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

Tdoc **x** S4-040711

CHANGE REQUEST						
æ	26.290 CR 001 # rev - [#]	Current version: 6.0.0 ^{BE}				
For <u>HELP</u> o	n 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 stereo bit allocation tables					
Source:	郑 TSG-SA WG4					
Work item code	amentary and a second s	Date: 器 14/12/2004				
Category:	 F Use <u>one</u> 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 <u>TR 21.900</u>. 	Release:XRel-6Use one 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 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	X A wrong interpretation of the bitstream and the location of the reserved bit may		
not approved:	lead to the possibility wrong decoding in the case of stereo bitstreams		
Clauses affected:	H 7		
	YN		
Other specs	X Other core specifications X 26.304		
affected:	X Test specifications		
	X O&M Specifications		

How to create CRs using this form:

Ж

Other comments:

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

- 1) Fill out the above form. The symbols above marked **B** 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 <u>ftp://ftp.3gpp.org/specs/</u> 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

	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			

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	<u>b0-b3</u>	<u>b0-b6</u>			
Midband gain	<u>b4-b5</u>	<u>b7-b11</u>			
Lowband stereo					
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>			
<u>reserved</u>	<u>b8</u>	<u>b14</u>			
Balance factor	<u>B9-b15</u>	<u>b15-b21</u>			
<u>Global gain</u>	<u>b16 ñ b22</u>	<u>b22-b28</u>			
Algebraic VQ	<u>b23 ñ bN₁</u>	<u>b29- bN₁</u>			

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	<u>N bits/frame > 76</u>		
	Midband stereo			
Midband filter	<u>b0-b3</u>	<u>b0-b6</u>		
Midband gain	<u>b4-b5</u>	<u>b7-b11</u>		
Lowband stereo				
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>		
reserved	<u>b8</u>	<u>b14</u>		
Balance factor	<u>B9-b15</u>	<u>b15-b21</u>		
<u>Algebraic VQ</u>	<u>b16 ñ bN₁</u>	<u>b22- bN₁</u>		

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	<u>N bits/frame <=76</u>	N bits/frame > 76		
	Midband stereo			
Midband filter	<u>b0-b3</u>	<u>b0-b6</u>		
<u>Midband gain</u>	<u>b4-b5</u>	<u>b7-b11</u>		
Lowband stereo				
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>		
<u>reserved</u>	<u>b8</u>	<u>b14</u>		
<u>Global gain</u>	<u>b9-b15</u>	<u>b15-b21</u>		
Algebraic VQ	<u>b16ñ bN₁</u>	<u>b22- bN₁</u>		

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 ₄				

	Bits (MSB-LSB)			
Description	N bits/frame <=76	N bits/frame > 76		
	Midband stereo			
Midband filter	<u>b0-b3</u>	<u>b0-b6</u>		
Midband gain	<u>b4-b5</u>	<u>b7-b11</u>		
Lowband stereo				
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>		
reserved	<u>b8</u>	<u>b14</u>		
Balance factor	<u>b9-b15</u>	<u>b15-b21</u>		
Algebraic VQ	<u>b16ñ bN₁</u>	<u>b22- bN₁</u>		

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 bN₁	b14- bN 1					

	<u>Bits (MSB-LSB)</u>			
Description	N bits/frame <=76	N bits/frame > 76		
	Midband stereo			
Midband filter	<u>b0-b3</u>	<u>b0-b6</u>		
<u>Midband gain</u>	<u>b4-b5</u>	<u>b7-b11</u>		
Lowband stereo				
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>		
<u>reserved</u>	<u>b8</u>	<u>b14</u>		
<u>Algebraic VQ</u>	<u>b9 ñ bN₁</u>	<u>b15- bN₁</u>		

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	<u>b0-b3</u>	<u>b0-b6</u>		
<u>Midband gain</u>	<u>b4-b5</u>	<u>b7-b11</u>		
	Lowband stereo			
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>		
<u>reserved</u>	<u>b8</u>	<u>b14</u>		
<u>Global gain</u>	<u>B9-b15</u>	<u>b15-b21</u>		
Algebraic VQ	<u>b16 ñ bN₁</u>	<u>b22- bN₁</u>		

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- bN₁					

	Bits (MSB-LSB)			
Description	N bits/frame <=76	N bits/frame > 76		
	Midband stereo			
Midband filter	<u>b0-b3</u>	<u>b0-b6</u>		
<u>Midband gain</u>	<u>b4-b5</u>	<u>b7-b11</u>		
	Lowband stereo			
Mode bits	<u>b6-b7</u>	<u>b12-b13</u>		
<u>reserved</u>	<u>b8</u>	<u>b14</u>		
Algebraic VQ	<u>b9 ñ bN₁</u>	<u>b15- bN₁</u>		

Tdoc **x** S4-040712

	CHANGE REQUEST							
æ	26.290 CR 002	rev - 🖁	Current version: 6.0.0					
For <u>HELP</u> o	n using this form, see bottom of this p	age or look at the	e pop-up text over the H symbols.					
Proposed chang	ge affects: UICC apps	ME <u>X</u> Radio A	ccess Network Core Network					
Title:	Correction of storage format for	AMR-WB+						
Source:	XI TSG-SA WG4							
Work item code	: # AMR-WB+		<i>Date:</i> <mark>೫ 14/12/2004</mark>					
Category:	 F Use <u>one</u> of the following categories: <i>F</i> (correction) <i>A</i> (corresponds to a correction i <i>B</i> (addition of feature), <i>C</i> (functional modification of feature), <i>C</i> (functional modification) D (editorial modification) Detailed explanations of the above categories 	n an earlier release ^t ure) tegories can	Release:Rel-6Use one Ph2of the following releases: (GSM Phase 2)e)R96(Release 1996)R97(Release 1997)R98(Release 1998)R99(Release 1999)Rel-4(Release 4)Rel-5(Release 5)Rel-6(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	# A wrong interpretation may lead to different implementations of the way the
not approved:	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:	第 8.3
Other specs affected:	Y N X Other core specifications X X Test specifications X X O&M Specifications X
Other comments:	₩

How to create CRs using this form:

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

1) Fill out the above form. The symbols above marked **B** 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 <u>ftp://ftp.3gpp.org/specs/</u> 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 <u>only</u> for file storage and defines <u>the bitstreama storage unit</u> contained in an AMR-WB+ sample of a 3GP file [9]. It is <u>quite</u> similar to transport format with the exception that the two-octet header is used once per superframe <u>for AMR-WB+ extension modes and once per frame for AMR-WB modes(a storage sample consist of 4</u> frames or one superframe). Note that in AMR-WB+, the operation code and internal sampling frequency can be switched only on <u>a</u> superframe <u>basis</u> 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 <u>the syntax of the basic</u> 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 $OC = 16\ddot{O} 47$.

