

Информационный стандарт  
Документ: id3v2.4.0-structure  
M. Nilsson  
1.11.2000  
Перевод: elp87  
Агуст 2013

Теги ID3 версия 2.4.0 - Основная структура

## Статус документа

Этот документ является информационным стандартом и заменяет стандарт ID3v2.3.0 [ID3v2]. Официальный стандарт будет использовать другой номер ревизии даже если содержание будет идентично с документом. Материалы этого документа могут меняться для пояснений, но не для добавления или изменения функциональности.

Распространение этого документа без ограничений

## Краткое описание

Этот документ описывает основную структуру ID3v2.4.0, который является исправленной версией информационного стандарта ID3v2 версии 2.3.0. ID3v2 предлагает гибкий способ хранения аудио мета-информации внутри аудио файла. Информация может быть технической, как уравнения кривых, так и название, исполнитель, копирайт и т.д.

ID3v2.4.0 создан наиболее возможно близким к ID3v2.3.0 в целях обеспечения осуществления исправления наиболее просто

### 1 Содержание

### 2 Условные обозначения в документе

Текст внутри “” является текстовой строкой в точности, как она отображается в тэге. Числа, начинающиеся с \$ - шестнадцатиричные, а числа, начинающиеся с % - двоичные. \$xx используется для обозначения байта с неизвестным содержимым. %x используется для обозначения бита с неизвестным содержимым. Наибольший значащий бит (MSB) байта называется 'бит 7', а наименьший значащий бит (LSB) называется 'бит 0'.

Тег - это весь тег, описанный в этом документе. Фрейм - блок информации тега. Тег состоит из заголовка, фреймов и необязательных вложений. Поле - это часть информации; одно значение, строка и т.д. Числовая строка - это строка состоящая только из символов “0123456789”.

Ключевые слова как "ДОЛЖЕН", "НЕ ДОЛЖЕН", "НЕОБХОДИМО", "ОБЯЗАН", "НЕ ДОЛЖЕН", "СЛЕДУЕТ", "НЕ СЛЕДУЕТ", "РЕКОМЕНДУЕТСЯ", "МОЖЕТ быть" и "НЕОБЯЗАТЕЛЬНО" в этом документе следует интерпретировать, как описано в RFC 2119 [KEYWORDS].

### 3 Обзор ID3v2

ID3v2 - это общий формат тегов для аудио, который делает возможным хранение мета-данных об аудио внутри аудио-файла. Тег ID3v2, описанный в этом документе, в основном нацелен на файлы, кодированные в MPEG-1/2 layer I, MPEG-1/2 layer II, MPEG-1/2 layer III и MPEG-2.5, но может работать и с другими типами кодировок аудио или как самостоятельный формат мета-данных аудио.

ID3v2 разработан наиболее возможно гибким и расширяем для удовлетворения в потребности новой мета-информации, которая может возникнуть. Для этого ID3v2 построен, как контейнер для нескольких информационных блоков, названных фреймами, формат которых не должен быть известен программному обеспечению, которое обнаруживает их. В начале каждого фрейма находится уникальный и предопределенный идентификатор, размер дескриптора, который позволяет программному обеспечению пропускать неизвестные фреймы и поле флагов. Флаги описывают детали кодировки, и если фрейм должен находиться в теге, это должно быть неизвестно программному обеспечению, если файл изменен.

Порядок битов в ID3v2 - сначала наиболее значащий бит (MSB). Порядок байтов в многобайтовых числах - сначала наиболее значащий байт (т.е. \$12345678 будет кодироваться как \$12 34 56 78), также известный как порядок от старшего к младшему или сетевым порядком байтов.

#### Общая структура тегов

1. Заголовок (10 байтов)
2. Расширенный заголовок (переменная длина, НЕОБЯЗАТЕЛЬНО)
3. Фреймы (переменная длина)
4. Дополнение (переменная длина, НЕОБЯЗАТЕЛЬНО)
5. Нижний колонтитул (10 байт, НЕОБЯЗАТЕЛЬНО)

В общем, дополнение и нижний колонтитул являются взаимоисключающими. Смотрите детали в разделах 3.3, 3.4 и 5

### 3.1 Заголовок ID3v2

Первая часть тега ID3v2 - 10 байтовый заголовок тега, сформулированный следующим образом

