HD Radio™ Air Interface Design Description Program Service Data

Rev. E

December 14, 2016

TRADEMARKS

HD Radio™ and the HD, HD Radio, and “Arc” logos are proprietary trademarks of iBiquity Digital Corporation.

“iBiquity”, "iBiquity Digital", and the iBiquity Digital logo are also proprietary trademarks of iBiquity.

All other trademarks, whether claimed or registered, are the exclusive property of their respective owners.

iBiquity Digital Corporation
6711 Columbia Gateway Drive, Suite 500
Columbia, MD 21046
Voice: 443-539-4290
Fax: 443-539-4291
E-mail address:
info@ibiquity.com

Doc. No.: SY_IDD_1028s

Table of Contents

Contents

List of Figures

List of Tables

1 Scope

1.1 System Overview

The iBiquity Digital Corporation HD Radio™ system is designed to permit a smooth evolution from current analog amplitude modulation (AM) and frequency modulation (FM) radio to a fully digital in-band on-channel (IBOC) system. This system delivers digital audio and data services to mobile, portable, and fixed receivers from terrestrial transmitters in the existing medium frequency (MF) and very high frequency (VHF) radio bands. Broadcasters may continue to transmit analog AM and FM simultaneously with the new, higher-quality, and more robust digital signals, allowing themselves and their listeners to convert from analog to digital radio while maintaining their current frequency allocations.

1.2 Document Overview

This document provides a description of the Program Service Data capability. Program Service Data refers to both Main Program Service Data (MPSD) and Supplemental Program Service Data (SPSD). A detailed description of ID3v2.3.0 message encoding is explained in Appendix A. Appendix A contains the ID3 standard as implemented in the HD Radio system and the relevant sections of the specifications.

Refer to Reference [NRSC-G200-A] for guidelines on harmonizing PSD data fields with RBDS/RDS transmissions.

2 Referenced Documents

STATEMENT
Each referenced document that is mentioned in this document shall be listed in the following iBiquity document:

- Reference Documents for the NRSC In-Band/On-Channel Digital Radio Broadcasting Standard Document Number: SY_REF_2690s

3 Abbreviations, Acronyms, and Conventions

3.1 Abbreviations and Acronyms

AAS	Advanced Application Services
AM	Amplitude Modulation
API	Application Programming Interface
CODEC	Coder Decoder
COMM	Comment (ID3 Frame ID)
COMR	Commercial (ID3 Frame ID)
FM	Frequency Modulation
HD RLS	HD Radio Link Subsystem
IBOC	In-Band On-Channel
ID	Identification
MF	Medium Frequency
MPEG	Motion Picture Experts Group
MPS	Main Program Service
MPSA	Main Program Service Audio
MPSD	Main Program Service Data
PDU	Protocol Data Unit
PSD	Program Service Data
REFID	Reference Identifier
SIS	Station Information Service
SPS	Supplemental Program Service
SPSA	Supplemental Program Service Audio
SPSD	Supplemental Program Service Data
TALB	Album (ID3 Frame ID)
TCON	Genre (ID3 Frame ID)
TIT2	Title (ID3 Frame ID)
TPE1	Artist (ID3 Frame ID)
UFID	Unique File Identifier (ID3 Frame ID)
URL	Uniform Resource Locator
VHF	Very High Frequency

3.2 Presentation Conventions

Unless otherwise noted, the following conventions apply to this document:

- All vectors are indexed starting with 0.
- The element of a vector with the lowest index is considered to be first.
- In drawings and tables, the leftmost bit is considered to occur first in time.
- Bit 0 of a byte or word is considered the least significant bit.
- When presenting the dimensions of a matrix, the number of rows is given first (e.g., an n x m matrix has n rows and m columns).
- In timing diagrams, earliest time is on the left.
- Binary numbers are presented with the most significant bit having the highest index.
- In representations of binary numbers, the least significant bit is on the right.

4 Overview

The Main Program Service (MPS) allows the transmission of existing analog radio programming in both analog and digital formats. MPS includes the Main Program Service Audio (MPSA) and the Main Program Service Data (MPSD). Similarly, the Supplemental Program Service Data includes the Supplemental Program Service Audio (SPSA) and the Supplemental Program Service Data (SPSD). Both MPSD and SPSD are generally referred to as Program Service Data (PSD). PSD provides additional information about the audio. The processing of the Supplemental Program Service Data (SPS Data) is exactly the same as the MPS Data.