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, tThe 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 80ISF/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		IS	F mode (5 bi	ts)	

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 x S4-040763

	CHANGE REQUEST							
æ	<mark>26.290</mark>	CR 003	ж rev	1 [#]	Current vers	ion: 6.	0.0	æ
For <u>HELP</u> on	using this fo	rm, see bottom of	this page or	look at the	e pop-up text	over the	<mark>ж</mark> syn	nbols.
Proposed change	e affects:	UICC apps <mark>#</mark>	ME X	Radio A	ccess Netwo	·k 📃 Co	ore Ne	twork
Title:	Beditorial of	changes						
Source:	K TSG-SA	WG4						
Work item code:	* AMRWB-	F			Date: 🕷	14/12/2	2004	
Category:	D Use one of F (cor A (cor B (add C (fund D (edu) D (edu) be found in	the following catego rection) responds to a correc dition of feature), actional modification itorial modification) planations of the abo 3GPP <u>TR 21.900</u> .	ries: ction in an ear of feature) ove categories	lier release s can	Release: Use <u>one</u> of Ph2 Ph2 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	Rel-6 the follown (GSM Ph. (Release (Release (Release (Release (Release (Release (Release	ing rele ase 2) 1996) 1997) 1998) 1999) 4) 5) 6) 7)	ases:
Reason for chang	A (cor B (add C (fur D (edi Detailed ex be found in	responds to a correct dition of feature), ictional modification itorial modification) planations of the abo 3GPP <u>TR 21.900</u> .	ction in an ear of feature) ove categories s that improv	rlier release s can ve the spe	e) R96 R97 R98 R99 Rel-4 Rel-5 Rel-5 Rel-6 Rel-7	(Release (Release (Release (Release (Release (Release (Release (Release	1996) 1997) 1998) 1999) 4) 5) 6) 7) nat the	

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 ca frame type.				
Summary of change: ೫	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 ℜ 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 <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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 <u>Transport Transport Interface</u> 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.

Operation	Core	Stereo	Bit rate	Octets per		
code Frame	mode	mode		frame		
type						
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					

Table 25: Normative operation codeframe typetable. Bit-rates assumes 25600 Hz internal samplingfrequency.

8.2 AMR-WB+ Transport Interface Format

<u>The transport interface format serves as an intermediate interface to the transport format.</u> The <u>transport transport</u> <u>interface</u> 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	TF	0 ISF mode (5 bits)						

Operation code-Frame type(FTOC) (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 frame is 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).

OCFT=14 (AUDIO_LOST) is used to indicate frames that are lost. NO_DATA (OCFT=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 FTOC 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 OCFT 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 (FTOC=26) with a frame length of 35 octets (280 bits). The internal sampling frequency in this example is 25.6 kHz (ISF mode = 8). $\frac{\text{OC-FT}}{26}$ 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).

Ì

	MSB							LSB	
Octet	bit 8	bit 7	bit 6	bit 5	Bit 4	bit 3	bit 2	bit 1	
_									
1				OC-F	<u> </u>				
	0	0	0	1	1	0	1	0	
2	TF	I=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	
427		AMR-WB+ data (octets 2 to 25)							
	b8	Ö	Ö	Ö	Ö	Ö	Ö	Ö	
28				AMR-WB+ d	ata (octet 26)			
	b200	b201	b202	b203	B204	b205	b206	b207	
29				AMR-WB+ d	ata (octet 27)			
	s0	s1	s2	s3	s4	s5	s6	s7	
3036			AN	/IR-WB+ data	octet 28 to	34)			
	s8	Ö	Ö	Ö	Ö	Ö	Ö	Ö	
37				AMR-WB+ d	ata (octet <u>35</u>)			
	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 <u>interface</u> 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 <u>frame type (OCFT)</u> (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	<u>type</u> (7 bits)		
2	0	0	0		IS	F mode (5 bit	ts)	

Operation code Frame type (FTOC) (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 $\underline{\text{FTOC}}$ 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 <u>frame</u> 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 <u>coding operation frame type</u> (<u>FTOC</u>=26) with a superframe length of $4\langle 35=140 \text{ octets}$. The internal sampling frequency in this example is 25.6 kHz (ISF mode = 8). <u>FTOC</u> 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 $\underline{\text{FT}}_{\Theta C}$ <10, i.e. AMR-WB frames, the original AMR-WB storage format should be preferred in order to ensure backward decoding compatibility.

П

1

	MSB							LSB
Octet	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1
1				<u>FT</u> OC	= 26			
	0	0	0	1	1	0	1	0
2		1	1	ISF	= 8	-		
	0	0	0	0	1	0	0	0
3			Fra	me 1 AMR-W	B+ data (oc	tet 1)		
	b0	b1	b2	b3	<u>b4</u>	<u>b5</u>	b6	b7
427	1.0	Ä	Frame	1 AMR-WB+	data (octets	; 2 to 25)	ä	Ä
	80	0				0	0	0
28	1-000	L 004	Frar		3+ data (oct	et 26)	1-000	L 6007
20	D200	D201	D202		B204	D205	D206	D207
29	- 0	- 4	Frar			et 27)		-7
20.26	SU	51	SZ		S4		50	57
3030	- 0	- A	⊢rame	1 AMR-WB+			ö	Ö
27	<u> </u>	0					0	0
37	004	0.05			$\frac{6+0ata(0ct}{2}$		070	074
20	564	505	500		<u> </u>	509	570	5/1
30	b.O.	6.4	Fia د ا	ne z Alvik-vv			hC	h7
20.62	00					CQ CQ	00	D7
3902	b 0	ő	Frame	Z AIVIR-VVB+		5 Z TO ZS)	Ö	Ö
62	Do	0				U 0	0	
03	h200	h201	Fiai	h202		el 20)	h206	h207
64	0200	0201			DZU4		0200	0207
04	- 0	-4	Fran	ne z Alvik-vve		et 27)		-7
6E 74	<u>su</u>	51	SZ Fromo		S4		50	57
03/1	~ 9	ň	- rame	Z AIVIR-VVD+		<u>20 (0 34)</u> Ö	Ö	Ö
70	50		Г U				0	0
12	864	SOF	Fran	ne z Alvik-we	$\frac{6+0ata(0ct}{2}$		670	674
72		500	500		000 Di data (aa	509 tot 1)	570	5/1
13	<u>هم</u>	h1	Fia د ا	he 3 AIVIR-VV			he	h7
74 07	DU		Eromo		U4 data (actob		00	07
7497	60	Ö	riame Ö	S AIVIR-WD+		5 Z 10 Z5)	Ö	Ö
0.9	Do	0				U	0	
90	b 200	h201	Fidi	hooo		b205	h200	h207
00	0200	1020	D2UZ		DZU4	D200	0206	0207
33								07
100 106			Eramo	2 AMD \A/D +	o 4	28 ± 24	50	57
100100	c 2	Ö				20 (0 34) Ö	Ö	Ö
107	30		Fran		U too) eteb +8	ot 35)	0	
107	S64	\$65			268		\$70	\$71
108	004	000	Fra	me 4 AMP-W/	B± data (oc	tot 1)	570	
100	b0	b1	h2		Бт цага (ОС БЛ	b5	h6	h7
100 132	00		Eramo		U 4 data (octote	2 to 25)	00	
109132		Ö	Ö			<u>δ Z (0 Z 3)</u>	Ö	Ö
133	00		Fran		U too) eteb ±8	ot 26)	0	
155	h200	h201	h202	h203		b205	h206	h207
134	0200	0201	Fran		Bz04	ot 27)	0200	0207
10-1	s0	e1				<u>~</u> 5	s6	۶7
135 141		31	Eramo		data (octot (28 to 34)	30	51
135141	<u>8</u>	Ö				<u>حە رە 54)</u> م	Ö	Ö
1/2	30		Erar			ot 35)	0	
142	564	 					\$70	Q71
				507		303	0/0	0/1

