www.archive.org
415.561.6767
415.840-0391 e-fax

Internet Archive
300 Funston Avenue
San Francisco, CA 94118

AFFIDAVIT OF CHRISTOPHER BUTLER

1. I am the Office Manager at the Internet Archive, located in San Francisco, California. I make this declaration of my own personal knowledge.

2. The Internet Archive is a website that provides access to a digital library of Internet sites and other cultural artifacts in digital form. Like a paper library, we provide free access to researchers, historians, scholars, and the general public. The Internet Archive has partnered with and receives support from various institutions, including the Library of Congress.

3. The Internet Archive has created a service known as the Wayback Machine. The Wayback Machine makes it possible to surf more than 450 billion pages stored in the Internet Archive's web archive. Visitors to the Wayback Machine can search archives by URL (i.e., a website address). If archived records for a URL are available, the visitor will be presented with a list of available dates. The visitor may select one of those dates, and then begin surfing on an archived version of the Web. The links on the archived files, when served by the Wayback Machine, point to other archived files (whether HTML pages or images). If a visitor clicks on a link on an archived page, the Wayback Machine will serve the archived file with the closest available date to the page upon which the link appeared and was clicked.

4. The archived data made viewable and browseable by the Wayback Machine is compiled using software programs known as crawlers, which surf the Web and automatically store copies of web files, preserving these files as they exist at the point of time of capture.

5. The Internet Archive assigns a URL on its site to the archived files in the format http://web.archive.org/web/[Year in yyyy][Month in mm][Day in dd][Time code in hh:mm:ss]/[Archived URL]. Thus, the Internet Archive URL http://web.archive.org/web/19970126045828/http://www.archive.org/ would be the URL for the record of the Internet Archive home page HTML file (http://www.archive.org/) archived on January 26, 1997 at 4:58 a.m. and 28 seconds (1997/01/26 at 04:58:28). A web browser may be set such that a printout from it will display the URL of a web page in the printout's footer. The date assigned by the Internet Archive applies to the HTML file but not to image files linked therein. Thus images that appear on a page may not have been archived on the same date as the HTML file. Likewise, if a website is designed with "frames," the date assigned by the Internet Archive applies to the frameset as a whole, and not the individual pages within each frame.

6. Attached hereto as Exhibit A are true and accurate copies of printouts of the Internet Archive's records of the HTML files or PDF files for the URLs and the dates specified in the footer of the printout (HTML) or attached coversheet (PDF).

7. I declare under penalty of perjury that the foregoing is true and correct.

DATE: 12/14/18

Christopher Butler

CALIFORNIA JURAT

See Attached Document.

A notary public or other officer completing this certificate verifies only the identity of the individual who signed the document to which this certificate is attached, and not the truthfulness, accuracy, or validity of that document.

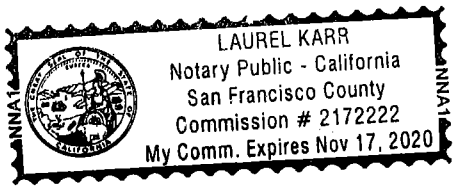
State of California
County of San Francisco

Subscribed and sworn to (or affirmed) before me on this

14 day of December, 2018, by

Christopher Butler,

proved to me on the basis of satisfactory evidence to be the person who appeared before me.



Signature: Laurel Karr

Exhibit A

ID3v2

The audience is informed

Introduction

A short explanation of ID3v2 and related standards

In depth information

Technical specifications and binaries.

Implementations

Existing and future programs and sources.

Contributors

Mailinglist

Copyright © 2000 Martin Nilsson - Latest changes 000730

Unified Patents
Exhibit 1013
Page 4 of 23

Developer Information

The ID3v2 documents

ID3v2.3.0 Informal standard - Plain text document describing ID3v2.3.0.
ID3v2.3.0 Informal standard - HTML document describing ID3v2.3.0.
ID3v2 Programming Guidlines - What is good and what is ungood.

ID3v2.4.0 Drafts - The drafts for probably the final major ID3v2 version

(ID3v2.2.0 Informal standard - Plain text document describing ID3v2.2.0 - obsolete)
(ID3v2.2.0 Commented informal standard - The ID3v2.2.0 HTML document with some notes - obsolete).

Related standards

ISO 639-2 - Language codes
ISO 4217 - Currency codes
ISO 3901 - International Standard Recording Code
Unicode - The Unicode Consortium (http://www.unicode.org/)
PNG - The PNG Home Page (http://www.cdrom.com/pub/png/)
zlib - The zlib Home Page (http://www.cdrom.com/pub/infozip/zlib/)

Example tags

There are currently no official example tags for ID3v2.3.0, but there is some experimental ones made available by the ID3Lib project.

ID3v2 logos

The ID3v2 logo is copyright © 1998 Martin Nilsson. You may freely use the ID3v2 logo on pages containing ID3v2 information and in programs supporting the ID3v2 standard.

id3v2.png - The original logo (2.6 kB, 156 x 149)
attilas_id3logo.jpg - A more colourful logo by Attila Sandor Fiko (3.6 kB, 132 x 132)
3d-logo.gif - A rendered rotating logo by Chris Nunn (44.7 kB, 132 x 132)
id3v2.ico - windows icon file by Chuck Zenkus (8.2 kB, 32 x 32 x 16/256, 48 x 48 x 16/256)

Unified Patents
Exhibit 1013

Informal standard M. Nilsson
Document: id3v2-00.txt 26th March 1998

ID3 tag version 2

Status of this document

This document is an Informal standard and is released so that implementors could have a set standard before the formal standard is set. The formal standard will use another version number if not identical to what is described in this document. The contents in this document may change for clarifications but never for added or altered functionallity.

Distribution of this document is unlimited.

Abstract

The recent gain of popularity for MPEG layer III audio files on the internet forced a standardised way of storing information about an audio file within itself to determinate its origin and contents.

Today the most accepted way to do this is with the so called ID3 tag, which is simple but very limited and in some cases very unsuitable. The ID3 tag has very limited space in every field, very limited numbers of fields, not expandable or upgradeable and is placed at the end of a the file, which is unsuitable for streaming audio. This draft is an attempt to answer these issues with a new version of the ID3 tag.

1. Table of contents

2. Conventions in this document

In the examples, text within "" is a text string exactly as it appears in a file. Numbers preceded with $ are hexadecimal and numbers preceded with % are binary. $xx is used to indicate a byte with unknown content. %x is used to indicate a bit with unknown content. The most significant bit (MSB) of a byte is called 'bit 7' and the least significant bit (LSB) is called 'bit 0'.

A tag is the whole tag described in this document. A frame is a block of information in the tag. The tag consists of a header, frames and optional padding. A field is a piece of information; one value, a string etc. A numeric string is a string that consists of the

Unified Patents
Exhibit 1013
Page 6 of 23

characters 0-9 only.

3. ID3v2 overview

The two biggest design goals were to be able to implement ID3v2 without disturbing old software too much and that ID3v2 should be expandable.