The main program audio and data are synchronized at the broadcast studio. That is, the PSD is transmitted so that radio receivers can acquire it and present it while its associated audio program is being heard by radio listeners.

4.1 Program Service Data Flow

Figure 4-1 illustrates the PSD flow through the HD Radio Broadcast System. The PSD is sent through the Service Interface as ID3 tags which are transmitted as data packets. The PSD is processed by the HD RLS mechanism within the PSD Transport [10] for framing and encapsulation as byte streams and then inserted into the audio streams by the Audio Transport as PSD PDUs (MPSD/SPSD PDUs). The Audio Transport [4] then multiplexes the PSD PDUs with the audio streams to generate MPS/SPS PDUs.

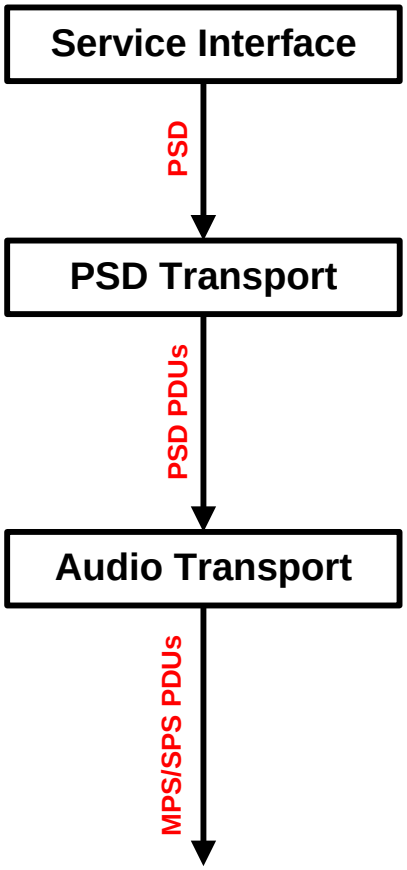

Figure 4-1: Program Service Data Flow

5 Program Service Data Description

Program Service Data (PSD) is transmitted along with the program audio. PSD is intended to describe or complement the audio program heard by the radio listener.

The following subsections provide:

- An introduction to the basic PSD content
- A description of broadcast PSD processing
- The format of PSD messages

5.1 Basic PSD Content

PSD consists of a general set of categories that describe the various programming content, such as a song, talk show, advertisement, or announcement. For example, the Title field can be used to describe the name of a song, topic of a talk show, advertisement, or announcement.

The PSD fields include the following:

- Title
- Artist
- Album
- Genre
- Comment
- Commercial
- Reference Identifier

A detailed description for the format of PSD messages is discussed in Subsection 5.3.

5.2 Broadcast PSD Processing

PSD can originate from a studio automation system or any other computing resource where program audio originates. Regardless of the source, the processing and interface to facilitate broadcast of PSD is consistent. Program Service Data providers input the desired content (for example, title, artist, album, etc.) and transfer the resulting PSD message to the Service Interface. The PSD data packets then undergo framing and encapsulation by the HD RLS as byte streams within the PSD Transport before being sent as PSD PDUs to the Audio Transport where they are multiplexed with the audio streams.

Some key considerations for Program Service Data are:

- Typical PSD messages contain less than 30 characters per field.
- PSD messages are continuously transmitted, with the most recent message transmitted repeatedly.
- PSD providers send a new PSD message only when the PSD content has changed. Note that radio receivers will continue to display information for a given ID3 frame until one of the following things occur:
    - A new ID3 tag is received with that particular frame omitted. In this case, the receiver shall clear the text for that frame.
    - A new ID3 tag is received with different text in that particular frame. In this case, the receiver shall display the updated text for that frame.
    - The signal is lost for an appreciable amount of time. In this case, all ID3 tag information shall be cleared by the receiver.

5.3 Format of PSD Messages

PSD is formatted using a subset of the standard called ID3v2.3.0 (Reference [25]). Historically, ID3 has been used to allow textual information, such as artist, title, and genre information, to co-exist within MPEG-3 (MP3) audio files. The HD Radio system uses ID3 to deliver Program Service Data along with real-time broadcast audio.

