Technical Specification Group Services and System Aspects **TSGS#26(04)0841**
Meeting #26, Athens, Greece
13-16 December 2004

Source: **TSG-SA WG4**

Title: CRs TS 26.290 and TS 26.304 on AMR-WB+ codec (Release 6)

Document for: **Approval**

Agenda Item: **7.4.3**

The following CRs, agreed at the TSG-SA WG4 meeting #33, are presented to TSG SA #26 for approval.

Spec	CR	Rev	Phase	Subject	Cat	Vers	WG	Meeting	S4 doc
26.290	001		Rel-6	Correction of stereo bit allocation tables	F	6.0.0	S4	TSG-SA WG4#33	S4-040711
26.290	002		Rel-6	Correction of storage format for AMR-WB+	F	6.0.0	S4	TSG-SA WG4#33	S4-040712
26.290	003	1	Rel-6	Editorial changes	D	6.0.0	S4	TSG-SA WG4#33	S4-040763
26.304	001		Rel-6	Incorrect definition of mode index for SID frames	F	6.0.0	S4	TSG-SA WG4#33	S4-040684
26.304	002		Rel-6	Correction of TCX coding selection for MMS encoder	F	6.0.0	S4	TSG-SA WG4#33	S4-040685
26.304	003		Rel-6	Misread of energy buffer in coding mode selection in MMS encoder. Correction of energy buffer initialisation	F	6.0.0	S4	TSG-SA WG4#33	S4-040686
26.304	004		Rel-6	Correction of stereo bit allocation tables	F	6.0.0	S4	TSG-SA WG4#33	S4-040713
26.304	005	1	Rel-6	Optimization of error concealment operation	F	6.0.0	S4	TSG-SA WG4#33	S4-040764
26.304	006	1	Rel-6	Stereo operation of pre-echo mode, saturation of gain_shape	F	6.0.0	S4	TSG-SA WG4#33	S4-040765
26.304	007		Rel-6	Stereo operation of pre-echo mode, alignement of encoder and decoder	F	6.0.0	S4	TSG-SA WG4#33	S4-040716
26.304	008	1	Rel-6	Addition of support for file formats and improved command line	D	6.0.0	S4	TSG-SA WG4#33	S4-040768
26.304	009	1	Rel-6	Source code editorial changes	D	6.0.0	S4	TSG-SA WG4#33	S4-040767
26.304	010		Rel-6	Removal of complexity counters	D	6.0.0	S4	TSG-SA WG4#33	S4-040719
26.304	011	1	Rel-6	Editorial changes	D	6.0.0	S4	TSG-SA WG4#33	S4-040780
26.304	012		Rel-6	Editorial changes	D	6.0.0	S4	TSG-SA WG4#33	S4-040722
26.304	013		Rel-6	Removal of the eid tool	D	6.0.0	S4	TSG-SA WG4#33	S4-040723
26.304	014	1	Rel-6	Addition of frame erasures simulation at the decoder	D	6.0.0	S4	TSG-SA WG4#33	S4-040766
26.304	015		Rel-6	Removal of two unused stereo rate	D	6.0.0	S4	TSG-SA WG4#33	S4-040725
26.304	016		Rel-6	Source code editorial changes	D	6.0.0	S4	TSG-SA WG4#33	S4-040726

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040711

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.290** **CR** **001** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Correction of stereo bit allocation tables
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	A reserved bit for the stereo is missing from the bit allocation tables
Summary of change:	⌘	The tables are updated with the correct location of the reserved bit.
Consequences if not approved:	⌘	A wrong interpretation of the bitstream and the location of the reserved bit may lead to the possibility wrong decoding in the case of stereo bitstreams

Clauses affected:	⌘	7			
Other specs affected:	⌘	Y: X / N:	Other core specifications	⌘	26.304
		Y: / N: X	Test specifications		
		Y: / N: X	O&M Specifications		
Other comments:	⌘				

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

7 Detailed bit allocation of the Extended AMR-WB codec

Ö <cut text> Ö

Table 18: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX48 frame type, mode 0 and 1

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
Balance factor	b8-b14	b14-b20
Global gain	b15 ñ b21	b21-b27
Algebraic VQ	b21 ñ bN_1	b28- bN_1

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Balance factor	B9-b15	b15-b21
Global gain	b16 ñ b22	b22-b28
Algebraic VQ	b23 ñ bN_1	b29- bN_1

Table 19a: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX96 frame type, mode 2 - First packet

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
Balance factor	b8-b14	b14-b20
Algebraic VQ	b15 ñ bN_1	b21- bN_1

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Balance factor	B9-b15	b15-b21
Algebraic VQ	b16 ñ bN_1	b22- bN_1

Table 19b: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX96 frame type, mode 2 - Second packet

	~~Bits (MSB-LSB)~~	
~~Description~~	~~N bits/frame <=76~~	~~N bits/frame > 76~~
~~Midband stereo~~		
~~Midband filter~~	~~b0-b3~~	~~b0-b6~~
~~Midband gain~~	~~b4-b5~~	~~b7-b11~~
~~Lowband stereo~~		
~~Mode bits~~	~~b6-b7~~	~~b12-b13~~
~~Global gain~~	~~b8-b14~~	~~b14-b20~~
~~Algebraic VQ~~	~~b15 ñ bN_1~~	~~b21- bN_1~~

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Global gain	b9-b15	b15-b21
Algebraic VQ	b16ñ bN_1	b22- bN_1

Table 20a: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX96 frame type, mode 3 - First packet

	~~Bits (MSB-LSB)~~	
~~Description~~	~~N bits/frame <=76~~	~~N bits/frame > 76~~
~~Midband stereo~~		
~~Midband filter~~	~~b0-b3~~	~~b0-b6~~
~~Midband gain~~	~~b4-b5~~	~~b7-b11~~
~~Lowband stereo~~		
~~Mode bits~~	~~b6-b7~~	~~b12-b13~~
~~Balance factor~~	~~b8-b14~~	~~b14-b20~~
~~Algebraic VQ~~	~~b15 ñ bN_1~~	~~b21- bN_1~~

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Balance factor	b9-b15	b15-b21
Algebraic VQ	b16ñ bN_1	b22- bN_1

Table 20b: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX96 frame type, mode 3 - Second packet

	~~Bits (MSB-LSB)~~	
~~Description~~	~~N bits/frame <=76~~	~~N bits/frame > 76~~
~~Midband stereo~~		
~~Midband filter~~	~~b0-b3~~	~~b0-b6~~
~~Midband gain~~	~~b4-b5~~	~~b7-b11~~
~~Lowband stereo~~		
~~Mode bits~~	~~b6-b7~~	~~b12-b13~~
~~Algebraic VQ~~	~~b8 ñ bN_1~~	~~b14- bN_1~~

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Algebraic VQ	b9 ñ bN_1	b15- bN_1

Table 20c: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX96 frame type, mode 3 - Third packet

	~~Bits (MSB-LSB)~~	
~~Description~~	~~N bits/frame <=76~~	~~N bits/frame > 76~~
~~Midband stereo~~		
~~Midband filter~~	~~b0-b3~~	~~b0-b6~~
~~Midband gain~~	~~b4-b5~~	~~b7-b11~~
~~Lowband stereo~~		
~~Mode bits~~	~~b6-b7~~	~~b12-b13~~
~~Global gain~~	~~b8-b14~~	~~b14-b20~~
~~Algebraic VQ~~	~~b15 ñ bN_1~~	~~b21- bN_1~~

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Global gain	B9-b15	b15-b21
Algebraic VQ	b16 ñ bN_1	b22- bN_1

Table 20d: Stereo encoder output parameters in order of occurrence and bit allocation within the audio frame of TCX96 frame type, mode 3 - Fourth packet

	~~Bits (MSB-LSB)~~	
~~Description~~	~~N bits/frame <=76~~	~~N bits/frame > 76~~
~~Midband stereo~~		
~~Midband filter~~	~~b0-b3~~	~~b0-b6~~
~~Midband gain~~	~~b4-b5~~	~~b7-b11~~
~~Lowband stereo~~		
~~Mode bits~~	~~b6-b7~~	~~b12-b13~~
~~Algebraic VQ~~	~~b8 ñ bN_1~~	~~b14- bN_1~~

	Bits (MSB-LSB)	
Description	N bits/frame <=76	N bits/frame > 76
Midband stereo		
Midband filter	b0-b3	b0-b6
Midband gain	b4-b5	b7-b11
Lowband stereo		
Mode bits	b6-b7	b12-b13
reserved	b8	b14
Algebraic VQ	b9 ñ bN_1	b15- bN_1

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040712

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.290** **CR** **002** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Correction of storage format for AMR-WB+
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMR-WB+
Date:	⌘	14/12/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The way of storing AMR-WB frames as an AMR-WB+ storage unit is undefined.
Summary of change:	⌘	An AMR-WB+ storage unit is clearly defined. It is specified that a storage unit corresponding to AMR-WB modes should contain only one frame, while a storage unit corresponding to the AMR-WB+ extensions is stored as a superframe.
Consequences if not approved:	⌘	A wrong interpretation may lead to different implementations of the way the AMR-WB frames are stored and read. Players may not be interoperable as some implementations may assume that a storage unit contains four AMR-WB frames.

Clauses affected:	⌘	8.3

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	26.244
			X	Test specifications		
			X	O&M Specifications		

Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

8.3 AMR-WB+ File Storage Format

This format is relevant only for file storage and defines ~~the bitstream~~a storage unit contained in an AMR-WB+ sample of a 3GP file [9]. It is quite similar to transport format with the exception that the two-octet header is used once per superframe for AMR-WB+ extension modes and once per frame for AMR-WB modes~~(a storage sample consist of 4 frames or one superframe)~~. Note that in AMR-WB+, the operation code and internal sampling frequency can be switched only on a superframe basis boundaries so the header octets are needed only once per superframe. ~~Note that in a transport format the superframe is split in 4 transport frames for robustness against packet loss.~~
All media streams in a 3GP file are stored in timed units called samples. This format defines the syntax of the basic component of a sample, which is here called a storage unit.~~the syntax of a sample for AMR-WB+~~.

A storage ~~sample~~ unit consists of a two-octet header followed by data octets corresponding to the whole superframe (4 transport frames).

to either:

1. Awhole superframe (4 transport frames) when OC = 10..13 or OC = 16Ö 47 .
2. A frame otherwise

~~The two-octet header contains the operation code (OC) (7 bits) and the ISF mode (5 bits). This is followed by the data octets corresponding to the whole superframe (4 transport frames).~~ For the first case, t~~T~~he number of data octets per superframe is given by 4 times the number of octets per frame (the right-most column in Table 25).

The length of an AMR-WB+ storage ~~sample~~ unit in ms (corresponding to one superframe) depends on the internal sample frequency and given by 80◊ISF/25600 where ISF is the internal sampling frequency in Hz (ISF modes are shown in Table 24).

The header in each ~~superframe~~ storage unit contains the following two octets.

	MSB							LSB
Octet	**bit 8**	**bit 7**	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**
1	0	Operation Code (7 bits)						
2	0	0	0	ISF mode (5 bits)				

Operation code (OC) (7 bits): Indicates the operational setting of the codec used for the corresponding frame (the combination of AMR-WB+ core and stereo mode, the AMR-WB mode, or comfort noise, as specified by Table 25 above).

ISF index (5 bits): Indicates the internal sampling frequency employed for the corresponding frame. The index values correspond to internal sampling frequency as specified in Table 24 above. This field SHALL be set to 0 for operation according to the AMR-WB+ modes defined in table 21 (Operation codes 0-13).

For operation according to OC 0-13 the ISF field shall be set 0 and has no meaning. The frame length for that operation is fixed to 20 ms in time.