The first criterion is met by the simple fact that the MPEG [MPEG] decoding software uses a syncsignal, embedded in the audiostream, to 'lock on to' the audio. Since the ID3v2 tag doesn't contain a valid syncsignal, no software will attempt to play the tag. If, for any reason, coincidence make a syncsignal appear within the tag it will be taken care of by the 'unsynchronisation scheme' described in section 5.

The second criterion has made a more noticeable impact on the design of the ID3v2 tag. It is constructed as a container for several information blocks, called frames, whose format need not be known to the software that encounters them. At the start of every frame there is an identifier that explains the frames's format and content, and a size descriptor that allows software to skip unknown frames.

If a total revision of the ID3v2 tag should be needed, there is a version number and a size descriptor in the ID3v2 header.

The ID3 tag described in this document is mainly targeted to files encoded with MPEG-2 layer I, MPEG-2 layer II, MPEG-2 layer III and MPEG-2.5, but may work with other types of encoded audio.

The bitorder in ID3v2 is most significant bit first (MSB). The byteorder in multibyte numbers is most significant byte first (e.g. $12345678 would be encoded $12 34 56 78).

It is permitted to include padding after all the final frame (at the end of the ID3 tag), making the size of all the frames together smaller than the size given in the head of the tag. A possible purpose of this padding is to allow for adding a few additional frames or enlarge existing frames within the tag without having to rewrite the entire file. The value of the padding bytes must be $00.

3.1. ID3v2 header

The ID3v2 tag header, which should be the first information in the file, is 10 bytes as follows:

```
  ID3/file identifier      "ID3"
  ID3 version              $02 00
  ID3 flags                %xx000000
  ID3 size             4 * %0xxxxxxx
```

The first three bytes of the tag are always "ID3" to indicate that this is an ID3 tag, directly followed by the two version bytes. The first byte of ID3 version is it's major version, while the second byte is its revision number. All revisions are backwards compatible while major versions are not. If software with ID3v2 and below support should encounter version three or higher it should simply ignore the whole tag. Version and revision will never be $FF.

The first bit (bit 7) in the 'ID3 flags' is indicating whether or not unsynchronisation is used (see section 5 for details); a set bit indicates usage.

The second bit (bit 6) is indicating whether or not compression is used; a set bit indicates usage. Since no compression scheme has been decided yet, the ID3 decoder (for now) should just ignore the entire tag if the compression bit is set.

The ID3 tag size is encoded with four bytes where the first bit (bit 7) is set to zero in every byte, making a total of 28 bits. The zeroed bits are ignored, so a 257 bytes long tag is represented as $00 00 02 01.

The ID3 tag size is the size of the complete tag after unsychronisation, including padding, excluding the header (total tag size - 10). The reason to use 28 bits (representing up to 256MB) for size description is that we don't want to run out of space here.

A ID3v2 tag can be detected with the following pattern:
  $49 44 33 yy yy xx zz zz zz zz
Where yy is less than $FF, xx is the 'flags' byte and zz is less than $80.

3.2. ID3v2 frames overview

The headers of the frames are similar in their construction. They consist of one three character identifier (capital A-Z and 0-9) and one three byte size field, making a total of six bytes. The header is excluded from the size. Identifiers beginning with "X", "Y" and "Z" are for experimental use and free for everyone to use. Have in mind

Unified Patents
Exhibit 1013
Page 7 of 23

that someone else might have used the same identifier as you. All other identifiers are either used or reserved for future use.

The three character frame identifier is followed by a three byte size descriptor, making a total header size of six bytes in every frame. The size is calculated as framesize excluding frame identifier and size descriptor (frame size - 6).

There is no fixed order of the frames' appearance in the tag, although it is desired that the frames are arranged in order of significance concerning the recognition of the file. An example of such order: UFI, MCI, TT2 ...

A tag must contain at least one frame. A frame must be at least 1 byte big, excluding the 6-byte header.

If nothing else is said a string is represented as ISO-8859-1 [ISO-8859-1] characters in the range $20 - $FF. All unicode strings [UNICODE] use 16-bit unicode 2.0 (ISO/IEC 10646-1:1993, UCS-2). All numeric strings are always encoded as ISO-8859-1. Terminated strings are terminated with $00 if encoded with ISO-8859-1 and $00 00 if encoded as unicode. If nothing else is said newline character is forbidden. In ISO-8859-1 a new line is represented, when allowed, with $0A only. Frames that allow different types of text encoding have a text encoding description byte directly after the frame size. If ISO-8859-1 is used this byte should be $00, if unicode is used it should be $01.

The three byte language field is used to describe the language of the frame's content, according to ISO-639-2 [ISO-639-2].

All URLs [URL] may be relative, e.g. "picture.png", "../doc.txt".

If a frame is longer than it should be, e.g. having more fields than specified in this document, that indicates that additions to the frame have been made in a later version of the ID3 standard. This is reflected by the revision number in the header of the tag.

4. Declared ID3v2 frames

The following frames are declared in this draft.

4.19 BUF Recommended buffer size

4.17 CNT Play counter
4.11 COM Comments
4.21 CRA Audio encryption
4.20 CRM Encrypted meta frame

4.6 ETC Event timing codes
4.13 EQU Equalization

4.16 GEO General encapsulated object

4.4 IPL Involved people list

4.22 LNK Linked information

4.5 MCI Music CD Identifier
4.7 MLL MPEG location lookup table

4.15 PIC Attached picture
4.18 POP Popularimeter

4.14 REV Reverb
4.12 RVA Relative volume adjustment

4.10 SLT Synchronized lyric/text
4.8 STC Synced tempo codes

4.2.1 TAL Album/Movie/Show title
4.2.1 TBP BPM (Beats Per Minute)
4.2.1 TCM Composer
4.2.1 TCO Content type
4.2.1 TCR Copyright message
4.2.1 TDA Date
4.2.1 TDY Playlist delay
4.2.1 TEN Encoded by
4.2.1 TFT File type
4.2.1 TIM Time
4.2.1 TKE Initial key
4.2.1 TLA Language(s)
4.2.1 TLE Length
4.2.1 TMT Media type
4.2.1 TOA Original artist(s)/performer(s)
4.2.1 TOF Original filename
4.2.1 TOL Original Lyricist(s)/text writer(s)
4.2.1 TOR Original release year
4.2.1 TOT Original album/Movie/Show title
4.2.1 TP1 Lead artist(s)/Lead performer(s)/Soloist(s)/Performing group
4.2.1 TP2 Band/Orchestra/Accompaniment
4.2.1 TP3 Conductor/Performer refinement

Unified Patents
Exhibit 1013
Page 8 of 23

```
    4.2.1 TP4 Interpreted, remixed, or otherwise modified by
    4.2.1 TPA Part of a set
    4.2.1 TPB Publisher
    4.2.1 TRC ISRC (International Standard Recording Code)
    4.2.1 TRD Recording dates
    4.2.1 TRK Track number/Position in set
    4.2.1 TSI Size
    4.2.1 TSS Software/hardware and settings used for encoding
    4.2.1 TT1 Content group description
    4.2.1 TT2 Title/Songname/Content description
    4.2.1 TT3 Subtitle/Description refinement
    4.2.1 TXT Lyricist/text writer
    4.2.2 TXX User defined text information frame
    4.2.1 TYE Year

    4.1   UFI Unique file identifier
    4.9   ULT Unsychronized lyric/text transcription

    4.3.1 WAF Official audio file webpage
    4.3.1 WAR Official artist/performer webpage
    4.3.1 WAS Official audio source webpage
    4.3.1 WCM Commercial information
    4.3.1 WCP Copyright/Legal information
    4.3.1 WPB Publishers official webpage
    4.3.2 WXX User defined URL link frame
```