The HD Radio system implements a specific subset of the ID3v2.3.0 parameters. The ID3v2.3.0 general structure is as follows:

- A complete ID3 message is called an ID3 tag.
- ID3 tags contain one or more content types referred to as frames. Frames contain individual pieces of information (for example, song, artist, title, etc.). Each frame has a four-character identifier. For example, the commercial frame is identified as COMR.
- Within frames, sub-elements called fields can exist. Fields further categorize the information within a frame. For example, the commercial frame has a field to specify sale price.

Figure 5-1 shows the general ID3 message structure.

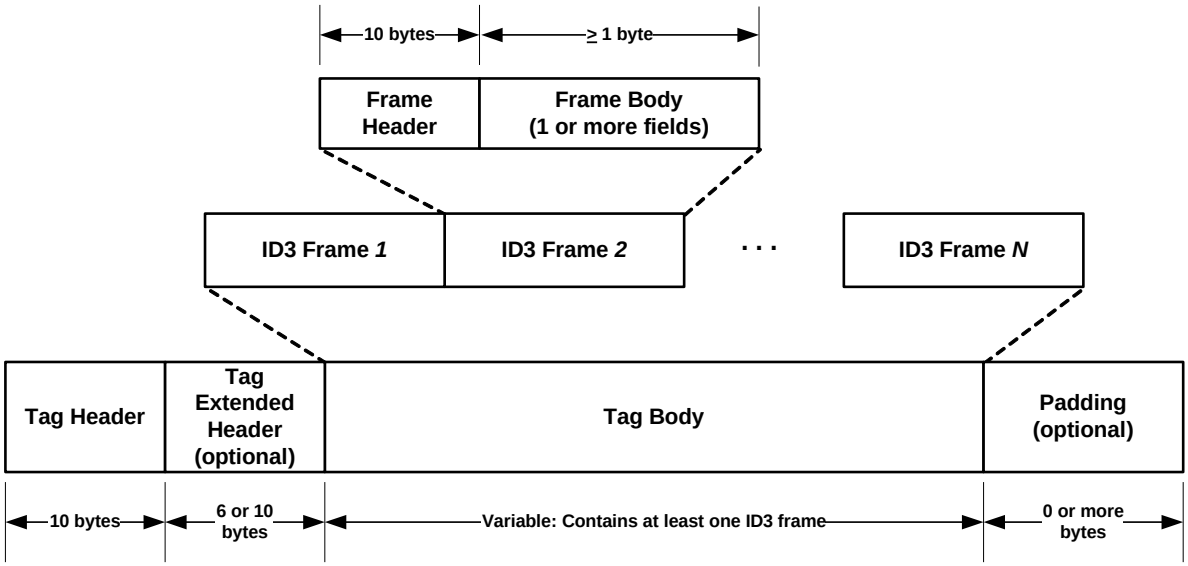

Figure 5-1: General ID3 Message Structure

Appendix A contains the ID3 standard as implemented in the HD Radio system and the relevant sections of the specification.

Table 5-1 gives a description of ID3 frames used for PSD. For a detailed description of ID3v2.3.0 message encoding used for PSD, see Appendix A.

Table 5-1: ID3 Frames Supported by PSD

	PSD	ID3	ID3	Description	Type		
	Attribute	**Frame ID**	**Field(s)**		**Music**	**Talk**	**Announcement**
1	Title	TIT2	Info	One-line Title Name	Song title	Talk Topic	Announcement or Advertisement Title
2	Artist	TPE1	Info	Performer, Originator, Author, Sponsor	Artist Name	Show Host	Author/Sponsor
3	Album	TALB	Info	Content Source	Album Name	Show Name	Sponsor Name
4	Genre	TCON	Info	Categorization of content This is an enumerated field of predefined types	(8) Jazz (17) Rock (32) Classical	(101) Speech	(101) Speech
5	Comment	COMM	Language	3-byte language code per ISO 639-2	3 byte code	3-byte code	3-byte code
			Short description field	One-line Title for Comment Description	Comment Title (or complete comment if Content field is omitted)	Comment Title (or complete comment if Content field is omitted)	Comment Title (or complete comment if Content field is omitted)