Идентификатор ID3v2/файла	"ID3"
Версия ID3v2	\$04 00
Флаг ID3v2	%abcd0000
Размер ID3v2	4 * %0xxxxxxx

Первые три байта тега всегда "ID3" для идентификации, что это тег ID3v2, непосредственно за которым идут два байта версии. Первый байт версии ID3v2 - это его основная версия, в то время как второй байт - номер ревизии. В данном случае это ID3v2.4.0. Все ревизии обратно совместимы, тогда как основные версии - нет. Если программное обеспечение с поддержкой ID3v2.4.0 или ниже должно встретить версию 5 или выше, то оно должно просто пропустить весь тег. Версия или ревизия никогда не будет \$FF.

За номером версии следует поле флагов, из которых в настоящее время используется четыре флага

- а Рассинхронизация  
Бит 7 в флагах ID3v2 показывает используется ли рассинхронизация на всех фреймах (см.раздел 6.1); установленный бит означает использование.
- б Расширенный заголовок  
Второй бит (бит 6) показывает идет ли за заголовко расширенный заголовок. Расширенный заголовок описан в разделе 3.2. Установленный бит указывает на наличие расширенного заголовка.
- в Экспериментальный индикатор  
Третий бит (бит 5) используется как 'экспериментальный индикатор'. Этот флаг ДОЛЖЕН всегда быть установлен, если тег находится в экспериментальной стадии.
- д Наличие нижнего колонтитула  
Бит 4 показывает, что нижний колонтитул (раздел 3.4) присутствует в самом конце тега. Установленный бит показывает наличие нижнего колонтитула.

Все остальные флаги ДОЛЖНЫ быть очищены. Если хотя бы один из этих неопределенных флагов установлен, тег может быть нечитабельным для парсера, который не знает функцию флага.

Размер тега ID3v2 хранится, как 32-битное synchsafe целое <sup>1</sup>(раздел 6.2) , дающее в сумме 28 эффективных бит (означающие до 256 мб).

Размер тега ID3v2 - это сумма длин расширенного заголовка, дополнения и фреймов после рассинхронизации. Если нижний колонтитул присутствует, то это равняется ("общий размер" - 20) байт, в противном случае ("общий размер" - 10) байт.

Тег ID3v2 можно обнаружить по следующему шаблону:

\$ 49 44 33 yy yy xx zz zz zz zz

Где yy - меньше, чем \$FF, xx - байт флага, а zz - меньше, чем \$80

### 3.2 Расширенный заголовок

Расширенный заголовок содержит информацию, которая может обеспечить дальнейшее проникновение в структуру тега, но не является необходимой для правильного анализа информации тега; следовательно, расширенный заголовок является необязательным.

Размер расширенного заголовка	4 * %0xxxxxxx
Количество байт флага	\$01
Расширенные флаги	\$xx

Где "Размер расширенного заголовка" - это размер всего расширенного заголовка, хранящийся как 32-битное synchsafe целое. Расширенный заголовок, таким образом, никогда не может иметь размер менее 6 байт.

Поле расширенного флага с его размером, описанным "количество байт флага" определяется как:

%0bcd0000

Каждый флаг, установленный в расширенном заголовке, имеет прикрепленные данные, которые идут в порядке, в котором встречаются флаги (т.е. данные для флага 'b' идут до данных для флага 'c'). Сброшенные флаги не могут иметь прикрепленные данные. Все неизвестные флаги ДОЛЖНЫ быть сброшены, и соответствующие данные удалены, если тег изменился.

Данные каждого установленного флага начинаются с байта длины, который содержит значение от 0 до 128 (\$00-\$7f), за которым следуют данные, которые имеют длину поля, отображаемую байтом длины. Если флаг не имеет прикрепленных данных, значение \$00 используется как байт длины

#### b Тег-обновление

Если этот флаг установлен, то настоящий тег - обновление тега, найденного ранее в настоящем файле или потоке. Если фреймы, определенные как уникальные, найдены в настоящем теге, они переопределяют любые соответствующие фреймы, найденные в более раннем теге. Этот флаг не имеет соответствующих данных