4.1. Unique file identifier

This frame's purpose is to be able to identify the audio file in a database that may contain more information relevant to the content. Since standardisation of such a database is beyond this document, all frames begin with a null-terminated string with a URL [URL] containing an email address, or a link to a location where an email address can be found, that belongs to the organisation responsible for this specific database implementation. Questions regarding the database should be sent to the indicated email address. The URL should not be used for the actual database queries. If a $00 is found directly after the 'Frame size' the whole frame should be ignored, and preferably be removed. The 'Owner identifier' is then followed by the actual identifier, which may be up to 64 bytes. There may be more than one "UFI" frame in a tag, but only one with the same 'Owner identifier'.

```
  Unique file identifier  "UFI"
  Frame size              $xx xx xx
  Owner identifier        <textstring> $00
  Identifier              <up to 64 bytes binary data>
```

4.2. Text information frames

The text information frames are the most important frames, containing information like artist, album and more. There may only be one text information frame of its kind in an tag. If the textstring is followed by a termination ($00 (00)) all the following information should be ignored and not be displayed. All the text information frames have the following format:

```
  Text information identifier  "T00" - "TZZ" , excluding "TXX",
                               described in 4.2.2.
  Frame size                   $xx xx xx
  Text encoding                $xx
  Information                  <textstring>
```

4.2.1. Text information frames - details

TT1
The 'Content group description' frame is used if the sound belongs to a larger category of sounds/music. For example, classical music is often sorted in different musical sections (e.g. "Piano Concerto", "Weather - Hurricane").

TT2
The 'Title/Songname/Content description' frame is the actual name of the piece (e.g. "Adagio", "Hurricane Donna").

TT3
The 'Subtitle/Description refinement' frame is used for information directly related to the contents title (e.g. "Op. 16" or "Performed live at wembley").

TP1
The 'Lead artist(s)/Lead performer(s)/Soloist(s)/Performing group' is used for the main artist(s). They are seperated with the "/" character.

TP2
The 'Band/Orchestra/Accompaniment' frame is used for additional information about the performers in the recording.

TP3
The 'Conductor' frame is used for the name of the conductor.

```
TP4
 The 'Interpreted, remixed, or otherwise modified by' frame contains
 more information about the people behind a remix and similar
 interpretations of another existing piece.

TCM
 The 'Composer(s)' frame is intended for the name of the composer(s).
 They are seperated with the "/" character.

TXT
 The 'Lyricist(s)/text writer(s)' frame is intended for the writer(s)
 of the text or lyrics in the recording. They are seperated with the
 "/" character.

TLA
 The 'Language(s)' frame should contain the languages of the text or
 lyrics in the audio file. The language is represented with three
 characters according to ISO-639-2. If more than one language is used
 in the text their language codes should follow according to their
 usage.

TCO
 The content type, which previously (in ID3v1.1, see appendix A) was
 stored as a one byte numeric value only, is now a numeric string. You
 may use one or several of the types as ID3v1.1 did or, since the
 category list would be impossible to maintain with accurate and up to
 date categories, define your own.
 References to the ID3v1 genres can be made by, as first byte, enter
 "(" followed by a number from the genres list (section A.3.) and
 ended with a ")" character. This is optionally followed by a
 refinement, e.g. "(21)" or "(4)Eurodisco". Several references can be
 made in the same frame, e.g. "(51)(39)". If the refinement should
 begin with a "(" character it should be replaced with "((", e.g. "((I
 can figure out any genre)" or "(55)((I think...)". The following new
 content types is defined in ID3v2 and is implemented in the same way
 as the numerig content types, e.g. "(RX)".

   RX  Remix
   CR  Cover

TAL
 The 'Album/Movie/Show title' frame is intended for the title of the
 recording(/source of sound) which the audio in the file is taken from.

TPA
 The 'Part of a set' frame is a numeric string that describes which
 part of a set the audio came from. This frame is used if the source
 described in the "TAL" frame is divided into several mediums, e.g. a
 double CD. The value may be extended with a "/" character and a
 numeric string containing the total number of parts in the set. E.g.
 "1/2".

TRK
 The 'Track number/Position in set' frame is a numeric string
 containing the order number of the audio-file on its original
 recording. This may be extended with a "/" character and a numeric
 string containing the total numer of tracks/elements on the original
 recording. E.g. "4/9".

TRC
 The 'ISRC' frame should contian the International Standard Recording
 Code [ISRC].

TYE
 The 'Year' frame is a numeric string with a year of the recording.
 This frames is always four characters long (until the year 10000).

TDA
 The 'Date' frame is a numeric string in the DDMM format containing
 the date for the recording. This field is always four characters
 long.

TIM
 The 'Time' frame is a numeric string in the HHMM format containing
 the time for the recording. This field is always four characters
 long.

TRD
 The 'Recording dates' frame is a intended to be used as complement to
 the "TYE", "TDA" and "TIM" frames. E.g. "4th-7th June, 12th June" in
 combination with the "TYE" frame.

TMT
 The 'Media type' frame describes from which media the sound
 originated. This may be a textstring or a reference to the predefined
 media types found in the list below. References are made within "("
 and ")" and are optionally followed by a text refinement, e.g. "(MC)
 with four channels". If a text refinement should begin with a "("
 character it should be replaced with "((" in the same way as in the
 "TCO" frame. Predefined refinements is appended after the media type,
 e.g. "(CD/S)" or "(VID/PAL/VHS)".
```

```
    DIG    Other digital media
      /A    Analog transfer from media

    ANA    Other analog media
      /WAC  Wax cylinder
      /8CA  8-track tape cassette

    CD     CD
      /A    Analog transfer from media
      /DD   DDD
      /AD   ADD
      /AA   AAD

    LD     Laserdisc
      /A     Analog transfer from media

    TT     Turntable records
      /33    33.33 rpm
      /45    45 rpm
      /71    71.29 rpm
      /76    76.59 rpm
      /78    78.26 rpm
      /80    80 rpm

    MD     MiniDisc
      /A    Analog transfer from media

    DAT    DAT
      /A    Analog transfer from media
      /1    standard, 48 kHz/16 bits, linear
      /2    mode 2, 32 kHz/16 bits, linear
      /3    mode 3, 32 kHz/12 bits, nonlinear, low speed
      /4    mode 4, 32 kHz/12 bits, 4 channels
      /5    mode 5, 44.1 kHz/16 bits, linear
      /6    mode 6, 44.1 kHz/16 bits, 'wide track' play

    DCC    DCC
      /A    Analog transfer from media

    DVD    DVD
      /A    Analog transfer from media

    TV     Television
      /PAL    PAL
      /NTSC   NTSC
      /SECAM  SECAM

    VID    Video
      /PAL    PAL
      /NTSC   NTSC
      /SECAM  SECAM
      /VHS    VHS
      /SVHS   S-VHS
      /BETA   BETAMAX

    RAD    Radio
      /FM   FM
      /AM   AM
      /LW   LW
      /MW   MW

    TEL    Telephone
      /I    ISDN

    MC     MC (normal cassette)
      /4    4.75 cm/s (normal speed for a two sided cassette)
      /9    9.5 cm/s
      /I    Type I cassette (ferric/normal)
      /II   Type II cassette (chrome)
      /III  Type III cassette (ferric chrome)
      /IV   Type IV cassette (metal)

    REE    Reel
      /9    9.5 cm/s
      /19   19 cm/s
      /38   38 cm/s
      /76   76 cm/s
      /I    Type I cassette (ferric/normal)
      /II   Type II cassette (chrome)
      /III  Type III cassette (ferric chrome)
      /IV   Type IV cassette (metal)

  TFT
   The 'File type' frame indicates which type of audio this tag defines.
   The following type and refinements are defined:

     MPG    MPEG Audio
       /1     MPEG 2 layer I
       /2     MPEG 2 layer II
       /3     MPEG 2 layer III
       /2.5   MPEG 2.5
       /AAC   Advanced audio compression
```