	PSD	ID3	ID3	Description	Type		
	Attribute	**Frame ID**	**Field(s)**		**Music**	**Talk**	**Announcement**
5	Comment	COMM	Content field	Comment Description Detailed explanation, user callback information or further information	Further information about the song such as recording date, artist bio info. Announcements of general interest such as the weather, concert dates.	Talk Show call-in number, or other show info Announcements of general interest such as the weather, concert dates.	Announcement or Advertising statement Point-of-sale or more info Announcements of general interest such as the weather, concert dates.
6	Commercial	COMR	Price	Price of merchandise	The commercial frame facilitates sale of products and services (Note: Binary pictures are not supported. Any images shall be sent separately via the AAS or LOT data transports.)		
			Valid until	Expiration data for transaction			
			Contact URL	URL identifier used to contact the seller Can be used to initiate purchase transaction via an external return channel, such as a cellular phone network			
			Received as	Method in which merchandise is received (e.g., over the internet)			
6	Commercial	COMR	Name of seller	Text identifying seller			
			Description	Textual description of advertisement			
			Picture	Picture of advertised item This item is not supported and shall not be broadcast.			
			Seller Logo	Binary graphic of seller logo This item is not supported and shall not be broadcast.			

	PSD	ID3	ID3	Description	Type		
	Attribute	**Frame ID**	**Field(s)**		**Music**	**Talk**	**Announcement**
7	Reference Identifier	UFID	Owner Identifier	Typically a URL that may be used to connect with the audio content in some way.	---		
			Identifier	Unique Identifier	An identifier code, in either binary or text format of up to 64 bytes in length.		

6 Requirements for Program Service Data

Some key requirements for Program Service Data are:

- Program audio and associated data shall be synchronized as tightly as possible so that receivers will render them both at the same time.
- PSD messages shall not exceed 1024 bytes, including HD RLS overhead as described in [5]. Therefore, ID3 tags shall be limited to no larger than 1,018 bytes.
- PSD shall utilize only the subset of the ID3v2.3.0 standard as shown in Table 5-1.
- PSD is not compatible with later versions of ID3; such as v2.4.0.
- Broadcasters providing PSD shall, at a minimum, transmit the Title and Artist information.
    - If dynamic program Info is not available, only the Song Title field shall be populated with static information. The Artist field shall be left empty.
- Title, Artist, Album, and Genre frames shall be limited to less than 128 characters (excluding the frame header).
- Comment, Commercial, and Reference Identifier frames shall be any length as long as the maximum ID3 tag size of 1,018 characters is not exceeded.
- ID3 frames shall not contain empty strings or null strings (single 0x00 byte) after the frame header. Frames should not be sent unless they contain at least one displayable character.
- The text encoding byte that is specified in the ID3 specification is required for all text-based frames. A value of 0x00 indicates ISO-8859-1 encoding and is highly recommended if possible. A value of 0x01 indicates Unicode, which is a much richer character set, but may not be supported by all receivers.
- All ID3 tags shall start with the characters "ID3" followed by version bytes of 0x03 0x00. Other versions are not supported.

7 ID3 Tag Implementation Considerations

This section provides operational constraints that should be applied to ID3 tag generation in order to ensure successful ID3 tag decoding and display by radio receivers.

The following features defined by the ID3 specification are not supported by the HD Radio system. Refer to Appendix A for details.

- Padding bytes are not supported
- Unsychronization is not supported. ID3 tags are always byte-aligned by the PSD data transport mechanism. No furteh rsynchronization is needed.
- CRC is not supported. The PSD data transport already includes its own CRC capability.
- Data compression is not supported. The maximum tag and frame size constraints stated in Section 6 shall always apply. In addition, it is suggested to keep ID3 tags and frames well below these limits in order to minimize transmission latency. For example, reference [G-200A] provides recommendations on reduced frame sizes in order to maintain compatibility with equivalent frames sent via RBDS.
- Data encryption at the ID3 frame or tag level is not supported.

7.1 Genre Frames

Genre should be considered non-displayable by radio receivers. The genre frame is intended to identify the specific genre of each individual song. The genre codes, as defined by the ID3 standard are not compatible with the program type codes used in the SIS transport, which could confuse the radio user. However, Genre could possibly be useful in supporting other receiver functions such as scanning or in prompting a receiver to display a genre-based image on the screen.

It is highly recommended to either omit this field or else just broadcast the genre code without additional supporting text.

7.2 Comment Frames

The comment filed is composed of three individual fields: language, short description, and the content field. This may be too complex for some radio displays to have both a short description and a content text field. If a Comment frame is broadcast, the language and short description fields are required to be populated with information. The content field may be a null string (single 0x00 byte) if desired.

