**3GPP TSG- Meeting #**

**, , -** Revision of S4-241125  
 Revision of S4-241061

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** |  | **rev** | **2** | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **x** | Radio Access Network |  | Core Network | **x** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** |  | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** |  | | | | |  | ***Date:*** | | |  |
|  |  | | | |  | |  | | |  |
| ***Category:*** |  |  | | | | | ***Release:*** | | |  |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | 1. The split rendering feature enabled with this CR was selected by SA4 according to the ISAR selection procedure. This CR replaces CR0001, as presented at SA4#127-bis-e 2. After the submission of the initial IVAS C-Code to SA4#125, continued and extended testing (both, mechanical and subjective) has revealed a number of issues that have to be corrected. These issues include crashes, address/memory sanitizer errors, undefined-behavior sanitizer errors, quality issues. 3. The listener orientation axes are not defined correctly. 4. The distance attenuation parameter of the renderer configuration is not defined. 5. Add provision of provision of HRIR / BRIR filter sets as control data for binaural audio rendering by means of Matlab scripts in order include custom HRIRs/BRIRs (inline wih IVAS-4) 6. IVAS specific Split Rendering solution: LC3plus bitrates don’t comply with Bluetooth BAP profile | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | 1. The feature is introduced in section 4.5 and fully described in a new subsection under section 7 (Functional description of the rendering, rendering control, and pre-rendering). 2. Corrections to the findings as described above, specifically corrections to:  * Various crashes, especially for bitrate switching and packet-loss conditions * Various sanitizer errors (address sanitizer, memory sanitizer, unbehaved behavior sanitizer) * Various improvements to memory allocation * Various corrections for bitrate switching * Code refactoring and improvements * Broken JBM functionality for OSBA * Correction to various quality issues   + CNA too high in ISM DTX with JBM (   + Wrong object panning for OMASA at 32 and 48 kbit/s (MR 1095)   + Loudness mitmatch for rendering 7.1.4 at 160 kbit/s to mono or stereo (MR 1069)   + Clicks for MDCT-Stereo DTX with mono output for transitions inactive -> active (MR 1173)   + Wrong rendering rendering 7.1.4 at 160 kbit/s to SBA (MR 1151)   + Distorted multi-channel output in 5ms rendering (MR 1311)   + Mismatch in OSBA output gain (MR 1341)   + ParamMC and ParamUpmix binaural room reverb with head rotation energy not correct (MR 1382)   + Level drop in high-bitrate ISM configurations (MR 1516)   + Glitches for border-cases in ParamMC output (MR 1563) * Corrections which impact interoperability with previous versions:   + Added missing SBA order bits for OSBA (MR 1501)   + Adjust OSBA discrete-coding x-over bitrate (MR 1347)   + For SBA, add signaling of input signal order to bitstream (MR 1539)   + Introduce an introduce finite precision comparison in MASA surround coherence entropy coding for 384 and 512 kbit/s (MR 1486); this issue was detected during the conversion to BASOP.   A complete list with all merge requests affecting the source code is available at <https://forge.3gpp.org/rep/ivas-codec-pc/ivas-codec/-/wikis/Documentation/MRs-in-20240522_delivery_SA4_128_final>   1. An incorrect specification of the listener orientation may lead to an incorrect rendering of the audio scene. 2. The use of the distance attenuation feature would not be documented and the parsing of the binary format renderer configuration may fail. 3. As part of the electronic attachment, a set of Matlab scripts is provided to provision custom HRIR / BRIR filter sets as control data for binaural audio rendering including new HRIRs as proposed and agreed in S4-241134. 4. IVAS specific Split Rendering solution: Correct LC3plus bitrates in order to comply with Bluetooth BAP profile (agreed S4-241004). | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | IVAS codec will not offer a split rendering feature and potentially not be available on lightweight end devices. Significant issues with IVAS C-code, i.e. crashes, memory issues, undefined behavior, impact on BASOP-port, quality issues, missing support for custom HRIRs/BRIRs. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 3.3, 4, 5.11, 5.14.2, 5.16, 5.17, Annex B.1, Annex B.2  Electronic attachment | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | | **X** |  | Other core specifications | | | | TS 26.253 CR 0003  TS 26.254 CR 0001  TS 26.255 CR 0001 | | |
| ***affected:*** | | **X** |  | Test specifications | | | | TS 26.258 CR 0001 | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | | Changes to the electronic attachment will be implemented in S4-241172. | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | | S4-241061: Initial Version  S4-241125: R1, with additional changes to the electronic attachment only  S4-241171: R2, with additional changes to the electronic attachment only | | | | | | | | |