Unified Patents
Exhibit 1013
Page 11 of 23

but other types may be used, not for these types though. This is used in a similar way to the predefined types in the "TMT" frame, but without parenthesis. If this frame is not present audio type is assumed to be "MPG".

TBP
BPM is short for beats per minute, and is easily computed by dividing the number of beats in a musical piece with its length. To get a more accurate result, do the BPM calculation on the main-part only. To acquire best result measure the time between each beat and calculate individual BPM for each beat and use the median value as result. BPM is an integer and represented as a numerical string.

TCR
The 'Copyright message' frame, which must begin with a year and a space character (making five characters), is intended for the copyright holder of the original sound, not the audio file itself. The absence of this frame means only that the copyright information is unavailable or has been removed, and must not be interpreted to mean that the sound is public domain. Every time this field is displayed the field must be preceded with "Copyright " (C) " ", where (C) is one character showing a C in a circle.

TPB
The 'Publisher' frame simply contains the name of the label or publisher.

TEN
The 'Encoded by' frame contains the name of the person or organisation that encoded the audio file. This field may contain a copyright message, if the audio file also is copyrighted by the encoder.

TSS
The 'Software/hardware and settings used for encoding' frame includes the used audio encoder and its settings when the file was encoded. Hardware refers to hardware encoders, not the computer on which a program was run.

TOF
The 'Original filename' frame contains the preferred filename for the file, since some media doesn't allow the desired length of the filename. The filename is case sensitive and includes its suffix.

TLE
The 'Length' frame contains the length of the audiofile in milliseconds, represented as a numeric string.

TSI
The 'Size' frame contains the size of the audiofile in bytes excluding the tag, represented as a numeric string.

TDY
The 'Playlist delay' defines the numbers of milliseconds of silence between every song in a playlist. The player should use the "ETC" frame, if present, to skip initial silence and silence at the end of the audio to match the 'Playlist delay' time. The time is represented as a numeric string.

TKE
The 'Initial key' frame contains the musical key in which the sound starts. It is represented as a string with a maximum length of three characters. The ground keys are represented with "A","B","C","D","E", "F" and "G" and halfkeys represented with "b" and "#". Minor is represented as "m". Example "Cbm". Off key is represented with an "o" only.

TOT
The 'Original album/Movie/Show title' frame is intended for the title of the original recording(/source of sound), if for example the music in the file should be a cover of a previously released song.

TOA
The 'Original artist(s)/performer(s)' frame is intended for the performer(s) of the original recording, if for example the music in the file should be a cover of a previously released song. The performers are seperated with the "/" character.

TOL
The 'Original Lyricist(s)/text writer(s)' frame is intended for the text writer(s) of the original recording, if for example the music in the file should be a cover of a previously released song. The text writers are seperated with the "/" character.

TOR
The 'Original release year' frame is intended for the year when the original recording, if for example the music in the file should be a cover of a previously released song, was released. The field is formatted as in the "TDY" frame.

4.2.2. User defined text information frame

Unified Patents
Exhibit 1013
Page 12 of 23

This frame is intended for one-string text information concerning the audiofile in a similar way to the other "T"xx frames. The frame body consists of a description of the string, represented as a terminated string, followed by the actual string. There may be more than one "TXX" frame in each tag, but only one with the same description.

User defined...	"TXX"
Frame size	$xx xx xx
Text encoding	$xx
Description	<textstring> $00 (00)
Value	<textstring>

4.3. URL link frames

With these frames dynamic data such as webpages with touring information, price information or plain ordinary news can be added to the tag. There may only be one URL [URL] link frame of its kind in an tag, except when stated otherwise in the frame description. If the textstring is followed by a termination ($00 (00)) all the following information should be ignored and not be displayed. All URL link frames have the following format:

URL link frame	"W00" - "WZZ" , excluding "WXX" (described in 4.3.2.)
Frame size	$xx xx xx
URL	<textstring>

4.3.1. URL link frames - details

WAF
The 'Official audio file webpage' frame is a URL pointing at a file specific webpage.

WAR
The 'Official artist/performer webpage' frame is a URL pointing at the artists official webpage. There may be more than one "WAR" frame in a tag if the audio contains more than one performer.

WAS
The 'Official audio source webpage' frame is a URL pointing at the official webpage for the source of the audio file, e.g. a movie.

WCM
The 'Commercial information' frame is a URL pointing at a webpage with information such as where the album can be bought. There may be more than one "WCM" frame in a tag.

WCP
The 'Copyright/Legal information' frame is a URL pointing at a webpage where the terms of use and ownership of the file is described.

WPB
The 'Publishers official webpage' frame is a URL pointing at the official wepage for the publisher.

4.3.2. User defined URL link frame

This frame is intended for URL [URL] links concerning the audiofile in a similar way to the other "W"xx frames. The frame body consists of a description of the string, represented as a terminated string, followed by the actual URL. The URL is always encoded with ISO-8859-1 [ISO-8859-1]. There may be more than one "WXX" frame in each tag, but only one with the same description.

User defined...	"WXX"
Frame size	$xx xx xx
Text encoding	$xx
Description	<textstring> $00 (00)
URL	<textstring>

4.4. Involved people list

Since there might be a lot of people contributing to an audio file in various ways, such as musicians and technicians, the 'Text information frames' are often insufficient to list everyone involved in a project. The 'Involved people list' is a frame containing the names of those involved, and how they were involved. The body simply contains a terminated string with the involvement directly followed by a terminated string with the involvee followed by a new involvement and so on. There may only be one "IPL" frame in each tag.

