3GPP TS 26.403 V11.0.0 (2012-09)

Technical Specification

3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
General audio codec audio processing functions;
Enhanced aacPlus general audio codec;
Encoder specification;
Advanced Audio Coding (AAC) part
(Release 11)

Keywords
UMTS, GSM, codec, LTE

3GPP

Postal address

3GPP support office address
650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet
http://www.3gpp.org

Contents

Foreword

The present document describes the detailed mapping of the general audio service employing the aacPlus general audio codec within the 3GPP system.

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of this TS, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 Indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the specification;

1 Scope

This Telecommunication Standard (TS) describes the AAC encoder part of the Enhanced aacPlus general audio codec [1].

2 Normative references

This TS incorporates by dated and undated reference, provisions from other publications. These normative references are cited in the appropriate places in the text and the publications are listed hereafter. For dated references, subsequent amendments to or revisions of any of these publications apply to this TS only when incorporated in it by amendment or revision. For undated references, the latest edition of the publication referred to applies.

[1] 3GPP TS 26.401: "Enhanced aacPlus general audio codec; General Description".

[2] ISO/IEC 14496-3:2001: "Information technology - Coding of audio-visual objects - Part 3: Audio".

[3] ISO/IEC 14496-3:2001/Amd.1:2003: "Bandwidth Extension".

[4] ISO/IEC 14496-3:2001/Amd.1:2003/DCOR1.

[5] ISO/IEC 14496-3:2001/ Amd.2:2004: "Parametric Coding for High Quality Audio".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of this TS, the following definitions apply:

frame: time segment associated with one AAC single channel or channel pair element

frequency coefficient: output value of the MDCT transform

scalefactor band: a group of consecutive frequency coefficients, that will be coded with the same quantizer step size

3.2 Symbols

For the purposes of this TS, the following symbols apply:

k is the current index for the spectral coefficients

$kOffset(n)$ is the index of the first spectral coefficient in scalefactorband n

n is the current scalefactor band

3.3 Abbreviations

For the purposes of this TS, the following abbreviations apply.

AAC	Advanced Audio Coding
aacPlus	Combination of MPEG-4 AAC and MPEG-4 Bandwidth extension (SBR)
Enhanced aacPlus	Combination of MPEG-4 AAC, MPEG-4 Bandwidth extension (SBR) and MPEG-4 Parametric Stereo
KBD	Kaiser-Bessel derived
PE	perceptual entropy
SBR	Spectral Band Replication
TNS	Temporal Noise Shaping

4 Outline description

This TS is structured as follows:

Section 5.1 gives an encoder overview description. Section 5.2 gives a detailed description of the stereo preprocessing. Section 5.3 gives a detailed description of the filterbank used in the encoder. Section 5.4 gives a detailed description of the psychoacoustic model. Section 5.5 gives a detailed description of the temporal noise shaping and mid/side stereo tools. Section 5.6 gives a detailed description of the quantization and coding procedure used in the encoder.

5 AAC Encoder

5.1 Overview

The AAC encoder acts as the core encoding algorithm of the aacPlus system encoding at half the sampling rate of aacPlus. Since aacPlus implements the High Efficiency AAC Profile at Level 2 as defined in [3], the AAC LC object type is used. The AAC LC object type does not implement the Long Term Predictor (LTP) tool. The Level 2 implies a restriction to a maximum of two channels. Furthermore in case of SBR being used, the maximum AAC sampling rate is restricted to 24 kHz whereas if SBR is not used the maximum AAC sampling rate is restricted to 48 kHz.

The basic layout is depicted below.