\* \* \* First Change \* \* \* \*

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

ACN Ambisonic Channel Number

CICP Coding-independent Code Points

CSV Comma Separated Values

EVS Enhanced Voice Services

FB Fullband

FEC Frame Erasure Concealment

HRTF Head Related Transfer Function

ISM Independent Stream with Metadata

IVAS Immersive Voice and Audio Services

ISAR Immersive Audio for Split Rendering Scenarios

JBM Jitter Buffer Management

LFE Low Frequency Enhancement

MASA Metadata-Assisted Spatial Audio

MC Multi-channel

NB Narrowband

OBA Object Based Audio

SBA Scene Based Audio

SID Silence Insertion Descriptor

SWB Super Wideband

WB Wideband

WMOPS Weighted Millions of Operations Per Second

\* \* \* Next Change \* \* \* \*

# 4 C code structure

This clause gives an overview of the structure of the floating-point C code and provides an overview of the contents and organization of the C code attached to the present document.

The C code has been verified on the following platforms:

- IBM PC compatible computers with Windows 10 operating systems and Microsoft Visual C++ 2017 compiler, 32-bit.

C was selected as the programming language because portability was desirable.

## 4.1 Contents of the C source code

The C code is organized as listed in Table 1:

Table 1: Source code directory structure

|  |  |
| --- | --- |
| **Directory** | **Description** |
| readme.txt | information on how to compile and use |
| Makefile | UNIX style encoder Makefile |
| Workspace\_msvc/ | Directory for the MSVC 2017 (or newer) project files |
| apps/ | Source code files used solely for the encoder/decoder/renderer applications; these applications make use of the libraries built from lib\_com, lib\_dec, lib\_enc, lib\_rend, and lib\_util |
| lib\_com/ | Source code files used both in encoder and decoder |
| lib\_dec/ | Source code files used solely in the decoder |
| lib\_enc/ | Source code files used solely in the encoder |
| lib\_isar/ | Source code files used solely for split rendering |
| lib\_lc3plus/ | Source code files used solely for split rendering |
| lib\_rend/ | Source code files used solely in the renderer |
| lib\_util/ | Source code files solely for utility functions used by the applications |

The distributed files with suffix "c" contain the source code and the files with suffix "h" are the header files. The ROM data is contained in files named "rom\_\*" and “ivas\_rom\_\*” with suffix "c".

Makefiles are provided for the platforms in which the C code has been verified (listed above). Once the software is installed, this directory will have a compiled version of the encoder (named IVAS\_cod), the decoder (named IVAS\_dec), the renderer (named IVAS\_rend) and the split rendering post-renderer (named ISAR\_post\_rend).

## 4.2 Program execution

The codec for Immersive Voice and Audio Services is implemented in four programs:

*-* IVAS\_cod: encoder;

*-* IVAS\_dec: decoder;

*-* IVAS\_rend: renderer;

- ISAR\_post\_rend: split rendering post-renderer.

The programs should be called like:

- IVAS\_cod [encoder options] <input file> <bitstream file>;

- IVAS\_dec [decoder options] <bitstream file> <output file>;

- IVAS\_rend [renderer options] -i <input file> -if <input format> -o <output file> -of <output format>;

- ISAR\_post\_rend [post-renderer options] -i < bitstream file or input file> -if <input format> -o <output file>.