Involved people list	"IPL"
Frame size	$xx xx xx
Text encoding	$xx
People list strings	<textstrings>

4.5. Music CD Identifier

Unified Patents
Exhibit 1013
Page 13 of 23

This frame is intended for music that comes from a CD, so that the CD can be identified in databases such as the CDDB [CDDB]. The frame consists of a binary dump of the Table Of Contents, TOC, from the CD, which is a header of 4 bytes and then 8 bytes/track on the CD making a maximum of 804 bytes. This frame requires a present and valid "TRK" frame. There may only be one "MCI" frame in each tag.

```
Music CD identifier   "MCI"
Frame size            $xx xx xx
CD TOC                <binary data>
```

4.6. Event timing codes

This frame allows synchronisation with key events in a song or sound. The head is:

```
Event timing codes   "ETC"
Frame size           $xx xx xx
Time stamp format    $xx
```

Where time stamp format is:

```
$01  Absolute time, 32 bit sized, using MPEG [MPEG] frames as unit
$02  Absolute time, 32 bit sized, using milliseconds as unit
```

Abolute time means that every stamp contains the time from the beginning of the file.

Followed by a list of key events in the following format:

```
Type of event   $xx
Time stamp      $xx (xx ...)
```

The 'Time stamp' is set to zero if directly at the beginning of the sound or after the previous event. All events should be sorted in chronological order. The type of event is as follows:

```
$00  padding (has no meaning)
$01  end of initial silence
$02  intro start
$03  mainpart start
$04  outro start
$05  outro end
$06  verse begins
$07  refrain begins
$08  interlude
$09  theme start
$0A  variation
$0B  key change
$0C  time change
$0D  unwanted noise (Snap, Crackle & Pop)

$0E-$DF  reserved for future use

$E0-$EF  not predefined sync 0-F

$F0-$FC  reserved for future use

$FD  audio end (start of silence)
$FE  audio file ends
$FF  one more byte of events follows (all the following bytes with
     the value $FF have the same function)
```

The 'Not predefined sync's ($E0-EF) are for user events. You might want to synchronise your music to something, like setting of an explosion on-stage, turning on your screensaver etc.

There may only be one "ETC" frame in each tag.

4.7. MPEG location lookup table

To increase performance and accuracy of jumps within a MPEG [MPEG] audio file, frames with timecodes in different locations in the file might be useful. The ID3 frame includes references that the software can use to calculate positions in the file. After the frame header is a descriptor of how much the 'frame counter' should increase for every reference. If this value is two then the first reference points out the second frame, the 2nd reference the 4th frame, the 3rd reference the 6th frame etc. In a similar way the 'bytes between reference' and 'milliseconds between reference' points out bytes and milliseconds respectively.

Each reference consists of two parts; a certain number of bits, as defined in 'bits for bytes deviation', that describes the difference between what is said in 'bytes between reference' and the reality and a certain number of bits, as defined in 'bits for milliseconds deviation', that describes the difference between what is said in 'milliseconds between reference' and the reality. The number of bits in every reference, i.e. 'bits for bytes deviation'+'bits for milliseconds deviation', must be a multiple of four. There may only be one "MLL" frame in each tag.

Unified Patents
Exhibit 1013

```
  Location lookup table          "MLL"
  ID3 frame size                 $xx xx xx
  MPEG frames between reference  $xx xx
  Bytes between reference        $xx xx xx
  Milliseconds between reference $xx xx xx
  Bits for bytes deviation       $xx
  Bits for milliseconds dev.     $xx
```

Then for every reference the following data is included;

```
  Deviation in bytes         %xxx....
  Deviation in milliseconds  %xxx....
```

4.8. Synced tempo codes

For a more accurate description of the tempo of a musical piece this frame might be used. After the header follows one byte describing which time stamp format should be used. Then follows one or more tempo codes. Each tempo code consists of one tempo part and one time part. The tempo is in BPM described with one or two bytes. If the first byte has the value $FF, one more byte follows, which is added to the first giving a range from 2 - 510 BPM, since $00 and $01 is reserved. $00 is used to describe a beat-free time period, which is not the same as a music-free time period. $01 is used to indicate one single beat-stroke followed by a beat-free period.

The tempo descriptor is followed by a time stamp. Every time the tempo in the music changes, a tempo descriptor may indicate this for the player. All tempo descriptors should be sorted in chronological order. The first beat-stroke in a time-period is at the same time as the beat description occurs. There may only be one "STC" frame in each tag.

```
  Synced tempo codes  "STC"
  Frame size          $xx xx xx
  Time stamp format   $xx
  Tempo data          <binary data>
```

Where time stamp format is:

```
  $01  Absolute time, 32 bit sized, using MPEG [MPEG] frames as unit
  $02  Absolute time, 32 bit sized, using milliseconds as unit
```

Abolute time means that every stamp contains the time from the beginning of the file.

4.9. Unsychronised lyrics/text transcription

This frame contains the lyrics of the song or a text transcription of other vocal activities. The head includes an encoding descriptor and a content descriptor. The body consists of the actual text. The 'Content descriptor' is a terminated string. If no descriptor is entered, 'Content descriptor' is $00 (00) only. Newline characters are allowed in the text. Maximum length for the descriptor is 64 bytes. There may be more than one lyrics/text frame in each tag, but only one with the same language and content descriptor.

```
  Unsynced lyrics/text "ULT"
  Frame size           $xx xx xx
  Text encoding        $xx
  Language             $xx xx xx
  Content descriptor   <textstring> $00 (00)
  Lyrics/text          <textstring>
```

4.10. Synchronised lyrics/text

This is another way of incorporating the words, said or sung lyrics, in the audio file as text, this time, however, in sync with the audio. It might also be used to describing events e.g. occurring on a stage or on the screen in sync with the audio. The header includes a content descriptor, represented with as terminated textstring. If no descriptor is entered, 'Content descriptor' is $00 (00) only.

```
  Synced lyrics/text   "SLT"
  Frame size           $xx xx xx
  Text encoding        $xx
  Language             $xx xx xx
  Time stamp format    $xx
  Content type         $xx
  Content descriptor   <textstring> $00 (00)
```

```
Encoding:   $00  ISO-8859-1 [ISO-8859-1] character set is used => $00
                 is sync identifier.
            $01  Unicode [UNICODE] character set is used => $00 00 is
                 sync identifier.

Content type:   $00 is other
                $01 is lyrics
                $02 is text transcription
```


Unified Patents
Exhibit 1013

```
                  $03 is movement/part name (e.g. "Adagio")
                  $04 is events (e.g. "Don Quijote enters the stage")
                  $05 is chord (e.g. "Bb F Fsus")
```

Time stamp format is:

```
  $01  Absolute time, 32 bit sized, using MPEG [MPEG] frames as unit
  $02  Absolute time, 32 bit sized, using milliseconds as unit
```

Abolute time means that every stamp contains the time from the beginning of the file.

The text that follows the frame header differs from that of the unsynchronised lyrics/text transcription in one major way. Each syllable (or whatever size of text is considered to be convenient by the encoder) is a null terminated string followed by a time stamp denoting where in the sound file it belongs. Each sync thus has the following structure:

```
  Terminated text to be synced (typically a syllable)
  Sync identifier (terminator to above string)   $00 (00)
  Time stamp                                     $xx (xx ...)
```

The 'time stamp' is set to zero or the whole sync is omitted if located directly at the beginning of the sound. All time stamps should be sorted in chronological order. The sync can be considered as a validator of the subsequent string.

Newline characters are allowed in all "SLT" frames and should be used after every entry (name, event etc.) in a frame with the content type $03 - $04.