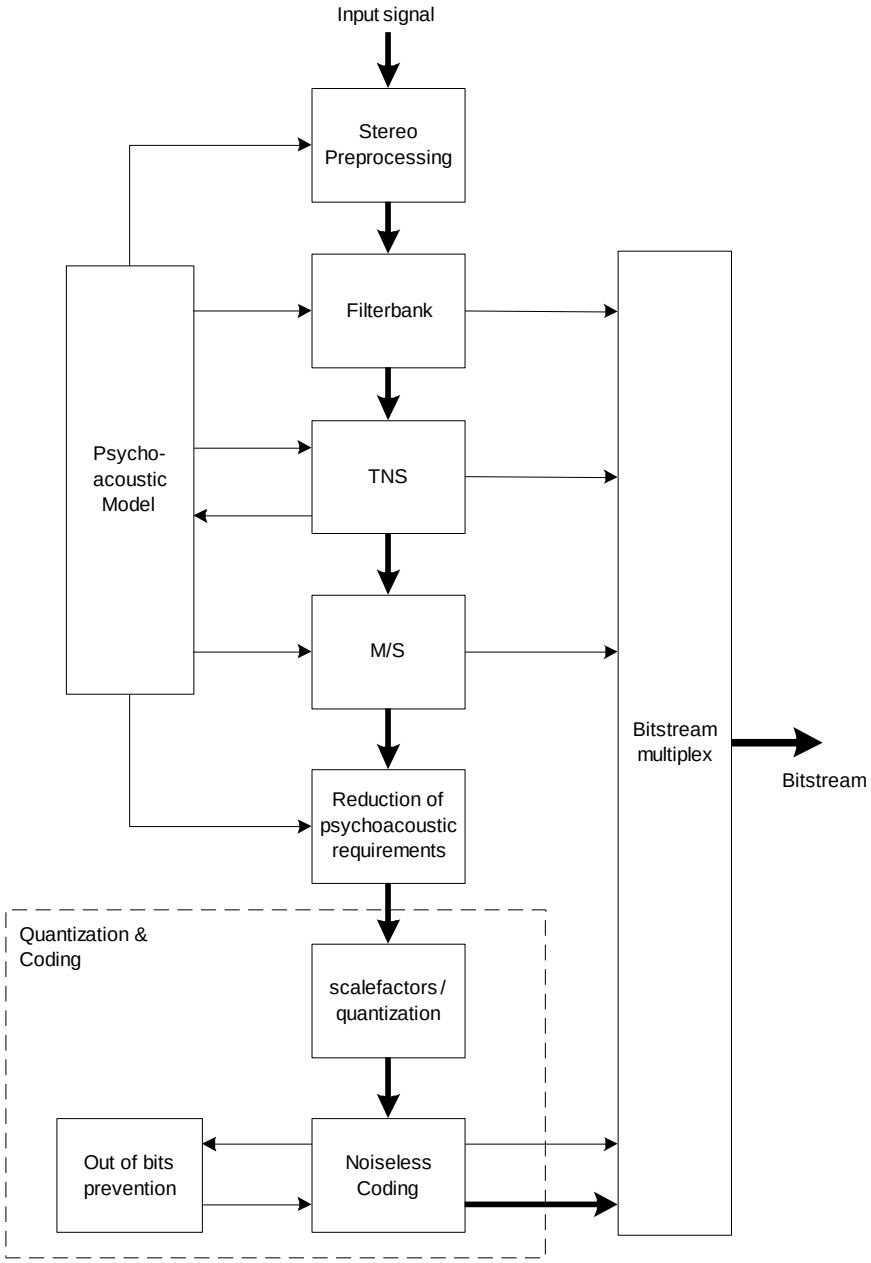

Figure 1: AAC Encoder Block Diagram

5.2 Stereo Preprocessing

With stereo preprocessing, the stereo width of difficult to encode signals at low bitrates is reduced. Stereo preprocessing is active for bitrates less than 60kbit/s.

The side channel is attenuated with influence of the following parameters:

- The total perceptual entropy pe_0 before the increase of the thresholds. This PE is smoothed over past frames and normalized. For a definition of the perceptual entropy see 5.6.1.1.3. .
- The energy ratio between side and mid channel smoothed over past frames. If the side channel is very strong, less attenuation of the side channel should happen.
- The energy ratio between the left and right channel. Less attenuation of the side channel occurs for signals that appear to be nearly on the left or the right.
- The smaller the bitrate, the more attenuation of the side channel takes place.

Depending on these parameters an attenuation factor $stereoAttFac$ is calculated for every frame. Always 1024 samples of one frame for the left and right channel are then modified as follows:

$$L' = \frac{1 + stereoAttFac}{2} \cdot L + \frac{1 - stereoAttFac}{2} \cdot R$$

$$R' = \frac{1 - stereoAttFac}{2} \cdot L + \frac{1 + stereoAttFac}{2} \cdot R$$

with L, R as left resp. right samples before and L', R' as modified left and right samples after stereo preprocessing.

5.3 Filterbank

The filterbank is an MDCT as described in [2]. The window length N of the MDCT is either 2048 for the ONLY_LONG_SEQUENCE, LONG_START_SEQUENCE and LONG_STOP_SEQUENCE window sequence or 256 for the EIGHT_SHORT_SEQUENCE window sequence. The spectral coefficients are defined as follows:

$$X(k) = 2 \cdot \sum_{n=0}^{N-1} z_n \cos\left(\frac{2\pi}{N}\left(n + \frac{N/2+1}{2}\right)\left(k + \frac{1}{2}\right)\right) \qquad \text{for } 0 \le k < N/2$$

with z_n as windowed input sequence, n as samples index and k as spectral coefficient index.

For long windows the window shape is always 1, that is a Kaiser-Bessel derived (KBD) window will be used. As window shape for the short windows always the sine window will be applied. For the definition of KBD and sine window see [2].

5.4 Psychoacoustic Model

The psychoacoustic model is simplified compared to the model presented in [2]. Note that the model works in combination with the quantization and coding strategy described in 5.6 below. The following sections describe the steps of the threshold calculation.

5.4.1 Blockswitching

The decision wether to use long windows with a window length of 2048 samples or a sequence of eight short blocks with a window length of 256 samples will be taken in the time domain. It is not possible to switch immediately between an ONLY_LONG_SEQUENCE and an EIGHT_SHORT_SEQUENCE. Thus when switching from the long window transform to frames with eight short windows a LONG_START_SEQUENCE has to be inserted, resp. when switching back from short to long a STOP_WINDOW_SEQUENCE is neeeded. Therefore there needs to be a lookahead of 1024+576 samples for the blockswitch decision (see figure below).

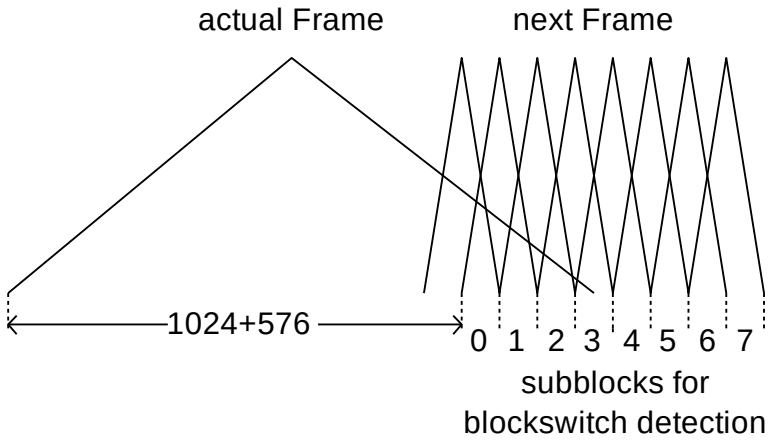

Figure 2: Blockswitch detection lookahead

A high pass IIR-Filter with the transfer function

$$H(z) = \frac{0.7548 \cdot (z-1)}{z - 0.5095}$$

is applied to the samples. After filtering, eight subblock energies are calculated by summing up 128 consecutive squared samples. These eight subblock energies represent the eight short windows of the next frame.

An attack is detected if one of these subblock energies exceeds a sliding average of the previous energies by a constant factor $attackRatio$ and is greater than a constant energy level $minAttackNrg = 1 \cdot 10^{-3}$. The value for $attackRatio$ depends on the bitrate and the number of channels:

$$\text{for mono: } attackRatio = \begin{cases} 18 & \text{for } br \le 24\text{kbps} \\ 10 & \text{for } br > 24\text{kbps} \end{cases}$$

$$\text{for stereo: } attackRatio = \begin{cases} 18 & \text{for } br \le 32\text{kbps} \\ 10 & \text{for } br > 32\text{kbps} \end{cases}$$

The window sequence of the next frame is now either set to ONLY_LONG_SEQUENCE or EIGHT_SHORT_SEQUENCE. Now the final window sequence of the actual frame can be determined by obeying the following the rules:

1. after a long window there can be a long or a start window
2. after a start window there will always be a short window sequence
3. after a short window sequence follows another short window sequence or a stop window
4. after a stop window there can be a long window or a start window