7.3 Commercial Frames

Commercial frames are fairly complex and are composed of several individual fields. If a Commercial frame is broadcast, the following rules shall be applied:

- Price string may be a null string (single 0x00 byte). If it is not null, the 3-character currency code is required. After the currency code, the only allowable characters are 0 to 9 and a decimal point. However, a decimal point is not required.
- Valid Until is required. It must be an 8-character date string in the format YYYYMMDD. If a valid date is not desired to be sent, for example, if a date is not applicable to the other content, a value of "00000000" may be used.
- Contact URL may be a null string (single 0x00 byte)
- Received as byte is required: can be any value
- Name of seller may be a null string
- Description may be a null string
- At least one of URL, Seller, and Description shall contain at least one displayable character. It is acceptable for a maximum of two of those fields to be null strings.

7.4 UFID Frames

If a UFID frame is broadcast, it is required to contain information in the Owner Identifier field. The Identifier data field may be empty if it is not needed.

Appendix A

Reference | ID3 web site: http://id3.org/id3v2.3.00

Appendix A provides an annotated version of the ID3 V2.3.0 Standard.

Appendix A shows the portions that are supported by the HD Radio System, not supported by the HD Radio System, and omitted.

ID3 Standard Reference for HD Radio Program Service Data

Informal standard M. Nilsson

Document: id3v2.3.0.txt 3rd February 1999

ID3 tag version 2.3.0

Status of this document

This document is an informal standard and replaces the ID3v2.2.0 standard [ID3v2]. The informal standard is released so that implementers could have a set standard before a formal standard is set. The formal standard will use another version or revision 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 functionality.

Distribution of this document is unlimited.

Abstract

This document describes the ID3v2.3.0, which is a more developed version of the ID3v2 informal standard [ID3v2] (version 2.2.0), evolved from the ID3 tagging system. The ID3v2 offers a flexible way of storing information about an audio file within itself to determine its origin and contents. The information may be technical information, such as equalization curves, as well as related meta information, such as title, performer, copyright etc.

7. References
8. Genre List from ID3v1
9. **omitted**

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 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 as flexible and expandable as possible.

The first criterion is met by the simple fact that the MPEG [MPEG] decoding software uses a sync signal, embedded in the audio stream, to 'lock on to' the audio. Since the ID3v2 tag doesn't contain a valid sync signal, no software will attempt to play the tag. If, for any reason, coincidence make a sync signal 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' 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 at files encoded with MPEG-1/2 layer I, MPEG-1/2 layer II, MPEG-1/2 layer III and MPEG-2.5, but may work with other types of encoded audio.

The bit order in ID3v2 is most significant bit first (MSB). The byte order in multi-byte 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:

ID3v2/file identifier	"ID3"
ID3v2 version	$03 00
ID3v2 flags	%abc00000
ID3v2 size	4 * %0xxxxxxx

The first three bytes of the tag are always "ID3" to indicate that this is an ID3v2 tag, directly followed by the two version bytes. The first byte of ID3v2 version is it's major version, while the second byte is its revision number. In this case this is ID3v2.3.0. All revisions are backwards compatible while major

versions are not. If software with ID3v2.2.0 and below support should encounter version three or higher it should simply ignore the whole tag. Version and revision will never be $FF.

The version is followed by one the ID3v2 flags field, of which currently only three flags are used.

a - Unsynchronisation **Not supported. Must be 0.**

Bit 7 in the 'ID3v2 flags' indicates whether or not unsynchronisation is used (see section 5 for details); a set bit indicates usage.

b - Extended header

The second bit (bit 6) indicates whether or not the header is followed by an extended header. The extended header is described in section 3.2.

c - Experimental indicator

The third bit (bit 5) should be used as an 'experimental indicator'. This flag should always be set when the tag is in an experimental stage.

All the other flags should be cleared. If one of these undefined flags are set that might mean that the tag is not readable for a parser that does not know the flags function.

The ID3v2 tag size is encoded with four bytes where the most significant 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 ID3v2 tag size is the size of the complete tag after unsynchronisation, including padding, excluding the header but not excluding the extended header (total tag size - 10). Only 28 bits (representing up to 256MB) are used in the size description to avoid the introduction of 'false sync signals'.

