

What is ID3 (v1)?

The audio format MPEG layer I, layer II and layer III (MP3) has no native way of saving information about the contents, except for some simple yes/no parameters like "private", "copyrighted" and "original home" (meaning this is the original file and not a copy). A solution to this problem was introduced with the program "Studio3" by Eric Kemp alias NamkraD in 1996. By adding a small chunk of extra data in the end of the file one could get the MP3 file to carry information about the audio and not just the audio itself.

The placement of the tag, as the data was called, was probably chosen as there were little chance that it should disturb decoders. In order to make it easy to detect a fixed size of 128 bytes was chosen. The tag has the following layout (as hinted by the scheme to the right):



Internal layout of an ID3v1 tagged file.

Song Title	30 characters
Artist	30 characters
Album	30 characters
Year	4 characters
Comment	30 characters
Genre	1 byte

If you one sum the the size of all these fields we see that 30+30+30+4+30+1 equals 125 bytes and not 128 bytes. The missing three bytes can be found at the very beginning of the tag, before the song title. These three bytes are always "TAG" and is the identification that this is indeed a ID3 tag. The easiest way to find a ID3v1/1.1 tag is to look for the word "TAG" 128 bytes from the end of a file.

As all artists doesn't have a 30 character name it is said that if there is some bytes left after the information is entered in the field, those bytes should be fille with the binary value 0. You might also think that you cannot write that much in the genre field, being one byte big, but it is more clever than that. The byte value you enter in the genre field corresponds to a value in a predefined list. The list that Eric Kemp created had 80 entries, ranging from 0 to 79.

What is ID3v1.1?

ID3v1 may well be easy to implement for programmers, but it sure is frustrating for those with their own, creative ideas. Since the ID3v1 tag had a fixed size and no space marked "Reserved for future use", there isn't really room for that much improvement, if you want to maintain compatibility with existing software.

One who found a way out was Michael Mutschler who made a quite clever improvement on ID3v1. Since all non-filled fields must be padded with zeroed bytes its a good assumption that all ID3v1 readers will stop reading the field when they encounter a zeroed byte. If the second last byte of a field is zeroed and the last one isn't we have an extra byte to fill with information. As the comments field is to short to write anything useful in the ID3v1.1 standard declares that this field should be 28 characters, that the next byte always should be zero and that the last byte before the genre byte should contain which track on the CD this music comes from.



Internal layout of an ID3v1.1 tagged file.