Длина данных флага \$00

#### c Наличие данных CRC

Если этот флаг установлен, то в расширенный заголовок включены данные CRC-32 [ISO-3309]. CRC рассчитывается на все данные между заголовком и нижним колонтитулом, как указано в поле длины тега в заголовке, кроме расширенного заголовка. Обратите внимание, что оно включает дополнение (если оно есть), но исключает нижний колонтитул. CRC-32 хранится как 35 битное synchsafe целое, оставляя верхние 4 бита всегда обнуленными

Длина данных флага \$05

Общий фрейм CRC 5 \* %0xxxxxxx

<sup>1</sup> <http://en.wikipedia.org/wiki/Synchsafe>

#### d Ограничения тега

Для некоторых приложение может быть желательно, ограничить тег большими способами, нежели налагаемыми спецификацией ID3v2. Обратите внимание, что наличие этих ограничений не влияет на то, как тег декодируется, только, как он был ограничен перед кодированием. Если флаг установлен, на тег накладываются следующие ограничения:

Длина данных флага     \$01  
Общий фрейм CRC        %ppqrstt

#### p Ограничения на размер тега

- 00 Не более 128 кадров и 1мб общего размера тега
- 01 Не более 64 кадров и 128кб общего размера тега
- 10 Не более 32 кадров и 40кб общего размера тега
- 11 Не более 32 кадров и 4кб общего размера тега

#### q Ограничения на кодировку текста

- 0 Без ограничений
- 1 Строки кодируются только в ISO-8859-1 [ISO-8859-1] или UTF-8 [UTF-8]

#### r Ограничения на размер текстового поля

- 00 Без ограничений
- 01 Строка не длинее 1024 символов
- 10 Строка не длинее 128 символов
- 11 Строка не длинее 30 символов

Обратите внимание, что ничего не сказано о том, сколько байт используется для представления этих символов, поскольку это зависит от кодирования. Если текстовый фрейм состоит более чем из одной строки, сумма строки ограничена, как указано.

#### s Ограничение на кодировку изображения

- 0 Без ограничений
- 1 Изображения кодируются только в PNG [PNG] или JPEG [JFIF]

#### t Ограничения на размер изображения

- 00 Без ограничений
- 01 Все изображения - 256x256 пиксел или менее
- 10 Все изображения - 64x64 пиксел или менее
- 11 Все изображения - именно 64x64 пикселей, если не требуется иначе.

### 3.3 Нижний колонтитул ID3v2

**НЕОБЯЗАТЕЛЬНО** включать дополнение после последнего фрейма (в конце тега ID3v2), делая размер всех кадров вместе меньше размера, заданного в теге заголовка. Возможной целью этого дополнения является разрешение на добавление нескольких дополнительных фреймов или расширение существующих фреймов внутри тега без необходимости переписывать весь файл. Значение байтов вложения должно быть \$00. Тег **НЕ ДОЛЖЕН** иметь каких-либо дополнений между фреймами или между заголовком тега и фреймами. Кроме того, в нем **НЕ ДОЛЖНО** быть каких-либо дополнений, когда в тег добавлен нижний колонтитул тега.

### 3.4 Нижний колонтитул ID3v2

Для увеличения скорости обнаружения тега ID3v2 при поиске с конца файла, в тег может быть добавлен нижний колонтитул. **НЕОБХОДИМО** добавлять нижний колонтитул к прикрепленному тегу, т.е. тегу, находящемуся после всех аудио-данных. Нижний колонтитул является копией заголовка, но имеет другой идентификатор

Идентификатор ID3v2/файла	"ID3"
Версия ID3v2	\$04 00
Флаг ID3v2	%abcd0000
Размер ID3v2	4 * %0xxxxxxx

## 4 Обзор фрейма ID3v2

Все фреймы ID3v2 состоят из одного заголовка фрейма, за которым следует один или более полей, содержащих фактическую информацию. Заголовок содержит всегда 10 байтов и сформулирован следующим образом:

ID фрейма	\$xx xx xx xx (4 символа)
Размер	4 * %0xxxxxxx
Флаги	\$xx xx

ID фрейма состоит из заглавных символов A-Z и 0-9. Идентификаторы, начинающиеся с "X", "Y" и "Z" - экспериментальные фреймы и свободны в использовании для всех, без необходимости устанавливать экспериментальный бит в заголовок тега. Имейте в виду, что кто-то другой может использовать тот же идентификатор, что вы. Все остальные идентификаторы или используются, или зарезервированы для будущего использования.