An 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 extended header

The extended header contains information that is not vital to the correct parsing of the tag information, hence the extended header is optional.

Extended header size $xx xx xx xx

Extended Flags $xx xx

Size of padding $xx xx xx xx **Not supported. Must be $00 00 00 00**

Where the 'Extended header size', currently 6 or 10 bytes, excludes itself. The 'Size of padding' is simply the total tag size excluding the frames and the headers, in other words the padding. The extended header is considered separate from the header proper, and as such is subject to unsynchronisation.

The extended flags are a secondary flag set which describes further attributes of the tag. These attributes are currently defined as follows

%x0000000 00000000

x - CRC data present **Not Supported**

If this flag is set four bytes of CRC-32 data is appended to the extended header. The CRC should be calculated before unsynchronisation on the data between the extended header and the padding, i.e. the frames and only the frames.

Total frame CRC $xx xx xx xx

3.3. ID3v2 frame overview

As the tag consists of a tag header and a tag body with one or more frames, all the frames consists of a frame header followed by one or more fields containing the actual information. The layout of the frame header:

Frame ID $xx xx xx xx (four characters)

Size $xx xx xx xx

Flags $xx xx

The frame ID made out of the characters capital A-Z and 0-9. Identifiers beginning with "X", "Y" and "Z" are for experimental use and free for everyone to use, without the need to set the experimental bit in the tag header. Have in mind that someone else might have used the same identifier as you. All other identifiers are either used or reserved for future use.

The frame ID is followed by a size descriptor, making a total header size of ten bytes in every frame. The size is calculated as frame size excluding frame header (frame size - 10).

In the frame header the size descriptor is followed by two flags bytes. These flags are described in section 3.3.1.

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: UFID, TIT2, MCDI, TRCK ...

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

If nothing else is said a string is represented as ISO-8859-1 [ISO-8859-1] characters in the range $20 - $FF. Such strings are represented as <text string>, or <full text string> if newlines are allowed, in the frame descriptions. All Unicode strings [UNICODE] use 16-bit Unicode 2.0 (ISO/IEC 10646-1:1993, UCS-2). Unicode strings must begin with the Unicode BOM ($FF FE or $FE FF) to identify the byte order.

All numeric strings and URLs [URL] 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 $01. Strings dependent on encoding is represented as <text string according to encoding>, or <full text string according to encoding> if newlines are allowed. Any empty Unicode strings which are NULL-terminated may have the Unicode BOM followed by a Unicode NULL ($FF FE 00 00 or $FE FF 00 00).

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 ID3v2 standard. This is reflected by the revision number in the header of the tag.

3.3.1. Frame header flags

In the frame header the size descriptor is followed by two flags bytes. All unused flags must be cleared. The first byte is for 'status messages' and the second byte is for encoding purposes. If an unknown flag is set in the first byte the frame may not be changed without the bit cleared. If an unknown flag is set in the second byte it is likely to not be readable. The flags field is defined as follows.

%abc00000 %ijk00000	**Default value = %00000000 %00000000. However, these bits shall be considered reserved for future applications.**

a - Tag alter preservation **Not supported**

This flag tells the software what to do with this frame if it is unknown and the tag is altered in any way. This applies to all kinds of alterations, including adding more padding and reordering the frames.

0 Frame should be preserved.

1 Frame should be discarded.

b - File alter preservation **Not supported**

This flag tells the software what to do with this frame if it is unknown and the file, excluding the tag, is altered. This does not apply when the audio is completely replaced with other audio data.

0 Frame should be preserved.

1 Frame should be discarded.

c - Read only Reserved **Not supported**

This bit is reserved. It may be set or cleared by external subsystems and should therefore be ignored.

i - Compression **Not supported**

This flag indicates whether or not the frame is compressed.

0 Frame is not compressed.

1 Frame is compressed using zlib [zlib] with 4 bytes for 'decompressed size' appended to the frame header.

j - Encryption **Not supported**

This flag indicates whether or not the frame is encrypted. If set one byte indicating with which method it was encrypted will be appended to the frame header. See section 4.26 for more information about encryption method registration.

0 Frame is not encrypted.

1 Frame is encrypted.

k - Grouping identity **Not supported**

This flag indicates whether or not this frame belongs in a group with other frames. If set a group identifier byte is added to the frame header. Every frame with the same group identifier belongs to the same group.