The input and output files contain 16-bit linear encoded PCM samples (headerless or in WAVE format) and the bitstream file contains encoded data.

The encoder, decoder, and renderer options will be explained by running the programs without any input arguments. See the file readme.txt for more information on how to run the *IVAS\_cod*, *IVAS\_dec*, *IVAS\_rend* and *ISAR\_post\_rend* programs.

\* \* \* Next Change \* \* \* \*

## 5.11 Head rotation trajectory file (decoder/renderer input)

In the reference implementation of the codec, input data representing the current rotation of the listeners head can be provided to the decoder in an ASCII formatted file comprising four columns separated by commas. These columns contain floating-point numbers representing either a quaternion or a Euler angle. The distinction between these two input formats is made by a magic number in the first column. If this value is set to -3.0, it is assumed that the remaining three columns contain three Euler angles. Otherwise, all four columns are interpreted as a Quaternion. The input is expected to have one line for each subframe of 5 ms.

In the case of Quaternion-based input, the columns are the w, x, y, z components of a unit quaternion. Proper normalization to 1 shall be maintained in the input. The coordinate system is defined such that the x-axis points into the direction of view, the y axis points right to the left ear, and the z axis point from bottom to top. The origin is in the center of the head. For example, an approximate 90-degree rotation around the horizontal (z) axis would be represented by the following input line:

0.707107,0.000000,0.000000,0.70710

In the case of Euler-angle input, the first column contains the magic number -3.0, and the next three columns are the Euler angles yaw, pitch, and roll. The rotations are applied in the order yaw-pitch-roll. The yaw angle rotates around the z axis. The pitch angle rotates around the new y axis. The roll angle rotates around the new x axis. The equivalent of the example line above is then:

-3.0,90.000035,0.000000,0.000000

\* \* \* Next Change \* \* \* \*

### 5.14.2 Text renderer config metadata format

The text based renderer configuration file contains the following syntax elements:

[general] header of general metadata

binaryConfig = path; path to the binary configuration file

[roomAcoustics] header of room acoustic metadata group

frequencyGridCount = N; number of frequency grids

acousticEnvironmentCount = N; number of acoustic environments

[frequencyGrid:N] header of a frequency grid, where N is a zero-based, sequential grid index

method = individualFrequencies | startHopAmount | defaultBanding;  
specifies frequency grid representation method

nrBands = N; number of frequency bands, applicable for individual frequencies and start-hop-amount representation methods

frequencies = [...]; center frequencies for individualFrequencies representation method, a comma separated list of N numeric values (ints or floats)

startFrequency = value; starting frequency for start-hop-amount representation method

frequencyHop = value; frequency hop for start-hop-amount representation method. Center frequencies for a grid are computed as fcn = fcn-1 \* hop

defaultGrid = N; default grid identifier. The available default grids are as in Annex B.1, Table B.4.

defaultGridOffset = N; it is possible to use a subset of a default grid by specifying an offset - index of the first center frequency of the default grid and

defaultGridNrBands = N; number of bands from the default grid to be used

[acousticEnvironment:N] header of an acoustic environment element, where N is a zero-based grid index (does not have to be sequential)

frequencyGridIndex = N; index of the frequency grid (see above) used for frequency dependent parameters

preDelay = value; a delay at which DSR (diffuse to source ratios) were measured

rt60 = [...]; RT60 values per frequency band

dsr = [...]; diffuse to source sound energy ratio per frequency band

earlyReflectionsSize = [x, y, z]; shoebox model room size in x, y, z dimension in meters

absorptionCoeffs = [x1, x2, y1, y2, z1, z2];  
early reflections absorption coefficients per wall

listenerOrigin = [x, y, z]; early reflections listener origin (optional) as offset from the room center

lowComplexity = TRUE | FALSE; early reflection low-complexity mode flag (FALSE by default)

[directivitySetting] header of the directivity data group

directivityCount = N; number of directivity components

[directivityPattern:N] header of a directivity pattern element, where N is a zero-based element index

directivity = [ia, oa, og]; directivity data: ia – inner angle, oa – outer angle, og – outer gain.