За ID фрейма следует размер дескриптора, содержащий размер данных в последнем фрейме после кодирования, сжатия и рассинхронизации. Размер не учитывает заголовок фрейма ('общий размер фрейма' - 10 байт) и хранится как 32-битное synchsafe целое.

После размера дескриптора в заголовке фрейма идет два байта флагов. Эти флаги описаны в разделе 4.1

Нет фиксированного порядка фрейма, идущих в теге, но желательно чтобы фреймы были расположены в порядке значимости для распознавания файла. Пример такого порядка: UFID, TIT2, MCDI, TRCK ...

Тег ДОЛЖЕН содержать по меньшей мере один фрейм. Фрейм должен быть по меньшей мере размером 1 байт, не считая заголовка.

Если не указано другое, то строки, включая числовые строки и адреса-URL [URL], представляются в виде ISO-8859-1[ISO-8859-1] символов в диапазоне \$20 - \$FF. Такие строки представлены в описании фрейма как <text string> или <full text string>, если разрешены переносы строк. Если не указано другое, символ переноса строки запрещен. В ISO-8859-1 символ переноса представлен, когда он разрешен, только как \$0A.

Кадры, которые разрешают разные типы кодировки текста содержат байт описания кодировки текста. Возможные кодировки:

\$00	ISO-8859-1 [ISO-8859-1]. Заканчивается на \$00.
\$01	UTF-16 [UTF-16] кодированное в Unicode [UNICODE] с BOM 2. Все строки в этом кадре ДОЛЖНЫ иметь этот же порядок байтов. Заканчивается на \$00 00
\$02	UTF-16BE [UTF-16] кодированное Unicode [UNICODE] без BOM. Заканчивается на \$00 00
\$03	UTF-8 [UTF-8] одированное Unicode [UNICODE]. Заканчивается на \$00.

Строки, зависящие от кодировки, представлены в описании кадра как <text string according to encoding> или <full text string according to encoding>, если разрешен перенос строки. Любые пустые строки типа \$01, которые оканчиваются NULL могут иметь Unicode BOM и далее Unicode NULL (\$FF FE 00 00 или \$FE FF 00 00).

Поля метки времени основаны на подмножестве ISO 8601. Если быть максимально точным формат времени строка гггг-ММ-ддТчч:мм:сс (год, " - ", " месяц", " - ", " день", "Т", час (из 24), ":", минут, ":", секунды), но точность может быть уменьшена путем удаления столько показателей времени, сколько хочется. Следовательно, допустимые временные метки гггг, гггг-ММ, гггг-ММ-дд, гггг-ММ-ддТчч, гггг-ММ-ддТчч:мм и гггг-ММ-ддТчч:мм:сс. Все временные метки в UTC. Для продолжительности используйте символ шпеш '/', как описано в 8601, а для нескольких несмежных дат используйте несколько строк, если это разрешено описанием фрейма.

3 байта поля языка, присутствующего в некоторых фреймах, используются для описания языка содержимого фрейма согласно ISO-639-2 [ISO-639-2]. Язык должен быть представлен в нижнем регистре. Если язык не известен, должна использоваться строка "XXX".

Все адреса-URL [URL] МОГУТ быть относительными, т.е. "picture.png", "../doc.txt".

Если фрейм длиннее, чем должен быть, т.е. имеет больше полей, чем указано в этом документе, это значит что дополнения к фрейму были сделаны в более поздней версии стандарта ID3v2. Это отражено в номере ревизии в заголовке тега

#### 4.1 Флаги заголовка фрейма

В заголовке фрейма за дескриптор размера следует два байта флагов. Все неиспользуемые флаги ДОЛЖНЫ быть очищены. Первый байт для "сообщений статуса", а второй - описание формата. Если неизвестный флаг установлен во втором байте, скорее всего он не будет читаться. Некоторые флаги во втором байте показывают, что дополнительная информация добавлена в заголовок. Эти поля дополнительной информации расположены в порядке как флаги, которые указывают на них. Поле флага определено следующим образом (1 и 0 исключены из-за их сходства с единицей и нулём):

%0abc0000 %0h00kmnp

Формат флагов некоторых фреймов указывает, что дополнительные информационные поля добавлены к фрейму. Эта информация добавляется после заголовка фрейма и перед данными фрейма в том же порядке, что и флаги, которые указывают на них. Т.е. четыре байта распакованного размера будут предшествовать байту метода шифрования. Эти дополнения влияют на поле размера фрейма, но не являются предметом шифрования или сжатия.