0 Frame does not contain group information

1 Frame contains group information

Some flags indicates that the frame header is extended with additional information. This information will be added to the frame header in the same order as the flags indicating the additions. I.e. the four bytes of decompressed size will precede the encryption method byte. These additions to the frame header, while not included in the frame header size but are included in the 'frame size' field, are not subject to encryption or compression.

3.3.2. Default flags **Omitted**

The default settings for the frames described in this document can be divided into the following classes. The flags may be set differently if found more suitable by the software.

1. Discarded if tag is altered, discarded if file is altered.

   None.

2. Discarded if tag is altered, preserved if file is altered.

   None.

3. Preserved if tag is altered, discarded if file is altered.

   AENC, ETCO, EQUA, MLLT, POSS, SYLT, SYTC, RVAD, TENC, TLEN, TSIZ

4. Preserved if tag is altered, preserved if file is altered.

   The rest of the frames.

4. This section has been omitted.

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 standardization 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 organization 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. The string "http://www.id3.org/dummy/ufid.html" should be used for tests. Software that isn't told otherwise may safely remove such frames. The 'Owner identifier' must be non-empty (more than just a termination). The 'Owner identifier' is then followed by the actual identifier, which may be up to 64 bytes. There may be more than one "UFID" frame in a tag, but only one with the same 'Owner identifier'.

<Header for 'Unique file identifier', ID: "UFID">

Owner identifier <text string> $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 text string is followed by a termination ($00 (00)) all the following information should be ignored and not be displayed. All text frame identifiers begin with "T". Only text frame identifiers begin with "T", with the exception of the "TXXX" frame. All the text information frames have the following format:

<Header for 'Text information frame', ID: "T000" - "TZZZ",

excluding "TXXX" described in 4.2.2.>

Text encoding $xx

Information <text string according to encoding>

4.2.1. Text information frames - details

This section only includes frames supported in MPS.

TALB

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.

TCON

The 'Content type', which previously 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 8.1, Appendix A.) 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 are defined in ID3v2 and are implemented in the same way as the numeric content types, e.g. "(RX)".

RX Remix

CR Cover

TIT2

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

TPE1

The 'Lead artist(s)/Lead performer(s)/Soloist(s)/Performing group' is

used for the main artist(s). They are separated with the "/"

character.

4.2.2. This section has been omitted.

4.3 thru 4.10 omitted

4.11. Comments

This frame is intended for any kind of full text information that does not fit in any other frame. It consists of a frame header 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.

<Header for 'Comment', ID: "COMM">

Text encoding $xx

Language $xx xx xx

Short content descrip. <text string according to encoding> $00 (00)

The actual text <full text string according to encoding>

4.12 thru 4.24 omitted

4.25. Commercial frame

This frame enables several competing offers in the same tag by bundling all needed information. That makes this frame rather complex but it's an easier solution than if one tries to achieve the same result with several frames. The frame begins, after the frame ID, size and encoding fields, with a price string field. A price is constructed by one three character currency code, encoded according to ISO 4217 [ISO-4217] alphabetic currency code, followed by a numerical value where "." is used as decimal separator. In the price string several prices may be concatenated, separated by a "/" character, but there may only be one currency of each type.

The price string is followed by an 8 character date string in the format YYYYMMDD, describing for how long the price is valid. After that is a contact URL, with which the user can contact the seller, followed by a one byte 'received as' field. It describes how the audio is delivered when bought according to the following list:

$00 Other

$01 Standard CD album with other songs

$02 Compressed audio on CD

$03 File over the Internet

$04 Stream over the Internet

$05 As note sheets

$06 As note sheets in a book with other sheets

$07 Music on other media

$08 Non-musical merchandise

Next follows a terminated string with the name of the seller followed by a terminated string with a short description of the product. The last thing is the ability to include a company logotype. The first of them is the 'Picture MIME type' field containing information about which picture format is used. In the event that the MIME media type name is omitted, "image/" will be implied. Currently only "image/png" and "image/jpeg" are allowed. This format string is followed by the binary picture data. This two last fields may be omitted if no picture is to attach.

