



INTERNATIONAL STANDARD ISO/IEC 23003-3:2012
TECHNICAL CORRIGENDUM 1

Published 2012-09-01

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — MPEG audio technologies —
Part 3:
Unified speech and audio coding

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Technologies audio MPEG —

Partie 3: Discours unifié et codage audio

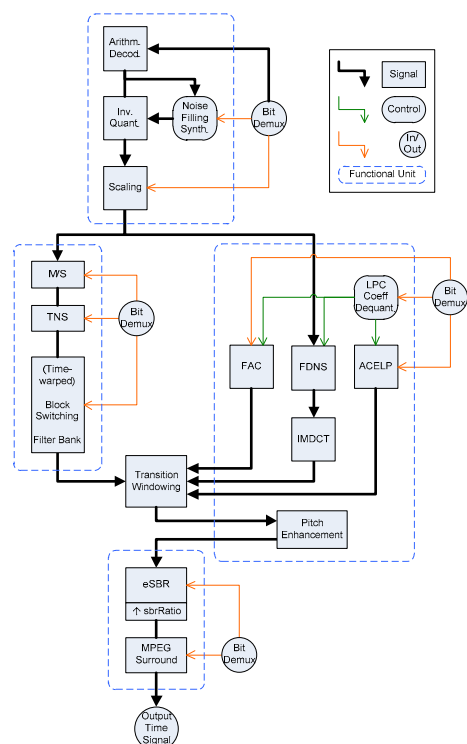
RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC 23003-3:2012 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

In 3.2 add at the end:

$v[] = \{a\}$ This expression indicates that all elements of the array **v** shall be set to the value **a**.


Documents
 Document Sharing Hub
 © ISO/IEC 2012 – All rights reserved



In 4.2. *replace*:

The filterbank / block switching tool applies the inverse of the frequency mapping that was carried out in the encoder. An inverse modified discrete cosine transform (IMDCT) is used for the filterbank tool. The IMDCT can be configured to support 120, 128, 240, 256, 480, 512, 960 or 1024 spectral coefficients.

with:

The filterbank / block switching tool applies the inverse of the frequency mapping that was carried out in the encoder. An inverse modified discrete cosine transform (IMDCT) is used for the filterbank tool. The IMDCT can be configured to support 96, 128, 192, 256, 384, 512, 768, or 1024 spectral coefficients.

In 5.2 in Table 6, *replace*:

<pre>[...] case: ID_USAC_EXT UsacExtElementConfig(); break; } }</pre>	
NOTE: UsacSingleChannelElementConfig(), UsacChannelPairElementConfig(), UsacLfeElementConfig() and UsacExtElementConfig() signaled at position elemIdx refer to the corresponding elements in UsacFrame() at the respective position elemIdx.	

with

<pre>[...] case: ID_USAC_EXT UsacExtElementConfig(); break; } }</pre>	
NOTE: UsacSingleChannelElementConfig(), UsacChannelPairElementConfig(), UsacLfeElementConfig() and UsacExtElementConfig() signaled at position elemIdx refer to the corresponding elements in UsacFrame() at the respective position elemIdx.	

In 5.3.1 in Table 17 *replace*:

<pre>[...] case: ID_USAC_EXT UsacExtElement(usacIndependencyFlag); break; } }</pre>	
---	--

with

<pre>[...] case: ID_USAC_EXT UsacExtElement(usacIndependencyFlag); break; } }</pre>	
---	--

In 5.3.1 in Table 21 replace:

```
[...]
    if (usacExtElementUseDefaultLength) {
        usacExtElementPayloadLength = usacExtElementDefaultLength;
    } else {
        usacExtElementPayloadLength = escapedValue(8,16,0);
    }
[...]
```

with:

```
[...]
    if (usacExtElementUseDefaultLength) {
        usacExtElementPayloadLength = usacExtElementDefaultLength;
    } else {
        usacExtElementPayloadLength;                8           uimsbf
        if (usacExtElementPayloadLength==255) {
            valueAdd                                16          uimsbf
            usacExtElementPayloadLength += valueAdd - 2;
        }
    }
[...]
```