[SPLITREND] header of split rendering group

BITRATE = R; split rendering bitrate

DOF = N; degree of freedom (N ranging from 0 to 3)

HQMODE = N; High quality mode for 3DOF (N can be 0 or 1), adds more complexity at pre-renderer

CODEC = X; split rendering transport codec (X can be LCLD or LC3plus or NONE)

FRAMESIZE = [5, 10, 20] frame size in ms of the split rendering transport codec. Note: LC3plus supports 5 and 10 ms framesize, LCLD supports 5, 10 and 20 ms framesize.

The config file format supports comments starting with a hash sign #. It also supports splitting data into multiple lines, useful in case of larger arrays.

\* \* \* Next Change \* \* \* \*

## 5.16 Split rendering pose correction file (decoder/renderer output, post-renderer input)

The split rendering pose correction file used with PCM split rendering audio data (output of decoder/renderer and input to post-renderer, mode BINAURAL\_SPLIT\_PCM) is described in TS 26.253, clause 7.6.2.3 and clause 7.6.7.

## 5.17 Split rendering bitstream file (decoder/renderer output, post-renderer input)

The split rendering bitstream file (output of decoder/renderer and input to post-renderer, mode BINAURAL\_SPLIT\_CODED) is described in TS 26.253, clause 7.6.7.

\* \* \* Next Change \* \* \* \*

Annex B (normative):  
Binary renderer config metadata format

# B.1 Definition of binary renderer config metadata format

The binary renderer config metadata format consists of acoustic environment, directivity payload components and distance attenuation components (payloadRendConfig, see Table B.1). The acoustic environment component (payloadAcEnv, see Table B.2) metadata syntax consists of a frequency grids element (payloadFreqGrid) containing single or multiple frequency grids, and a single or multiple acoustic environments. An acoustic environment contains a late reverb element (payloadLateReverb), and optionally a shoebox model element for early reflections synthesis (payloadEarlyReflections). This construction allows for dynamic switching between acoustic environments by selecting an environment using its identifier (revAcEnvID). This facilitates multiple use cases, such as scenes with multiple, fully independent rooms, dynamic scene changes, or user selectable acoustics environments. The payload syntax of the payloadAcEnv() and its elements are shown in the tables below. Locally atomic data components are marked bold with their respective size in bits and mnemonic format, and their descriptions are provided below the payload element tables. The complex payload elements are provided in subsequent tables.

**Table B.1: Syntax of payloadRendConfig**

|  |  |  |
| --- | --- | --- |
| **Syntax** | **Bits** | **Mnemonic** |
| payloadRendConfig() { |  |  |
| if ( **hasAcEnv** ) { | 1 | bslbf |
| payloadAcEnv(); |  |  |
| }  if ( **hasDirectivity** ) {  payloadDirectivity();  }  if ( **hasDistanceAttenuation** ) {  payloadDistanceAttenuation();  } | 1  1 | bslbf  bslbf |
| } |  |  |

…

Table B.9: Syntax of GetFrequency

| Syntax | Bits | Mnemonic |
| --- | --- | --- |
| frequency = GetFrequency() { |  |  |
| frequency = LUT( **frequencyCode** ); | var | vlclbf |
| if ( **moreAccuracy** ) { | 1 | bslbf |
| frequency = frequency \* 2^((**frequencyRefine** + 1) / 51); | 4 | uimsbf |
| return frequency; |  |  |
| } |  |  |

**frequencyCode** Code that indicates a center frequency in Hz of a one-third octave band (see Table B.20)

**moreAccuracy** Flag that indicates whether data for a more accurate frequency is transmitted.

**frequencyRefine** Field that indicates a value for refining the frequency value.

The payloadDirectivity() elements describe the source directivity pattern. Each pattern has an ID and the objects can be assigned to use a specific ID.