The audio data follows the header octets. The number of data octets per ~~superframe~~ storage unit corresponding to ~~a certain~~ operation codes 10..13 and 16Ö 47 ~~is~~ are given as 4 times the number of octets per frame (right-most column in Table 25), for the other operation codes, the number of octets are those corresponding to 1 frame only..

It should be noticed that when OC <10, i.e. AMR-WB frames, the original AMR-WB storage format should be preferred in order to ensure backward decoding compatibility.

Example

The following diagram (Table 27) shows a storage sample of AMR-WB+ using 14 kbit/s coding operation (OC=26) with a superframe length of 4◊35=140 octets. The internal sampling frequency in this example is 25.6 kHz (ISF mode = 8). OC 26 corresponds to mono mode 0 (208 bits/frame) and stereo mode 4 (72 bits/frame).

The data octets are packetized according to the detailed bit allocation given in tables 14 to 20. The first bit of the AMR-WB+ data b0 is placed in bit 8 of octet 3.

~~It should be noticed that when OC <10, i.e. AMR-WB frames, the original AMR-WB storage format should be preferred in order to ensure backward decoding compatibility.~~

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040763

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.290** **CR** **003** ⌘**rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME [X] Radio Access Network ☐ Core Network ☐

Title:	⌘	Editorial changes
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change:	⌘	Some editorial changes that improve the specification and clarify that the transport format is actually just an interface to a transport data format. Alignement with the AMR-WB specification, where the operation code is called frame type.
Summary of change:	⌘	Transport format is renamed to transport interface format Operation code is renamed to frame type
Consequences if not approved:	⌘	May lead to a misinterpretation with the RTP payload format. May lead to misinterpretation of Operation code with respect to the AMR-WB specification

Clauses affected:	⌘	8, 8.1,8.2,8.3

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

8 Storage and ~~Transport~~ Transport Interface formats

The AMR-WB+ codec storage and ~~transport formats~~ transport interface formats are described in this section.

8.1 Available Modes and Bitrates

The AMR-WB+ format contains the AMR-WB modes and a set of AMR-WB+ extension modes.
The AMR-WB+ codec includes the AMR-WB modes, as shown in Table 21 below.

Ö <cut text> Ö

The ~~operation code~~ frame type is used to identify the content of an AMR-WB+ encoded frame. This ~~code~~ type indicates if it is; an AMR-WB mode, Comfort noise, NO_DATA, AMR-WB+ core mode in mono usage, or a combination of a core mode and a stereo mode. The ~~operation codes~~frame types are presented in Table 25 below. The core mode and stereo mode index values are according to Table 22 and 23 respectively. The bit-rate value assumes an internal sampling frequency of 25600 Hz.

Table 25: Normative ~~operation code~~frame type table. Bit-rates assumes 25600 Hz internal sampling frequency.

~~Operation code~~Frame type	Core mode	Stereo mode	Bit rate	Octets per frame
0-15	As specified in Table 21			
16	0	None	10.4	26
17	1	None	12.0	30
18	2	None	13.6	34
19	3	None	15.2	38
20	4	None	16.8	42
21	5	None	19.2	48
22	6	None	20.8	52
23	7	None	24.0	60
24	0	0	12.4	31
25	0	1	12.8	32
26	0	4	14	35
27	1	1	14.4	36
28	1	3	15.2	38
29	1	5	16	40
30	2	2	16.4	41
31	2	4	17.2	43
32	2	6	18	45
33	3	3	18.4	46
34	3	5	19.2	48
35	3	7	20	50
36	4	4	20.4	51
37	4	6	21.2	53
38	4	9	22.4	56
39	5	5	23.2	58
40	5	7	24	60
41	5	11	25.6	64
42	6	8	26	65
43	6	10	26.8	67
44	6	15	28.8	72
45	7	9	29.6	74
46	7	10	30	75
47	7	15	32	80
48-127	Reserved			

8.2 AMR-WB+ ~~Transport~~ Transport Interface Format

The transport interface format serves as an intermediate interface to the transport format. The ~~transport~~ transport interface frame contains a two-octet header followed by data octets.

The header in each frame contains the following two octets.

	MSB							LSB
Octet	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1
1	0	~~Operation code~~Frame type (FT)						
2	TFI		0	ISF mode (5 bits)				

~~**Operation code**~~ **Frame type**(FT~~OC~~) (7 bits): Indicates the ~~operational~~ frame type setting of the codec used for the corresponding frame (the combination of AMR-WB+ core and stereo mode, the AMR-WB mode, or comfort noise, as specified by Table 25 above).

Transport Frame Index (TFI) (2 bits): An index from 0 (first) to 3 (last) indicating this transport frameís position in the superframe.

ISF index (5 bits): Indicates the internal sampling frequency employed for the corresponding frame. The index values correspond to internal sampling frequency as specified in Table 24 above. This field SHALL be set to 0 for operation according to the AMR-WB+ modes defined in table 21 (~~Operation codes~~Frame types 0-13).

~~OC~~FT=14 (AUDIO_LOST) is used to indicate frames that are lost. NO_DATA (~~OC~~FT=15) frame could mean either that there is no data produced by the audio encoder for that frame or that no data for that frame is transmitted in the current packet (i.e., valid data for that frame could be sent in either an earlier or later packet). The duration for these non-included frames is dependent on the internal sampling frequency indicated by the ISF mode field.

For operation according to FT~~OC~~ 0-13 the ISF field shall be set 0 and has no meaning. The frame length for that operation is fixed to 20 ms in time.

If receiving a frame with an ~~OC~~FT value not defined the whole frame SHOULD be discarded and assumed erased.

The AMR-WB+ SCR/DTX is identical with AMR-WB SCR/DTX described in [8] and SHALL only be used in combination with the AMR-WB modes (0-8).

The audio data follows the header octets. The number of data octets per frame corresponding to a certain ~~operation code~~ frame type is given in Table 25.

Example

The following diagram (Table 26) shows a ~~packet~~ frame of AMR-WB+ using 14 kbit/s ~~coding operation~~frame type (FT~~OC~~=26) with a frame length of 35 octets (280 bits). The internal sampling frequency in this example is 25.6 kHz (ISF mode = 8). ~~OC~~ FT 26 corresponds to mono mode 0 (208 bits/frame) and stereo mode 4 (72 bits/frame). The frame is the first frame in the superframe (TFI=0).

The data octets are ~~packetized~~ placed according to the detailed bit allocation given in tables 14 to 20. The first bit of the AMR-WB+ data b0 is placed in bit 8 of octet 3.

Table 26: AMR-WB+ ~~packet~~ transport interface format for 14 kbit/s operation with ISF mode 8 (bit rate factor=1).

	MSB							LSB
Octet	**bit 8**	**bit 7**	**bit 6**	**bit 5**	**Bit 4**	**bit 3**	**bit 2**	**bit 1**
1	~~OC~~ FT = 26							
	0	0	0	1	1	0	1	0
2	TFI=0			ISF = 8				
	0	0	0	0	1	0	0	0
3	AMR-WB+ data (octet 1)							
	b0	b1	b2	b3	b4	b5	b6	b7
4..27	AMR-WB+ data (octets 2 to 25)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
28	AMR-WB+ data (octet 26)							
	b200	b201	b202	b203	B204	b205	b206	b207
29	AMR-WB+ data (octet 27)							
	s0	s1	s2	s3	s4	s5	s6	s7
30..36	AMR-WB+ data (octet 28 to 34)							
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
37	AMR-WB+ data (octet 35)							
	S64	S65	S66	S67	S68	S69	S70	S71

8.3 AMR-WB+ File Storage Format

This format is relevant for file storage and defines the bitstream contained in an AMR-WB+ sample of a 3GP file [9]. It is similar to the transport interface format with the exception that the two-octet header is used once per superframe (a storage sample consist of 4 frames or one superframe). Note that in AMR-WB+, the operation code and internal sampling frequency can be switched only on superframe boundaries so the header octets are needed only once per superframe. Note that in a transport format the superframe is split in 4 transport frames for robustness against packet loss.
All media streams in a 3GP file are stored in timed units called samples. This format defines the syntax of a sample for AMR-WB+.

A storage sample consists of a two-octet header followed by data octets corresponding to the whole superframe (4 transport frames).

The two-octet header contains the ~~operation code~~ frame type (~~OC~~FT) (7 bits) and the ISF mode (5 bits). This is followed by the data octets corresponding to the whole superframe (4 transport frames). The number of data octets per superframe is given by 4 times the number of octets per frame (the right-most column in Table 25).

The length of an AMR-WB+ storage sample in ms (corresponding to one superframe) depends on the internal sample frequency and given by 80◊ISF/25600 where ISF is the internal sampling frequency in Hz (ISF modes are shown in Table 24).

The header in each superframe contains the following two octets.

	MSB							LSB
Octet	**bit 8**	**bit 7**	**bit 6**	**bit 5**	**bit 4**	**bit 3**	**bit 2**	**bit 1**
1	0	~~Operation Code~~ Frame type(7 bits)						
2	0	0	0	ISF mode (5 bits)				

~~Operation code~~**Frame type** (FT~~OC~~) (7 bits): Indicates the ~~operational~~ frame type setting of the codec used for the corresponding frame (the combination of AMR-WB+ core and stereo mode, the AMR-WB mode, or comfort noise, as specified by Table 25 above).

ISF index (5 bits): Indicates the internal sampling frequency employed for the corresponding frame. The index values correspond to internal sampling frequency as specified in Table 24 above. This field SHALL be set to 0 for operation according to the AMR-WB+ modes defined in table 21 (frame types ~~Operation codes~~ 0-13).

For ~~operation~~ frame types according to FT~~OC~~ 0-13 the ISF field shall be set 0 and has no meaning. The frame length for that operation is fixed to 20 ms in time.

The audio data follows the header octets. The number of data octets per superframe corresponding to a certain frame type ~~operation code~~ is given as 4 times the number of octets per frame (right-most column in Table 25).

Example

The following diagram (Table 27) shows a storage sample of AMR-WB+ using 14 kbit/s ~~coding operation~~frame type (FT~~OC~~=26) with a superframe length of 4◊35=140 octets. The internal sampling frequency in this example is 25.6 kHz (ISF mode = 8). FT~~OC~~ 26 corresponds to mono mode 0 (208 bits/frame) and stereo mode 4 (72 bits/frame).

The data octets are packetized according to the detailed bit allocation given in tables 14 to 20. The first bit of the AMR-WB+ data b0 is placed in bit 8 of octet 3.

It should be noticed that when FT~~OC~~ <10, i.e. AMR-WB frames, the original AMR-WB storage format should be preferred in order to ensure backward decoding compatibility.

Table 27: AMR-WB+ storage sample (superframe) for 14 kbit/s operation with ISF mode 8 (bit rate factor=1).

