

ID3 Tags

Introduction

These functions let you read and manipulate ID3 tags. ID3 tags are used in MP3 files to store title of the song, as well as information about the artist, album, genre, year and track number.

Since version 0.2 it is also possible to extract text frames from ID3 v2.2+ tags.

Installing/Configuring

Requirements

No external libraries are needed to build this extension.

Installation

id3 is part of PECL and can be installed using the PEAR installer. To compile PHP with id3 support, download the sourcecode, put it in *php-src/ext/id3* and compile PHP using *--enable-id3*.

Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

Resource Types

This extension has no resource types defined.

Predefined Constants

Most of the id3 functions either let you specify or return a tag version. In order to specify the version please use one of these constants.

ID3_V1_0 ([integer](#))

ID3_V1_0 is used if you are working with ID3 V1.0 tags. These tags may contain the fields title, artist, album, genre, year and comment.

ID3_V1_1 ([integer](#))

ID3_V1_1 is used if you are working with ID3 V1.1 tags. These tags may all information contained in v1.0 tags plus the track number.

ID3_V2_1 ([integer](#))

ID3_V2_1 is used if you are working with ID3 V2.1 tags.

ID3_V2_2 ([integer](#))

ID3_V2_2 is used if you are working with ID3 V2.2 tags.

ID3_V2_3 ([integer](#))

ID3_V2_3 is used if you are working with ID3 V2.3 tags.

ID3_V2_4 ([integer](#))

ID3_V2_4 is used if you are working with ID3 V2.4 tags.

ID3_BEST ([integer](#))

ID3_BEST is used if you would like to let the id3 functions determine which tag version should be used.

ID3 Functions

id3_get_frame_long_name

id3_get_frame_long_name -- Get the long name of an ID3v2 frame

Description

string **id3_get_frame_long_name** (string \$frameId)

[id3_get_frame_long_name\(\)](#) returns the long name for an ID3v2 frame.

Parameters

frameId

An ID3v2 frame

Return Values

Returns the frame long name or **FALSE** on errors.

Examples

Example #1 - [id3_get_frame_long_name\(\)](#) example

```
<?php
$longName = id3_get_frame_long_name("TOLY");
echo $longName;
?>
```

The above example will output:

```
Original lyricist(s)/text writer(s)
```

See Also

- [id3_get_frame_short_name\(\)](#)

id3_get_frame_short_name

id3_get_frame_short_name -- Get the short name of an ID3v2 frame

Description

string **id3_get_frame_short_name** (string \$frameId)

[id3_get_frame_short_name\(\)](#) returns the short name for an ID3v2 frame.

Parameters

frameId

An ID3v2 frame

Return Values

Returns the frame short name or **FALSE** on errors.

The values returned by **id3_get_short_name()** are used in the array returned by [id3_get_tag\(\)](#).

Examples

Example #2 - [id3_get_frame_short_name\(\)](#) example

```
<?php
$shortName = id3_get_frame_short_name("TOLY");
echo $shortName;
?>
```

The above example will output:

```
originalLyricist
```

See Also

- [id3_get_frame_long_name\(\)](#)

id3_get_genre_id

id3_get_genre_id -- Get the id for a genre

Description

int **id3_get_genre_id** (string \$genre)

[id3_get_genre_id\(\)](#) returns the id for a genre.

Parameters

genre

An integer ranging from 0 to 147

Return Values

The genre id or **FALSE** on errors.

Examples

Example #3 - [id3_get_genre_id\(\)](#) example

```
<?php
$id = id3_get_genre_id("Alternative");
echo $id;
?>
```

The above example will output:

20

See Also

- [id3_get_genre_name\(\)](#)
- [id3_get_genre_list\(\)](#)

id3_get_genre_list

id3_get_genre_list -- Get all possible genre values

Description

array **id3_get_genre_list** (void)

[id3_get_genre_list\(\)](#) returns an array containing all possible genres that may be stored in an ID3 tag. This list has been created by Eric Kemp and later extended by WinAmp.

This function is useful to provide you users a list of genres from which they may choose one. When updating the ID3 tag you will always have to specify the genre as an integer ranging from 0 to 147.