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

			CI	HANG	E RE	EQL	JES	Т				CR-Form-v7
[#] TS	26.	. <mark>304</mark>	CR <mark>0</mark>	01	жre	€V	- *	Cur	rent vers	sion:	6.0.0	H
For <u>HELP</u> on us	sing t	his fori	m, see b	ottom of t	this pag	e or lo	ok at i	the pop	o-up text	t over	the ¥ sy	mbols.
Proposed change a	affect	t s: U	IICC app	s#	M	EX	Radio	Acces	s Netwo	rk	Core N	etwork
Title: ೫	Inco	orrect o	definition	of mode	index fo	or SID	frame	S				
Source: ೫	TSC	<mark>G-SA V</mark>	VG4									
Work item code: ೫	AM	RWB+							Date: ೫	200	04-12-14	
Category: Ж	F Use <u>c</u> Detai be fo	one of t F (corr A (corr B (add C (func D (edit iled exp und in 3	he followi ection) esponds ition of fea tional mod lanations 3GPP <u>TR</u>	ng categor to a correc ature), dification o fication) of the abo <u>21.900</u> .	ries: ction in a of feature ove categ	n earli e) gories (<i>er relea</i> can	Rel Us ase)	lease: % se <u>one</u> of 2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-6	Financial Relevance (Relevance) (Relevance	L-6 Illowing rei A Phase 2, pase 1996, pase 1997, pase 1999, pase 4) pase 5) pase 6)	leases:))))
Posson for change		In the		B+ floatir	na point	C-00	do (co	ntaine	d in TS 2	06 30/	1) the de	finitions
neason for change.		of MF	RDTX an ged to be	d NUM_C consiste	DF_MOI	DES L TS 26	used fo 3.201.	or AMR	R-WB coo	ding n	nodes are	e
Summary of change	е: Ж	The c code	definition (TS 26.3	s of mode 304) are c	e index a correcte	and th d.	e num	ber of	modes i	in the	floating p	ooint C-
Consequences if not approved:	es if # 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 th number of modes.) This may raise confusion and even interoperability problems when implementing the generic AMR-WB interface formats of TS 26.201.						des used S 26.304 rmats be index Also the 11 as the problems 1.					
Clauses affected.	ቋ	dec i	fcenc	dtx c en	<u>cifca</u>	nc m	ain c fi	les				
Other specs affected:	# 	Y N X X X	Other co Test spo O&M Sp	ore specification	fications	60_111 }	#					
Other comments:	ж											

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

			C	HANG	GE RI	EQL	JES	ST				CR-Form-v7
* <mark>T</mark>	<mark>6 26</mark>	. <mark>304</mark>	CR	002	жг	ev	-	ж	Current ver	sion:	6.0.0	ж
For <u>HELP</u> on u	ising t	this for	m, see	bottom of	f this pag	e or lo	ok a	t the	pop-up tex	t over	the X syr	nbols.
Proposed change	affec	ts: L	JICC ap	ps#	M	X	Radi	o Ac	cess Netwo	ork	Core Ne	etwork
Title: #	Co	rection	of TC	X coding s	selection	for MI	MS e	enco	der			
Source: #	TS	<mark>G-SA V</mark>	VG4									
Work item code: ¥	AM	RWB+							Date: ೫	e <mark>20</mark>	04-12-14	
Category: #	F Use Deta be fo e: #	one of t F (corr A (corr B (add C (func D (edit iled exp und in (TCX2 the si used Wher	he follo ection) respond ition of i ctional n orial mo lanation 3GPP <u>1</u> 256 exc uperfra	wing categ s to a corre feature), nodificatior dification) is of the at <u>R 21.900</u> . Station en me and the coding is s	ories: ection in a n of feature pove categ coding is ne flag is selected	n earlie e) gories o not e set inc	er rele can dicat	ease, ited, ing 1	Release: # Use <u>one</u> o 2) R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 When TCX TCX1024 co	f the fc (GSM (Rela (Re) (Rela (Re) (Re) (Re) (Re) (Re) (Re) (Re) (Re)	L-6 M Phase 2) Pase 1996) Pase 1997) Pase 1998) Pase 1999) Pase 4) Pase 5) Pase 6) g is selected option can	ed for not be
possibilities are not tried. If the flag is not set enabling the use of TCX10 either TCX512 and TCX1024 is selected for the encoding. If the flag is s disabling the use of TCX1024, either TCX256 and TCX512 is selected for encoding. The change request conserns the addition of TCX256 in the la case.					24, et or the atter							
Consequences if not approved:	Ħ	тсх	coding	selection	is not ex	ecute	d coi	rrect	ly in MMS e	encode	er.	
Clauses offented	مە	acd	fha									
Clauses affected:	ፙ											
Other specs affected:	ж	Y N X X X X	Other Test s O&M \$	core spec pecificatio Specificat	cifications ons ions	;	ж					
Other comments:	ж											