In 5.3.2 in Table 23 replace:

```
[...]
    if (nrChannels == 2) {
        StereoCoreToolInfo(core_mode);
    }
[...]
```

with

```
[...]
    if (nrChannels == 2) {
        StereoCoreToolInfo(core_mode, indepFlag);
    }
[...]
```

In 5.3.2 in Table 24 replace:

Syntax	No. of bits	Mnemonic
<pre> StereoCoreToolInfo(core_mode) { if (core_mode[0] == 0 && core_mode[1] == 0) { tns_active; common_window; if (common_window) { ics_info(); common_max_sfb; if (common_max_sfb == 0) { </pre>		
tns_active;	1	uimsbf
common_window;	1	uimsbf
common_max_sfb;	1	uimsbf

if (window_sequence == EIGHT_SHORT_SEQUENCE) {		
max_sfb1;	4	uimbsf
} else {		
max_sfb1;	6	uimbsf
}		
} else {		
max_sfb1 = max_sfb;		
}		
max_sfb_ste = max(max_sfb, max_sfb1);		
ms_mask_present;	2	uimbsf
if (ms_mask_present == 1) {		
for (g = 0; g < num_window_groups; g++) {		
for (sfb = 0; sfb < max_sfb; sfb++) {		
ms_used[g][sfb];	1	uimbsf
}		
}		
}		
if (ms_mask_present == 3) {		
cplx_pred_data();		
} else {		
alpha_q_re[g][sfb] = 0;		
alpha_q_im[g][sfb] = 0;		
}		
}		
[...]		

with

Syntax	No. of bits	Mnemonic
StereoCoreToolInfo(core_mode, indepFlag)		
{		
if (core_mode[0] == 0 && core_mode[1] == 0) {		
tns_active;	1	uimbsf
common_window;	1	uimbsf
if (common_window) {		
ics_info();		
common_max_sfb;	1	uimbsf
if (common_max_sfb == 0) {		
if (window_sequence == EIGHT_SHORT_SEQUENCE) {		
max_sfb1;	4	uimbsf
} else {		
max_sfb1;	6	uimbsf
}		
} else {		
max_sfb1 = max_sfb;		
}		
max_sfb_ste = max(max_sfb, max_sfb1);		
ms_mask_present;	2	uimbsf
if (ms_mask_present == 1) {		
for (g = 0; g < num_window_groups; g++) {		
for (sfb = 0; sfb < max_sfb_ste; sfb++) {		
ms_used[g][sfb];	1	uimbsf
}		
}		
}		
if (ms_mask_present == 3) {		
cplx_pred_data(max_sfb_ste, indepFlag);		
} else {		

```

        alpha_q_re[] = {0};
        alpha_q_im[] = {0};
    }
}
[...]
```

In 5.3.2 in Table 38 – Syntax of *arith_data*, replace:

`pki = arith_get_pk(c + esc_nb<<17)`

with:

`pki = arith_get_pk(c + (esc_nb<<17));`

In 6.1.1.1 replace:

usacSamplingFrequency Output sampling frequency of the decoder coded as unsigned integer value in case `usacSamplingFrequencyIndex` equals zero.

with:

usacSamplingFrequency Output sampling frequency of the decoder coded as unsigned integer value in case `usacSamplingFrequencyIndex` is equal to the escape value.

In 6.1.1.1 add definition of **bs_pvc** by replacing:

bs_interTes This flag signals the usage of the inter-TES tool in SBR.

with

bs_interTes This flag signals the usage of the inter-TES tool in SBR.

bs_pvc This flag signals the usage of the PVC tool in SBR.

In 6.1.1.2, replace:

bsStereoSbr This flag signals the usage of the stereo SBR in combination with MPEG Surround decoding.

with

bsStereoSbr This flag signals the usage of the stereo SBR in combination with MPEG Surround decoding. The value of `bsStereoSbr` is defined by `stereoConfigIndex` (see Table 72).