A few considerations regarding whitespace characters: Whitespace separating words should mark the beginning of a new word, thus occurring in front of the first syllable of a new word. This is also valid for new line characters. A syllable followed by a comma should not be broken apart with a sync (both the syllable and the comma should be before the sync).

An example: The "ULT" passage

```
  "Strangers in the night" $0A "Exchanging glances"
```

would be "SLT" encoded as:

```
  "Strang" $00 xx xx "ers" $00 xx xx " in" $00 xx xx " the" $00 xx xx
  " night" $00 xx xx 0A "Ex" $00 xx xx "chang" $00 xx xx "ing" $00 xx
  xx "glan" $00 xx xx "ces" $00 xx xx
```

There may be more than one "SLT" frame in each tag, but only one with the same language and content descriptor.

4.11. Comments

This frame replaces the old 30-character comment field in ID3v1. It consists of a frame head followed by encoding, language and content descriptors and is ended with the actual comment as a text string. Newline characters are allowed in the comment text string. There may be more than one comment frame in each tag, but only one with the same language and content descriptor.

```
  Comment                   "COM"
  Frame size                $xx xx xx
  Text encoding             $xx
  Language                  $xx xx xx
  Short content description <textstring> $00 (00)
  The actual text           <textstring>
```

4.12. Relative volume adjustment

This is a more subjective function than the previous ones. It allows the user to say how much he wants to increase/decrease the volume on each channel while the file is played. The purpose is to be able to align all files to a reference volume, so that you don't have to change the volume constantly. This frame may also be used to balance adjust the audio. If the volume peak levels are known then this could be described with the 'Peak volume right' and 'Peak volume left' field. If Peakvolume is not known these fields could be left zeroed or completely omitted. There may only be one "RVA" frame in each tag.

```
  Relative volume adjustment    "RVA"
  Frame size                    $xx xx xx
  Increment/decrement           %000000xx
  Bits used for volume descr.   $xx
  Relative volume change, right $xx xx (xx ...)
  Relative volume change, left  $xx xx (xx ...)
  Peak volume right             $xx xx (xx ...)
  Peak volume left              $xx xx (xx ...)
```

In the increment/decrement field bit 0 is used to indicate the right

Unified Patents
Exhibit 1013

channel and bit 1 is used to indicate the left channel. 1 is increment and 0 is decrement.

The 'bits used for volume description' field is normally $10 (16 bits) for MPEG 2 layer I, II and III [MPEG] and MPEG 2.5. This value may not be $00. The volume is always represented with whole bytes, padded in the beginning (highest bits) when 'bits used for volume description' is not a multiple of eight.

4.13. Equalisation

This is another subjective, alignment frame. It allows the user to predefine an equalisation curve within the audio file. There may only be one "EQU" frame in each tag.

Equalisation	"EQU"
Frame size	$xx xx xx
Adjustment bits	$xx

The 'adjustment bits' field defines the number of bits used for representation of the adjustment. This is normally $10 (16 bits) for MPEG 2 layer I, II and III [MPEG] and MPEG 2.5. This value may not be $00.

This is followed by 2 bytes + ('adjustment bits' rounded up to the nearest byte) for every equalisation band in the following format, giving a frequency range of 0 - 32767Hz:

Increment/decrement	%x (MSB of the Frequency)
Frequency	(lower 15 bits)
Adjustment	$xx (xx ...)

The increment/decrement bit is 1 for increment and 0 for decrement. The equalisation bands should be ordered increasingly with reference to frequency. All frequencies don't have to be declared. Adjustments with the value $00 should be omitted. A frequency should only be described once in the frame.

4.14. Reverb

Yet another subjective one. You may here adjust echoes of different kinds. Reverb left/right is the delay between every bounce in ms. Reverb bounces left/right is the number of bounces that should be made. $FF equals an infinite number of bounces. Feedback is the amount of volume that should be returned to the next echo bounce. $00 is 0%, $FF is 100%. If this value were $7F, there would be 50% volume reduction on the first bounce, yet 50% on the second and so on. Left to left means the sound from the left bounce to be played in the left speaker, while left to right means sound from the left bounce to be played in the right speaker.

'Premix left to right' is the amount of left sound to be mixed in the right before any reverb is applied, where $00 id 0% and $FF is 100%. 'Premix right to left' does the same thing, but right to left. Setting both premix to $FF would result in a mono output (if the reverb is applied symmetric). There may only be one "REV" frame in each tag.

Reverb settings	"REV"
Frame size	$00 00 0C
Reverb left (ms)	$xx xx
Reverb right (ms)	$xx xx
Reverb bounces, left	$xx
Reverb bounces, right	$xx
Reverb feedback, left to left	$xx
Reverb feedback, left to right	$xx
Reverb feedback, right to right	$xx
Reverb feedback, right to left	$xx
Premix left to right	$xx
Premix right to left	$xx

4.15. Attached picture

This frame contains a picture directly related to the audio file. Image format is preferably "PNG" [PNG] or "JPG" [JFIF]. Description is a short description of the picture, represented as a terminated textstring. The description has a maximum length of 64 characters, but may be empty. There may be several pictures attached to one file, each in their individual "PIC" frame, but only one with the same content descriptor. There may only be one picture with the picture type declared as picture type $01 and $02 respectively. There is a possibility to put only a link to the image file by using the 'image format' "-->" and having a complete URL [URL] instead of picture data. The use of linked files should however be used restrictively since there is the risk of separation of files.

Attached picture	"PIC"
Frame size	$xx xx xx
Text encoding	$xx
Image format	$xx xx xx
Picture type	$xx

Unified Patents
Exhibit 1013

```
  Description        <textstring> $00 (00)
  Picture data       <binary data>


Picture type:  $00  Other
               $01  32x32 pixels 'file icon' (PNG only)
               $02  Other file icon
               $03  Cover (front)
               $04  Cover (back)
               $05  Leaflet page
               $06  Media (e.g. lable side of CD)
               $07  Lead artist/lead performer/soloist
               $08  Artist/performer
               $09  Conductor
               $0A  Band/Orchestra
               $0B  Composer
               $0C  Lyricist/text writer
               $0D  Recording Location
               $0E  During recording
               $0F  During performance
               $10  Movie/video screen capture
               $11  A bright coloured fish
               $12  Illustration
               $13  Band/artist logotype
               $14  Publisher/Studio logotype
```

4.16. General encapsulated object

In this frame any type of file can be encapsulated. After the header, 'Frame size' and 'Encoding' follows 'MIME type' [MIME] and 'Filename' for the encapsulated object, both represented as terminated strings encoded with ISO 8859-1 [ISO-8859-1]. The filename is case sensitive. Then follows a content description as terminated string, encoded as 'Encoding'. The last thing in the frame is the actual object. The first two strings may be omitted, leaving only their terminations. MIME type is always an ISO-8859-1 text string. There may be more than one "GEO" frame in each tag, but only one with the same content descriptor.

```
  General encapsulated object "GEO"
  Frame size                  $xx xx xx
  Text encoding               $xx
  MIME type                   <textstring> $00
  Filename                    <textstring> $00 (00)
  Content description         <textstring> $00 (00)
  Encapsulated object         <binary data>
```