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)) {
```

							CP Form V
			CHA	NGE RE	QUES	Г	CR-rom-vi
ж	TS 2	<mark>26.304</mark>	CR <mark>003</mark>	жrev	۲ - ^ж	Current vers	sion: 6.0.0 [%]
For <u>HELP</u>	on usii	ng this for	m, see bottom	of this page	or look at ti	he pop-up text	over the # symbols.
Proposed cha	ange af	fects: L	JICC apps ೫ <mark>_</mark>	ME	X Radio	Access Netwo	rk Core Network
Title:	Ħ	Misread o energy bu	f energy buffe ffer initialisatio	r in coding m on	ode selectio	on in MMS end	coder. Correction of
Source:	ж	<mark>TSG-SA V</mark>	VG4				
Work item co	de: ೫	AMRWB+				Date: ೫	2004-12-14
Category:	₩ U D b	F Jse <u>one</u> of t F (corr A (corr B (ada C (fund D (edit Detailed exp e found in 3	the following cat rection) responds to a co lition of feature), ctional modificatio orial modificatio lanations of the 3GPP <u>TR 21.90</u>	egories: prrection in an ion of feature) n) above catego <u>0</u> .	earlier releas ries can	Release: % Use <u>one</u> of 2 se) R96 R97 R98 R99 Rel-4 Rel-5 Rel-6	REL-6 the following releases: (GSM Phase 2) (Release 1996) (Release 1997) (Release 1998) (Release 1999) (Release 4) (Release 5) (Release 6)
Reason for ch	hando.	와 In the	a coding mode	selection th	enerav h	uffer is read at	incorrect position. The
	lange.	initial	isation of the e	energy buffer	is also cor	rected	inconcer position. The
Summary of c	change.	: # In the searc of bu buffe	e coding mode ched from the ffer, therefore r initialisation a	selection, th energy buffer searched ma and reading is	e maximum . The energ ximum energes s corrected	n energy value gy buffer is not ergy value is in	of the last frames is read at the beginning correct. The energy
Consequence not approved	es if I:	第 Maxi at the	mum energy v e beginning of	alue is calcul energy buffe	ated incorre	ectly, because gy buffer is no	pointer is not pointing t correctly initialised.
Clauses affec	ted:	策 nclas	S.C				
Other specs affected:		¥ N 米 X ス メ	Other core sp Test specifica O&M Specific	pecifications ations cations	¥		
Other comme	ents:	ж					

1) Change in nclass.c

Before the change:

- The initialisation of the energy buffer (line 39) stClass->TotalEnergy[0]= 0;
 - 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++) {

		CHANGE	EREQ	UES	т			CR-Form-v7.1
ж	<mark>26.304</mark>	CR 004	ж rev	- भ	€ Curr	ent versi	^{ion:} 6.0.(<mark>) ж</mark>
For <u>HELP</u> on	using this fo	rm, see bottom of thi	s page or	look at	the pop	-up text	over the ¥ s	ymbols.
Proposed change	e affects:	UICC apps 	ME X	Radio	Access	s Networ	k Core	Network
Title:	f Correctio	n of stereo bit allocat	tion tables					
Source:	∜ <mark>TSG-SA</mark>	WG4						
Work item code:	<mark>€ AMRWB</mark> -	+				Date: ೫	14/12/2004	ł
Category: S	F Use <u>one</u> of <i>F</i> (cor <i>A</i> (cor <i>B</i> (add <i>C</i> (fur <i>D</i> (edd Detailed ex be found in	the following categorie rection) rresponds to a correction dition of feature), actional modification of itorial modification) planations of the above 3GPP <u>TR 21.900</u> .	s: on in an ear feature) e categories	rlier relea s can	Rele Us ase)	ease: # e <u>one</u> of t Ph2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-5 Rel-6 Rel-7	Rel-6 the following r (GSM Phase (Release 199 (Release 199 (Release 199 (Release 4) (Release 5) (Release 6) (Release 7)	eleases: 2) 6) 7) 8) 9)

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 #	A wrong interpretation of the bitstream and the location of the reserved bit may							
not approved:	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							
	YN							
Other specs #	X Other core specifications # 26.290							
affected:	X Test specifications							
	X O&M Specifications							
Other comments: ೫								

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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++) {</pre>
  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++) {</pre>
  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 */</pre>
```

```
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++)</pre>
         {
            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 = 2i
         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*/
                                      ptr += 7;
         prm[1] = bin2int(7, ptr);
         pack4bits_d(nbits_AVQ[1], ptr, prm_AVQ+j);
         /\,{}^{\star} demultiplex and decode tcx parameters \,{}^{\star}/
         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++)</pre>
         ł
            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 */
                                      ptr += 7;
         prm[1] = bin2int(7, ptr);
         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);
```

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++)</pre>
         {
            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 = 2i
         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*/
                                     ptr += 7;
         prm[0] = bin2int(7, ptr);
         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++)</pre>
         {
           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 = 4i
         nbits AVO[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 */
                                      ptr += 7;
         prm[0] = bin2int(7, ptr);
         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++)</pre>
            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);
         /\,{}^{\star} set pointer to bit stream {}^{\star}/
         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 */
                                    ptr += 2;
         int2bin(mode, 2, ptr);
          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 */
                             ptr += 2;
    int2bin(mode, 2, ptr);
    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 */</pre>
```

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;
                                      ptr += 2;
          int2bin(mode, 2, ptr);
                                    ptr += 7;
          int2bin(prm[0], 7, ptr);
          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;
                                ptr += 7;
    int2bin(prm[0], 7, ptr);
    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 += 1;
       *ptr = 0;
    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 += 1;
       *ptr = 0;
                               ptr += 2;
    int2bin(mode, 2, ptr);
                               ptr += 7;
    int2bin(prm[1], 7, ptr);
    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 += 1;
       *ptr = 0;
    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*