Octet	MSB bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	LSB bit 1
1	FT~~OC~~ = 26							
	0	0	0	1	1	0	1	0
2	ISF = 8							
	0	0	0	0	1	0	0	0
3	Frame 1 AMR-WB+ data (octet 1)							
	b0	b1	b2	b3	b4	b5	b6	b7
4..27	Frame 1 AMR-WB+ data (octets 2 to 25)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
28	Frame 1 AMR-WB+ data (octet 26)							
	b200	b201	b202	b203	B204	b205	b206	b207
29	Frame 1 AMR-WB+ data (octet 27)							
	s0	s1	s2	s3	s4	s5	s6	s7
30..36	Frame 1 AMR-WB+ data (octet 28 to 34)							
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
37	Frame 1 AMR-WB+ data (octet 35)							
	S64	S65	S66	S67	S68	S69	S70	S71
38	Frame 2 AMR-WB+ data (octet 1)							
	b0	b1	b2	b3	b4	b5	b6	b7
39..62	Frame 2 AMR-WB+ data (octets 2 to 25)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
63	Frame 2 AMR-WB+ data (octet 26)							
	b200	b201	b202	b203	B204	b205	b206	b207
64	Frame 2 AMR-WB+ data (octet 27)							
	s0	s1	s2	s3	s4	s5	s6	s7
65..71	Frame 2 AMR-WB+ data (octet 28 to 34)							
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
72	Frame 2 AMR-WB+ data (octet 35)							
	S64	S65	S66	S67	S68	S69	S70	S71
73	Frame 3 AMR-WB+ data (octet 1)							
	b0	b1	b2	b3	b4	b5	b6	b7
74..97	Frame 3 AMR-WB+ data (octets 2 to 25)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
98	Frame 3 AMR-WB+ data (octet 26)							
	b200	b201	b202	b203	B204	b205	b206	b207
99	Frame 3 AMR-WB+ data (octet 27)							
	s0	s1	s2	s3	s4	s5	s6	s7
100..106	Frame 3 AMR-WB+ data (octet 28 to 34)							
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
107	Frame 3 AMR-WB+ data (octet 35)							
	S64	S65	S66	S67	S68	S69	S70	S71
108	Frame 4 AMR-WB+ data (octet 1)							
	b0	b1	b2	b3	b4	b5	b6	b7
109..132	Frame 4 AMR-WB+ data (octets 2 to 25)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
133	Frame 4 AMR-WB+ data (octet 26)							
	b200	b201	b202	b203	B204	b205	b206	b207
134	Frame 4 AMR-WB+ data (octet 27)							
	s0	s1	s2	s3	s4	s5	s6	s7
135..141	Frame 4 AMR-WB+ data (octet 28 to 34)							
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
142	Frame 4 AMR-WB+ data (octet 35)							
	S64	S65	S66	S67	S68	S69	S70	S71

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040684

CR-Form-v7

CHANGE REQUEST

⌘ **TS 26.304 CR 001** ⌘ **rev -** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Incorrect definition of mode index for SID frames
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	2004-12-14
Category:	⌘	**F**
Release:	⌘	REL-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	In the AMR-WB+ floating point C-code (contained in TS 26.304), the definitions of MRDTX and NUM_OF_MODES used for AMR-WB coding modes are changed to be consistent with TS 26.201.
Summary of change:	⌘	The definitions of mode index and the number of modes in the floating point C-code (TS 26.304) are corrected.
Consequences if not approved:	⌘	The definitions for mode indice for the SID frame and the number of modes used for AMR-WB coding modes generated by the floating point C-code of TS 26.304 are not in line with the indices used in the generic AMR-WB interface formats explained in TS 26.201. (E.g., according to TS 26.201 the SID frame type index is 9, while the C-code of TS 26.304 produces index 10 for SID frames. Also the actual number of modes is 10, while the C-code of TS 26.304 produces 11 as the number of modes.) This may raise confusion and even interoperability problems when implementing the generic AMR-WB interface formats of TS 26.201.

Clauses affected:	⌘	dec_if.c, enc_dtx.c, enc_if.c, enc_main.c files

Other specs affected: ⌘

Y	N	
	X	Other core specifications
	X	Test specifications
	X	O&M Specifications

⌘

Other comments: ⌘

1) Change in dec_if.c, enc_dtx.c, enc_if.c, enc_main.c files

Before the change:

```
#define MRDTX        10
```

After the change:

```
#define MRDTX        9
```

2) Change in dec_if.c

Before the change:

```
#define NUM_OF_MODES 11
```

After the change:

```
#define NUM_OF_MODES 10
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040685

CR-Form-v7

CHANGE REQUEST

⌘ **TS 26.304 CR 002** ⌘ **rev -** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Correction of TCX coding selection for MMS encoder
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	2004-12-14
Category:	⌘	**F**
Release:	⌘	REL-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	TCX256 excitation encoding is not exploited, when TCX coding is selected for the superframe and the flag is set indicating TCX1024 coding option cannot be used.
Summary of change:	⌘	When TCX coding is selected for all frames in superframe, all TCX coding possibilities are not tried. If the flag is not set enabling the use of TCX1024, either TCX512 and TCX1024 is selected for the encoding. If the flag is set disabling the use of TCX1024, either TCX256 and TCX512 is selected for the encoding. The change request conserns the addition of TCX256 in the latter case.
Consequences if not approved:	⌘	TCX coding selection is not executed correctly in MMS encoder.

Clauses affected:	⌘	cod_lf_b.c

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

1) Change in cod_lf_b.c

Before the change (line 251):

```
if (coding_mod[k] == 0 ||
   (coding_mod[k] == 1 && st->stClass->NbOfAcelps != 0) ||
   (st->stClass->NoMtcx[i1] != 0 &&  st->stClass->NbOfAcelps == 0)) {
```

After the change (line 251):

```
if (coding_mod[k] == 0 ||
   (coding_mod[k] == 1 && st->stClass->NbOfAcelps != 0)) {
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040686

CR-Form-v7

CHANGE REQUEST

⌘ **TS 26.304 CR 003** ⌘ **rev -** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Misread of energy buffer in coding mode selection in MMS encoder. Correction of energy buffer initialisation
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	2004-12-14
Category:	⌘	**F**
Release:	⌘	REL-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	In the coding mode selection, the energy buffer is read at incorrect position. The initialisation of the energy buffer is also corrected
Summary of change:	⌘	In the coding mode selection, the maximum energy value of the last frames is searched from the energy buffer. The energy buffer is not read at the beginning of buffer, therefore searched maximum energy value is incorrect. The energy buffer initialisation and reading is corrected.
Consequences if not approved:	⌘	Maximum energy value is calculated incorrectly, because pointer is not pointing at the beginning of energy buffer. The energy buffer is not correctly initialised.

Clauses affected:	⌘	nclass.c

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

1) Change in nclass.c

Before the change:

- **The initialisation of the energy buffer (line 39)**

```
stClass->TotalEnergy[0]= 0;
stClass->TotalEnergy[1]= 0;
stClass->TotalEnergy[2]= 0;
stClass->TotalEnergy[3]= 0;
stClass->TotalEnergy[4]= 0;
stClass->TotalEnergy[5]= 0;
stClass->TotalEnergy[6]= 0;
```

- **The energy buffer reading in classification algorithm (line 486)**

```
for(i=5;i<7;i++) {
```

After the change:

- **The initialisation of the energy buffer (line 39)**

```
stClass->TotalEnergy[0]= 0;
stClass->TotalEnergy[1]= 0;
stClass->TotalEnergy[2]= 0;
stClass->TotalEnergy[3]= 0;
stClass->TotalEnergy[4]= 0;
```

- **The energy buffer reading in classification algorithm (line 486)**

```
for(i=0;i<4;i++) {
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040713

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **004** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Correction of stereo bit allocation tables
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	A reserved bit for the stereo is missing from the bit allocation tables
Summary of change:	⌘	The tables are updated with the correct location of the reserved bit.
Consequences if not approved:	⌘	A wrong interpretation of the bitstream and the location of the reserved bit may lead to the possibility wrong decoding in the case of stereo bitstreams

Clauses affected:	⌘	d_stereo_x.c tables_stereo.c cod_tcx_stereo.c c_stereo_x.c				
Other specs affected:	⌘	**Y**	**N**			
		X		Other core specifications	⌘	26.290
			X	Test specifications		
			X	O&M Specifications		
Other comments:	⌘					

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

1. How the code is changed in the file *d_stereo_x.c*

Lines 156-158 before the change

```
nbits = (StereoNbits[brMode] + (2*nbits_bwe))/4;
hiband_mode = 0;
if (StereoNbits[brMode] > 300)
{
   hiband_mode = 1;
}
```

Lines 156-xx after the change

```
nbits = (StereoNbits[brMode] + (2*nbits_bwe))/4;
hiband_mode = 0;
if (StereoNbits[brMode]-4 > 300)
{
   hiband_mode = 1;
}
```

Lines 156-158 before the change

```
/* decode the mode */
for(k=0;k<NB_DIV;k++) {
   ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
   if(bad_frame[k] == 0) {
      mod[k] = bin2int(2, ptr);
   }
   else {
      mod[k] = -1;
   }
}
```

Lines 156-xx after the change

```
/* decode the mode */
for(k=0;k<NB_DIV;k++) {
   ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +1; /* +1 reserved bit*/
   if(bad_frame[k] == 0) {
      mod[k] = bin2int(2, ptr);
   }
   else {
      mod[k] = -1;
   }
}
```

Lines xx-xx before the change

```
      k = 0;
      while (k < NB_DIV)
      {
         /* set pointer to parameters */
         prm = (param +4+NPRM_STEREO_HI_X*NB_DIV)+ (k*NPRM_DIV_TCX_STEREO);
         if ((mod[k] == 1) || (mod[k]==0)){
            n_pack =1;
            nbits_AVQ[0] = (StereoNbits[brMode]/4)-7-7-2-hf_bits;
            /* decode 20ms TCX */
```

```
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +2; /* +2 for the
mode*/
          prm[0] = bin2int(7, ptr); ptr += 7;
          prm[1] = bin2int(7, ptr); ptr += 7;
          pack4bits_d(nbits_AVQ[0], ptr, prm_AVQ);
          /* demultiplex and decode */
          AVQ_demuxdec(n_pack,prm_AVQ, nbits_AVQ, tmp_prm, TOT_PRM_20/8,
bad_frame+k);
          /* convert to integer */
          for(i=0;i< TOT_PRM_20;i++)
          {
            prm[i+2] = (int)tmp_prm[i];
          }
          k++;
        } /* end of mode 0/1 */
        else if (mod[k] == 2) {
          /* decode and demultiplex a 40 ms frame */
          n_pack = 2;
          nbits_AVQ[0] = (StereoNbits[brMode]/4)-2-7-hf_bits;
          nbits_AVQ[1] = (StereoNbits[brMode]/4)-2-7-hf_bits;
          /* decode first 20ms packet */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +2; /* +2 for the
mode*/
          prm[0] = bin2int(7, ptr);    ptr += 7;
          j = pack4bits_d(nbits_AVQ[0], ptr, prm_AVQ);
          /* decode second 20ms packet */
          ptr = serial + (k+2)*nbits_pack - nbits + hf_bits +2; /* +2 for the
mode*/
          prm[1] = bin2int(7, ptr);    ptr += 7;
          pack4bits_d(nbits_AVQ[1], ptr, prm_AVQ+j);
          /* demultiplex and decode tcx parameters */
          AVQ_demuxdec(n_pack,prm_AVQ, nbits_AVQ, tmp_prm, TOT_PRM_40/8,
bad_frame+k);
          /* convert to integer */
          for(i=0;i< TOT_PRM_40;i++)
          {
            prm[i+2] = (int)tmp_prm[i];
          }
          k+=2;
        } /* end of mode 2 */
        else if (mod[k] == 3) {
          /* encode and multiplex 80ms TCX */
          n_pack = 4;
          nbits_AVQ[0] = (StereoNbits[brMode]/4)-7-2-hf_bits;
          nbits_AVQ[1] = (StereoNbits[brMode]/4)-2-hf_bits;
          nbits_AVQ[2] = (StereoNbits[brMode]/4)-7-2-hf_bits;
          nbits_AVQ[3] = (StereoNbits[brMode]/4)-2-hf_bits;
          /* set pointer to bit stream */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +2 ;
          /* decode first 20 ms frame */
          prm[0] = bin2int(7, ptr);    ptr += 7;
          j = pack4bits_d(nbits_AVQ[0], ptr, prm_AVQ);
          /* set pointer to bit stream */
          ptr = serial + (k+2)*nbits_pack - nbits + hf_bits +2;
          /* decode second 20 ms frame */
          j += pack4bits_d(nbits_AVQ[1], ptr, prm_AVQ+j);
          /* set pointer to bit stream */
          ptr = serial + (k+3)*nbits_pack - nbits + hf_bits +2;
          /* decode third 20 ms frame */
          prm[1] = bin2int(7, ptr);    ptr += 7;
          j += pack4bits_d(nbits_AVQ[2], ptr, prm_AVQ+j);
          /* set pointer to bit stream */
          ptr = serial + (k+4)*nbits_pack - nbits + hf_bits +2;
          /* decode forth 20 ms frame */
          j += pack4bits_d(nbits_AVQ[3], ptr, prm_AVQ+j);
```

