэддинга. Размер указывается с учётом десинхронизации, если она проводилась.

Флаги

a – наличие/отсутствие десинхронизации. Флаг ставится, только если все фреймы десинхронизированы

b – наличие/отсутствие расширенного заголовка

c – ставится, если тег находится в экспериментальном состоянии (this flag have to be set if the tag is in an experimental stage)

d – наличие/отсутствие футера

[[к структуре]](#id3v24_strct)

Расширенный заголовок

|  |  |  |
| --- | --- | --- |
| Количество байт | Поле | Значение |
| 4 | Размер в байтах[[3]](#fn3) | 4 \* %0xxxxxxx |
| 1 | Количество байтов с флагами | $01 |
| 1 | Байт с [флагами](#flags) | %0[b](#extd_flb)[c](#extd_flc)[d](#extd_fld)0000 |

[3] – synchsafe integer. Размер всего расширенного хедера.

Флаги

Каждому установленному флагу соответствуют несколько байт с информацией. Эти наборы байт идут в таком же порядке, в каком установлены флаги. Каждый набор начинается с байта, в котором указано количество следующих за ним байт. Все эти байты идут сразу после [байта с флагами.](#byte_w_flags_extd_h)

|  |  |  |  |
| --- | --- | --- | --- |
| Флаг | Если установлен, то: | Длина информации в байтах | Информация |
| b | Данный тег является апдейтом предыдущего. Все уникальные фреймы из этого тега должны заменить соответствующие им фреймы из предыдущего тега. | $00 | - |
| c | Расширенный хедер содержит контрольную сумму тега (CRC-32), посчитанную на фреймах и пэддинге. Контрольная сумма хранится в пяти synchsafe байтах. | $05 | 5 \* %0xxxxxxx |
| d | Расширенный хедер содержит байт с информацией об ограничениях для данного тега. | $01 | %[pp](#rst_pp) [q](#rst_q) [rr](#rst_rr) [s](#rst_s) [tt](#rst_tt) |

Ограничения

pp – ограничения на размер тега

00 – не более 128 фреймов; общий размер не более 1 Мб

01 – не более 64 фреймов; общий размер не более 128 Кб

10 – не более 32 фреймов; общий размер не более 40 Кб

11 – не более 32 фреймов; общий размер не более 4 Кб

q – ограничения по кодировке текста

0 – ограничений нет

1 – только [ISO-8859-1](https://ru.wikipedia.org/wiki/ISO_8859-1) или [UTF-8](https://ru.wikipedia.org/wiki/UTF-8)

rr – ограничения по длине текстовых полей (то есть ограничения по количеству символов в текстовом фрейме)

00 – ограничений нет

01 – не более 1024 символов

10 – не более 128 символов

11 – не более 30 символов

s – ограничения по формату изображений

0 – ограничений нет

1 – только PNG или JPEG

tt – ограничения по размеру изображений

00 – ограничений нет

01 – 256х256 пикселей или менее

10 – 64х64 пикселя или менее

11 – строго 64х64 пикселя, пока не указано иное

[[к структуре]](#id3v24_strct) [[к началу раздела]](#extdheader)

Пэддинг (padding)

Поскольку тег ID3v2.4 чаще всего находится в начале файла, при его изменении нужно перезаписывать весь файл. Чтобы этого избежать, можно после фреймов добавлять какое-то количество нулевых байтов, на место которых при расширении тега будут записываться значащие байты. Эти нулевые байты называются padding. Если тег содержит padding, то он не может содержать футер.

[[к структуре]](#id3v24_strct)

Футер (footer)

Футер – аналог хедера, только в конце тега. Может быть добавлен, чтобы быстрее находить тег при поиске с конца файла. Футер состоит из 10 байт, из которых первые три – идентификатор хедера наоборот, остальные – копия соответствующих байтов хедера.

|  |  |  |
| --- | --- | --- |
| Количество байт | Поле | Значение |
| 3 | Идентификатор ID3v2 | “3DI” |
| 1 | Версия ID3v2 | $04 |
| 1 | Субверсия[[1]](#fn1) (revision number) | Меньше $FF |
| 1 | Флаги | %[a](#fla)[b](#flb)[c](#flc)1[[4]](#fn4)0000 |
| 4 | Размер тега в байтах[[2]](#fn2) | 4 \* %0xxxxxxx |

[4] – очевидно, что если футер есть, то флаг футера должен быть установлен

Если тег содержит футер, то он не может содержать padding.

[[к структуре]](#id3v24_strct)

Фреймы

Фреймы состоят из заголовка и [информации как таковой](#v24_frame_types).

Заголовок

|  |  |  |
| --- | --- | --- |
| Количество байт | Поле | Значение |
| 4 | Четырёхсимвольный идентификатор фрейма | $xx xx xx xx |
| 4 | Размер фрейма[[5]](#fn5) | 4 \* %0xxxxxxx |
| 1 | Флаги статуса (status flags) | %0 [a](#status_fla) [b](#status_flb) [c](#status_flc) 0000 |
| 1 | Флаги формата (format flags) | %0 [h](#format_flh) 00 [k](#format_flk) [m](#format_flm) [n](#format_fln) [p](#format_flp) |

Итого 10 байт.