If the current window sequence is and EIGHT_SHORT_SEQUENCE, the eight windows will be grouped to reduce the sideinfo for transmitting scalefactors etc. If no attack has been detected in the actual short window sequence, there will be 3 groups with the first 3 subwindows in the first group, the next 3 in the second group and the last 2 subwindows will be in the third group. For short window sequences with attack there will always be 4 groups. The number of subwindows in each group depends on the position of the attack subwindow:

Table 1: Grouping of subwindows in an EIGHT_SHORT_SEQUENCE

position of attack	nr of subwindows group 1	nr of subwindows group 2	nr of subwindows group 3	nr of subwindows group 4
0	1	3	3	1
1	1	1	3	3
2	2	1	3	2
3	3	1	3	1
4	3	1	1	3
5	3	2	1	2
6	3	3	1	1
7	3	3	1	1

In case of stereo encoding, the blocktype for both channels must be the same to be able to apply the M/S stereo tool. The final common blocktype is chosen as shown in the following table:

Table 2: window sequence synchronization for stereo

blocktype channel 0	blocktype channel 1	final blocktype for both channels
Long	long	long
Long	start	start
Long	short	short
Long	stop	stop
Start	long	start
Start	start	start
Start	short	short
Start	stop	short
Short	long	short
Short	start	short
Short	short	short
Short	stop	short
Stop	long	stop
Stop	start	short
Stop	short	short
Stop	stop	stop

If the final window sequence is EIGHT_SHORT_SEQUENCE the grouping is chosen from the channel containing the higher maximum subblock energy.

5.4.2 Threshold Calculation

The following are the necessary steps for the calculation of the psychoacoustic threshold $thr(n)$, which is an upper limit for the quantization noise of the coder.

5.4.2.1 Calculation of the energy spectrum

The energy spectrum in the coder scalefactor band domain $en(n)$ is calculated by using the output values $X(k)$ of the MDCT-transform that are later quantized and coded. This is done by the following equation:

$$en(n) = \sum_{k=kOffset(n)}^{kOffset(n+1)-1} X(k) \cdot X(k)$$

Here $kOffset(n)$ is the first spectral line of scalefactor band n.

In this psychoacoustic model no threshold partition is used. The threshold calculation is performed directly in the scalefactor band domain.

5.4.2.2 From energy to threshold

No difference is made between tonal and noisy components in the signal. Therefore the "worst case" is assumed, i.e. the signal is tonal for the complete frequency range. Thresholds must be achieved that result in a "transparent" audio quality.

The decrease of the energy is done by a constant required signal to noise ratio SNR which is 29dB for AAC . The scaled thresholds $thr_{scaled}(n)$ are:

$$thr_{scaled}(n) = \frac{en(n)}{SNR}$$

5.4.2.3 Spreading

Instead of a convolution of the spectral energy with a spreading function, a simpler spreading is calculated. Here the slope to higher frequencies is created by weighting the previous threshold value with a frequency dependent factor $s_h(n)$ and by building the maximum of the threshold value of the actual band with this weighted threshold of the previous band.

$$thr'_{spr}(n) = \max(thr_{scaled}(n), s_h(n) \cdot thr_{scaled}(n-1))$$

Accordingly the steeper slope towards the low frequencies is computed by another pass beginning at the highest band and weighting the energy values by a factor $s_l(n)$.

$$thr_{spr}(n) = \max(thr'_{spr}(n), s_l(n) \cdot thr'_{spr}(n+1))$$

The values for $s_h(n)$ resp. $s_l(n)$ are calculated by the distance of the adjacent bands in Bark and a constant slope that is 15dB/Bark for the first and 30dB/Bark for the second equation.

5.4.2.4 Threshold in quiet

The comparison with the threshold in quiet $thr_{quiet}(n)$ is a simple maximum operation.

$$thr_q(n) = \max(thr_{spr}(n), thr_{quiet}(n))$$

The threshold in quiet is given as array for the Bark scale. Because of the difference of the scalefactor band scale compared to the Bark scale, the minimum of the threshold in quiet for the Bark values at the lower and the upper end of the scalefactor band is used.

5.4.2.5 Pre-echo control

The pre-echo control operates as in the psychoacoustic model of [2]. To avoid pre-echos the actual threshold $thr_q(n)$ is compared to the previous threshold $thr_{q,-1}(n)$:

$$thr(n) = \max(rpmin \cdot thr_q(n), \min(thr_q(n), rpelev \cdot thr_{q,-1}(n)))$$

with the parameters $rpelev = 2$ and $rpmin = 0.01$.

No pre-echo control can be calculated in case of blockswitching, because the psychoacoustic model doesn't calculate the thresholds for both long and short blocks simultaneous and the pre-echo control needs the thresholds of the previous block with the same scalefactor band partition as the actual block. Thus the pre-echo control is inactive for the first short window (but not all short windows in a short frame) after a start block and for all frames with a stop window sequence.

5.4.3 Spreaded Energy Calculation

After an eventual filtering of the mdct spectrum with the TNS analysis filter (see 5.5.1), the energy calculation of section 5.4.2.1 has to be performed again. Spreading this energy $en(n)$ the same way as the thresholds (see 5.4.2.3) yields the spreaded energy $es(n)$ in the scalefactor band domain. The values for $s_h(n)$ resp. $s_l(n)$ are dependent on the blocktype and are derived from constant slopes in the bark domain.

For long block $s_l(n) = 30\text{dB/bark} \cdot \Delta bark(n)$ and

$$s_h(n) = \begin{cases} 20\text{dB/bark} \cdot \Delta bark(n) & \text{for } bitrate/channel > 22\text{kbit/s} \\ 15\text{dB/bark} \cdot \Delta bark(n) & \text{for } bitrate/channel \le 22\text{kbit/s} \end{cases}$$

and for short blocks $s_l(n) = 20\text{dB/bark} \cdot \Delta bark(n)$ and $s_h(n) = 15\text{dB/bark} \cdot \Delta bark(n)$, with

$\Delta bark(n)$ as the width in bark of the scalefactor band n

Both the scalefactor band energy $en(n)$ after TNS and the spreaded energy $es(n)$ are important input values for the determination of which bands must not be quantized to zero (see section 5.6.1.1.2. for more details on the avoidance of spectral holes).