```
          /* demultiplex and decode tcx parameters */
          AVQ_demuxdec(n_pack,prm_AVQ, nbits_AVQ, tmp_prm, TOT_PRM_80/8,
bad_frame);
          /* convert to integer */
          for(i=0;i< TOT_PRM_80;i++)
          {
            prm[i+2] = (int)tmp_prm[i];
          }
          k+=4;
        }
      }
```

Lines xx-xx after the change

while (k < NB_DIV)

```
      {
        /* set pointer to parameters */
        prm = (param +4+NPRM_STEREO_HI_X*NB_DIV)+ (k*NPRM_DIV_TCX_STEREO);
        if ((mod[k] == 1) || (mod[k]==0)){
          n_pack =1;
          nbits_AVQ[0] = ((StereoNbits[brMode]-4)/4)-7-7-2-hf_bits;
          /* decode 20ms TCX */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +2 +1; /* +2 for the
mode +1 reserved bit*/
          prm[0] = bin2int(7, ptr); ptr += 7;
          prm[1] = bin2int(7, ptr); ptr += 7;
          pack4bits_d(nbits_AVQ[0], ptr, prm_AVQ);
          /* demultiplex and decode */
          AVQ_demuxdec(n_pack,prm_AVQ, nbits_AVQ, tmp_prm, TOT_PRM_20/8,
bad_frame+k);
          /* convert to integer */
          for(i=0;i< TOT_PRM_20;i++)
          {
            prm[i+2] = (int)tmp_prm[i];
          }
          k++;
        } /* end of mode 0/1 */
        else if (mod[k] == 2) {
          /* decode and demultiplex a 40 ms frame */
          n_pack = 2;
          nbits_AVQ[0] = ((StereoNbits[brMode]-4)/4)-2-7-hf_bits;
          nbits_AVQ[1] = ((StereoNbits[brMode]-4)/4)-2-7-hf_bits;
          /* decode first 20ms packet */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +2 +1; /* +2 for the
mode +1 reserved bit*/
          prm[0] = bin2int(7, ptr);    ptr += 7;
          j = pack4bits_d(nbits_AVQ[0], ptr, prm_AVQ);
          /* decode second 20ms packet */
          ptr = serial + (k+2)*nbits_pack - nbits + hf_bits +2 +1 ; /* +2 for
the mode +1 reserved bit*/
          prm[1] = bin2int(7, ptr);    ptr += 7;
          pack4bits_d(nbits_AVQ[1], ptr, prm_AVQ+j);
          /* demultiplex and decode tcx parameters */
          AVQ_demuxdec(n_pack,prm_AVQ, nbits_AVQ, tmp_prm, TOT_PRM_40/8,
bad_frame+k);
          /* convert to integer */
          for(i=0;i< TOT_PRM_40;i++)
          {
            prm[i+2] = (int)tmp_prm[i];
          }
          k+=2;
        } /* end of mode 2 */
        else if (mod[k] == 3) {
          /* encode and multiplex 80ms TCX */
```

```
          n_pack = 4;
          nbits_AVQ[0] = ((StereoNbits[brMode]-4)/4)-7-2-hf_bits;
          nbits_AVQ[1] = ((StereoNbits[brMode]-4)/4)-2-hf_bits;
          nbits_AVQ[2] = ((StereoNbits[brMode]-4)/4)-7-2-hf_bits;
          nbits_AVQ[3] = ((StereoNbits[brMode]-4)/4)-2-hf_bits;
          /* set pointer to bit stream */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits +2 +1;
          /* decode first 20 ms frame */
          prm[0] = bin2int(7, ptr);    ptr += 7;
          j = pack4bits_d(nbits_AVQ[0], ptr, prm_AVQ);
          /* set pointer to bit stream */
          ptr = serial + (k+2)*nbits_pack - nbits + hf_bits +2 +1;
          /* decode second 20 ms frame */
          j += pack4bits_d(nbits_AVQ[1], ptr, prm_AVQ+j);
          /* set pointer to bit stream */
          ptr = serial + (k+3)*nbits_pack - nbits + hf_bits +2 +1;
          /* decode third 20 ms frame */
          prm[1] = bin2int(7, ptr);    ptr += 7;
          j += pack4bits_d(nbits_AVQ[2], ptr, prm_AVQ+j);
          /* set pointer to bit stream */
          ptr = serial + (k+4)*nbits_pack - nbits + hf_bits +2 +1;
          /* decode forth 20 ms frame */
          j += pack4bits_d(nbits_AVQ[3], ptr, prm_AVQ+j);
          /* demultiplex and decode tcx parameters */
          AVQ_demuxdec(n_pack,prm_AVQ, nbits_AVQ, tmp_prm, TOT_PRM_80/8,
bad_frame);
          /* convert to integer */
          for(i=0;i< TOT_PRM_80;i++)
          {
            prm[i+2] = (int)tmp_prm[i];
          }
          k+=4;
        }
      }
```

2. How the code is changed in the file *c_stereo_x.c*

Lines xx-xx before the change

```
  if (StereoNbits[brMode] > 300)
  {
    st->filt_hi_pmsvq = &filt_hi_pmsvq7;
    st->gain_hi_pmsvq = &gain_hi_pmsvq5;
  }
```

Lines xx-xx after the change

```
  if (StereoNbits[brMode]-4 > 300)
  {
    st->filt_hi_pmsvq = &filt_hi_pmsvq7;
    st->gain_hi_pmsvq = &gain_hi_pmsvq5;
  }
```

Lines xx-xx before the change

```
    nbits = (StereoNbits[brMode] + (2*nbits_bwe))/4;
    hiband_mode = 0;
    if (StereoNbits[brMode] > 300) {
      hiband_mode = 1;
    }
```

Lines xx-xx after the change

```
     nbits = (StereoNbits[brMode] + (2*nbits_bwe))/4;
     hiband_mode = 0;
     if (StereoNbits[brMode]-4 > 300) {
       hiband_mode = 1;
     }
```

Lines xx-xx before the change

```
k = 0;
     while (k < NB_DIV)
     {
       mode = mod[k];
       /* set pointer to parameters */
       prm = (param +4+NPRM_STEREO_HI_X*NB_DIV)+ (k*NPRM_DIV_TCX_STEREO);
       if ((mode == 1) || (mode == 0)){
         /* encode 20ms TCX */
         n_pack = 1;
         nbits_AVQ[0] = (StereoNbits[brMode]/4)-7-2-7-hf_bits;
         AVQ_encmux(n_pack, prm+2, prm_AVQ, nbits_AVQ, TOT_PRM_20/8);
         /* set pointer to bit stream */
         ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
         /* encode the mode */
         int2bin(mode, 2, ptr);       ptr += 2;
         int2bin(prm[0], 7, ptr);     ptr += 7;
         int2bin(prm[1], 7, ptr);     ptr += 7;
         unpack4bits_d(nbits_AVQ[0], prm_AVQ, ptr);
         ptr += nbits_AVQ[0];
         k++;
       } /* end of mode 0/1 */
       else if (mode == 2) {
         /* encode and multiplex 40ms TCX */
         n_pack = 2;
         nbits_AVQ[0] = (StereoNbits[brMode]/4)-2-7-hf_bits;
         nbits_AVQ[1] = (StereoNbits[brMode]/4)-2-7-hf_bits;
         AVQ_encmux(n_pack, prm+2, prm_AVQ, nbits_AVQ, TOT_PRM_40/8);
         /* set pointer to bit stream */
         ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
         /* encode first 20 ms frame */
         int2bin(mode, 2, ptr);     ptr += 2;
         int2bin(prm[0], 7, ptr);     ptr += 7;
         j= unpack4bits_d(nbits_AVQ[0], prm_AVQ, ptr);
         /* set pointer to bit stream */
         ptr = serial + (k+2)*nbits_pack - nbits + hf_bits;
         /* encode second 20 ms frame */
         int2bin(mode, 2, ptr);     ptr += 2;
         int2bin(prm[1], 7, ptr);     ptr += 7;
         unpack4bits_d(nbits_AVQ[1], prm_AVQ+j, ptr);
         k+=2;
       } /* end of mode 2 */
       else if (mode == 3) {
         /* encode and multiplex 80ms TCX */
         n_pack = 4;
         nbits_AVQ[0] = (StereoNbits[brMode]/4)-7-2-hf_bits;
         nbits_AVQ[1] = (StereoNbits[brMode]/4)-2-hf_bits;
         nbits_AVQ[2] = (StereoNbits[brMode]/4)-7-2-hf_bits;
         nbits_AVQ[3] = (StereoNbits[brMode]/4)-2-hf_bits;
         AVQ_encmux(n_pack, prm+2, prm_AVQ, nbits_AVQ, TOT_PRM_80/8);
         /* set pointer to bit stream */
         ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
         /* encode first 20 ms frame */
         int2bin(mode, 2, ptr);     ptr += 2;
         int2bin(prm[0], 7, ptr);     ptr += 7;
         j= unpack4bits_d(nbits_AVQ[0], prm_AVQ, ptr);
```

```
          /* set pointer to bit stream */
          ptr = serial + (k+2)*nbits_pack - nbits + hf_bits;
          /* encode second 20 ms frame */
          int2bin(mode, 2, ptr);     ptr += 2;
          j += unpack4bits_d(nbits_AVQ[1], prm_AVQ+j, ptr);
          /* set pointer to bit stream */
          ptr = serial + (k+3)*nbits_pack - nbits + hf_bits;
          /* encode third 20 ms frame */
          int2bin(mode, 2, ptr);     ptr += 2;
          int2bin(prm[1], 7, ptr);     ptr += 7;
          j += unpack4bits_d(nbits_AVQ[2], prm_AVQ+j, ptr);
          /* set pointer to bit stream */
          ptr = serial + (k+4)*nbits_pack - nbits + hf_bits;
          /* encode forth 20 ms frame */
          int2bin(mode, 2, ptr);     ptr += 2;
          unpack4bits_d(nbits_AVQ[3], prm_AVQ+j, ptr);
          k+=4;
        }  /* end of mode 3 */
      } /* end of while k < NB_DIV */
```

Lines xx-xx after the change