Установка по умолчанию флагов статуса для фрейма (за исключением если указано иное) - "сохраняется если тег изменен" и "сохраняется если файл изменен", т.е. %0000000000.

##### 4.1.1 Флаги статуса фрейма

###### a Сохранение изменения тега

Этот флаг говорит разборщику тега что делать с этим фреймом, если он не известен, а тег каким-либо образом изменен. Это касается всех видов изменений, в том числе добавление дополнений и обновление фреймов.

- 0 Фрейм должен быть сохранен
- 1 Фрейм должен быть отброшен

###### b Сохранение изменения файла

Этот флаг говорит разборщику тега что делать с этим фреймом, если он не известен, а файл, включая тег, был изменен. Это не касается случая, когда аудио было полностью изменено на другие аудио-данные.

- 0 Фрейм должен быть сохранен
- 1 Фрейм должен быть отброшен

###### c Только чтение

Этот флаг, если установлен, говорит программному обеспечению, что содержимое этого фрейма предназначено только для чтения. Изменение содержимого может что-то сломать, например подпись. Если содержимое изменилось, без знания того, почему фрейм был помечен только для чтения и не принимая надлежащие средства для компенсации, например. пересчет подписи, бит ДОЛЖЕН быть очищен.

##### 4.1.2 Флаги формата фрейма

###### h Групповая идентичность

Этот флаг указывает входит ли этот фрейм или нет в группу с другими фреймами. Если установлен, байт идентификатора группы добавляется к фрейму. Каждый фрейм с тем же идентификатором группы принадлежит к той же группе.

- 0 Фрейм не содержит информацию о группе
- 1 Фрейм содержит информацию о группе

#### **k Сжатие**

Этот флаг указывает был ли фрейм сжат. Байт индикатор длины данных ДОЛЖЕН быть включен в фрейм.

- 0 Фрейм не сжат
- 1 Фрейм сжат с использованием deflate метода zlib [zlib]. Если установлен, то требуется чтобы также был установлен бит "Индикатор длины данных".

#### **m Шифрование**

Этот флаг указывает был ли зашифрован фрейм или нет. Если установлен, один байт с указанием метода шифрования будет добавлен к фрейму. Смотрите описание фрейма ENCR для получения большей информации о записи метода шифрования. Шифрование должно делаться после сжатия. Установлен этот флаг или нет, он требует наличия "Индикатора длины данных", зависящий от конкретного используемого алгоритма.

- 0 Фрейм не шифрован
- 1 Фрейм шифрован

#### **n Рассинхронизация**

Этот флаг указывает была ли применена рассинхронизация к этому фрейму. См. раздел 6 для подробностей о рассинхронизации. Если этот флаг установлен, то все данные от конца этого заголовка до конца этого фрейма была рассинхронизирована. Хотя и желательно, но не обязательно наличие индикатора длины данных для рассинхронизации.

- 0 Фрейм не был рассинхронизирован
- 1 Фрейм был рассинхронизирован

#### **p Индикатор длины данных**

Этот флаг указывает, что индикатор длины данных был добавлен к фрейму. Индикатор длины данных - значение, которое можно было бы написать как "Длина фрейма", если бы все флаги формата были обнулены; представляется как 32-битное synchsafe целое

- 0 Индикатор длины данных отсутствует
- 1 Индикатор длины данных был добавлен к фрейму

## **5 Расположение тега**

**По умолчанию, тег ID3v2 располагается перед аудио, это может принести пользу проигрывателям получить информацию при потоковых данных. Также возможно добавить тег в конец или сделать сочетание начало/конец. При принятии решения, где извлеченный тег должен быть расположен, следующий порядок предпочтений ДОЛЖЕН быть рассмотрен.**

1. Тег присоединен спереди
2. Тег присоединен спереди со всей необходимой информацией, и второй тег добавлен в конце файла до тегов из других систем тегов. Первому тегу необходимо иметь фрейм SEEK.
3. Тег добавлен в конце файла до тегов из других систем тегов.