<Header for 'Commercial frame', ID: "COMR">

Text encoding $xx

Price string <text string> $00

Valid until <text string>

Contact URL <text string> $00

Received as $xx

Name of seller <text string according to encoding> $00 (00)

Description <text string according to encoding> $00 (00)

Picture MIME type <string> $00	**Not Supported**
Seller logo <binary data>	**Not Supported**

4.26. Encryption method registration

To identify with which method a frame has been encrypted the encryption method must be registered in the tag with this frame. The 'Owner identifier' is 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 organization responsible for this specific encryption method. Questions regarding the encryption method should be sent to the indicated email address. The 'Method symbol' contains a value that is associated with this method throughout the whole tag. Values below $80 are reserved. The 'Method symbol' may optionally be followed by encryption specific data. There may be several "ENCR" frames in a tag but only one containing the same symbol and only one containing the same owner identifier. The method must be used somewhere in the tag. See section 3.3.1, flag j for more information.

<Header for 'Encryption method registration', ID: "ENCR">

Owner identifier <text string> $00

Method symbol $xx

Encryption data <binary data>

4.27 thru 4.28 omitted

5. The 'unsynchronisation scheme' **Omitted**

The only purpose of the 'unsynchronisation 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 synchronization is found within the tag, one zeroed byte is inserted after the first false synchronization 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 unsynchronisation.

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

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

If the last byte in the tag is $FF, and there is a need to eliminate false synchronizations in the tag, at least one byte of padding should be added.

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>

[ID3v2] Martin Nilsson, "ID3v2 informal standard".
<url:http://www.id3.org/id3v2-00.txt>

[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-4217] ISO 4217:1995.
Codes for the representation of currencies and funds. Technical committee / subcommittee: TC 68

[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

[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>

[ZLIB] P. Deutsch, Aladdin Enterprises & J-L. Gailly, "ZLIB Compressed Data Format Specification version 3.3", RFC 1950, May 1996.
<url:ftp://ftp.isi.edu/in-notes/rfc1950.txt>

8. Appendix

8.1 Appendix A – Genre List from ID3v1

The following genres is defined in ID3v1		The following genres are Winamp extensions
0.Blues	46.Instrumental Pop	80.Folk
1.Classic Rock	47.Instrumental Rock	81.Folk-Rock
2.Country	48.Ethnic	82.National Folk
3.Dance	49.Gothic	83.Swing
4.Disco	50.Darkwave	84.Fast Fusion
5.Funk	51.Techno-Industrial	85.Bebob
6.Grunge	52.Electronic	86.Latin
7.Hip-Hop	53.Pop-Folk	87.Revival
8.Jazz	54.Eurodance	88.Celtic
9.Metal	55.Dream	89.Bluegrass
10.New Age	56.Southern Rock	90.Avantgarde
11.Oldies	57.Comedy	91.Gothic Rock
12.Other	58.Cult	92.Progressive Rock
13.Pop	59.Gangsta	93.Psychedelic Rock
14.R&B	60.Top 40	94.Symphonic Rock
15.Rap	61.Christian Rap	95.Slow Rock
16.Reggae	62.Pop/Funk	96.Big Band
17.Rock	63.Jungle	97.Chorus
18.Techno	64.Native American	98.Easy Listening
19.Industrial	65.Cabaret	99.Acoustic
20.Alternative	66.New Wave	100.Humour
21.Ska	67.Psychadelic	101.Speech
22.Death Metal	68.Rave	102.Chanson
23.Pranks	69.Showtunes	103.Opera
24.Soundtrack	70.Trailer	104.Chamber Music
25.Euro-Techno	71.Lo-Fi	105.Sonata
26.Ambient	72.Tribal	106.Symphony
27.Trip-Hop	73.Acid Punk	107.Booty Bass
28.Vocal	74.Acid Jazz	108.Primus
29.Jazz+Funk	75.Polka	109.Porn Groove
30.Fusion	76.Retro	110.Satire
31.Trance	77.Musical	111.Slow Jam
32.Classical	78.Rock & Roll	112.Club
33.Instrumental	79.Hard Rock	113.Tango
34.Acid		114.Samba
35.House		115.Folklore
36.Game		116.Ballad
37.Sound Clip		117.Power Ballad
38.Gospel		118.Rhythmic Soul
39.Noise		119.Freestyle
40.AlternRock		120.Duet
41.Bass		121.Punk Rock
42.Soul		122.Drum Solo
43.Punk		123.Acapella
44.Space		124.Euro-House
45.Meditative		125.Dance Hall

9. This section has been omitted