Return Values

Returns an array containing all possible genres that may be stored in an ID3 tag.

Examples

Example #4 - [id3_get_genre_list\(\)](#) example

```
<?php
$genres = id3_get_genre_list();
print_r($genres);
?>
```

The above example will output:

```
Array
(
    [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] => Alternative Rock
[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 US
[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
[76] => Retro
[77] => Musical

[78] => Rock & Roll
[79] => Hard Rock
[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] => Acapella
[124] => Euro-House
[125] => Dance Hall
[126] => Goa
[127] => Drum & Bass
[128] => Club-House
[129] => Hardcore
[130] => Terror
[131] => Indie
[132] => BritPop
[133] => Negerpunk
[134] => Polsk Punk
[135] => Beat
[136] => Christian Gangsta
[137] => Heavy Metal

```
[138] => Black Metal
[139] => Crossover
[140] => Contemporary C
[141] => Christian Rock
[142] => Merengue
[143] => Salsa
[144] => Thrash Metal
[145] => Anime
[146] => JPop
[147] => SynthPop
)
```

See Also

- [id3_get_genre_name\(\)](#)
- [id3_get_genre_id\(\)](#)

id3_get_genre_name

id3_get_genre_name -- Get the name for a genre id

Description

string **id3_get_genre_name** (int \$genre_id)

[id3_get_genre_name\(\)](#) returns the name for a genre id.

Parameters

genre_id

An integer ranging from 0 to 147

Return Values

Returns the name as a string.

Examples

Example #5 - [id3_get_genre_name\(\)](#) example

```
<?php
$genre = id3_get_genre_name(20);
echo $genre;
?>
```

The above example will output:

Alternative

See Also

- [id3_get_genre_list\(\)](#)
- [id3_get_genre_id\(\)](#)

id3_get_tag

id3_get_tag -- Get all information stored in an ID3 tag

Description

array **id3_get_tag** (string \$filename [, int \$version])

[id3_get_tag\(\)](#) is used to get all information stored in the id3 tag of the specified file.

Parameters

filename

The path to the MP3 file Instead of a filename you may also pass a valid stream resource

version

Allows you to specify the version of the tag as MP3 files may contain both, version 1.x and version 2.x tags Since version 0.2 [id3_get_tag\(\)](#) also supports ID3 tags of version 2.2, 2.3 and 2.4. To extract information from these tags, pass one of the constants ID3_V2_2, ID3_V2_3 or ID3_V2_4 as the second parameter. ID3 v2.x tags can contain a lot more information about the MP3 file than ID3 v1.x tags.

Return Values

Returns an associative array with various keys like: *title*, *artist*, ..

The key *genre* will contain an integer between 0 and 147. You may use [id3_get_genre_name\(\)](#) to convert it to a human readable string.

Examples

Example #6 - [id3_get_tag\(\)](#) example

```
<?php
$tag = id3_get_tag( "path/to/example.mp3" );
print_r($tag);
?>
```

The above example will output something similar to:

```
Array
(
    [title] => DN-38416
    [artist] => Re:\Legion
    [album] => Reflections
```

```
[year] => 2004
[genre] => 19
)
```

Example #7 - [id3_get_tag\(\)](#) example

```
<?php
$tag = id3_get_tag( "path/to/example2.mp3", ID3_V2_3 );
print_r($tag);
?>
```

The above example will output something similar to:

```
Array
(
    [copyright] => Dirty Mac
    [originalArtist] => Dirty Mac
    [composer] => Marcus Götze
    [artist] => Dirty Mac
    [title] => Little Big Man
    [album] => Demo-Tape
    [track] => 5/12
    [genre] => (17)Rock
    [year] => 2001
)
```

See Also

- [id3_set_tag\(\)](#)
- [id3_remove_tag\(\)](#)
- [id3_get_version\(\)](#)

id3_get_version

id3_get_version -- Get version of an ID3 tag

Description

int **id3_get_version** (string \$filename)

[id3_get_version\(\)](#) retrieves the version(s) of the ID3 tag(s) in the MP3 file.

If a file contains an ID3 v1.1 tag, it always contains a 1.0 tag, as version 1.1 is just an extension of 1.0.

Parameters

filename

The path to the MP3 file Instead of a filename you may also pass a valid stream resource

Return Values

Returns the version number of the ID3 tag of the file. As a tag can contain ID3 v1.x and v2.x tags, the return value of this function should be bitwise compared with the predefined constants **ID3_V1_0**, **ID3_V1_1** and **ID3_V2**.

Examples

Example #8 - [id3_get_version\(\)](#) example