5.4.4 Grouping

If the window sequence of the current frame is an EIGHT_SHORT_SEQUENCE, a grouping configuration has been determined by the blockswitching algorithm described above. The psychoacoustic model calculates thresholds, energies and other variables in the subwindow domain. The scalefactor band based thresholds and energies are grouped by adding up the values of all subwindows belonging to one group. The spectrum has to be reordered to match the new combined scalefactor bands.

5.5 Tools

5.5.1 Temporal Noise Shaping (TNS)

For a general description of TNS see [2]. If TNS is active in this encoder, only one filter per MDCT-spectrum will be applied. The steps in TNS encoding are described below. TNS is always calculated on a per subwindow basis, so in case of an EIGHT_SHORT_SEQUENCE window sequence these steps have to be applied once for each of the eight subwindows.

5.5.1.1 TNS detection

Out of the spectral coefficients $X(k)$ a weighted spectrum $X_w(k) = X(k) \cdot wfac(k)$ is calculated. The weighting factors are determined from the energy of the appropriate scalefactor band $wfac(k) = \frac{1}{\sqrt{en(n)}}$. For a definition of the scale factor band energy $en(n)$ see section 5.4.2.1. The factors are smoothed by filtering down:

```
for (k=lpcStopLine-2; k>=lpcStartLine; k--) {
  wfac[k] = (wfac[k] + wfac[k+1]) / 2;
}
```

and up:

```
for (k=lpcStartLine+1; k<lpcStopLine; k++) {
  wfac[k] = (wfac[k] + wfac[k-1]) / 2;
}
```

The lower and upper limits `lpcStartLine` and `lpcStopLine` depend on the bitrate and the blocktype. Next steps are an autocorrelation calculation and a LPC calculation using the Levinson-Durbin algorithm. As result so called parcor or reflection coefficients rq and the prediction gain are available. TNS will be used only if the prediction gain is greater than a given threshold, which is bitrate dependent and varies between 1.2 and 1.41.

5.5.1.2 TNS Stereo Synchronization

If prediction gains for the left and right channel differ only less than 3%, the same TNS filter coefficients are chosen for both channels by copying the TNS data of the left channel to the right channel.

5.5.1.3 TNS Order

The TNS parcor coefficients will be quantized with a resolution of 4 bits for long blocks and 3 bits for short blocks. The order of the coefficients is now determined by going down from the maximum order until the first coefficient that exceeds an absolute value of 0.1 has been reached.

5.5.1.4 TNS Filtering

The spectral coefficients $X(k)$ will now be replaced by filtering with the parcor coefficients. The first scalefactor band affected corresponds to a frequency of 1275Hz for long blocks resp. 2750Hz for short blocks. The filtering is done with the help of a so called lattice filter, no conversion from parcor coefficients rq to linear prediction coefficients is required.

5.5.1.5 Threshold modification

In the frequency range from 380Hz to the start frequency of the TNS filter the coding demands will be increased by multiplying a factor of 0.25 to the thresholds $thr(n)$ calculated by the psychoacoustic model.

5.5.2 Mid/Side Stereo

Normal stereo operation, and thus Mid/Side Stereo, is only required when operating the encoder at bitrates at or above 44 kbit/s. Below 44 kbit/s the Parametric Stereo coding tool [5] is used instead where the AAC core is operated in mono.

Within Mid/Side Stereo, for each scalefactor band the left and right channel coefficients are either coded as L and R or as mid and side channel

$$M = \frac{L+R}{2} \text{ and } S = \frac{L-R}{2}.$$

For stereo in the psychoacoustic model in addition to the left and right energies $en_{L,R}(n)$ also the mid and side energies $en_{M,S}(n)$ are calculated. The threshold for coding mid and side channel is simply the minimum of left and right thresholds $thr_{L,R}(n)$. M/S coding is actually used if

$$\frac{\min(thr_L(n), thr_R(n))^2}{en_M(n) \cdot en_S(n)} \geq \frac{thr_L(n) \cdot thr_R(n)}{en_L(n) \cdot en_R(n)}$$

is fulfilled. In such a case left channel values for spectral coefficients, energies and thresholds will be replaced by the mid channel values, resp. right channel values will be replaced by the side channel values. The spreaded energy $es(n)$ for mid and side channel will be the minium of the spreaded energy of left and right channel.

5.6 Quantization and coding

5.6.1 Reduction of psychoacoustic requirements

Usually the requirements of the psychoacoustic model are too strong for the desired bitrate. Thus a threshold reduction strategy is necessary, i.e. the strategy reduces the requirements by increasing the thresholds given by the psychoacoustic model. An overcoding, i.e. decreasing the thresholds for a finer quantization, doesn't take place in this encoder.
In this section the strategy to reduce the requirements for the quantization accuracy is presented. The first section explains the technique to modify the thresholds calculated by the psychoacoustic model. The reduction strategy has to operate with estimations of the bit demand to avoid multiple quantization and bit counting. This is done by an estimation of the used bits by the perceptual entropy described in the second section. Finally the method how to find the amount of threshold reduction for a given bit demand is presented.

5.6.1.1 Principle of the threshold reduction strategy

5.6.1.1.1 Addition of noise with equal loudness

An increase of the thresholds $thr(n)$ from the psychoacoustic model is done in the form that the loudness of the disturbance is equal for all bands. Here the loudness l for additional noise is approximated by the equation $l = n^{0.25}$, where n is the energy of the noise. To increase the masking threshold equally loud over the whole frequency range, to each scalefactor band the constant loudness r will be added:

$$thr_r(n) = (thr(n)^{0.25} + r)^4$$

The thresholds are converted to the loudness domain and after the addition of the constant loudness there is a conversion back to the energy domain.