Table B.9a1: Syntax of payloadDirectivity

|  |  |  |
| --- | --- | --- |
| Syntax | Bits | Mnemonic |
| payloadDirectivity() { |  |  |
| **directivityCount** = GetCountOrIndex (); | var | vlclbf |
| for ( i = 0; i < directivityCount; i++ ) { |  |  |
| **directivityIndex** = GetCountOrIndex (); | var | vlclbf |
| ia[directivityIndex] = getAngle(); |  |  |
| oa[directivityIndex] = getAngle(); |  |  |
| og[directivityIndex] = getOuterGain(); |  |  |
| } |  |  |
| } |  |  |

Table B.9a2: Syntax of GetAngle

| Syntax | Bits | Mnemonic |
| --- | --- | --- |
| angle = GetAngle() { |  |  |
| angle = **angleCode** \* 20.0; | 5 | bslbf |
| return angle; |  |  |
| } |  |  |

Table B.9a3: Syntax of GetOuterGain

| Syntax | Bits | Mnemonic |
| --- | --- | --- |
| outerGain = GetAngle() { |  |  |
| log\_gain = **-**90.0 + **outerGainCode** \* 3.0; | 5 | bslbf |
| outerGain = 10^(log\_gain/20); |  |  |
| return outerGain; |  |  |
| } |  |  |

The payloadDistanceAttenuation describes the distance attenuation parameters that is used for all objects being rendered.

Table B.9a4: Syntax of payloadDistanceAttenuation

|  |  |  |
| --- | --- | --- |
| Syntax | Bits | Mnemonic |
| payloadDistanceAttenuation() { |  |  |
| rd = GetRefDistMeters(); |  |  |
| md = GetMaxDistMeters(); |  |  |
| rf = GetRolloffFactor(); |  |  |
| } |  |  |
| } |  |  |

Table B.9a5: Syntax of GetRefDistMeters

| Syntax | Bits | Mnemonic |
| --- | --- | --- |
| refDist = GetRefDistMeters() { |  |  |
| refDist = (**refDistCode** + 1) \* 0.1; | 6 | bslbf |
| return refDist; |  |  |
| } |  |  |

Table B.9a6: Syntax of GetMaxDistMeters

| Syntax | Bits | Mnemonic |
| --- | --- | --- |
| maxDist = GetMaxDistMeters() { |  |  |
| maxDist = (**maxDistCode** + 1) \* 1.0; | 6 | bslbf |
| return maxDist; |  |  |
| } |  |  |

Table B.9a7: Syntax of GetRolloffFactor

| Syntax | Bits | Mnemonic |
| --- | --- | --- |
| rolloffFactor = GetRolloffFactor() { |  |  |
| rolloffFactor = **rolloffFactorCode**\* 0.1; | 6 | bslbf |
| return rolloffFactor; |  |  |
| } |  |  |

\* \* \* Next Change \* \* \* \*

# B.2 Support Elements Look-up Tables

This clause contains the look-up tables used in the binary renderer config metadata.