```
k = 0;
     while (k < NB_DIV)
     {
       mode = mod[k];
       /* set pointer to parameters */
       prm = (param +4+NPRM_STEREO_HI_X*NB_DIV)+ (k*NPRM_DIV_TCX_STEREO);
       if ((mode == 1) || (mode == 0)){
          /* encode 20ms TCX */
          n_pack = 1;
          nbits_AVQ[0] = ((StereoNbits[brMode]-4)/4)-7-2-7-hf_bits;
          AVQ_encmux(n_pack, prm+2, prm_AVQ, nbits_AVQ, TOT_PRM_20/8);
          /* set pointer to bit stream */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
          /* encode the mode */
             *ptr = 0;                  ptr += 1;
          int2bin(mode, 2, ptr);       ptr += 2;
          int2bin(prm[0], 7, ptr);     ptr += 7;
          int2bin(prm[1], 7, ptr);     ptr += 7;
          unpack4bits_d(nbits_AVQ[0], prm_AVQ, ptr);
          ptr += nbits_AVQ[0];
          k++;
       } /* end of mode 0/1 */
       else if (mode == 2) {
          /* encode and multiplex 40ms TCX */
          n_pack = 2;
          nbits_AVQ[0] = ((StereoNbits[brMode]-4)/4)-2-7-hf_bits;
          nbits_AVQ[1] = ((StereoNbits[brMode]-4)/4)-2-7-hf_bits;
          AVQ_encmux(n_pack, prm+2, prm_AVQ, nbits_AVQ, TOT_PRM_40/8);
          /* set pointer to bit stream */
          ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
          /* encode first 20 ms frame */
             *ptr = 0;                  ptr += 1;
          int2bin(mode, 2, ptr);     ptr += 2;
          int2bin(prm[0], 7, ptr);     ptr += 7;
          j= unpack4bits_d(nbits_AVQ[0], prm_AVQ, ptr);
          /* set pointer to bit stream */
          ptr = serial + (k+2)*nbits_pack - nbits + hf_bits;
          /* encode second 20 ms frame */
             *ptr = 0;                  ptr += 1;
          int2bin(mode, 2, ptr);     ptr += 2;
          int2bin(prm[1], 7, ptr);     ptr += 7;
```

```
        unpack4bits_d(nbits_AVQ[1], prm_AVQ+j, ptr);
        k+=2;
      } /* end of mode 2 */
      else if (mode == 3) {
        /* encode and multiplex 80ms TCX */
        n_pack = 4;
        nbits_AVQ[0] = ((StereoNbits[brMode]-4)/4)-7-2-hf_bits;
        nbits_AVQ[1] = ((StereoNbits[brMode]-4)/4)-2-hf_bits;
        nbits_AVQ[2] = ((StereoNbits[brMode]-4)/4)-7-2-hf_bits;
        nbits_AVQ[3] = ((StereoNbits[brMode]-4)/4)-2-hf_bits;
        AVQ_encmux(n_pack, prm+2, prm_AVQ, nbits_AVQ, TOT_PRM_80/8);
        /* set pointer to bit stream */
        ptr = serial + (k+1)*nbits_pack - nbits + hf_bits;
        /* encode first 20 ms frame */
          *ptr = 0;                  ptr += 1;
        int2bin(mode, 2, ptr);     ptr += 2;
        int2bin(prm[0], 7, ptr);     ptr += 7;
        j= unpack4bits_d(nbits_AVQ[0], prm_AVQ, ptr);
        /* set pointer to bit stream */
        ptr = serial + (k+2)*nbits_pack - nbits + hf_bits;
        /* encode second 20 ms frame */
          *ptr = 0;                  ptr += 1;
        int2bin(mode, 2, ptr);     ptr += 2;
        j += unpack4bits_d(nbits_AVQ[1], prm_AVQ+j, ptr);
        /* set pointer to bit stream */
        ptr = serial + (k+3)*nbits_pack - nbits + hf_bits;
        /* encode third 20 ms frame */
          *ptr = 0;                  ptr += 1;
        int2bin(mode, 2, ptr);     ptr += 2;
        int2bin(prm[1], 7, ptr);     ptr += 7;
        j += unpack4bits_d(nbits_AVQ[2], prm_AVQ+j, ptr);
        /* set pointer to bit stream */
        ptr = serial + (k+4)*nbits_pack - nbits + hf_bits;
        /* encode forth 20 ms frame */
          *ptr = 0;                  ptr += 1;
        int2bin(mode, 2, ptr);     ptr += 2;
        unpack4bits_d(nbits_AVQ[3], prm_AVQ+j, ptr);
        k+=4;
      }  /* end of mode 3 */
    } /* end of while k < NB_DIV */
```

3. How the code is changed in the file *cod_tcx_stereo.c*

Lines xx-xx before the change

```
nbits = StereoNbits[brMode]-24;
if (StereoNbits[brMode] > 300) nbits -= 24;
```

Lines xx-xx after the change

```
nbits = StereoNbits[brMode]-24-4;
if (StereoNbits[brMode]-4 > 300) nbits -= 24;
```

4. How the code is changed in the file *cod_tcx_stereo.c*

Lines xx-xx before the change

```
const int StereoNbits[18] = {
 (int)(1.2*80)-4,  /* 1.2 + 0.8 = 2.0 kbps */
 (int)(1.4*80)-4,  /* 1.4 + 0.8 = 2.2 kbps */
 (int)(1.6*80)-4,  /* 1.6 + 0.8 = 2.4 kbps */
 (int)(1.8*80)-4,  /* 1.8 + 0.8 = 2.6 kbps */
 (int)(2.0*80)-4,  /* 2.0 + 0.8 = 2.8 kbps */
 (int)(2.4*80)-4,  /* 2.4 + 0.8 = 3.2 kbps */
 (int)(2.8*80)-4,  /* 2.8 + 0.8 = 3.6 kbps */
 (int)(3.2*80)-4,  /* 3.2 + 0.8 = 4.0 kbps */
 (int)(3.6*80)-4,  /* 3.6 + 0.8 = 4.4 kbps */
 (int)(4.0*80)-4,  /* 4.0 + 0.8 = 4.8 kbps */
 (int)(4.4*80)-4,  /* 4.4 + 0.8 = 5.2 kbps */
 (int)(4.8*80)-4,  /* 4.8 + 0.8 = 5.6 kbps */
 (int)(5.2*80)-4,  /* 5.2 + 0.8 = 6.0 kbps */
 (int)(5.6*80)-4,  /* 5.6 + 0.8 = 6.4 kbps */
 (int)(6.0*80)-4,  /* 6.0 + 0.8 = 6.8 kbps */
 (int)(6.4*80)-4,  /* 6.4 + 0.8 = 7.2 kbps */
 (int)(6.8*80)-4,  /* 6.8 + 0.8 = 7.6 kbps */
 (int)(7.2*80)-4}; /* 7.2 + 0.8 = 8.0 kbps */
```

Lines xx-xx after the change

```
const int StereoNbits[18] = {
  (int)(1.2*80),  /* 1.2 + 0.8 = 2.0 kbps */
  (int)(1.4*80),  /* 1.4 + 0.8 = 2.2 kbps */
  (int)(1.6*80),  /* 1.6 + 0.8 = 2.4 kbps */
  (int)(1.8*80),  /* 1.8 + 0.8 = 2.6 kbps */
  (int)(2.0*80),  /* 2.0 + 0.8 = 2.8 kbps */
  (int)(2.4*80),  /* 2.4 + 0.8 = 3.2 kbps */
  (int)(2.8*80),  /* 2.8 + 0.8 = 3.6 kbps */
  (int)(3.2*80),  /* 3.2 + 0.8 = 4.0 kbps */
  (int)(3.6*80),  /* 3.6 + 0.8 = 4.4 kbps */
  (int)(4.0*80),  /* 4.0 + 0.8 = 4.8 kbps */
  (int)(4.4*80),  /* 4.4 + 0.8 = 5.2 kbps */
  (int)(4.8*80),  /* 4.8 + 0.8 = 5.6 kbps */
  (int)(5.2*80),  /* 5.2 + 0.8 = 6.0 kbps */
  (int)(5.6*80),  /* 5.6 + 0.8 = 6.4 kbps */
  (int)(6.0*80),  /* 6.0 + 0.8 = 6.8 kbps */
  (int)(6.4*80),  /* 6.4 + 0.8 = 7.2 kbps */
  (int)(6.8*80),  /* 6.8 + 0.8 = 7.6 kbps */
  (int)(7.2*80)}; /* 7.2 + 0.8 = 8.0 kbps */
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040764

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **005** ⌘ **rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Optimization of error concealment operation
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**F**
Release:	⌘	Release 6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Parametric stereo error concealment for low-band stereo part uses data from a frame history array to reduce stereo image when frequent frame losses occur. The implementation contains a bug, which makes it almost never used. There is no quality gain or loss if the bug is corrected.
Summary of change:	⌘	Removal of code used to reduce stereo image and removal of frame history array.
Consequences if not approved:	⌘	Unneccessary large static memory allocation (0.5 kW) and Uncessary additional complexity.

Clauses affected:	⌘	dec_tcx_stereo.c mem.h

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

1. How the code is changed in the file *dec_tcx_stereo.c*

Lines before the change

```
        st->mem_balance = 0;
        for (i = 0; i < FER_STAT_LENGTH; i++) {
           st->fer_hist[i] = 0;
        }
        st->fer_mean = 0;
        st->fer_hist_ptr = 0;
```

Lines after the change

```
        st->mem_balance = 0;
```

Lines before the change

```
        int i,k, mod[4];
        int bfi;
        int *prm;
        float alpha,s;
         int ptr;
         float tmp0,tmp1,tmp2;
         int j=4;
```

Lines after the change

```
        int i,k, mod[4];
        int bfi;
        int *prm;
```

Lines before the change

```
while(k<NB_DIV)
   {
    ptr = st->fer_hist_ptr;

     if(mod[k]==0 || mod[k] ==1)
      j = 1;
     if(mod[k]==2)
      j = 2;
     if(mod[k]==3)
      j = 4;
   for (i=0; i < j; i++) {
      st->fer_hist[st->fer_hist_ptr] = bad_frame[k+i];    MOVE(1);INDIRECT(1);
      st->fer_hist_ptr++;                                 BRANCH(1);ADD(1);
      if (st->fer_hist_ptr == FER_STAT_LENGTH)
     {
        st->fer_hist_ptr = 0;                             MOVE(1);
     }
    }
    tmp0 = 0;                                             MOVE(2);
    tmp2 = 0;                                             LOOP(1);
    for (j=0; j < FER_STAT_LENGTH; j++) {
      tmp1 = my_min(1.0f,(FER_STAT_LENGTH-j+FER_STAT_LENGTH/2.0f)/FER_STAT_LENGTH);
                MOVE(1);FUNC(2);ADD(1);DIV(1);MULT(1);
      tmp0 += st->fer_hist[ptr]*tmp1;                     MAC(1);INDIRECT(1);
      tmp2 += tmp1;                                       ADD(1);
      ptr++;                                              BRANCH(1); ADD(1);
      if (ptr == FER_STAT_LENGTH)
     {
        ptr = 0;                                          MOVE(1);
     }
   }
```

```
    st->fer_mean = 0.9f*st->fer_mean+0.1f*tmp0/tmp2; pessimize();
                                                    DIV(1);MAC(1);MULT(1);
    alpha = my_min(1.0,my_max(0.0,1.0-my_min(0.7,0.7*(st->fer_mean*100.0-1.0)/4.0)));
             MOVE(1);FUNC(2);FUNC(2);FUNC(2);SHIFT(1);MULT(1);ADD(2);

                                                    BRANCH(2);ADD(2);
     if(mod[k]==0 || mod[k] ==1)
     {                                              LOOP(1);PTR_INIT(4);
        for(i=k*L_FRAME_2k/4;i<k*L_FRAME_2k/4+L_FRAME_2k/4;i++)
        {
           s = synth_side[i]*alpha;                 MULT(1);
           left_2k[i] = synth_2k[i] + s;            ADD(2);STORE(2);
           right_2k[i] = synth_2k[i] - s;
        }
     k++;
     }
     else if(mod[k] ==2)
     {                                              LOOP(1);PTR_INIT(4);
        for(i=k*L_FRAME_2k/4;i<k*L_FRAME_2k/4+L_FRAME_2k/2;i++)
        {
           s = synth_side[i]*alpha;                 MULT(1);
           left_2k[i] = synth_2k[i] + s;            ADD(2);STORE(2);
           right_2k[i] = synth_2k[i] - s;
        }
     k+=2;
     }
     else
     {                                              LOOP(1);PTR_INIT(4);

        for(i=0;i<L_FRAME_2k;i++)
        {
           s = synth_side[i]*alpha;                 MULT(1);
           left_2k[i] = synth_2k[i] + s;            ADD(2);STORE(2);
           right_2k[i] = synth_2k[i] - s;
        }
     k+=4;
     }

   }
                                                    FLC_sub_end();

   Dyn_Mem_Out();
   return;
}
```