5.6.1.1.2 Avoidance of spectral holes

The basic form of threshold reduction described above is insufficient to guarantee an adequate audio quality. There will be too many bands where the quantization sets all spectral values to zero, i.e. audible holes in the frequency domain will occur. This problem can be solved with the help of an additional strategy to avoid holes. The bands that must not be quantized to zero are selected. The value of the increased threshold $thr_r(n)$, which is determined by the previous equation, in such bands must not exceed the energy $en(n)$ in this band diminished by a minimum signal to noise ratio $minSnr(n)$. This is done by building the minimum:

$$thr_r(n) = \min((thr(n)^{0.25} + r)^4, \frac{en(n)}{minSnr(n)})$$

The minimum requirements $minSNR(n)$ are frequency dependent and will be calculated for the given bitrate on initialization of the encoder. First the number of average bits per channel and transform block $avgChBits$ have to be converted in a perceptual entropy pe by calculating:

$$pe = 1.18 \cdot avgChBits$$

Of this total pe 60% are equally divided among the number of active barks $nBark$, that is determined by the maximum bandwidth of the AAC encoder.

$$barkPe_{min} = \frac{0.6 \cdot pe}{nBark}$$

For each scalefactor band the corresponding part of the perceptual entropy called $sfbPe_{min}(n)$ is calculated by converting $barkPe$ to the width in Bark of the appropriate band. With the following equation $minSNR(n)$ is calculated from the $sfbPe_{min}(n)$:

$$minSnr(n) = \frac{1}{2^{sfbPe_{min}(n)/w(n)} - 1.5}$$

The value $w(n)$ holds the number of spectral lines in the scalefactor band n.

Finally the value of $minSNR(n)$ is limited to a maximum of 25dB and a minimum of 1dB.

A signal dependent modification of the minimum requirements is performed by increasing the minimum distance between energy and threshold on local peaks and decrease it for local valleys of the scalefactor band energy $en(n)$.

In case of M/S stereo another modification of $minSNR(n)$ takes place. Depending on the energy difference between mid and side channel the requirements for the weaker channel will be released.

5.6.1.1.3 Relation between bit demand and perceptual entropy

As there is only one quantization and the used bits will only be counted thereafter, the reduction strategy works with an estimation of these used bits. This is done by using the so called perceptual entropy PE. The PE used in this encoder is computed per scalefactor band:

$$sfbPe = nl \cdot \begin{cases} \log_2(\frac{en}{thr}) & \text{for } \log_2(\frac{en}{thr}) \geq c1 \\ (c2 + c3 \cdot \log_2(\frac{en}{thr})) & \text{for } \log_2(\frac{en}{thr}) < c1 \end{cases}$$

with $c1 = \log_2(8)$, $c2 = \log_2(2.5)$, $c3 = 1 - c2/c1$. The estimated number of lines that won't be zero after the quantization is called nl. This number is derived from the form factor $ffac(n) = \sum_{k=kOffset(n)}^{kOffset(n+1)-1} \sqrt{|X(k)|}$ that is also needed by the scalefactor estimator:

$$nl = \frac{ffac(n)}{(\frac{en(n)}{kOffset(n+1) - kOffset(n)})^{0.25}}$$

The total PE pe of one frame is the sum of the scalefactor band perceptual entropies:

$$pe = peOffset + \sum_n sfbPe(n)$$

To get a more linear relation between PE and the number of bits needed a constant value $peOffset$ is added to the scalefactorband perceptual entropies. The value for $peOffset$ is determined at initialization time:

$$peOffset = \begin{cases} 0 & \text{for } chBitrate > 32000 \\ \max(50, 100 - \frac{100}{32000} \cdot chBitrate) & \text{for } chBitrate \leq 32000 \end{cases}$$

with $chBitrate$ as the bitrate per channel in bits per second

An approximation for the conversion from actual needed bits to perceptual entropy is:

$$pe = 1.18 \cdot bits$$

5.6.1.2 Calculation of Bit Demand

Since the AAC encoder uses a bitreservoir technique, the number of bits used for the actual frame will be variable. The bit demand of the current frame depends on:

- the current fullness of the bitreservoir, a number between 0 (empty) and 1 (full)
- the relative difficulty of the frame, a measure for this is the perceptual entropy pe_0 based on the unmodified psychoacoustic thresholds

The steps to calculate the bit demand for the current frame are:

With help of the bitreservoir fullness, the variables $bitSave$ and $bitSpend$ are calculated according to the two figures below:

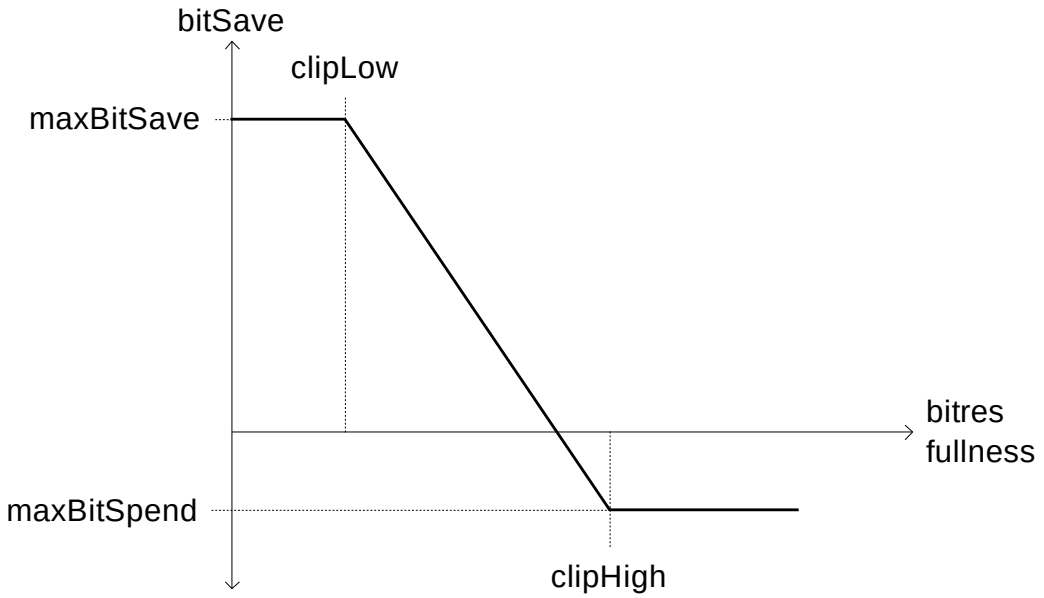

Figure 3: Calclation of bitSave

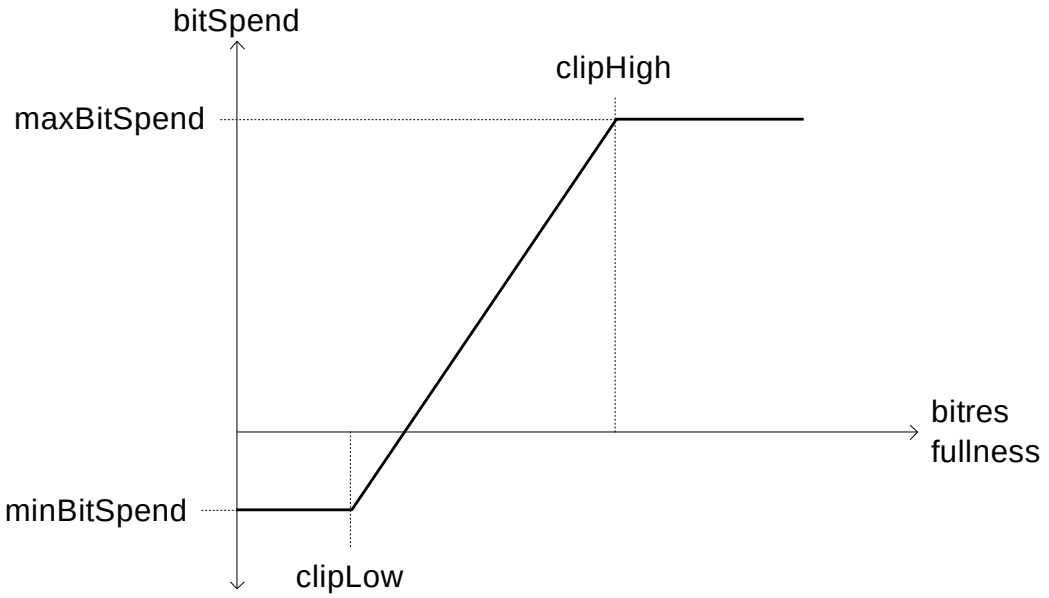

Figure 4: Calclation of bitSpend

The parameters are different for long and short blocks:

Table 3: Parameters for bitreservoir control

Parameters for bitSave calculation		
	Long block parameters	**Short block parameters**
clipLow	0.2	0.2
clipHigh	0.95	0.75
minBitSave	-0.05	0
maxBitSave	0.3	0.2
Parameters for bitSpend calculation		
	Long block parameters	**Short block parameters**
clipLow	0.2	0.2
clipHigh	0.95	0.75
minBitSpend	-0.1	-0.05
maxBitSpend	0.4	0.5

A factor $bitFac$ is calculated out of $bitSave$, $bitSpend$ and the current perceptual entropy pe_0. For a relatively low perceptual entropy (easy to encode frame) this means that bitfac is less than 1 and bits will be put to the bitreservoir. If the perceptual entropy is above the average which is an indication of a diffcult frame, bits will be taken out of the bitreservoir (bitfac>1). See the next figure on how to calculate this factor.

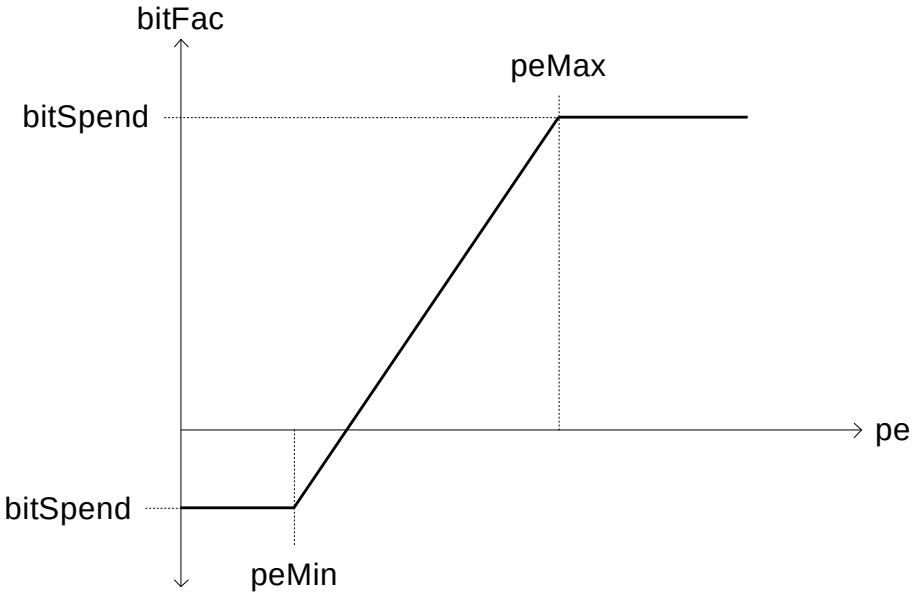

Figure 5: Calclation of bitFac

The variables $peMin$ and $peMax$ are adjusted after each calculation of $bitFac$.

The desired number of bits for the actual frame is:

$$desiredBits = \min(avgFrameBits \cdot bitFac, avgFrameBits + bitreservoirBits)$$

where $avgFrameBits$ is the average number of bits per frame matching the given constant bitrate

and $bitreservoirBits$ is the actual number of bits in the bitreservoir.

5.6.1.3 Calculation of the reduction value

The actual problem of the reduction strategy is to find the loudness value r of the equation defined in section 5.6.1.1.2. so that the requirements of the granted bits resp. the appropriate PE pe_r are fulfilled. In the following an iterative process to find the reduction value r and the thresholds used for the quantization is described.

5.6.1.3.1 Preparatory steps of the perceptual entropy calculation

The calculation of the scalefactor band PEs $sfbPe(n)$ can be split up in a constant part $a(n)$ and a variable, threshold dependent part $b(n) \cdot \log_2(thr(n))$.

$$sfbPe(n) = a(n) - b(n) \cdot \log_2(thr(n))$$

$$\text{with } a(n) = \begin{cases} nl \cdot \log_2(en) & \text{for } \log_2(\frac{en}{thr}) \geq c1 \\ nl \cdot (c2 + c3 \cdot \log_2(en) & \text{for } \log_2(\frac{en}{thr}) < c1 \end{cases}$$

$$\text{and } b(n) = \begin{cases} nl & \text{for } \log_2(\frac{en}{thr}) \geq c1 \\ nl \cdot c3 & \text{for } \log_2(\frac{en}{thr}) < c1 \end{cases}$$

It is possible to calculate the constant part once at the start of the iteration.