4.17. Play counter

This is simply a counter of the number of times a file has been played. The value is increased by one every time the file begins to play. There may only be one "CNT" frame in each tag. When the counter reaches all one's, one byte is inserted in front of the counter thus making the counter eight bits bigger. The counter must be at least 32-bits long to begin with.

```
  Play counter   "CNT"
  Frame size     $xx xx xx
  Counter        $xx xx xx xx (xx ...)
```

4.18. Popularimeter

The purpose of this frame is to specify how good an audio file is. Many interesting applications could be found to this frame such as a playlist that features better audiofiles more often than others or it could be used to profile a persons taste and find other 'good' files by comparing people's profiles. The frame is very simple. It contains the email address to the user, one rating byte and a four byte play counter, intended to be increased with one for every time the file is played. The email is a terminated string. The rating is 1-255 where 1 is worst and 255 is best. 0 is unknown. If no personal counter is wanted it may be omitted. When the counter reaches all one's, one byte is inserted in front of the counter thus making the counter eight bits bigger in the same away as the play counter ("CNT"). There may be more than one "POP" frame in each tag, but only one with the same email address.

```
  Popularimeter   "POP"
  Frame size      $xx xx xx
  Email to user   <textstring> $00
  Rating          $xx
  Counter         $xx xx xx xx (xx ...)
```

4.19. Recommended buffer size

Sometimes the server from which a audio file is streamed is aware of transmission or coding problems resulting in interruptions in the audio stream. In these cases, the size of the buffer can be recommended by the server using this frame. If the 'embedded info

Unified Patents
Exhibit 1013

flag' is true (1) then this indicates that an ID3 tag with the maximum size described in 'Buffer size' may occur in the audiostream. In such case the tag should reside between two MPEG [MPEG] frames, if the audio is MPEG encoded. If the position of the next tag is known, 'offset to next tag' may be used. The offset is calculated from the end of tag in which this frame resides to the first byte of the header in the next. This field may be omitted. Embedded tags is currently not recommended since this could render unpredictable behaviour from present software/hardware. The 'Buffer size' should be kept to a minimum. There may only be one "BUF" frame in each tag.

Recommended buffer size	"BUF"
Frame size	$xx xx xx
Buffer size	$xx xx xx
Embedded info flag	%0000000x
Offset to next tag	$xx xx xx xx

4.20. Encrypted meta frame

This frame contains one or more encrypted frames. This enables protection of copyrighted information such as pictures and text, that people might want to pay extra for. Since standardisation of such an encryption scheme is beyond this document, all "CRM" frames begin with a terminated string with a URL [URL] containing an email address, or a link to a location where an email adress can be found, that belongs to the organisation responsible for this specific encrypted meta frame.

Questions regarding the encrypted frame should be sent to the indicated email address. If a $00 is found directly after the 'Frame size', the whole frame should be ignored, and preferably be removed. The 'Owner identifier' is then followed by a short content description and explanation as to why it's encrypted. After the 'content/explanation' description, the actual encrypted block follows.

When an ID3v2 decoder encounters a "CRM" frame, it should send the datablock to the 'plugin' with the corresponding 'owner identifier' and expect to receive either a datablock with one or several ID3v2 frames after each other or an error. There may be more than one "CRM" frames in a tag, but only one with the same 'owner identifier'.

Encrypted meta frame	"CRM"
Frame size	$xx xx xx
Owner identifier	<textstring> $00 (00)
Content/explanation	<textstring> $00 (00)
Encrypted datablock	<binary data>

4.21. Audio encryption

This frame indicates if the actual audio stream is encrypted, and by whom. Since standardisation of such encrypion scheme is beyond this document, all "CRA" frames begin with a terminated string with a URL containing an email address, or a link to a location where an email address can be found, that belongs to the organisation responsible for this specific encrypted audio file. Questions regarding the encrypted audio should be sent to the email address specified. If a $00 is found directly after the 'Frame size' and the audiofile indeed is encrypted, the whole file may be considered useless.

After the 'Owner identifier', a pointer to an unencrypted part of the audio can be specified. The 'Preview start' and 'Preview length' is described in frames. If no part is unencrypted, these fields should be left zeroed. After the 'preview length' field follows optionally a datablock required for decryption of the audio. There may be more than one "CRA" frames in a tag, but only one with the same 'Owner identifier'.

Audio encryption	"CRA"
Frame size	$xx xx xx
Owner identifier	<textstring> $00 (00)
Preview start	$xx xx
Preview length	$xx xx
Encryption info	<binary data>

4.22. Linked information

To keep space waste as low as possible this frame may be used to link information from another ID3v2 tag that might reside in another audio file or alone in a binary file. It is recommended that this method is only used when the files are stored on a CD-ROM or other circumstances when the risk of file seperation is low. The frame contains a frame identifier, which is the frame that should be linked into this tag, a URL [URL] field, where a reference to the file where the frame is given, and additional ID data, if needed. Data should be retrieved from the first tag found in the file to which this link points. There may be more than one "LNK" frame in a tag, but only one with the same contents. A linked frame is to be considered as part of the tag and has the same restrictions as if it was a physical part of the tag (i.e. only one "REV" frame allowed, whether it's linked or not).

Unified Patents
Exhibit 1013

```
  Linked information  "LNK"
  Frame size          $xx xx xx
  Frame identifier    $xx xx xx
  URL                 <textstring> $00 (00)
  Additional ID data  <textstring(s)>
```

Frames that may be linked and need no additional data are "IPL", "MCI", "ETC", "LLT", "STC", "RVA", "EQU", "REV", "BUF", the text information frames and the URL link frames.

The "TXX", "PIC", "GEO", "CRM" and "CRA" frames may be linked with the content descriptor as additional ID data.

The "COM", "SLT" and "ULT" frames may be linked with three bytes of language descriptor directly followed by a content descriptor as additional ID data.

5. The 'unsynchronisation scheme'

The only purpose of the 'unsychronisation scheme' is to make the ID3v2 tag as compatible as possible with existing software. There is no use in 'unsynchronising' tags if the file is only to be processed by new software. Unsynchronisation may only be made with MPEG 2 layer I, II and III and MPEG 2.5 files.

Whenever a false synchronisation is found within the tag, one zeroed byte is inserted after the first false synchronisation byte. The format of a correct sync that should be altered by ID3 encoders is as follows:

```
%11111111 111xxxxx
```

And should be replaced with:

```
%11111111 00000000 111xxxxx
```

This has the side effect that all $FF 00 combinations have to be altered, so they won't be affected by the decoding process. Therefore all the $FF 00 combinations have to be replaced with the $FF 00 00 combination during the unsynchonisation.

To indicate usage of the unsynchronisation, the first bit in 'ID3 flags' should be set. This bit should only be set if the tag contained a, now corrected, false synchronisation. The bit should only be clear if the tag does not contain any false synchronisations.

Do bear in mind, that if a compression scheme is used by the encoder, the unsyncronisation scheme should be applied *afterwards*. When decoding a compressed, 'unsyncronised' file, the 'unsyncronisation scheme' should be parsed first, compression afterwards.

6. Copyright

Copyright (C) Martin Nilsson 1998. 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.