Lines after the change

```
                                                    LOOP(1);PTR_INIT(4);
        for(i=0;i<L_FRAME_2k;i++)
        {
           left_2k[i] = synth_2k[i] + synth_side[i];    ADD(2);STORE(2);
           right_2k[i] = synth_2k[i] - synth_side[i];
        }
                                                         FLC_sub_end();


   Dyn_Mem_Out();
   return;
}
```

2. How the code is changed in the file *mem.h*

Line before the change

```
       float mem_balance;

       int fer_hist[FER_STAT_LENGTH];
```

```
int fer_hist_ptr;
float fer_mean;
//E_TCX_FILL
float old_xri[L_TCX];
```

Line after the change

```
float mem_balance;
//E_TCX_FILL
float old_xri[L_TCX];
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040765

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **006** ⌘ **rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Stereo operation of pre-echo mode, saturation of gain_shape
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Due to fixed point implementation issues it is beneficial to limit the gain_shape values to the range [0.5..2.0]. In order to avoid potential floating point to fixed point interoperability problems it is recommended to include this modification to the floating point code aswell.
Summary of change:	⌘	A limitation to the range [0.5..2.0] is added to the gain_shape computation
Consequences if not approved:	⌘	Potential interoperability problems between if signal encoded with floating point code and decoded with fixed point code, might lead to artefacts in the sound.

Clauses affected:	⌘	cod_tcx_stereo.c, dec_tcx_stereo.c

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

1. How the code is changed in the file *dec_tcx_stereo.c*

Lines before the change

```
for(i=0;i<lg/16;i++){
  gain_shap[i] = (float)sqrt(gain_shap[i]*pow(10.0f,-tmp));
}
```

Lines after the change

```
for(i=0;i<lg/16;i++){
  gain_shap[i] = (float)sqrt(gain_shap[i]*pow(10.0f,-tmp));
  gain_shap[i] = my_min(2.0,my_max(0.5,gain_shap[i]));
}
```

2. How the code is changed in the file *cod_tcx_stereo.c*

Lines before the change

```
for(i=0;i<lg;i+=16){
  gain_shap[i/16] = (float) sqrt(gain_shap[i/16]*pow(10.0f,-tmp));
  for(k=0;k<16;k++)
```

Lines after the change

```
for(i=0;i<lg;i+=16){
  gain_shap[i/16] = (float) sqrt(gain_shap[i/16]*pow(10.0f,-tmp));
  gain_shap[i/16] = my_min(2.0,my_max(0.5,gain_shap[i/16]));
  for(k=0;k<16;k++)
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, xx-yy November 2004

Tdoc ⌘S4-040716

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **007** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Stereo operation of pre-echo mode, alignement of encoder and decoder
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	Pre-echo stereo mode operation is selected in closed loop. The operation performed in the encoder should be the same as that one performed in the decoder.
Summary of change:	⌘	gain_shape computation is done after signal windowing in both the encoder and the decoder. Mono signal is compensated by gain_shape in the encoder
Consequences if not approved:	⌘	Potential problem may arise when pre-echo mode is selected since there is a mismatch between the encoder and the decoder

Clauses affected:	⌘	cod_tcx_stereo.c, dec_tcx_stereo.c

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

1. How the code is changed in the file *dec_tcx_stereo.c*

Lines xx-xx before the change

```
/* windowed mono  */
for (i=0; i<lg; i++){
  wm[i] = mono[i];
}
if(pre_echo){
  /* compensate for the gain */
  tmp = 0.0f;
  for(i=0;i<lg;i+=16){
    gain_shap[i/16] = 0.001f;
    for(k=0;k<16;k++)
    {
      gain_shap[i/16] += wm[i+k]*wm[i+k];
    }
    /* average log gain in frame */
    tmp += (float)log10(gain_shap[i/16]);
  }
  tmp /= (float)(lg/16);
  for(i=0;i<lg/16;i++){
    gain_shap[i] = (float)sqrt(gain_shap[i]*pow(10.0f,-tmp));
  }
}
/* windowing for TCX overlap and correlation */
for (i=0; i<ovlp_size; i++)
{
  wm[i] *= window[i];
}
for (i=0; i<lext; i++) {
  wm[L_frame+i] *= window[ovlp_size+i];
}
```

Lines xx-xx after the change

```
/* windowed mono  */
for (i=0; i<lg; i++){
  wm[i] = mono[i];
}
 /* windowing for TCX overlap and correlation */
for (i=0; i<ovlp_size; i++)
{
  wm[i] *= window[i];
}
for (i=0; i<lext; i++) {
  wm[L_frame+i] *= window[ovlp_size+i];
}
if(pre_echo){
  /* compensate for the gain */
  tmp = 0.0f;
  for(i=0;i<lg;i+=16){
    gain_shap[i/16] = 0.001f;
    for(k=0;k<16;k++)
    {
      gain_shap[i/16] += wm[i+k]*wm[i+k];
    }
```

```
          /* average log gain in frame */
          tmp += (float)log10(gain_shap[i/16]);
        }
        tmp /= (float)(lg/16);
        for(i=0;i<lg/16;i++){
          gain_shap[i] = (float)sqrt(gain_shap[i]*pow(10.0f,-tmp));
        }
      }
```

2. How the code is changed in the file *cod_tcx_stereo.c*

Lines xx-xx before the change

```
  if(pre_echo)
  {
    /* compensate for the gain */
    tmp = 0.0f;
    for(i=0;i<lg;i+=16){
      gain_shap[i/16] = 0.001f;
      for(k=0;k<16;k++)
      {
        gain_shap[i/16] += wm[i+k]*wm[i+k];
      }
      /* average log gain in frame */
      tmp += (float)log10(gain_shap[i/16]);
    }
    tmp /= (float)(lg/16);
    for(i=0;i<lg;i+=16){
      gain_shap[i/16] = (float) sqrt(gain_shap[i/16]*pow(10.0f,-tmp));
      for(k=0;k<16;k++)
      {
        xn[i+k] /= gain_shap[i/16];
      }
    }
  }
   for (i=0; i<lext; i++) {
#ifndef COS_FAC
     xn[L_frame+i] *= window[ovlp_size+i];
     wm[L_frame+i] *= window[ovlp_size+i];
#else
     tmpfloat=cos_fac(ovlp_size+i,ovlp_size,lext);
     xn[L_frame+i] *= tmpfloat;
     wm[L_frame+i] *= tmpfloat;
#endif
  }
```

Lines xx-xx after the change

```
   for (i=0; i<lext; i++) {
#ifndef COS_FAC
     xn[L_frame+i] *= window[ovlp_size+i];
     wm[L_frame+i] *= window[ovlp_size+i];
#else
     tmpfloat=cos_fac(ovlp_size+i,ovlp_size,lext);
     xn[L_frame+i] *= tmpfloat;
     wm[L_frame+i] *= tmpfloat;
#endif
  }
  if(pre_echo)
  {
    /* compensate for the gain */
```

```
    tmp = 0.0f;
    for(i=0;i<lg;i+=16){
      gain_shap[i/16] = 0.001f;
      for(k=0;k<16;k++)
      {
        gain_shap[i/16] += wm[i+k]*wm[i+k];
      }
      /* average log gain in frame */
      tmp += (float)log10(gain_shap[i/16]);
    }
    tmp /= (float)(lg/16);
    for(i=0;i<lg;i+=16){
      gain_shap[i/16] = (float) sqrt(gain_shap[i/16]*pow(10.0f,-tmp));
      for(k=0;k<16;k++)
      {
        xn[i+k] /= gain_shap[i/16];
         wm[i+k] /= gain_shap[i/16];
      }
    }
  }
```

Lines xx-xx before the change

```
 /*-------------------------------------------------------------*
  * find and quantize gain, multiply xnq[] by gain.           *
  * windowing of xnq[] for TCX overlap.                       *
  *-------------------------------------------------------------*/
  if (pre_echo) {
    for(i=0;i<lg;i+=16){
      for(k=0;k<16;k++) {
        xnq[i+k] *= gain_shap[i/16];
        xn[i+k] *= gain_shap[i/16];
      }
    }
  }
```