```
CR page 10
```

```
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 */
                  /* 1.4 + 0.8 = 2.2 kbps */
  (int)(1.4*80),
                  /* 1.6 + 0.8 = 2.4 kbps */
  (int)(1.6*80),
                  /* 1.8 + 0.8 = 2.6 kbps */
  (int)(1.8*80),
                  /* 2.0 + 0.8 = 2.8 kbps */
  (int)(2.0*80),
                  /* 2.4 + 0.8 = 3.2 kbps */
  (int)(2.4*80),
                  /* 2.8 + 0.8 = 3.6 kbps */
  (int)(2.8*80),
                  /* 3.2 + 0.8 = 4.0 kbps */
  (int)(3.2*80),
  (int)(3.6*80), /* 3.6 + 0.8 = 4.4 \text{ kbps } */
  (int)(4.0*80), /* 4.0 + 0.8 = 4.8 \text{ kbps } */
  (int)(4.4*80), /* 4.4 + 0.8 = 5.2 \text{ kbps } */
  (int)(4.8*80), /* 4.8 + 0.8 = 5.6 \text{ kbps } */
  (int)(5.2*80), /* 5.2 + 0.8 = 6.0 \text{ kbps } */
  (int)(5.6*80), /* 5.6 + 0.8 = 6.4 \text{ 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 */
```
Tdoc **#S4-040764**

		CHANC	GE REQ	UE	ST			C	CR-Form-v7.1		
ж	26.30 4	CR 005	ж rev	1	ж	Current vers	sion:	6.0.0	ж		
For <u>HELP</u> on	using this fo	orm, see bottom of	this page or	look a	at th	e pop-up text	over	the X syr	nbols.		
Proposed change affects: UICC apps # ME X Radio Access Network Core Network											
Title:	f Optimiza	ation of error conce	alment oper	ation							
Source:	∜ <mark>TSG-SA</mark>	WG4									
Work item code: 8	€ <mark>AMRWE</mark>	+				Date: ೫	14/	12/2004			
Category:		f the following catego rrection) orresponds to a corre Idition of feature), nctional modification itorial modification) xplanations of the ab o 3GPP <u>TR 21.900</u> .	ories: ection in an ea of feature) nove categorie:	rlier re s can	lease	Release: % Use <u>one</u> of Ph2 9) R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	Rel the fo. (GSN (Rele (Rele (Rele (Rele (Rele (Rele (Rele	ease 6 llowing rele 1 Phase 2) ase 1996) ase 1997) ase 1998) ase 1999) ase 4) ase 5) ase 5) ase 6) ase 7)	eases:		

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								
Other specs % affected:	Y N X Other core specifications # X Test specifications # X O&M Specifications #								
Other comments: ೫									

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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)</pre>
  {
    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);
                                                     BRANCH(1);ADD(1);
      st->fer_hist_ptr++;
      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++) {</pre>
      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);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++)</pre>
           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);
        \texttt{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++)</pre>
        {
           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;
     ł
   }
                                                     FLC_sub_end();
  Dyn_Mem_Out();
  return;
Lines after the change
                                                     LOOP(1); PTR_INIT(4);
        for(i=0;i<L_FRAME_2k;i++)</pre>
        {
           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;
```

}

}

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

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];

Tdoc **#**S4-040765

			(CHANGE	REQ	UE	ST	1		C	R-Form-v7.1	
æ	26	5 <mark>.304</mark>	CR	006	жrev	1	Ħ	Current vers	ion:	6.0.0	ж	
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:	ដ St	ereo op	eratior	n of pre-echo r	node, sat	turatio	on of	gain_shape				
Source:	ж <mark>т</mark>	SG-SA	NG4									
Work item code:	ж <mark>А</mark>	MRWB+	-					Date: ೫	14/	12/2004		
Category:	策 F Use Det be f	e <u>one</u> of F (con A (cor B (add C (fun D (edi ailed exp found in	the follo rection) respond lition of ctional i torial m olanatio 3GPP	owing categories ds to a correction feature), modification of f odification) ns of the above <u>FR 21.900</u> .	s: n in an ea cature) categorie	rlier re s can	elease	Release: # Use <u>one</u> of Ph2 P) R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	Rele the foi (GSM (Relea (Relea (Relea (Relea (Relea (Relea	-6 llowing rele 1 Phase 2) ase 1996) ase 1997) ase 1998) ase 1999) ase 4) ase 5) ase 5) ase 6) ase 7)	pases:	

Reason for change: #	Due to fixed point implementation issues it is beneficial to limit the gain_shape values to the range [0.52.0]. 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: ೫	A limitation to the range [0.52.0] 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:	YNXOther core specifications#XTest specificationsXO&M Specifications
Other comments: ೫	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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));
}</pre>
```

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]));
}</pre>
```

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++)</pre>
```

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++)</pre>
```

CHANGE REQUEST												
¥		26.304	CR	007	ж rev	-	Ħ	Current vers	sion:	6.0.0	ж	
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:	ж	Stereo op	eratior	of pre-echo r	mode, al	gnem	ient o	of encoder an	d deo	coder		
Source:	ж	TSG-SA	NG4									
Work item code.	: Ж	AMRWB-	-					Date: #	14	/12/2004		
Category:	æ	F Use <u>one</u> of F (con A (cor B (add C (fun D (edi Detailed exp be found in	the follo rection) respond lition of ctional m torial m blanatio 3GPP]	owing categories ds to a correctio feature), modification of t odification) ns of the above <u>FR 21.900</u> .	s: on in an ea feature) e categorie	arlier re	elease	Release: # Use <u>one</u> of Ph2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	Ref (GSI (Rela (Rela (Rela (Rela (Rela (Rela (Rela (Rela	I-6 Dilowing rele M Phase 2) pase 1996) pase 1997) pase 1998) pase 1999) pase 4) pase 5) pase 6) pase 7)	eases:	

Reason for change: ೫	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 #	Potential problem may arise when pre-echo mode is selected since there is a							
not approved:	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 <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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++){</pre>
  wm[i] = mono[i];
if(pre_echo){
  /* compensate for the gain */
  tmp = 0.0f;
  for(i=0;i<lg;i+=16){</pre>
     gain_shap[i/16] = 0.001f;
     for(k=0;k<16;k++)</pre>
     {
        gain_shap[i/16] += wm[i+k]*wm[i+k];
     }
     /\,\star average log gain in frame \,\star\,/\,
     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++)</pre>
{
  wm[i] *= window[i];
}
for (i=0; i<lext; i++) {</pre>
  wm[L_frame+i] *= window[ovlp_size+i];
}
```

Lines xx-xx after the change

```
/* windowed mono */
for (i=0; i<lg; i++){</pre>
  wm[i] = mono[i];
}
 /* windowing for TCX overlap and correlation */
for (i=0; i<ovlp_size; i++)</pre>
ł
  wm[i] *= window[i];
}
for (i=0; i<lext; i++) {</pre>
  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){</pre>
     gain_shap[i/16] = 0.001f;
     for(k=0;k<16;k++)</pre>
     {
       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));
}
</pre>
```

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

Lines xx-xx before the change