[5] – synchsafe integer. Содержит размер всего фрейма после шифрования, сжатия и десинхронизации, за исключением заголовка.

Флаги статуса

|  |  |
| --- | --- |
| Флаг | Если установлен, то: |
| a | Фрейм должен быть удалён при любом изменении тега |
| b | Фрейм должен быть удалён при любом изменении файла, не затрагивающем тег |
| c | Тег является read only; при изменении тега этот флаг должен быть сброшен |

Флаги формата

Любой из флагов [h](#format_flh), [m](#format_flm) и [p](#format_flp) имеет приложенную информацию, если он установлен. Байты с информацией идут в том же порядке, в каком и флаги. Байты с информацией идут сразу после заголовка фрейма и перед содержимым фрейма.

|  |  |  |
| --- | --- | --- |
| Флаг | Если установлен, то: | Приложенная информация |
| h | Данный фрейм принадлежит к группе фреймов. Более подробно про группы в описании фрейма GRID | Метка группы – 1 байт |
| k | Фрейм был сжат при помощи метода zlib. В этом случае флаг p также должен быть установлен. | - |
| m | Фрейм был зашифрован и содержит метку метода шифрования. Более подробно про шифрование в описании фрейма ENCR. | Метка метода шифрования – 1 байт |
| n | Все байты от конца заголовка до конца фрейма были десинхронизированы. | - |
| p | К заголовку приложен размер информации фрейма, представленный в виде 4-байтного synchsafe целого числа. Это то же самое, что размер фрейма, если бы все флаги были сброшены. | 4 \* %0xxxxxxx |

Фрейм сначала должен быть сжат, затем зашифрован, затем десинхронизирован.

В дальнейшем информация, приложенная к флагам формата, будет считаться частью заголовка, потому что так удобнее. По сути, это то же самое, что расширенный заголовок тега.

Скорее всего, все строки в фреймах должны заканчиваться 0-терминатором, но это не точно. Можно точно сказать, что 0-терминатор используется в качестве разделителя строк.

[[к началу раздела]](#frames)

Типы фреймов

Фреймы начинаются с четырёх байт идентификатора, в которых могут использоваться заглавные A-Z и 0-9. Идентификаторы, начинающиеся с X, Y и Z, могут свободно использоваться для любых пользовательских фреймов. Общий вид фреймов описан в таблице. Здесь написано то, что не помещается в таблицу. По умолчанию строки имеют формат [ISO-8859-1](https://ru.wikipedia.org/wiki/ISO_8859-1).

Обозначения

Уникальный фрейм – фрейм, который может встречаться в теге не более одного раза

Общий вид фреймов будет описываться без учёта заголовка, то есть будет описан только вид информации в фрейме

Текстовые фреймы

Идентификатор начинается с T; все текстовые фреймы уникальны.

Общий вид:

|  |  |
| --- | --- |
| Байт кодировки | $eb |
| Информация | <строка в соответствии с кодировкой> |

Байт кодировки:

$00 – [ISO-8859-1](https://ru.wikipedia.org/wiki/ISO_8859-1), заканчивается на $00

$01 – UTF-16 в кодировке Unicode; первые два символа – метка порядка байтов. Если они имеют вид $FE FF, то левый байт является старшим, правый – младшим (big-endian). Например, число 387 в таком виде запишется как 00000001 10000011. Если первые два символа - $FF FE, то правый байт является старшим, левый – младшим (little-endian). 38710 = 10000011 000000012 . Заканчивается на $00 00

$02 – UTF-16 big-endian в кодировке Unicode. Заканчивается на $00 00

$03 – UTF-8 в кодировке Unicode. Заканчивается на $00

Нетипичные текстовые фреймы

TKEY – музыкальный ключ, с которого начинается звук. Максимум 3 символа, не считая $00.

A, B, C, D, E, F, G – ground keys(?)

b, # - halfkeys(?)

m – минор(?)

o – offkey(?)

TCON – жанр. Может быть представлен в виде числовой строки с одним из значений ID3v1, в виде обычной строки с названием жанра, в виде ключевого слова – RX – remix, CR – cover; также все эти виды данных могут быть скомбинированы в любом виде.

TFLT – тип аудиофайла. По умолчанию MPEG Audio. Значения:

|  |  |
| --- | --- |
| MIME | Далее следует [тип MIME](https://ru.wikipedia.org/wiki/Список_MIME-типов#audio) |
| MPG | MPEG Audio |
| MPG/1 | MPEG ½ layer I |
| MPG/2 | MPEG ½ layer II |
| MPG/3 | MPEG ½ layer III |
| MPG/2.5 | MPEG 2.5 |
| MPG/AAC | Advanced audio compression |
| VQF | https://ru.wikipedia.org/wiki/TwinVQ |
| PCM | https://ru.wikipedia.org/wiki/Импульсно-кодовая\_модуляция |