Lines xx-xx after the change

```
 /*-------------------------------------------------------------*
  * find and quantize gain, multiply xnq[] by gain.           *
  * windowing of xnq[] for TCX overlap.                       *
  *-------------------------------------------------------------*/
  if (pre_echo) {
    for(i=0;i<lg;i+=16){
      for(k=0;k<16;k++) {
        xnq[i+k] *= gain_shap[i/16];
        xn[i+k] *= gain_shap[i/16];
       wm[i+k] *= gain_shap[i/16];
      }
    }
  }
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040768

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **008** ⌘ **rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Addition of support for file formats and improved command line
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The AMR-WB+ 3GPP reference code should support the mandatory file format described in 26.244 To improve the user interface by simplifying the command line of the AMR-WB+ 3GPP floating point reference code, and by the addition of the support of bitstream transport interface file and switching of frame types by means of an auxiliary file with frame type and ISF information.
Summary of change:	⌘	Added two libraries and 1 header file to the source code distribution allowing to read and write 3gp files containing AMR-WB/AMR-WB+ tracks Modifying the command line such that the user can now specify the traget bit rate and mono/stereo operation. Replacing the unpacked bitstream file with transport interface file format. Add the functionality of frame type and ISF switching my means of a switching file.
Consequences if not approved:	⌘	The AMR-WB+ 3GPP reference code will not support the mandatory 3gp file format. May be confusing to the user how to select the best configuration for a certain bit rate target. May become not straightforward to convert the bitstream file into a transport file format. May become difficult to test the switching capability of the codec.

Clauses affected:	⌘	enc_wbplus.c dec_wbplus.c cnst.h, prot_func.h, table_plus.c			
		Y	**N**		
Other specs affected:	⌘		**X**	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

There are too many editorial changes for them all to be listed. The changes are bitexact. Additionally, the source code distribution contains additional libraries for supporting writing to and reading from a 3gp file.

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040767

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **009** ⌘ **rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Source code editorial changes
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The AMR-WB+ 3GPP floating point reference code contains some comments and typos that are unnecessary. The AMR-WB+ 3GPP floating point reference code contains copyright statements that are unnecessary.
Summary of change:	⌘	Removal of comments and copyright statements
Consequences if not approved:	⌘	Unecessary comments in the code may degrade the readability of the code. Unecessary copyright statements may contradict with 3GPP policies.

Clauses affected:	⌘	qpisf_2s.c util_stereo_x.c d_stereo_x.c cod_ace.c cod_hi_stereo.c code_tcx_stereo.c enc_wb_plus.c cod_main.c, enc_main.h, prot_func.h, c_steteo_x.c re8_cod.c, get_gain.c, wb_vad.c, dec_prm.c, alf_emph.c, bits.c, deci12k8.c, fft9.c, isp_isf.c, read_data.c, wave_file_tools.c, windows.c, write_data.c, wb_vad.h, wb_vad_c.h

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

There are too many editorial changes for them all to be listed. The changes do not add any functionality and are bitexact.

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040719

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **010** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Removal of complexity counters
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change:	⌘	The AMR-WB+ 3GPP floating point reference code contains complexity and memory counters that were used for verification purposes. The output of these counters has been documented and there is no need to keep the code heavily instrumented.
Summary of change:	⌘	The complexity and memory counters are removed. The counting tools are also removed
Consequences if not approved:	⌘	The AMR-WB+ 3GPP floating point reference code with the complexity counters is hard to read. When the counting is enabled the execution time is very slow. In addition, the code can be very cumbersome to include as part of an application.

Clauses affected:	⌘	All files (*.c and *.h)

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

1. How the code is changed in the files *.c*.h

Complexity counters are removed. Compelxity counting tools are also removed:
flc.c flc.h dynmem.c dynmem.h.

#include statements relative to the files flc.h and dynmem.h are also removed from the source code.

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040780

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **011** ⌘ **rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Editorial changes
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The description of the bitstream file format is not aligned with the floating point C-code.
Summary of change:	⌘	Editorial changes in Section 5.2 to include the correct bitstream file format.
Consequences if not approved:	⌘	May be confusing if the bitstream file format in 26.304 is not aligned with the C-code.

Clauses affected:	⌘	Section 2. Table 3. Section 5.2.

		Y	N			
Other specs affected:	⌘	X		Other core specifications	⌘	26.290
			X	Test specifications		
			X	O&M Specifications		

Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document.*

[1] 3GPP TS 26.273: "ANSI-C code for the Fixed-point Extended AMR Wideband codec".

[2] 3GPP TS 26.290: " Audio codec processing functions; Extended AMR Wideband codec; Transcoding functions ".

[3] 3GPP TS 26.xxx: "3GPP audio codecs, Conformance".

[4] 3GPP TS 26.201: " AMR Wideband speech codec; frame structure".

[5] IETF Internet Draft: "Real-Time Transport Protocol (RTP) Payload Format for Extended AMR Wideband (AMR-WB+) Audio Codec", Sjoberg J., Westerlund M. and Lakaniemi A., http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-amrwbplus-01.txt, July 2004.

[6] 3GPP TS 26193: " AMR Wideband speech codec; Source controlled rate operation".

[7] 3GPP TS 26.244: "Transparent end-to-end packet switched streaming service (PSS); 3GPP file format (3GP)"

Ö <cut text>

4.4 Variables, constants and tables

4.4.1 Description of fixed tables used in the C-code

This clause contains a listing of all fixed tables declared in tables_plus.c and tables_stereo.c files.

Table 3: Encoder fixed tables

Format	Table name	Size	Description
Float32	NBITS_CORE	8	Core bit-rates
Float32	T_sin	1152	FFT Sine table
Float32	T_cos	1152	FFT Cosine table
Float32	filter_32k	61	FIR table for decimation/oversampling
Float32	filter_32k_hf	61	FIR table for decimation/oversampling
Float32	filter_32k_7k	61	FIR table for decimation/oversampling
Float32	filter_48k	185	FIR table for decimation/oversampling
Float32	Filter_48k_hf	185	FIR table for decimation/oversampling
Float32	filter_8k	61	FIR table for decimation/oversampling
Float32	isf_init	16	Initial ISF memory
Float32	Mean_isf	16	Means of ISFs
Float32	Dico1_isf	2304	1st stage codebook, isf0 to isf8
Float32	Dico2_isf	1792	1st stage codebook, isf9 to isf15
Float32	Dico21_isf	192	2nd stage codebook, isf2_0 to isf 2_2
Float32	Dico22_isf	384	2nd stage codebook, isf2_3 to isf 2_5
Float32	Dico23_isf	384	2nd stage codebook, isf2_6 to isf 2_8
Float32	Dico24_isf	96	2nd stage codebook, isf2_9 to isf 2_11
Float32	Dico25_isf	128	2nd stage codebook, isf2_12 to isf 2_15
Float32	Dico21_isf_36b	640	1st stage codebook, (36b) split 1
Float32	Dico22_isf_36b	512	1st stage codebook, (36b) split 2
Float32	Dico23_isf_36b	448	1st stage codebook, (36b) split 3
Float32	Dico_gain_hf	512	Quantization table for one-stage HF gain
Float32	Mean_isf_hf_12k8	8	Means of ISFs (full band)
Float32	dico1_isf_hf_12k8	32	1nd stage isf codebook (full band)
Float32	mean_isf_hf_low_rate	8	Means of isfs
Float32	Dico1_isf_hf_low_rate	32	1st stage isf codebook
Float32	dico2_isf_hf	1024	2nd stage isf codebook
Float32	Lag_window	17	Lag window
Float32	Filt_lp	13	Low-pass fir filter for bass post filter
Float32	Sin20	20	Random phase
Float32	Inter4_2	65	º resolution interpolation filter
Float32	VadFiltBandFreqs	12	Open-loop classifier
Float32	Bw	12	Open-loop classifier
Float32	Lwg	8	Open-loop claissifier
Float32	~~Gain_if_ramp~~Gain_hf_ramp	64	HF gain ramp for wb->wb+ switiching
Float32	Inter2_coef	12	Filter coefficients for band join/split
Float32	Filter_LP180	2341	Filter for 48 kHz interpolation
Float32	StereoNbits	18	Stereo bit-rates
Float32	Filter_2k	321	2k decimation filter
Float32	Cb_filt_hi_mean	9	Average filter
Float32	Filt_hi_mscb4a	16*9	
Float32	Filt_hi_mscb_7a	16*9	
Float32	Filt_hi_mscb_7b	8*9	
Float32	Cb_gain_hi_mean	2	Average gain vector
Float32 Float32 TBC	Gain_hi_mscb_2a Gain_hi_mscb_5a	4*2 32*2	

Ö <cut text>

5 File formats

This clause describes the file formats used by the encoder and decoder programs.

5.1 Audio file (encoder input/decoder output)

Audio files read by the encoder must be formatted as 16 bits PCM wave (*.wav) files. The decoder output is written as a 16 bit PCM wave file (*.wav).

Note that the decoder, with proper command line switch, can produce a mono file from a stereo bit-stream.

5.2 Parameter bitstream file (encoder output/decoder input)

For AMR-WB+ operation, the files produced by the audio encoder/expected by the audio decoder are either according to the transport interface format defined in Reference [2], Section 8.2, or according to the 3GP file format [7], whereby the storage sample definition is found in Reference [2], Section 8.3.

~~contain an arbitrary number of frames containing a header and data octets in the following format.~~

~~MONO_RATE~~	~~STEREO EXTENSION RATE~~	~~FREQUENCY SCALE~~	~~B1~~	~~B2~~	~~…~~	~~Bmn~~	~~S1~~	~~S2~~	~~…~~	~~Ssn~~

~~Each box corresponds to one octet (Word8) value in the bitstream file, for a total of 3+mn+sn octets per frame, where mn is the number of encoded octets in the frame for the mono rate and sn is the number of encoded octets in the frame for the stereo extension rate. For mono encoding the value of sn is equal to zero.~~

~~For AMR-WB modes, the file has the following format:~~

~~MONO_RATE~~	~~AMR-WB bitstream in IF2 format [4]~~

~~The header fields have the following meaning:~~

~~**MONO_RATE:**~~

~~The rate of AMR-WB or mono rate of Extended AMR-WB. The values of MONO_RATE are given in Table 7 below.~~

~~Table 7: Description of MONO_RATE header field.~~

~~MONO RATE MODE~~	~~*Mono rate(incl. BWE)*~~ ~~(bits/frame)~~	~~*Number of data bytes*~~
~~0x00~~	~~AMR-WB 6.60 kbit/s mode~~	~~18~~
~~0x01~~	~~AMR-WB 8.85 kbit/s mode~~	~~23~~
~~0x02~~	~~AMR-WB 12.65 kbit/s mode~~	~~33~~
~~0x03~~	~~AMR-WB 14.25 kbit/s mode~~	~~37~~
~~0x04~~	~~AMR-WB 15.85 kbit/s mode~~	~~41~~
~~0x05~~	~~AMR-WB 18.25 kbit/s mode~~	~~47~~

0x06	AMR-WB 19.85 kbit/s mode	51
0x07	AMR-WB 23.05 kbit/s mode	59
0x08	AMR-WB 23.85 kbit/s mode	61
0x09	AMR-WB SID	6
0x0A-0x0D	RESERVED	
0x0E	AMR-WB FRAME_ERASURE	0
0x0F	AMR-WB NO_DATA	0
0x10	AMR-WB+ 208 bit/frame	26
0x11	AMR-WB+ 240 bit/frame	30
0x12	AMR-WB+ 272 bit/frame	34
0x13	AMR-WB+ 304 bit/frame	38
0x14	AMR-WB+ 336 bit/frame	42
0x15	AMR-WB+ 384 bit/frame	48
0x16	AMR-WB+ 416 bit/frame	52
0x17	AMR-WB+ 480 bit/frame	60
0x18-0x1D	RESERVED	
0x1E	FRAME_ERASURE	0
0x1F	NO_DATA	0

STEREO_EXTENSION_RATE:

The mode of the stereo extension bit rate. The values of STEREO_EXTENSION_RATE are given in Table 8 below.

Table 8: Description of STEREO_EXTENSION_RATE header field.

STEREO EXTENSION RATE MODE	*Stereo extension rate(incl. BWE)* (bits/frame)	*Number of data octets*
0xFF	No Stereo Extension	0
0x00	40 bits/frame	5
0x01	48 bits/frame	6
0x02	56 bits/frame	7
0x03	64 bits/frame	8
0x04	72 bits/frame	9
0x05	80 bits/frame	10
0x06	88 bits/frame	11
0x07	96 bits/frame	12
0x08	104 bits/frame	13

0x09	112 bits/frame	14
0x0A	120 bits/frame	15
0x0B	128 bits/frame	16
0x0C	136 bits/frame	17
0x0D	144 bits/frame	18
0x0E	152 bits/frame	19
0x0F	160 bits/frame	20

FREQUENCY_SCALE

This field is related to the internal sampling frequency of the audio codec, which in its turn is related to the frame size in ms. The internal sampling frequency in kHz is given by

Fs = FREQUENCY_SCALE x 25.6/96 kHz.

For a value FREQUENCY_SCALE=96, the internal sampling frequency is 25.6 kHz and the 2048-sample encoded super frame corresponds to 80 ms, giving a packet size of 20 ms. For a value FREQUENCY_SCALE=120, the internal sampling frequency is 32 kHz and the 2048-sample encoded super frame corresponds to 64 ms, giving a packet size of 16 ms. The value of FREQUENCY_SCALE is limited to the range 48-144 corresponding to internal sampling frequency range of 12.8-38.4 kHz.

The AMR-WB+ packet is formed as a concatenation of AMR-WB+ Header and AMR-WB+ data (mono followed by stereo). The data octets in each packet are packetized according to the detailed bit allocation given in [2], tables 14 to 20.

For AMR-WB+ operation, the first three octets contain the header fields MONO_RATE, STEREO_EXTENSION_RATE and FREQUENCY_SCALE. The nm+ns data octets follow. The first bit of the AMR-WB+ data b0 is placed in bit 8 of octet 4. Table 9 shows the composition for the example of AMR-WB+ packet with 272 bits/frame mono rate, 88 bits/frame stereo extension rate, and FREQUENCY_SCALE=96 corresponding to 25.6 internal sampling frequency and 20 ms packets (80 ms superframe).

Table 9: AMR-WB+ packet for 272 bits/frame mono rate, 88 bits/frame stereo extension rate, and FREQUENCY_SCALE=96.

	MSB							LSB
Octet	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1
1	**MONO_RATE =18 (272bits/frame)**							
	0	0	0	1	0	0	1	0
2	STEREO_EXTENSION_RATE =7 (88 bits/frame)							
	0	0	0	0	0	1	1	1
3	**FREQUENCY_SCALE=96**							
	0	1	1	0	0	0	0	0
4	**AMR-WB+ data (octet 1)**							
	b0	b1	b2	b3	b4	b5	b6	b7

5..36	AMR-WB+ data (octets 2 to 33)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
37	AMR-WB+ data (octet 34)							
	b264	b265	b266	b267	b268	b269	b270	b271
38	AMR-WB+ data (octet 35)							
	s0	s1	s2	s3	s4	s5	s6	s7
39..47	AMR-WB+ data (octet 36 to 44)							
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö
48	AMR-WB+ data (octet 45)							
	s80	s81	s82	s83	s84	s85	s86	s87

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040722

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.290** **CR** **012** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: | UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Editorial changes
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change:	⌘	One editorial change that adds an new Internal Sampling Frequency (ISF) to limited set of ISFs used in 3GP file format and in transport.
Summary of change:	⌘	The ISF value 36000 is added to Table 24.
Consequences if not approved:	⌘	Limits the flexibility of the codec in storage and transport.

Clauses affected:	⌘	8.1			
Other specs affected:	⌘	Y / N	Other core specifications	⌘	26.304
		☐ / ☒	Test specifications		
		☐ / ☒	O&M Specifications		
Other comments:	⌘				

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

8 Storage and Transport formats

The AMR-WB+ codec storage and transport formats are described in this section.

8.1 Available Modes and Bitrates

The AMR-WB+ format contains the AMR-WB modes and a set of AMR-WB+ extension modes.
The AMR-WB+ codec includes the AMR-WB modes, as shown in Table 21 below.

Ö <cut text> Ö

Table 24: Internal sampling frequencies and corresponding frame lengths in time

ISF Index	*Internal Sampling Rate (Hz)*	*Frame duration (ms)*	*Bit Rate factor*
0	N/A	20	N/A
1	12800	40	Ω
2	14400	35.55	9/16
3	16000	32	5/8
4	17067	30	2/3
5	19200	26.67	æ
6	21333	24	5/6
7	24000	21.33	15/16
8	25600	20	1
9	28800	17.78	9/8
10	32000	16	5/4
11	34133	15	4/3
12	36000	14.22	45/32
~~12~~13	38400	13.33	3/2

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040723

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **013** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Removal of the eid tool
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change:	⌘	The AMR-WB+ 3GPP floating point reference code contains a tool for error insertion into the bitstream file. The tool has been used for verification purposes and is not part of the codec. Additionaly the tool is based on a specific file format used in the selection phase and which is no longuer supported.
Summary of change:	⌘	The eid tool directory is removed from the source code distribution
Consequences if not approved:	⌘	The eid tool may be used for frame errasure insertion but the results would not be correct.
Clauses affected:	⌘	The eid directory in the source code distribution

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