Table B.10: countOrIndexLoCode look-up table

| Code | Value | Code | Value | Code | Value | Code | Value | Code | Value |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0111 | 0 | 001010 | 13 | 111101 | 26 | 1101000 | 39 | 1011011 | 52 |
| 100 | 1 | 001001 | 14 | 111100 | 27 | 1100111 | 40 | 1011010 | 53 |
| 01100 | 2 | 001000 | 15 | 111011 | 28 | 1100110 | 41 | 1011001 | 54 |
| 01101 | 3 | 000111 | 16 | 111010 | 29 | 1100101 | 42 | 1011000 | 55 |
| 01010 | 4 | 000110 | 17 | 111001 | 30 | 1100100 | 43 | 1010111 | 56 |
| 01011 | 5 | 000101 | 18 | 111000 | 31 | 1100011 | 44 | 1010110 | 57 |
| 01000 | 6 | 000100 | 19 | 1101111 | 32 | 1100010 | 45 | 1010101 | 58 |
| 01001 | 7 | 000011 | 20 | 1101110 | 33 | 1100001 | 46 | 1010100 | 59 |
| 001111 | 8 | 000010 | 21 | 1101101 | 34 | 1100000 | 47 | 1010011 | 60 |
| 001110 | 9 | 000001 | 22 | 1101100 | 35 | 1011111 | 48 | 1010010 | 61 |
| 001101 | 10 | 000000 | 23 | 1101011 | 36 | 1011110 | 49 | 1010001 | 62 |
| 001100 | 11 | 111111 | 24 | 1101010 | 37 | 1011101 | 50 | 1010000 | 63 |
| 001011 | 12 | 111110 | 25 | 1101001 | 38 | 1011100 | 51 |  |  |

…

Table B.22: absorptionCode look-up table

|  |  |
| --- | --- |
| Code | Value |
| 110 | 0 |
| 100 | 0.1 |
| 101 | 0.2 |
| 0110 | 0.3 |
| 0111 | 0.4 |
| 111 | 0.5 |
| 0100 | 0.6 |
| 0101 | 0.7 |
| 0010 | 0.8 |
| 0011 | 0.9 |
| 000 | 1 |











Table B.26: angleCode look-up table

| Code | Value | Code | Value | Code | Value | Code | Value |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 00000 | 0 | 00101 | 5 | 01010 | 10 | 01111 | 15 |
| 00001 | 1 | 00110 | 6 | 01011 | 11 | 10000 | 16 |
| 00010 | 2 | 00111 | 7 | 01100 | 12 | 10001 | 17 |
| 00011 | 3 | 01000 | 8 | 01101 | 13 | 10010 | 18 |
| 00100 | 4 | 01001 | 9 | 01110 | 14 |  |  |

Table B.27: outerGainCode look-up table

| Code | Value | Code | Value | Code | Value | Code | Value | Code | Value | Code | Value |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 00000 | 0 | 00110 | 6 | 01100 | 12 | 10010 | 18 | 11000 | 24 | 11110 | 30 |
| 00001 | 1 | 00111 | 7 | 01101 | 13 | 10011 | 19 | 11001 | 25 |  |  |
| 00010 | 2 | 01000 | 8 | 01110 | 14 | 10100 | 20 | 11010 | 26 |  |  |
| 00011 | 3 | 01001 | 9 | 01111 | 15 | 10101 | 21 | 11011 | 27 |  |  |
| 00100 | 4 | 01010 | 10 | 10000 | 16 | 10110 | 22 | 11100 | 28 |  |  |
| 00101 | 5 | 01011 | 11 | 10001 | 17 | 10111 | 23 | 11101 | 29 |  |  |