5.6.1.3.2 Calculation of the desired perceptual entropy

The goal to spend the number of bits calculated with the method described above in 5.6.1.2 is approximately achieved by increasing the tresholds in such a way that the resulting perceptual entropy equals the desired PE pe_r. The desired PE is determined by the relation between bit demand and perceptual enropy (5.6.1.1.3.).

$$pe_r = 1.18 \cdot desiredBits$$

The actual number of used bits will only approximately match the desired bits. Small differences will be balanced out with the help of the bitreservoir. Systematic differences are compensated by applying a correction factor to the desired perceptual entropy. This factor is obtained by taking into account the real relation between number of used bits and perceptual entropy of the previous frames. The allowed range of the correction factor is between 0.85 and 1.15.

5.6.1.3.3 Selection of the bands for avoidance of holes

First all bands are marked to participate in an eventually occurring avoidance of spectral holes. Then by means of different criteria individual bands are excluded.

- For long blocks no avoidance of holes in band n, if the spreaded energy $es(n) \cdot 0.5$ exceeds the energy $en(n)$
- For short blocks no avoidance of holes in band n, if the spreaded energy $es(n) \cdot 0.63$ exceeds the energy $en(n)$
- The minimum requirement $minSNR(n)$ has to be greater than 0dB

For a definition of the scalefactor band energy $en(n)$ and the spreaded energy $es(n)$, see section 5.4.2.1 resp. section 5.4.3 .

5.6.1.3.4 First Estimation of the reduction value

In the following an approximation for the total PE is used. It is derived from the equation of the scalefactor band PE:

$$pe = a - b \cdot \log_2(t^{0.25} + r)^4 = a - b \cdot 4 \cdot \log_2(t^{0.25} + r)$$

with $a = \sum_n a(n)$ as the sum of the constant parts of $sfbPe(n)$ and $b = \sum_n b(n)$ as the total number of estimated lines that will be unequal zero after quantization. The estimated average total threshold is t.

In the first iteration the loudness $t^{0.25}$ of this average threshold is calculated with help of the PE pe_0 without reduction ($r = 0$):

$$t^{0.25} = 2^{\frac{a - pe_0}{4b}}$$

The estimation of the reduction value r_1 follows from the desired PE pe_r:

$$r_1 = 2^{\frac{a - pe_r}{4b}} - t^{0.25}$$

With this r_1 the new thresholds $thr_1(n)$ can be calculated using the equation from section 5.6.1.1.2. and also the resulting PE pe_1.

5.6.1.3.5 Second Estimation of the reduction value

Usually the value of pe_1 will be greater than the desired PE pe_r. In scalefactor bands that avoid holes after the first iteration the thresholds can not be reduced further. By repeating the two equations above a second guess r_2 for the reduction value can be found, if only the bands that actually do not avoid holes are considered. Therefore the contributions of bands with active avoidance of holes have to be subtracted from a, b, pe_r and pe_1. The modified values are then called a_{nah}, b_{nah}, $pe_{r,nah}$ and $pe_{1,nah}$. The loudness $t_{nah}^{0.25}$ of a new average threshold is calculated by:

$$t_{nah}^{0.25} = 2^{\frac{a_{nah} - pe_{1,nah}}{4 \cdot b_{nah}}}$$

The second estimation of the reduction value r_2 is:

$$r_2 = r_1 + 2^{\frac{a_{nah} - pe_{r,nah}}{4 \cdot b_{nah}}} - t_{nah}^{0.25}$$

Using r_2, new thresholds $thr_2(n)$ and PE pe_2 can be calculated.

The calculation of r_2 may be repeated once more if the absolute difference between desired and actual PE is greater than 5%.

5.6.1.3.6 Final threshold modification by linearization

If the PE resulting from the second iteration pe_2 is already close to the desired value pe_r (that means the difference to the desired PE is less than 15%), the desired PE can be reached by a linearization of the logarithms.

The formula for the PE after the second guess is:

$$pe_2 = \sum_n a(n) - b(n) \cdot \log_2(thr_2(n)) = \sum_n a(n) - b(n) \cdot 4 \cdot \log_2(thr_2(n)^{0.25})$$

Accordingly the desired PE pe_r can be written as:

$$pe_r = \sum_n a(n) - b(n) \cdot 4 \cdot \log_2(thr_2(n)^{0.25} + \Delta r)$$

with Δr as the difference between the reduction value r and the latest guess r_2.
The difference of the two perceptual entropies is:

$$\Delta pe = pe_2 - pe_r = \sum_n b(n) \cdot 4 \cdot \log_2\left(1 + \frac{\Delta r}{thr_2(n)^{0.25}}\right)$$

A linearization of the logarithm around the zero results to the following:

$$\Delta pe \approx \frac{4}{\Delta r \cdot \ln(2)} \sum_n \frac{b(n)}{thr_2(n)^{0.25}}$$

Now the difference of the total PE can be divided among the individual bands, that actuallly do not avoid holes:

$$\Delta sfbPe(n) = \frac{b(n)}{thr_2(n)^{0.25}} \cdot \frac{1}{normFac} \cdot \Delta pe$$

with

$$normFac = \sum_n \frac{b(n)}{thr_2(n)^{0.25}}$$

For each band a final modification of the threshold is performed:

$$thr_3(n) = thr_2(n) \cdot 2^{\Delta sfbPe(n)/b(n)}$$

5.6.1.3.7 Further perceptual entropy reduction

If the conditions for section 5.6.1.3.6. can not be reached, i.e. the actual perceptual entropy pe_2 exceeds the desired pe_r by more than 15%, then it seems that the constraints given by the minum requirements $minSNR(n)$ or the number of bands with active avoidance of holes are too strong for the desired PE.

In a first step the values of $minSNR(n)$ are limited to a maximum value of 1dB starting from the scalefactor band with the highest frequency. By doing so, thresholds can be increased and the perceptual entropy will decrease.

If the actual perceptual entropy is still too large after having changed $minSNR(n)$ for the whole spectrum, more spectral holes have to be allowed. In case of M/S stereo always the scalefactor band of the channel with less energy is now quantized to zero. Afterwards for mono and stereo subsequently bands with low engergies get erased. Therefore the bands are partitioned into four categories with different energy levels. Starting from the highest band all bands falling in the category with the lowest category get erased. This process is eventually repeated with the next energy categories, until the resulting perceptual entropy is as small as the desired value of pe_r.