```
if(pre_echo)
  ł
     /* compensate for the gain */
     tmp = 0.0f;
     for(i=0;i<lg;i+=16){</pre>
       gain_shap[i/16] = 0.001f;
       for(k=0;k<16;k++)</pre>
       {
          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){</pre>
       gain_shap[i/16] = (float) sqrt(gain_shap[i/16]*pow(10.0f,-tmp));
       for(k=0;k<16;k++)
       ł
          xn[i+k] /= gain_shap[i/16];
        }
     }
  }
   for (i=0; i<lext; i++) {</pre>
#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++) {</pre>
#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){</pre>
    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){</pre>
    gain_shap[i/16] = (float) sqrt(gain_shap[i/16]*pow(10.0f,-tmp));
     for(k=0;k<16;k++)
     ł
       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];
    }
 }
}</pre>
```

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<l6;k++) {
         xnq[i+k] *= gain_shap[i/16];
         xn[i+k] *= gain_shap[i/16];
         wm[i+k] *= gain_shap[i/16];
         }
   }
}</pre>
```

	CHANGE REQUEST													
H	2	<mark>6.304</mark>	CR	008	жre	v	1	ж	Curr	ent ve	ersion:	6.0	.0	ж
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the # symbols.														
							- tao							
Title:	ж <mark>А</mark>	ddition c	of supp	ort for file form	nats ar	nd im	pro	ved o	comm	nand li	ine			
Source:	ж Т	<mark>SG-SA N</mark>	NG4											
Work item code:	ж <mark>А</mark>	<mark>MRWB+</mark>	-						I	Date:	೫ <mark>1</mark> 4	<mark>/12/20(</mark>)4	
Category:	H Ds Us De be	e <u>one</u> of t F (corr A (corr B (add C (fund D (edit tailed exp found in t	the follo rection) respond lition of ctional I torial mo lanatio 3GPP]	owing categories ds to a correctio feature), modification of f odification) ns of the above <u>FR 21.900</u> .	s: n in an feature) categc	earlie) pries	<i>er re</i> can	lease	Rele Us	e <u>one</u> Ph2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	ж R (GS (Re (Re (Re (Re (Re (Re (Re (Re	el-6 following M Phase lease 19 lease 19 lease 19 lease 19 lease 5) lease 5) lease 7)	rele e 2) 196) 197) 198) 199)	ases:

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:	ж	enc_wbplus.c dec_wbplus.c cnst.h, prot_func.h, table_plus.c
Other specs affected:	ж	YNXOther core specifications#XTest specificationsXO&M Specifications
Other comments:	ж	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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.

			(CHANGE	REQ	UE	ST			C	CR-Form-v7.1	
ж		26.304	CR	009	жrev	1	ж	Current ver	sion:	6.0.0	ж	
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:	ж	Source co	ode edi	torial changes	6							
Source:	ж	TSG-SA \	NG4									
Work item code	:Ж	AMRWB+	-					Date: ೫	3 <mark>14</mark>	14/12/2004		
Category:	ж	D Use <u>one</u> of <i>F</i> (corr A (cor B (add C (fun D (edi Detailed exp be found in	the folic rection) respond lition of ctional i torial m blanatio 3GPP]	owing categories ds to a correctio feature), modification of f odification) ns of the above <u>FR 21.900</u> .	s: on in an ea feature) e categorie	rlier re s can	elease	Release: # Use <u>one</u> o Ph2 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	G Ree (GSI (Rela (Rela (Rela (Rela (Rela (Rela (Rela	H-6 bllowing rele M Phase 2) ease 1996) ease 1997) ease 1998) ease 1999) ease 4) ease 5) ease 6) ease 7)	pases:	

Reason for change: ೫	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: ೫	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: #	<pre>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</pre>									
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 <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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

Tdoc *****S4-040719

æ		26.	<mark>304</mark> CR	<mark>010</mark>	жr	ev	-	ж	Curre	ent ver	sion:	6.0	0.0	ж
For <mark>HE</mark>	<u>LP</u> on u	sing th	nis form, se	ee bottom of	this pag	ge or i	look i	at the	e pop-	up tex	t over	the S	₩ syn	nbols.
Proposed change affects: UICC apps # ME X Radio Access Network Core Network														
Title:	ж	Rem	noval of co	mplexity cou	Inters									
Source:	ж	TSG	- <mark>SA WG4</mark>											
Work item	code: Ж	AMF	RWB+						D	ate: 🖁	8 14	/12/2	004	
Category:	ж	D Use o F E C Detail be fou	ne of the fo (correction (correspo (addition (functiona (editorial ed explanat and in 3GPF	llowing catego n) nds to a corre of feature), nl modification modification) ions of the ab 2 <u>TR 21.900</u> .	ories: ection in a of featur ove cate	an ear re) gories	lier re	lease	Rele Use F F F F F F F F	ase: \$ <u>one</u> o 2h2 796 798 798 799 7el-4 7el-5 7el-6 7el-7	f the fo (GSI (Rela (Rela (Rela (Rela (Rela (Rela (Rela (Rela	I-6 ollowin M Pha ease ease ease ease ease ease ease ea	ng rele ise 2) 1996) 1997) 1998) 1999) 4) 5) 5) 7)	ases:
Reason fo	r change	e: #	The AMR memory c counters I	WB+ 3GPP ounters that	floating were us cumente	point sed fo ed an	t refe or ver d the	renc ificat re is	e code tion pu no ne	e conta irpose ed to l	ains c s. The keep t	ompl e outp he co	exity a out of ode he	and these eavily

	counters has been documented and there is no need to keep the code heavily instrumented.
Summary of change: ೫	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: ೫	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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 **x** S4-040780

[#]	26.304 CR 011 #re	ev 1 ^(#) Current version: 6.0.0 ^(#)								
For <u>HELP</u> of	n using this form, see bottom of this pag	e or look at the pop-up text over the \Re symbols.								
Proposed chang	le affects: │ UICC apps <mark>೫ │ </mark> M	E X Radio Access Network Core Network								
Title:	Editorial changes									
Source:	X TSG-SA WG4									
Work item code	X AMRWB+	<i>Date:</i> <mark>೫</mark> 14/12/2004								
Category:	 D Use <u>one</u> of the following categories: F (correction) A (corresponds to a correction in a B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories be found in 3GPP <u>TR 21.900</u>. 	Release: X Rel-6Use one of the following releases: Ph2 (GSM Phase 2)on earlier release)R96 (Release 1996)R97 (Release 1997)R97 (Release 1997)e)R98 (Release 1998)R99 (Release 1999)R99 (Release 1999)gories canRel-4 (Release 4)Rel-5 (Release 5)Rel-6 (Release 6)Rel-7 (Release 7)								