7. References

[CDDB] Compact Disc Data Base

<url:http://www.cddb.com>

[ISO-639-2] ISO/FDIS 639-2.
Codes for the representation of names of languages, Part 2: Alpha-3 code. Technical committee / subcommittee: TC 37 / SC 2

[ISO-8859-1] ISO/IEC DIS 8859-1.
8-bit single-byte coded graphic character sets, Part 1: Latin alphabet No. 1. Technical committee / subcommittee: JTC 1 / SC 2

[ISRC] ISO 3901:1986
International Standard Recording Code (ISRC).
Technical committee / subcommittee: TC 46 / SC 9

Unified Patents
Exhibit 1013
Page 20 of 23

[JFIF] JPEG File Interchange Format, version 1.02

<url:http://www.w3.org/Graphics/JPEG/jfif.txt>

[MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

<url:ftp://ftp.isi.edu/in-notes/rfc2045.txt>

[MPEG] ISO/IEC 11172-3:1993.
Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s, Part 3: Audio.
Technical committee / subcommittee: JTC 1 / SC 29
and
ISO/IEC 13818-3:1995
Generic coding of moving pictures and associated audio information, Part 3: Audio.
Technical committee / subcommittee: JTC 1 / SC 29
and
ISO/IEC DIS 13818-3
Generic coding of moving pictures and associated audio information, Part 3: Audio (Revision of ISO/IEC 13818-3:1995)

[PNG] Portable Network Graphics, version 1.0

<url:http://www.w3.org/TR/REC-png-multi.html>

[UNICODE] ISO/IEC 10646-1:1993.
Universal Multiple-Octet Coded Character Set (UCS), Part 1: Architecture and Basic Multilingual Plane. Technical committee / subcommittee: JTC 1 / SC 2

<url:http://www.unicode.org>

[URL] T. Berners-Lee, L. Masinter & M. McCahill, "Uniform Resource Locators (URL).", RFC 1738, December 1994.

<url:ftp://ftp.isi.edu/in-notes/rfc1738.txt>

8. Appendix

A. Appendix A - ID3-Tag Specification V1.1

ID3-Tag Specification V1.1 (12 dec 1997) by Michael Mutschler <amiga2@info2.rus.uni-stuttgart.de>, edited for space and clarity reasons.

A.1. Overview

The ID3-Tag is an information field for MPEG Layer 3 audio files. Since a standalone MP3 doesn't provide a method of storing other information than those directly needed for replay reasons, the ID3-tag was invented by Eric Kemp in 1996.

A revision from ID3v1 to ID3v1.1 was made by Michael Mutschler to support track number information is described in A.4.

A.2. ID3v1 Implementation

The Information is stored in the last 128 bytes of an MP3. The Tag has got the following fields, and the offsets given here, are from 0-127.

Field	Length	Offsets
Tag	3	0-2
Songname	30	3-32
Artist	30	33-62
Album	30	63-92
Year	4	93-96
Comment	30	97-126
Genre	1	127

The string-fields contain ASCII-data, coded in ISO-Latin 1 codepage. Strings which are smaller than the field length are padded with zero-bytes.

Tag: The tag is valid if this field contains the string "TAG". This has to be uppercase!

Songname: This field contains the title of the MP3 (string as above).

Artist: This field contains the artist of the MP3 (string as above).

Album: this field contains the album where the MP3 comes from (string as above).

Unified Patents
Exhibit 1013
Page 21 of 23

```
      Year: this field contains the year when this song has originally
         been released (string as above).

      Comment: this field contains a comment for the MP3 (string as
         above). Revision to this field has been made in ID3v1.1. See
         A.4.

      Genre: this byte contains the offset of a genre in a predefined
         list the byte is treated as an unsigned byte. The offset is
         starting from 0. See A.3.


A.3.   Genre List

   The following genres is defined in ID3v1

       0.Blues
       1.Classic Rock
       2.Country
       3.Dance
       4.Disco
       5.Funk
       6.Grunge
       7.Hip-Hop
       8.Jazz
       9.Metal
      10.New Age
      11.Oldies
      12.Other
      13.Pop
      14.R&B
      15.Rap
      16.Reggae
      17.Rock
      18.Techno
      19.Industrial
      20.Alternative
      21.Ska
      22.Death Metal
      23.Pranks
      24.Soundtrack
      25.Euro-Techno
      26.Ambient
      27.Trip-Hop
      28.Vocal
      29.Jazz+Funk
      30.Fusion
      31.Trance
      32.Classical
      33.Instrumental
      34.Acid
      35.House
      36.Game
      37.Sound Clip
      38.Gospel
      39.Noise
      40.AlternRock
      41.Bass
      42.Soul
      43.Punk
      44.Space
      45.Meditative
      46.Instrumental Pop
      47.Instrumental Rock
      48.Ethnic
      49.Gothic
      50.Darkwave
      51.Techno-Industrial
      52.Electronic
      53.Pop-Folk
      54.Eurodance
      55.Dream
      56.Southern Rock
      57.Comedy
      58.Cult
      59.Gangsta
      60.Top 40
      61.Christian Rap
      62.Pop/Funk
      63.Jungle
      64.Native American
      65.Cabaret
      66.New Wave
      67.Psychadelic
      68.Rave
      69.Showtunes
      70.Trailer
      71.Lo-Fi
      72.Tribal
      73.Acid Punk
      74.Acid Jazz
      75.Polka
```

Unified Patents
Exhibit 1013

```
     76.Retro
     77.Musical
     78.Rock & Roll
     79.Hard Rock

   The following genres are Winamp extensions

     80.Folk
     81.Folk-Rock
     82.National Folk
     83.Swing
     84.Fast Fusion
     85.Bebob
     86.Latin
     87.Revival
     88.Celtic
     89.Bluegrass
     90.Avantgarde
     91.Gothic Rock
     92.Progressive Rock
     93.Psychedelic Rock
     94.Symphonic Rock
     95.Slow Rock
     96.Big Band
     97.Chorus
     98.Easy Listening
     99.Acoustic
    100.Humour
    101.Speech
    102.Chanson
    103.Opera
    104.Chamber Music
    105.Sonata
    106.Symphony
    107.Booty Bass
    108.Primus
    109.Porn Groove
    110.Satire
    111.Slow Jam
    112.Club
    113.Tango
    114.Samba
    115.Folklore
    116.Ballad
    117.Power Ballad
    118.Rhythmic Soul
    119.Freestyle
    120.Duet
    121.Punk Rock
    122.Drum Solo
    123.A capella
    124.Euro-House
    125.Dance Hall
```

A.4. Track addition - ID3v1.1

In ID3v1.1, Michael Mutschler revised the specification of the comment field in order to implement the track number. The new format of the comment field is a 28 character string followed by a mandatory null ($00) character and the original album tracknumber stored as an unsigned byte-size integer. In such cases where the 29th byte is not the null character or when the 30th is a null character, the tracknumber is to be considered undefined.

9. Author's Address

Martin Nilsson
Rydsvägen 246 C. 30
S-584 34 Linköping
Sweden

Email: nilsson@id3.org

Co-authors:

Johan Sundström Email: johan@id3.org