Table B.28: refDistCode look-up table

| Code | Value | Code | Value | Code | Value | Code | Value | Code | Value | Code | Value |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 000000 | 0 | 001100 | 12 | 011000 | 24 | 100100 | 36 | 110000 | 48 | 111100 | 60 |
| 000001 | 1 | 001101 | 13 | 011001 | 25 | 100101 | 37 | 110001 | 49 | 111101 | 61 |
| 000010 | 2 | 001110 | 14 | 011010 | 26 | 100110 | 38 | 110010 | 50 | 111110 | 62 |
| 000011 | 3 | 001111 | 15 | 011011 | 27 | 100111 | 39 | 110011 | 51 | 111111 | 63 |
| 000100 | 4 | 010000 | 16 | 011100 | 28 | 101000 | 40 | 110100 | 52 |  |  |
| 000101 | 5 | 010001 | 17 | 011101 | 29 | 101001 | 41 | 110101 | 53 |  |  |
| 000110 | 6 | 010010 | 18 | 011110 | 30 | 101010 | 42 | 110110 | 54 |  |  |
| 000111 | 7 | 010011 | 19 | 011111 | 31 | 101011 | 43 | 110111 | 55 |  |  |
| 001000 | 8 | 010100 | 20 | 100000 | 32 | 101100 | 44 | 111000 | 56 |  |  |
| 001001 | 9 | 010101 | 21 | 100001 | 33 | 101101 | 45 | 111001 | 57 |  |  |
| 001010 | 10 | 010110 | 22 | 100010 | 34 | 101110 | 46 | 111010 | 58 |  |  |
| 001011 | 11 | 010111 | 23 | 100011 | 35 | 101111 | 47 | 111011 | 59 |  |  |

Table B.29: maxDistCode look-up table

| Code | Value | Code | Value | Code | Value | Code | Value | Code | Value | Code | Value |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 000000 | 0 | 001100 | 12 | 011000 | 24 | 100100 | 36 | 110000 | 48 | 111100 | 60 |
| 000001 | 1 | 001101 | 13 | 011001 | 25 | 100101 | 37 | 110001 | 49 | 111101 | 61 |
| 000010 | 2 | 001110 | 14 | 011010 | 26 | 100110 | 38 | 110010 | 50 | 111110 | 62 |
| 000011 | 3 | 001111 | 15 | 011011 | 27 | 100111 | 39 | 110011 | 51 | 111111 | 63 |
| 000100 | 4 | 010000 | 16 | 011100 | 28 | 101000 | 40 | 110100 | 52 |  |  |
| 000101 | 5 | 010001 | 17 | 011101 | 29 | 101001 | 41 | 110101 | 53 |  |  |
| 000110 | 6 | 010010 | 18 | 011110 | 30 | 101010 | 42 | 110110 | 54 |  |  |
| 000111 | 7 | 010011 | 19 | 011111 | 31 | 101011 | 43 | 110111 | 55 |  |  |
| 001000 | 8 | 010100 | 20 | 100000 | 32 | 101100 | 44 | 111000 | 56 |  |  |
| 001001 | 9 | 010101 | 21 | 100001 | 33 | 101101 | 45 | 111001 | 57 |  |  |
| 001010 | 10 | 010110 | 22 | 100010 | 34 | 101110 | 46 | 111010 | 58 |  |  |
| 001011 | 11 | 010111 | 23 | 100011 | 35 | 101111 | 47 | 111011 | 59 |  |  |

Table B.30: rolloffFactorCode look-up table

| Code | Value | Code | Value | Code | Value | Code | Value |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 000000 | 0 | 001100 | 12 | 011000 | 24 | 100100 | 36 |
| 000001 | 1 | 001101 | 13 | 011001 | 25 | 100101 | 37 |
| 000010 | 2 | 001110 | 14 | 011010 | 26 | 100110 | 38 |
| 000011 | 3 | 001111 | 15 | 011011 | 27 | 100111 | 39 |
| 000100 | 4 | 010000 | 16 | 011100 | 28 | 101000 | 40 |
| 000101 | 5 | 010001 | 17 | 011101 | 29 |  |  |
| 000110 | 6 | 010010 | 18 | 011110 | 30 |  |  |
| 000111 | 7 | 010011 | 19 | 011111 | 31 |  |  |
| 001000 | 8 | 010100 | 20 | 100000 | 32 |  |  |
| 001001 | 9 | 010101 | 21 | 100001 | 33 |  |  |
| 001010 | 10 | 010110 | 22 | 100010 | 34 |  |  |
| 001011 | 11 | 010111 | 23 | 100011 | 35 |  |  |

\* \* \* End of Change \* \* \* \*