Reason for change:	光 T C	he description of the bitstream file format is not aligned with the floating point C- ode.					
Summary of change	<mark>بر ال</mark>	Editorial changes in Section 5.2 to include the correct bitstream file format.					
Consequences if not approved:	Consequences if May be confusing if the bitstream file format in 26.304 is not aligned with code.						
Clauses affected:	ж S	Section 2.					
	T S	Table 3. Section 5.2.					
	Y	Ν					
Other specs	ЖХ	Other core specifications # 26.290					
affected:		X Test specifications X O&M Specifications					
Other comments:	ж						

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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., <u>http://www.ietf.org/internet-drafts/draft-ietf-avt-rtp-amrwbplus-01.txt</u>, 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_rampGain_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	Gain_hi_mscb_2a	4*2	
Float32	Gain_hi_mscb_5a	32*2	
	TBC		
	.50		

Ö <cut text>

I

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 <u>are either according</u> 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	FREQUENCY	B1	B2	 Bmn	51	52	 Ssn
	EXTENSION	SCALE						
	RATE							

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)	Number of data bytes
	(bits/frame)	
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
0x0 4	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	θ
0x0F	AMR-WB NO_DATA	θ
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	θ
0x1F	NO_DATA	θ

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	Stereo extension rate(incl. BWE)	Number of data octets
EXTENSION RATE MODE	(bits/frame)	
0xFF	No Stereo Extension	θ
0x00	40 bits/frame	5
0x01	4 8 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	4	0		
2	STEREO_	EXTENSION	 RATE =7 ((88 bits/frame))					
	0	0	0	0	0	1	1	1		
3	FREQUE	NCY_SCAL	E=96							
	0	1	1	0	0	0	0	0		
4	AMR-WB	+ data (octe	⊨1)							
	b0	b1	b2	b3	b 4	b5	b6	b7		

536	AMR-W	AMR-WB+ data (octets 2 to 33)										
	b8	Ö-	Ö-	Ö-	Ö-	Ö-	Ö-	Ö -				
37	AMR-W	AMR-WB+ data (octet 34)										
	b26 4	b265	b266	b267	b268	b269	b270	b271				
38	AMR-W	AMR-WB+ data (octet 35)										
	s0	s1	s2	s3	s 4	s5	s6	s7				
3947	AMR-W	AMR-WB+ data (octet 36 to 44)										
	58	Ö-	Ö-	Ö -	Ö -	Ö -	Ö -	Ö-				
48	AMR-W	AMR-WB+ data (octet 45)										
	s80	s81	s82	\$83	s84	s85	s86	s87				

Tdoc **x** S4-040722

							CR-Form-v7.1	
æ	26.290 CF	R <mark>012</mark>	жrev	-	ß C	Current versi	^{on:} 6.0.0	æ
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the \mathfrak{X} symbols.							mbols.	
Proposed change affects: UICC apps ME X Radio Access Network Core Network								
Title:	ដ Editorial chan	jes						
Source:	<mark>೫ TSG-SA WG4</mark>							
Work item code.	<mark>ំង AMRWB+</mark>					Date: 🕷	14/12/2004	
Category:	 D Use <u>one</u> of the formation of	ollowing categorie n) onds to a correcti of feature), al modification of modification) tions of the abov P <u>TR 21.900</u> .	es: on in an ear feature) e categories	ilier rele	I ase)	Release: Use <u>one</u> of t Ph2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	Rel-6 he following re. (GSM Phase 2) (Release 1996) (Release 1998) (Release 1999) (Release 4) (Release 5) (Release 5) (Release 6) (Release 7)	leases:))))

Reason for change:	Cone 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:	H 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:	器 8.1
Other specs affected:	Y N X Other core specifications X Test specifications X O&M Specifications
Other comments:	æ

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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
<u>12</u>	<u>36000</u>	<u>14.22</u>	<u>45/32</u>
<u>1213</u>	38400	13.33	3/2

Tdoc **#S4-040723**

CHANGE REQUEST							R-Form-v7.1				
æ		<mark>26.30</mark>	<mark>4</mark> CR	013	жrev	-	Ħ	Current ver	rsion:	6.0.0	ж
For HELP on using this form, see bottom of this page or look at the pop-up text over the # symbols.							nbols.				
Proposed change affects: UICC apps # ME X Radio Access Network Core Network											
Title:	ж	Remov	al of the	eid tool							
Source:	ж	TSG-S	A WG4								
Work item code:	ж	AMRW	B+					Date: 8	€ <mark>14</mark>	/12/2004	
Category:	អ ((៤	D Jse <u>one</u> F (c A (c B (a C (t D (e Detailed De found	of the folk correction) correspon addition of functional editorial m explanatio in 3GPP	owing categories ds to a correctio feature), modification of f odification) ons of the above <u>TR 21.900</u> .	s: n in an ea feature) categorie	rlier re s can	eleas	Release: 8 Use <u>one</u> c Ph2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	f the for (GSI (Relo (Relo (Relo (Relo (Relo (Relo (Relo	21-6 ollowing rele M Phase 2) ease 1996) ease 1997) ease 1998) ease 1999) ease 4) ease 5) ease 6) ease 7)	eases:

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. Additionally the tool is based on a specific file format used in the selection phase and which is no longuer supported.				
• • • •					
Summary of change: #	I he eld tool directory is removed from the source code distribution				
Consequences if #	The eid tool may be used for frame errasure insertion but the results would not				
not approved:	he correct				
not approved.	De correct.				
Clauses affected: #	The eid directory in the source code distribution				
	YN				
Other specs #	X Other core specifications #				
offected					
anecied	A rest 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 <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

The directory eid/ is removed from the source code distribution
Tdoc **#S4-040766**