```
<?php
$version = id3_get_version( "path/to/example.mp3" );
if ($version & ID3_V1_0) {
    echo "Contains a 1.x tag\n";
}
if ($version & ID3_V1_1) {
    echo "Contains a 1.1 tag\n";
}
if ($version & ID3_V2) {
    echo "Contains a 2.x tag\n";
}
?>
```

The above example will output something similar to:

```
Contains a 1.x tag
Contains a 1.1 tag
```


See Also

- [id3_set_tag\(\)](#)
- [id3_get_tag\(\)](#)
- [id3_remove_tag\(\)](#)

id3_remove_tag

id3_remove_tag -- Remove an existing ID3 tag

Description

bool **id3_remove_tag** (string \$filename [, int \$version])

[id3_remove_tag\(\)](#) is used to remove the information stored of an ID3 tag.

Parameters

filename

The path to the MP3 file Instead of a filename you may also pass a valid stream resource

version

Allows you to specify the version of the tag as MP3 files may contain both, version 1.x and version 2.x tags.

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #9 - [id3_remove_tag\(\)](#) example

```
<?php
$result = id3_remove_tag( "path/to/example.mp3", ID3_V1_0 );
if ( $result === true ) {
    echo "Tag succesfully removed\n";
}
?>
```

If the file is writable and contained a 1.0 tag, this will output:

```
Tag succesfully removed
```

Notes

Note

Currently [id3_remove_tag\(\)](#) only supports version 1.0 and 1.1. If you choose to remove a 1.0 tag and the file contains a 1.1 tag, this tag will be removed, as v1.1 is only an extension of 1.0.

See Also

- [id3_set_tag\(\)](#)
- [id3_get_tag\(\)](#)
- [id3_get_version\(\)](#)

id3_set_tag

id3_set_tag -- Update information stored in an ID3 tag

Description

bool **id3_set_tag** (string \$filename, array \$tag [, int \$version])

[id3_set_tag\(\)](#) is used to change the information stored of an ID3 tag. If no tag has been present, it will be added to the file.

Parameters

filename

The path to the MP3 file Instead of a filename you may also pass a valid stream resource

tag

An associative array of tag keys and values The following keys may be used in the associative array:

Keys in the associative array

key	possible value	available in version
title	string with maximum of 30 characters	v1.0, v1.1
artist	string with maximum of 30 characters	v1.0, v1.1
album	string with maximum of 30 characters	v1.0, v1.1
year	4 digits	v1.0, v1.1
genre	integer value between 0 and 147	v1.0, v1.1
comment	string with maximum of 30 characters (28 in v1.1)	v1.0, v1.1
track	integer between 0 and 255	v1.1

version

Allows you to specify the version of the tag as MP3 files may contain both, version 1.x and

version 2.x tags

Return Values

Returns **TRUE** on success or **FALSE** on failure.

Examples

Example #10 - [id3_set_tag\(\)](#) example

```
<?php
$data = array(
    "title" => "Re:Start",
    "artist" => "Re:\Legion",
    "comment" => "A nice track"
);
$result = id3_set_tag( "path/to/example.mp3", $data, ID3_V1_0 );
if ($result === true) {
    echo "Tag succesfully updated\n";
}
?>
```

If the file is writable, this will output:

```
Tag succesfully updated
```

Notes

Note

Currently [id3_remove_tag\(\)](#) only supports version 1.0 and 1.1.

See Also

- [id3_remove_tag\(\)](#)
- [id3_get_tag\(\)](#)
- [id3_get_version\(\)](#)