**В случаях 2 и 3 тег может быть просто присоединен к концу, если ни один другой тег не представлен. Предложенный метод, чтобы найти ID3v2-теги:**

1. Искать присоединенный в начало тег используя шаблон из раздела 3.1
2. Если был обнаружен фрейм SEEK, использовать его значение для руководства дальнейшим поиском.
3. Искать нижний колонтитул тега, сканируя с конца файла.

**При нахождении каждого нового тега, старый тег должен быть сброшен, за исключением если флаг обновления установлен в расширенном заголовке (раздел 3.2).**

## 6 Рассинхронизация

Единственным назначением рассинхронизации является сделать тег ID3v2 максимально совместимым с существующим м и аппаратным обеспечением. От "рассинхронизации" тега не будет никакой пользы, если файл будет обрабатываться только ID3v2 совместимым программным и аппаратным обеспечением

### 6.1 Схема рассинхронизации

В случае нахождения ложной синхронизации внутри тега, один нулевой байт вставляется после байта ложной синхронизации. Формат синхронизаций, которые должны быть заменены декодером ID3v2 следующий:

%11111111 111xxxxx

должен быть заменен на:

%11111111 00000000 111xxxxx

Это имеет сторонний эффект, что все комбинации \$FF 00 должны быть заменены, следовательно на них не будет оказано влияние процессом декодирования. Поэтому все комбинации \$FF 00 должны быть заменены на комбинацию \$FF 00 00 во время рассинхронизации.

Для отображения использования рассинхронизации в заголовке фрейма должен быть установлен флаг рассинхронизации. Этот бит ДОЛЖЕН быть установлен, если фрейм был изменен рассинхронизацией, и НЕ ДОЛЖЕН быть установлен, если не изменен. Если все фреймы в теги были рассинхронизированы, флаг рассинхронизации ДОЛЖЕН быть установлен в заголовке тега. Он НЕ ДОЛЖЕН установлен, если в теге есть фреймы, которые не были рассинхронизированы.

Предположим, что первый байт аудио - \$FF. В особом случае, когда последний байт последнего фрейма - \$FF и не используется ни дополнение, ни нижний колонтитул, то будет введена ложная синхронизация. Это может быть решено добавлением нижнего колонтитула, дополнения или рассинхронизацией фрейма и добавлением \$00 в конец данных фрейма, таким образом добавляя дополнительный байт в размер фрейма, нежели было нормальной рассинхронизацией. Хотя и не предпочтительно, но разрешается использовать последний метод во всех фреймах, кончающихся на \$FF

Предпочтительно, чтобы тег был или полностью рассинхронизирован, или не рассинхронизирован вообще. Полностью рассинхронизированный тег не имеет ложных синхронизаций, как отмечено выше, и не заканчивается на \$FF. Полностью нерассинхронизированный тег содержит нерассинхронизированные фреймы и их флаг рассинхронизации в заголовке очищены.

Имейте в виду, что если используется сжатие или шифрование, впоследствии схема рассинхронизации ДОЛЖНА быть использована. При декодировании рассинхронизированных фреймов, сначала ДОЛЖНА быть отменена схема рассинхронизации, а потом шифрование и декомпрессия.

### 6.2 Synchsafe целые

В некоторых частях тега проблематично использовать схему рассинхронизации, потому что размер рассинхронизированных данных заранее не известен, что особенно проблематично с описателями размера. Решение в ID3v2 - использование synchsafe целых, в которых никто не может быть ложных синхронизаций. Synchsafe целые - целые числа, где старший бит (бит 7) обнулен, делая доступными 7 бит из 8. Таким образом, 32-битное synchsafe целое может хранить 28 бит информации.

Пример:

255 (%11111111) кодируется как 16-битное synchsafe целое 383 (%00000001 01111111).



## 7 Авторские права

Copyright (C) Martin Nilsson 2000. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that a reference to this document is included on all such copies and derivative works. However, this document itself may not be modified in any way and reissued as the original document.

The limited permissions granted above are perpetual and will not be revoked.

This document and the information contained herein is provided on an 'AS IS' basis and THE AUTHORS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright (C) Перевод на русский язык elp87 2013. All Rights Reserved.

Данный документ распространяется без каких-либо ограничений. Данный документ не является стандартом ID3V2, а только переводом на русский язык и предназначен для общего ознакомления. Для ознакомления со стандартом "ID3 tag version 2.4.0 - Main Structure" посетите сайт <http://www.id3.org/>