CHANGE REQUEST										
æ	26.30	04 CR (014	жrev	1	ж	Current vers	sion:	6.0.0	ж
For <u>HELP</u> on using this form, see bottom of this page or look at the pop-up text over the # symbols.										
Proposed chang	e affects:	UICC ap	ops#	ME	Rad	dio A	ccess Netwo	rk	Core Ne	twork
Title:	육 Additic	n of frame	erasures sim	ulation a	t the	deco	der			
Source:	<mark>೫ TSG-S</mark>	# TSG-SA WG4								
Work item code:	<mark>អ AMRN</mark>	/B+					<i>Date:</i> ೫	14/1	12/2004	
Category:		of the follow correction) correspond addition of t functional m editorial mo explanation in 3GPP <u>T</u>	ving categories s to a correction ieature), nodification of fe dification) is of the above <u>R 21.900</u> .	: n in an ea eature) categorie:	rlier re s can	elease	Release: ₩ Use <u>one</u> of Ph2 R96 R97 R98 R99 Rel-4 Rel-5 Rel-6 Rel-7	Rel- the fol (GSM (Relea (Relea (Relea (Relea (Relea (Relea	-6 llowing rele 1 Phase 2) ase 1996) ase 1997) ase 1998) ase 1999) ase 4) ase 5) ase 5) ase 6) ase 7)	ases:

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.						
Summery of changes	Simulation of frame arrangers is done in the decoder						
Summary of change: #	Simulation of frame effastires is done in the decoder						
Consequences if #	Not possible to simulate frame errasures with the AMR-WB+ 3GPP reference						
not approved:	code						
Clauses affected: #	dec wbplus.c mem.h						
	YN						
Other specs #	X Other core specifications #						
offected:							
anected:							
	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 <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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

```
{
                                /* AMR_WB core mode: 0..8 */
   short mode;
   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
ł
                                /* AMR_WB core mode: 0..8 */
   short mode;
                                /* 0=AMRWB, 1=mono, 2=stereo20%, 3=stereo25% */
   short extension;
                                /* stereo mode 0..13 */
  short st_mode;
   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]);
        }
    }
}</pre>
```

Tdoc **#S4-040725**

ж	2	2 <mark>6.304</mark>	CR	015	жrev	-	Ħ	Curren	t versi	ion:	6.0.0	ж
For <u>HELP</u> or	n usir	ng this for	m, see	bottom of this	s page or	look	at the	e pop-u	p text	over	the ¥ sy	mbols.
Proposed chang	e aff	ects:	JICC a	ррѕж	ME	Rad	dio A	ccess N	letworl	k 📃	Core Ne	etwork
Title:	жI	Removal	of two	unused stered	o rate							
Source:	ж -	TSG-SA	NG4									
Work item code:	ж /	AMRWB	-					Da	te: ೫	14/	12/2004	
Category:	₩ <mark>U</mark> U	D se <u>one</u> of F (cor A (cor B (add C (fun D (edi etailed exp e found in	the follo rection) respond lition of ctional r torial mo blanation 3GPP <u>1</u>	wing categories ls to a correctio feature), nodification of f odification) ns of the above <u>R 21.900</u> .	s: n in an ea feature) categorie	rlier re s can	elease	Releas Use <u>c</u> Ph RS RS RS RS RS RS RS RS RS RS RS RS RS	se: # one of t o2 06 07 08 09 09 09 09 09 09 09 09 09 09 09 09 09	Rel the fo. (GSN (Rele (Rele (Rele (Rele (Rele (Rele	-6 Ilowing rel I Phase 2) ase 1997) ase 1998) ase 1999) ase 4) ase 5) ase 6) ase 7)	eases:

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 #	Mismatch between written specification and implementation in ANSI-C code.						
not approved:							
Clauses affected: #	tables_stereo.c table_decl.h						
	Y N						
Other specs #	X Other core specifications %						
affected:	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 <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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.

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 \text{ kbps } */
  (int)(1.6*80), /* 1.6 + 0.8 = 2.4 kbps */
  (int)(1.8*80), /* 1.8 + 0.8 = 2.6 \text{ 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 \text{ kbps } */
  (int)(1.6*80), /* 1.6 + 0.8 = 2.4 kbps */
  (int)(2.0*80), /* 2.0 + 0.8 = 2.8 \text{ kbps } */
  (int)(2.4*80), /* 2.4 + 0.8 = 3.2 \text{ kbps } */
  (int)(2.8*80), /* 2.8 + 0.8 = 3.6 \text{ kbps } */
  (int)(3.2*80), /* 3.2 + 0.8 = 4.0 \text{ kbps } */
  (int)(3.6*80), /* 3.6 + 0.8 = 4.4 \text{ kbps } */
  (int)(4.0*80), /* 4.0 + 0.8 = 4.8 \text{ kbps } */
  (int)(4.4*80), /* 4.4 + 0.8 = 5.2 kbps */
                 /* 4.8 + 0.8 = 5.6 kbps */
  (int)(4.8*80),
                  /* 5.2 + 0.8 = 6.0 kbps */
  (int)(5.2*80),
  (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];
```

Tdoc *****S4-040726

CHANGE REQUEST																
ж		26	<mark>.30</mark> 4	4 CR	016	ж	rev	-	Ħ	Cu	irrent v	versi	on:	6.0	0.0	ж
For HELP on using this form, see bottom of this page or look at the pop-up text over the # symbols.																
Proposed chang	je a	affec	ts:	UICC	apps# 🦲]	MEX	Rad	dio A	cce	ss Net	twork	< 📃	Со	re Ne	etwork
Title:	ж	So	urce	code eq	ditorial cha	inges										
Source:	ж	TS	TSG-SA WG4													
Work item code:	:Ж	AN	IRWE	3+							Date	e: X	14/	12/20	004	
Category:	ж	D Use Deta be fo	<u>one</u> c F (cc A (c B (a C (f L D (e iled e bund i	of the follo prrection prespor ddition c unctional ditorial n xplanati n 3GPP	lowing cates) ads to a corri f feature), I modification, ons of the a <u>TR 21.900</u> .	gories: rection in on of feati) above cat	an ea ure) tegorie	rlier re s can	eleas	Re (elease Use <u>ond</u> Ph2 R96 R97 R98 R99 Rel-d Rel-d Rel-d	e: 36 e of t (((((4 (5 6 6 7 (Rel he fo (GSN (Rele (Rele (Rele (Rele (Rele (Rele	<mark>-6</mark> Ilowin A Pha ase 1 ase 1 ase 1 ase 5 ase 6 ase 6 ase 7	ng rele se 2) (996) (997) (998) (999) () () () () () () () () () () () () ()	ases:

Reason for change: ೫	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 #	I hused parameters may degrade the readability of the code						
not approved:	onused 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: ೫							

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at <u>http://www.3gpp.org/specs/CR.htm</u>. 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 <u>ftp://ftp.3gpp.org/specs/</u> 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 */
```