5.6.1.3.8 Possible failures

In general the difference of the resulting perceptual entropy to the desired pe_r is negligible. The described algorithm works fine for reasonable combinations of bitrate, samplerate and bandwidth. Normally inaccuracies especially in the relation between the perceptual entropy and the really used bits, can be balanced out by the bitreservoir. But there is no guarantee that there are always enough bits available to fulfill the requirements of the increased threshold. For measures that are available to avoid an abort of the encoder in such cases, see section 5.6.4 .

5.6.2 Scalefactor determination

The scalefactors determine the quanization step size for each scalefactor band. By changing the scalefactor, the quantization noise will be controlled. The equation for the quantization of the spectral coefficients is:

$$X_{quant}(k) = \mathrm{sgn}\left(X(k)\right) \cdot \mathrm{int}\left(\left(\left|X(k)\right| \cdot 2^{\frac{1}{4}\cdot(scf - globalGain)}\right)^{\frac{3}{4}} + MAGIC\_NUMBER\right)$$

$MAGIC\_NUMBER$ is defined to 0.4054 and $X(k)$ is one of the spectral coefficients that is calculated from the MDCT filterbank. In the following three steps of combined scalefactor determination and quantization, always $scfGain(n) = globalGain - scf(n)$ is calculated. Only at the end, scalefactors $scf(n)$ and the $globalGain$ values are derived from the values for $scfGain(n)$.

The formula for the inverse quantization is:

$$X_{invquant}(k) = \mathrm{sgn}\left(X_{quant}(k)\right) \cdot \left|X_{quant}(k)\right|^{\frac{4}{3}} \cdot 2^{-\frac{1}{4}\cdot(scf - globalGain)}$$

It is needed for calculating the quantization noise.

5.6.2.1 Scalefactor Estimation

A first guess of $scfGain(n)$ that results in quantization noise approximately equal to the threshold $thr_r(n)$ is given by the following equation:

$$scfGain(n) = \mathrm{floor}\left(8.8585 \cdot \left(\lg\left(6.75 \cdot thr_r(n)\right) - \lg(ffac(n))\right)\right)$$

with the form factor $ffac(n) = \sum_{k=kOffset(n)}^{kOffset(n+1)-1} \sqrt{|X(k)|}$ that is also needed by the calculation of the perceptual entropy (see 5.6.1.1.3.).

5.6.2.2 Scalefactor Improvement by Quantization

The following steps of scalefactor changes include always a quantization and inverse quantization procedure to be able to calculate and compare the quantization error. The equation for the distortion is:

$$sfbDist(n) = \sum_{k=kOffset(n)}^{kOffset(n+1)-1} \left(X(k) - X_{invquant}(k)\right)^2$$

After quantizing with the scalefactor value calculated in 5.6.2.1, the resulting distortion $sfbDist(n)$ may be greater than the threshold $thr_r(n)$. By trying increased and decreased values for $scfGain(n)$ a lower distortion is searched for. If the distortion was already below the threshold, a search will only be done for smaller values of $scfGain(n)$ to try to further improve the distortion.

5.6.2.3 Scalefactor Difference Reduction

Above each scale factor band is treated individually. The next algorithms take into acount that finally the difference of the scalefactors will be encoded. A smaller difference between two adjacent scale factors costs less bits.

In a first step it is searched for single scalefactor bands, where the number of bits gained by using a smaller $scfGain(n)$ is greater than the estimated increased bit demand for the noiseless coding of the quantized spectral coefficients. The estimation of needed bits for the noiseless coding is based on the equation for the perceptual entropy:

$$ldRatio = \log_2\left(en(n)\right) - 0.375 \cdot scfGain(n)$$

$$nBits_{estim} = \begin{cases} 0.7 \cdot nLines \cdot ldRatio & \text{for } ldRatio \geq c1 \\ 0.7 \cdot nLines \cdot (c2 + c3 \cdot ldRatio) & \text{for } ldRatio < c1 \end{cases}$$

with $c1 = \log_2(8)$, $c2 = \log_2(2.5)$, $c3 = 1 - c2/c1$. With nl the estimated number of lines that won't be zero after the quantization is meant. This number has already been calculated during the adaptation of the thresholds to the bitrate, for a definition of nl see 5.6.1.1.3.

If such a band is found and in addition the quantization error is smaller, the new value for $scfGain(n)$ is accepted.

In a second assimilation step the same procedure as above is repeated but now trying to increase the $scfGain(n)$ values for a complete region of scale factor bands instead of improving only single bands.

5.6.2.4 Final scalefactor determination

The conversion from $scfGain(n)$ to scalefactor values $scf(n)$ and a value for $globalGain$ is done after limitation of the maximum scalefactor difference to a value of 60. This is achieved by limiting all values of $scfGain(n)$ to a maximum allowed value of $minScfGain + 60$ with $minScfGain$ as the miniumover all bands.

The value for $globalGain$ is now chosen as the maximum of all $scfGain(n)$ values and the scalefactors result in:

$$scf(n) = globalGain - scfGain(n)$$

5.6.3 Noiseless coding

Coding of the quantized spectral coefficients is done by the noiseless coding. The encoder uses a so called greedy merge algorithm to segment the 1024 coefficients of a frame into section and to find the best huffman codebook for each section. For the sectioning and the huffman codebooks see [2].

5.6.4 Out of Bits Prevention

Only after the MDCT values are quantized according to the increased thresholds $thr_r(n)$ and after the following noiseless coding, the number of really needed bits is counted. If this number is too high, the number of bits have to be reduced. This is achieved by increasing the global gain value and a new quantization of the whole spectrum plus additional noiseless coding in a loop until the bit demand is small enough to match the constraints of the bitreservoir.

Annex A (informative): Change history

Change history							
Date	TSG SA#	TSG Doc.	CR	Rev	Subject/Comment	Old	New
2004-09	25	SP-040635			Approved at SA#25 Plenary	2.0.0	6.0.0
2006-06	32	SP-060360	0002		Modification of written specification: Change of encoder bitrate border for Parametric Stereo usage	6.0.0	7.0.0
2008-12	42				Version for Release 8	7.0.0	8.0.0
2009-12	46				Version for Release 9	8.0.0	9.0.0
2011-03	51				Version for Release 10	9.0.0	10.0.0
2012-09	57				Version for Release 11	10.0.0	11.0.0