```
The directory eid/ is removed from the source code distribution
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040766

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **014** ⌘ **rev** **1** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Addition of frame erasures simulation at the decoder
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:
Ph2 (GSM Phase 2)
R96 (Release 1996)
R97 (Release 1997)
R98 (Release 1998)
R99 (Release 1999)
Rel-4 (Release 4)
Rel-5 (Release 5)
Rel-6 (Release 6)
Rel-7 (Release 7)

Reason for change:	⌘	The AMR-WB+ 3GPP floating point reference code contains a tool for error insertion into the bitstream file. The tool has been used for verification purposes and is not part of the codec. Additionaly the tool is based on a specific file format used in the selection phase and which is no longuer supported.
Summary of change:	⌘	Simulation of frame errasures is done in the decoder
Consequences if not approved:	⌘	Not possible to simulate frame errasures with the AMR-WB+ 3GPP reference code

Clauses affected:	⌘	dec_wbplus.c mem.h

Other specs affected: ⌘	Y	N	
		X	Other core specifications
		X	Test specifications
		X	O&M Specifications

⌘

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

Additional functionality for the decoder on the command line level is implemented. The changes are editorial and do not affect bit-exactness.

1. How the code is changed in the file *mem.h*

Lines before the change

```
typedef struct
{
   short mode;                  /* AMR_WB core mode: 0..8 */
   short extension;             /* 0=AMRWB, 1=mono, 2=stereo20%, 3=stereo25% */
   short st_mode;               /* stereo mode 0..13 */
   short fscale;
   long fs;
   int mono_dec_stereo;
   int limiter_on;
   short FileFormat;

} DecoderConfig;
```

Lines after the change

```
typedef struct
{
   short mode;                  /* AMR_WB core mode: 0..8 */
   short extension;             /* 0=AMRWB, 1=mono, 2=stereo20%, 3=stereo25% */
   short st_mode;               /* stereo mode 0..13 */
   short fscale;
   long fs;
   int mono_dec_stereo;
   int limiter_on;
   short FileFormat;
  short fer_sim;                /* frame errasures simulation */
} DecoderConfig;
```

2. How the code is changed in the file *dec_wbplus.c*

After the change, the bad frame indicator is read from a file

```
      /* set the tfi */
      tfi = frame%4;
      /* read frame erasures every forth frame */
      if(conf.fer_sim) {
         if(tfi == 0)
         {
            for(i=0;i<4;i++) {
               fscanf(f_fer,"%d",&bfi[i]);
            }
         }
       }
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040725

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **015** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ [ ] ME [X] Radio Access Network [ ] Core Network [ ]

Title:	⌘	Removal of two unused stereo rate
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	There are two invalid entries in the table for parametric stereo bit allocation which are not present in the written specification.
Summary of change:	⌘	Removal of two entries in stereo bit allocation table.
Consequences if not approved:	⌘	Mismatch between written specification and implementation in ANSI-C code.

Clauses affected:	⌘	tables_stereo.c table_decl.h

	Y	N		
Other specs affected: ⌘		X	Other core specifications	⌘
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

1. How the code is changed in the file *tables_stereo.c*

Lines 5-23 before the change

```
const int StereoNbits[18] = {
  (int)(1.2*80),  /* 1.2 + 0.8 = 2.0 kbps */
  (int)(1.4*80),  /* 1.4 + 0.8 = 2.2 kbps */
  (int)(1.6*80),  /* 1.6 + 0.8 = 2.4 kbps */
  (int)(1.8*80),  /* 1.8 + 0.8 = 2.6 kbps */
  (int)(2.0*80),  /* 2.0 + 0.8 = 2.8 kbps */
  (int)(2.4*80),  /* 2.4 + 0.8 = 3.2 kbps */
  (int)(2.8*80),  /* 2.8 + 0.8 = 3.6 kbps */
  (int)(3.2*80),  /* 3.2 + 0.8 = 4.0 kbps */
  (int)(3.6*80),  /* 3.6 + 0.8 = 4.4 kbps */
  (int)(4.0*80),  /* 4.0 + 0.8 = 4.8 kbps */
  (int)(4.4*80),  /* 4.4 + 0.8 = 5.2 kbps */
  (int)(4.8*80),  /* 4.8 + 0.8 = 5.6 kbps */
  (int)(5.2*80),  /* 5.2 + 0.8 = 6.0 kbps */
  (int)(5.6*80),  /* 5.6 + 0.8 = 6.4 kbps */
  (int)(6.0*80),  /* 6.0 + 0.8 = 6.8 kbps */
  (int)(6.4*80),  /* 6.4 + 0.8 = 7.2 kbps */
  (int)(6.8*80),  /* 6.8 + 0.8 = 7.6 kbps */
  (int)(7.2*80)}; /* 7.2 + 0.8 = 8.0 kbps */
```

Lines 5-21 after the change

```
const int StereoNbits[16] = {
  (int)(1.2*80),  /* 1.2 + 0.8 = 2.0 kbps */
  (int)(1.6*80),  /* 1.6 + 0.8 = 2.4 kbps */
  (int)(2.0*80),  /* 2.0 + 0.8 = 2.8 kbps */
  (int)(2.4*80),  /* 2.4 + 0.8 = 3.2 kbps */
  (int)(2.8*80),  /* 2.8 + 0.8 = 3.6 kbps */
  (int)(3.2*80),  /* 3.2 + 0.8 = 4.0 kbps */
  (int)(3.6*80),  /* 3.6 + 0.8 = 4.4 kbps */
  (int)(4.0*80),  /* 4.0 + 0.8 = 4.8 kbps */
  (int)(4.4*80),  /* 4.4 + 0.8 = 5.2 kbps */
  (int)(4.8*80),  /* 4.8 + 0.8 = 5.6 kbps */
  (int)(5.2*80),  /* 5.2 + 0.8 = 6.0 kbps */
  (int)(5.6*80),  /* 5.6 + 0.8 = 6.4 kbps */
  (int)(6.0*80),  /* 6.0 + 0.8 = 6.8 kbps */
  (int)(6.4*80),  /* 6.4 + 0.8 = 7.2 kbps */
  (int)(6.8*80),  /* 6.8 + 0.8 = 7.6 kbps */
  (int)(7.2*80)}; /* 7.2 + 0.8 = 8.0 kbps */
```

2. How the code is changed in the file *table_decl.h*

Line 83 before the change

```
  extern const int StereoNbits[18];
```

Line 83 after the change

```
  extern const int StereoNbits[16];
```

3GPP TSG-SA4 Meeting #33
Helsinki, Finland, 22-26 November 2004

Tdoc ⌘S4-040726

CR-Form-v7.1

CHANGE REQUEST

⌘ **26.304** **CR** **016** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☒ Radio Access Network ☐ Core Network ☐

Title:	⌘	Source code editorial changes
Source:	⌘	TSG-SA WG4
Work item code:	⌘	AMRWB+
Date:	⌘	14/12/2004
Category:	⌘	**D**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction 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.

Use one of the following releases:

Ph2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*
Rel-7	*(Release 7)*

Reason for change:	⌘	The AMR-WB+ 3GPP floating point reference code contains a declared vector that that is not used
Summary of change:	⌘	Removal of unused vector extStMode
Consequences if not approved:	⌘	Unused parameters may degrade the readability of the code.

Clauses affected:	⌘	encwb_plus.c			
Other specs affected:	⌘	**Y**	**N**		
			X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	
Other comments:	⌘				

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at http://www.3gpp.org/specs/CR.htm.
Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

Changes to the C-code:

The following lines of code are deleted:

```
const int extStMode[2*8] = {
    /* stereo mode 1: stereo rate approx. 20% of total bitrate */
    0,  /* 9.6  + 2.0 + 1.6 = 13.2 kbps */
    1,  /* 11.2 + 2.4 + 1.6 = 15.2 kbps */
    2,  /* 12.8 + 2.8 + 1.6 = 17.2 kbps */
    3,  /* 14.4 + 3.2 + 1.6 = 19.2 kbps */
    4,  /* 16.0 + 3.6 + 1.6 = 21.2 kbps */
    5,  /* 18.4 + 4.0 + 1.6 = 24.0 kbps */
    8,  /* 20.0 + 5.2 + 1.6 = 26.8 kbps */
    8,  /* 23.2 + 5.2 + 1.6 = 30.0 kbps */
    /* stereo mode 2: stereo rate approx. 25% of total bitrate */
    2,  /* 9.6  + 2.8 + 1.6 = 14.0 kbps */
    3,  /* 11.2 + 3.2 + 1.6 = 16.0 kbps */
    4,  /* 12.8 + 3.6 + 1.6 = 18.0 kbps */
    5,  /* 14.4 + 4.0 + 1.6 = 20.0 kbps */
    7,  /* 16.0 + 4.8 + 1.6 = 22.4 kbps */
    9,  /* 18.4 + 5.6 + 1.6 = 25.6 kbps */
    12, /* 20.0 + 6.8 + 1.6 = 28.4 kbps */
    13};/* 23.2 + 7.2 + 1.6 = 32.0 kbps */
```